To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Application Note

78K0R/LH3

Sample Program (Operational Amplifier)

Operational Amplifier Application Program

This document generally describes the operation and use of a sample program. This sample program is an application for an on-chip op amp (operational amplifier). The operation examples describe cases where an op amp has been connected to a resistor or capacitor and uses an on-chip D/A converter for AC or DC signals that have been input. There are also operation examples that include an oscillator connected to a resistor or capacitor.

Target device

78K0R/LH3 microcontroller

CONTENTS

Document No. U20228EJ1V0AN00 (1st edition)
Date Published March 2010 N

CHAPTER 1 OVERVIEW

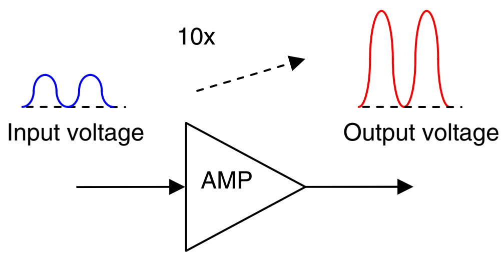
The following is a general description of the sample program.

The CHx that appears under [Circuit configuration] indicates the waveform CH from **CHAPTER 5 EXAMPLE OF OPERATION CHECK USING DEVICE**.

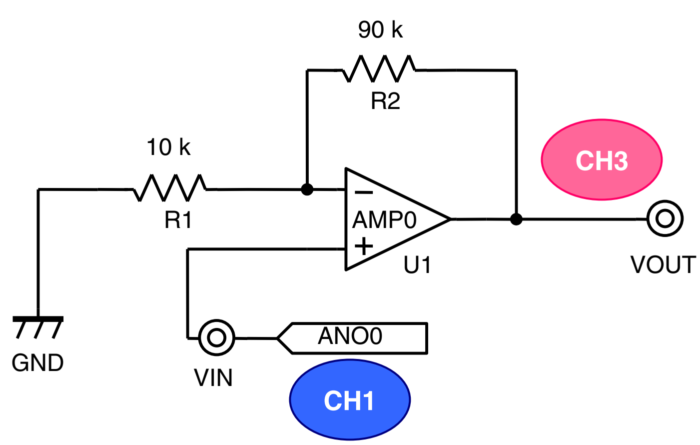
1.1 Non-Inverting Amplifier Circuit

[Overview]

Using the ground as the reference, the DAC0 output voltage (ANO0 pin) is amplified tenfold.



[Circuit configuration]



	Pin	Center Potential	Amplitude	Frequency
Reference voltage	(GND)	–	–	–
Input voltage	ANO0	0.1 V	±0.1 V	1 kHz
Output voltage	AMP0O	1.0 V	±1.0 V	1 kHz

(1) Primary initial settings

<Option byte settings>

- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Settings during initialization immediately after a reset ends>

- Setting up I/O ports (all unused ports are set to low-level output)
- Setting CPU clock to X1 oscillator operation (20 MHz)
- Stopping the internal high-speed oscillator
- Setting up timer array unit 0
  - Setting channel 1 for interval operation at a 500 μs interval
  - Setting channel 4 for interval operation at a 10 μs interval
- Ensuring that the supply voltage is at least 2.7 V by using the low-voltage detector
- Setting up voltage reference
  - Setting the voltage reference output voltage to 2.0 V
  - Operation stabilization wait time (approximately 17 ms) for voltage reference that uses channel 1 of the timer array unit
- Setting up op amp
  - Setting P20/AMP0–, P21/AMP0O, and P22/AMP0+ as analog input pins
  - Enabling operation of op amp 0
- Setting up D/A converter
  - Setting positive reference voltage as voltage reference output voltage
  - Setting P110/ANO0 for analog output
  - Setting operating mode of channel 0 to real-time output mode
  - Setting resolution of channel 0 to 12 bits
- Setting up DMA controller
  - Setting DMA start trigger for INTTM04 interrupt
  - Setting transfer data size to 16 bits and number of transfers to 100
  - Setting transfer source to RAM area and transfer destination to DACSW0 register
  - Enabling operation of DMA channel 0

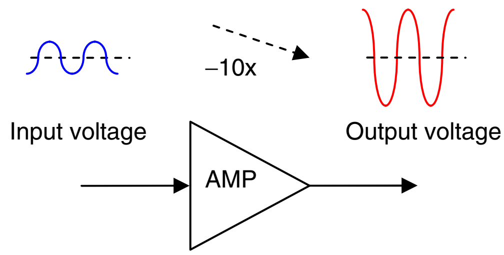
(2) Processing after main loop

After initialization, operation of D/A converter channel 0 is enabled, as are the count operation of timer array unit 0 channel 4 and the INTDMA0 interrupt, and then the mode switches to HALT mode. Subsequently, each time an INTDMA0 interrupt occurs, the DMA controller is reset and the mode is switched back to HALT mode.

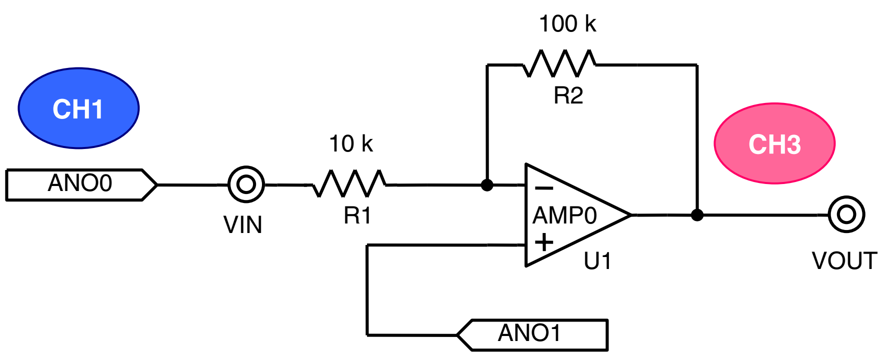
1.2 Inverting Amplifier Circuit

[Overview]

Using the DAC1 output voltage (ANO1 pin) as the reference, the DAC0 output voltage (ANO0 pin) is inverting amplified tenfold.



[Circuit configuration]



	Pin	Center Potential	Amplitude	Frequency
Reference voltage	ANO1	1.0 V	–	–
Input voltage	ANO0	1.0 V	±0.1 V	1 kHz
Output voltage	AMP0O	1.0 V	±1.0 V	1 kHz

(1) Primary initial settings

<Option byte settings>

- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Settings during initialization immediately after a reset ends>

- Setting up I/O ports (all unused ports are set to low-level output)
- Setting CPU clock to X1 oscillator operation (20 MHz)
- Stopping the internal high-speed oscillator
- Setting up timer array unit 0
  - Setting channel 1 for interval operation at a 500 μs interval
  - Setting channel 4 for interval operation at a 10 μs interval
- Ensuring that the supply voltage is at least 2.7 V by using the low-voltage detector
- Setting up voltage reference
  - Setting the voltage reference output voltage to 2.0 V
  - Operation stabilization wait time (approximately 17 ms) for voltage reference that uses channel 1 of the timer array unit
- Setting up op amp
  - Setting P20/AMP0–, P21/AMP0O, and P22/AMP0+ as analog input pins
  - Enabling operation of op amp 0
- Setting up D/A converter
  - Setting positive reference voltage as voltage reference output voltage
  - Setting P110/ANO0 and P111/ANO1 for analog output
  - Setting operating mode of channels 0 and 1 to normal mode
  - Setting resolution of channels 0 and 1 to 12 bits
- Setting up DMA controller
  - Setting DMA start trigger for INTTM04 interrupt
  - Setting transfer data size to 16 bits and number of transfers to 100
  - Setting transfer source to RAM area and transfer destination to DACSW0 register
  - Enabling operation of DMA channel 0

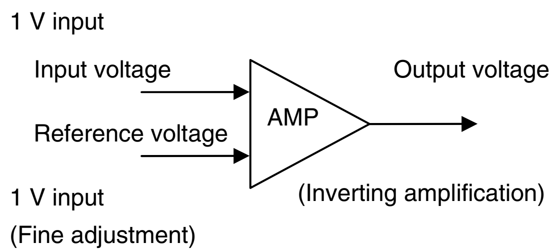
(2) Processing after main loop

After initialization, operation of D/A converter channels 0 and 1 is enabled, as are the count operation of timer array unit 0 channel 4 and the INTDMA0 interrupt, and then the mode switches to HALT mode. Subsequently, each time an INTDMA0 interrupt occurs, the DMA controller is reset and the mode is switched back to HALT mode.

1.3 Offset Adjustment of Inverting Amplifier Circuit

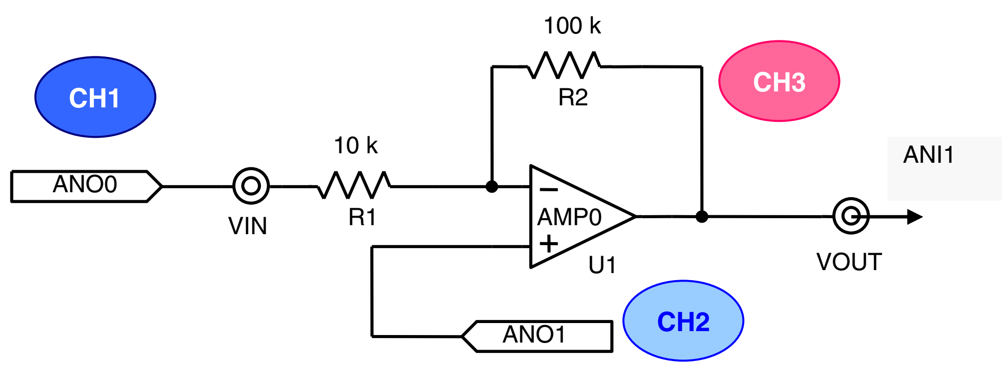
[Overview]

An offset adjustment flow has been added to the end of the flow described in **1.2 Inverting Amplifier Circuit**. For offset adjustment, the reference voltage is finely adjusted and this reference voltage is fixed at the point where the input voltage (1 V) and output voltage match.



Use AD to check whether output voltage is 1 V.
If it is not 1 V, finely adjust the reference voltage at approximately 1 V until the output voltage becomes 1 V.

[Circuit configuration]



	Pin	Center Potential	Amplitude	Frequency
Reference voltage	ANO1	Adjust from 0.98 V	–	–
Input voltage	ANO0	1.0 V	±0.1 V	1 kHz
Output voltage	AMP0O	1.0 V	±1.0 V	1 kHz

(1) Primary initial settings

<Option byte settings>

- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Settings during initialization immediately after a reset ends>

- Setting up I/O ports (all unused ports are set to low-level output)
- Setting CPU clock to X1 oscillator operation (20 MHz)
- Stopping the internal high-speed oscillator
- Setting up timer array unit 0
  - Setting channel 1 for interval operation at a 500 μs interval
  - Setting channel 4 for interval operation at a 10 μs interval
- Ensuring that the supply voltage is at least 2.7 V by using the low-voltage detector
- Setting up voltage reference
  - Setting the voltage reference output voltage to 2.0 V
  - Operation stabilization wait time (approximately 17 ms) for voltage reference that uses channel 1 of the timer array unit
- Setting up op amp
  - Setting P20/AMP0–, P21/AMP0O, and P22/AMP0+ as analog input pins
  - Enabling operation of op amp 0
- Setting up D/A converter
  - Setting positive reference voltage as voltage reference output voltage
  - Setting P110/ANO0 and P111/ANO1 for analog output
  - Setting operating mode of channel 0 to real-time output mode
  - Setting operating mode of channel 1 to normal mode
  - Setting resolution of channels 0 and 1 to 12 bits
- Setting up DMA controller
  - Setting DMA start trigger for INTTM04 interrupt
  - Setting transfer data size to 16 bits and number of transfers to 100
  - Setting transfer source to RAM area and transfer destination to DACSW0 register
  - Enabling operation of DMA channel 0

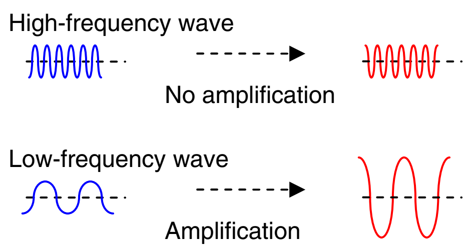
(2) Processing after main loop

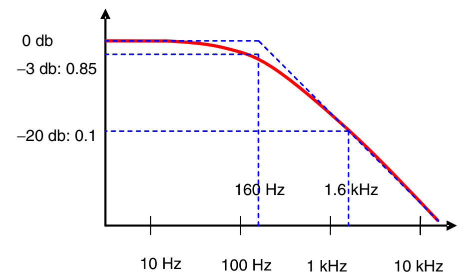
After initialization, operation of D/A converter channels 0 and 1 is enabled, as are the count operation of timer array unit 0 channel 4 and the INTDMA0 interrupt, and then the mode switches to HALT mode. Subsequently, each time an INTDMA0 interrupt occurs, the A/D converter is used to adjust the ANO1 output (op amp reference voltage), then the DMA controller is reset and the mode is switched back to HALT mode.

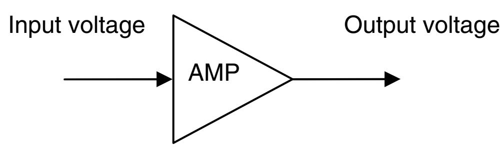
1.4 Offset Adjustment of Inverting Amplifier Circuit + Low-Pass Filter

[Overview]

A filter circuit (C1) has been added to the circuit described in **1.3 Offset Adjustment of Inverting Amplifier Circuit**. This enables us to confirm that a reduction has been made in the amplification of 1 kHz input. The amplification can also be checked for 100 Hz (low-frequency wave) input.



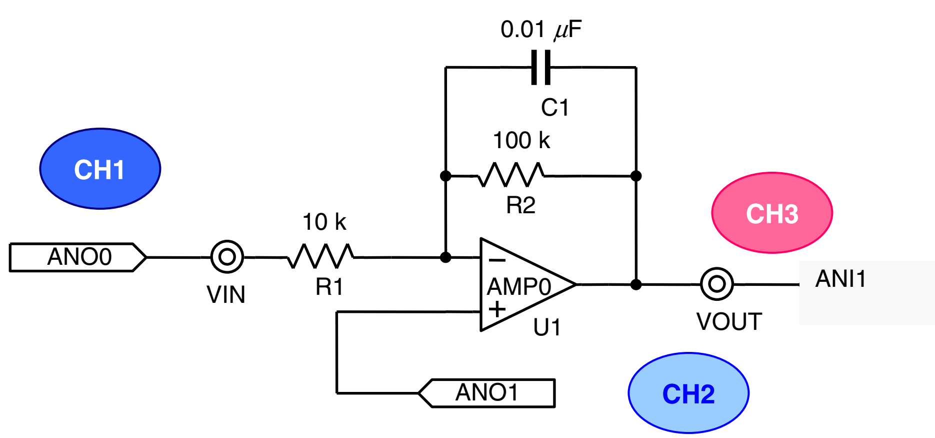




Cutoff frequency: Fc = 1/(2πCR) = 160 Hz

- 160 Hz or above (cut off) → 1 kHz
- 160 Hz or below (not cut off) → 100 Hz

[Circuit configuration]



	Pin	Center Potential	Amplitude	Frequency
Reference voltage	ANO1	Adjust from 0.98 V	–	–
Input voltage	ANO0	1.0 V	±0.1 V	1 kHz/100 Hz
Output voltage	AMP0O	1.0 V	Reduce/ Do not reduce	1 kHz/100 Hz

(1) Primary initial settings

<Option byte settings>

- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Settings during initialization immediately after a reset ends>

- Setting up I/O ports (all unused ports are set to low-level output)
- Setting CPU clock to X1 oscillator operation (20 MHz)
- Stopping the internal high-speed oscillator
- Setting up timer array unit 0
  - Setting channel 1 for interval operation at a 500 μs interval
  - Setting channel 4 for interval operation at a 10 μs interval
- Ensuring that the supply voltage is at least 2.7 V by using the low-voltage detector
- Setting up voltage reference
  - Setting the voltage reference output voltage to 2.0 V
  - Operation stabilization wait time (approximately 17 ms) for voltage reference that uses channel 1 of the timer array unit
- Setting up op amp
  - Setting P20/AMP0–, P21/AMP0O, and P22/AMP0+ as analog input pins
  - Enabling operation of op amp 0
- Setting up D/A converter
  - Setting positive reference voltage as voltage reference output voltage
  - Setting P110/ANO0 and P111/ANO1 for analog output
  - Setting operating mode of channel 0 to real-time output mode
  - Setting operating mode of channel 1 to normal mode
  - Setting resolution of channels 0 and 1 to 12 bits
- Setting up DMA controller
  - Setting DMA start trigger for INTTM04 interrupt
  - Setting transfer data size to 16 bits and number of transfers to 100
  - Setting transfer source to RAM area and transfer destination to DACSW0 register
  - Enabling operation of DMA channel 0

(2) Processing after main loop

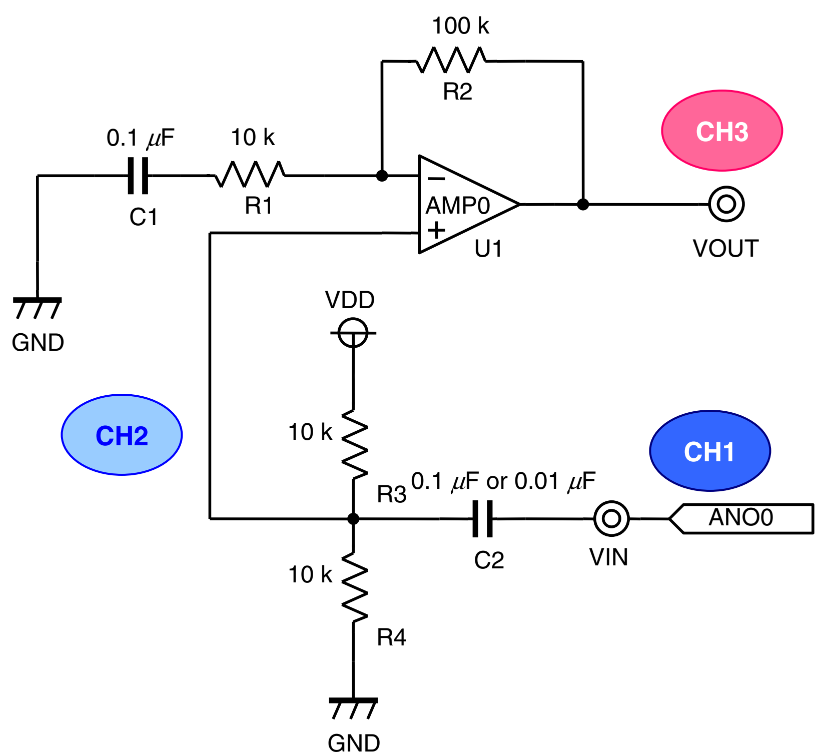
After initialization, operation of D/A converter channels 0 and 1 is enabled, as are the count operation of timer array unit 0 channel 4 and the INTDMA0 interrupt, and then the mode switches to HALT mode. Subsequently, each time an INTDMA0 interrupt occurs, the A/D converter is used to adjust the ANO1 output (op amp reference voltage), and then the DMA controller is reset. Also, periodically, the channel 4 count value of timer array unit 0 is updated and the frequency of the ANO0 output is changed.

1.5 Non-Inverting Amplifier Circuit (AC)

[Overview]

This circuit performs amplification when there is a large amount of variation (in frequency) in the input voltage. When a high-frequency wave and a low-frequency wave having the same amplitude are input, their differential can be checked.

[Circuit configuration]



	Pin	Center Potential	Amplitude	Frequency
Reference voltage	(GND)	–	–	–
Input voltage	ANO0	1.0 V	±0.1 V	High/Low
Output voltage	AMP0O	1.5 V	Yes/No	–

(1) Primary initial settings

<Option byte settings>

- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Settings during initialization immediately after a reset ends>

- Setting up I/O ports (all unused ports are set to low-level output)
- Setting CPU clock to X1 oscillator operation (20 MHz)
- Stopping the internal high-speed oscillator
- Setting up timer array unit 0
  - Setting channel 1 for interval operation at a 500 μs interval
  - Setting channel 4 for interval operation at a 100 μs interval
- Ensuring that the supply voltage is at least 2.7 V by using the low-voltage detector
- Setting up voltage reference
  - Setting the voltage reference output voltage to 2.0 V
  - Operation stabilization wait time (approximately 17 ms) for voltage reference that uses channel 1 of the timer array unit
- Setting up op amp
  - Setting P20/AMP0–, P21/AMP0O, and P22/AMP0+ as analog input pins
  - Enabling operation of op amp 0
- Setting up D/A converter
  - Setting positive reference voltage as voltage reference output voltage
  - Setting P110/ANO0 for analog output
  - Setting operating mode of channel 0 to real-time output mode
  - Setting resolution of channel 0 to 12 bits
- Setting up DMA controller
  - Setting DMA start trigger for INTTM04 interrupt
  - Setting transfer data size to 16 bits and number of transfers to 25
  - Setting transfer source to RAM area and transfer destination to DACSW0 register
  - Enabling operation of DMA channel 0

(2) Processing after main loop

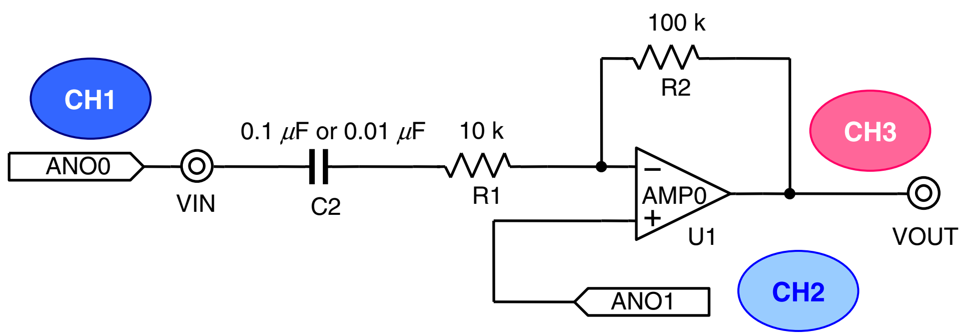
After initialization, operation of D/A converter channel 0 is enabled, as are the count operation of timer array unit 0 channel 4 and the INTDMA0 interrupt, and then the mode switches to HALT mode. Subsequently, each time an INTDMA0 interrupt occurs, the DMA controller transfer count and the channel 4 count of timer array unit 0 are updated and the frequency of the ANO0 output is changed.

1.6 Inverting Amplifier Circuit (AC)

[Overview]

This circuit performs inverting amplification when there is a large amount of variation (in frequency) in the input voltage. When a high-frequency wave and a low-frequency wave having the same amplitude are input, their differential can be checked.

[Circuit configuration]



	Pin	Center Potential	Amplitude	Frequency
Reference voltage	ANO1	1.0 V	–	–
Input voltage	ANO0	1.0 V	+0.1 V	High/Low
Output voltage	AMP0O	1.0 V	Yes/No	–

(1) Primary initial settings

<Option byte settings>

- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Settings during initialization immediately after a reset ends>

- Setting up I/O ports (all unused ports are set to low-level output)
- Setting CPU clock to X1 oscillator operation (20 MHz)
- Stopping the internal high-speed oscillator
- Setting up timer array unit 0
  - Setting channel 1 for interval operation at a 500 μs interval
  - Setting channel 4 for interval operation at a 100 μs interval
- Ensuring that the supply voltage is at least 2.7 V by using the low-voltage detector
- Setting up voltage reference
  - Setting the voltage reference output voltage to 2.0 V
  - Operation stabilization wait time (approximately 17 ms) for voltage reference that uses channel 1 of the timer array unit
- Setting up op amp
  - Setting P20/AMP0–, P21/AMP0O, and P22/AMP0+ as analog input pins
  - Enabling operation of op amp 0
- Setting up D/A converter
  - Setting positive reference voltage as voltage reference output voltage
  - Setting P110/ANO0 and P111/ANO1 for analog output
  - Setting operating mode of channel 0 to real-time output mode
  - Setting operating mode of channel 1 to normal mode
  - Setting resolution of channels 0 and 1 to 12 bits
- Setting up DMA controller
  - Setting DMA start trigger for INTTM04 interrupt
  - Setting transfer data size to 16 bits and number of transfers to 25
  - Setting transfer source to RAM area and transfer destination to DACSW0 register
  - Enabling operation of DMA channel 0

(2) Processing after main loop

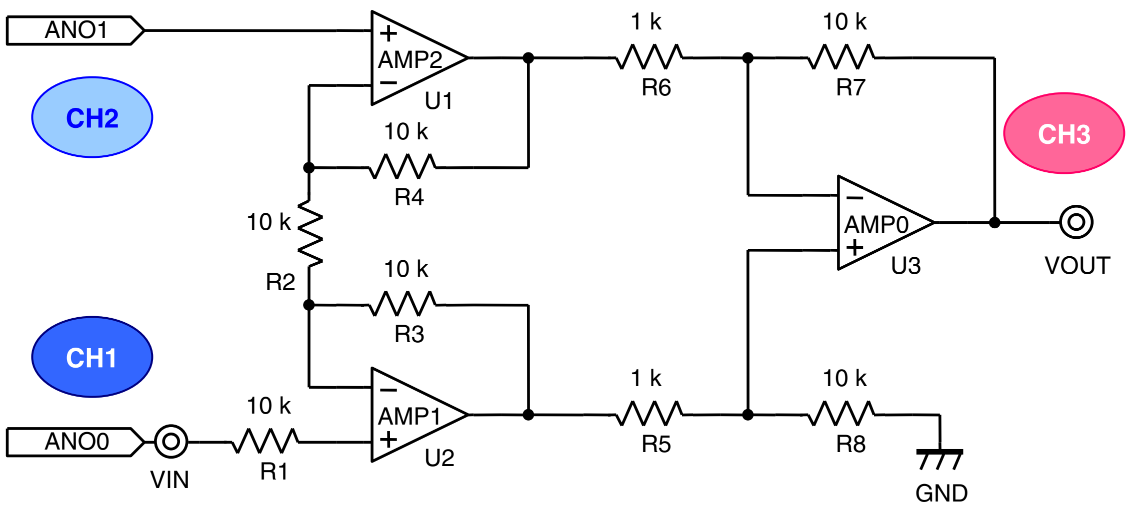
After initialization, operation of D/A converter channels 0 and 1 is enabled, as are the count operation of timer array unit 0 channel 4 and the INTDMA0 interrupt, and then the mode switches to HALT mode. Subsequently, each time an INTDMA0 interrupt occurs, the DMA controller transfer count and the channel 4 count of timer array unit 0 are updated and the frequency of the ANO0 output is changed.

1.7 Instrumentation Amplifier

[Overview]

This amplifies the potential difference between DAC1 output (ANO1 pin) and DAC0 output (ANO0 pin). With DAC1 output (ANO1 pin) fixed at 0.5 V, DAC0 output (ANO0 pin) is switched between 0.52 V and 0.54 V. The amplifier shown below performs 30x amplification. Since the input potential differences are 0.02 V and 0.04 V, the respective output voltages are 0.6 V and 1.2 V. Since the instrumentation amplifier is able to amplify the potential differences regardless of the input level, it can be used for amplification of bridge-type two-input sensors.

[Circuit configuration]



	Pin	Center Potential	Amplitude	Frequency
Reference voltage	ANO1	0.5 V	–	–
Input voltage	ANO0	0.52 V/0.54 V	–	–
Output voltage	AMP0O	0.6 V/1.2 V	–	–

(1) Primary initial settings

<Option byte settings>

- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Settings during initialization immediately after a reset ends>

- Setting up I/O ports (all unused ports are set to low-level output)
- Setting CPU clock to XT1 oscillator operation (32.768 kHz)
- Stopping the internal high-speed oscillator
- Ensuring that the supply voltage is at least 2.7 V by using the low-voltage detector
- Setting up real-time counter
  - Oscillation stabilization wait time for subsystem clock that uses channel 1 of timer array unit 0
  - Setting of periodic interrupts (1 second cycle)
- Setting up voltage reference
  - Setting the voltage reference output voltage to 2.0 V
  - Operation stabilization wait time (approximately 17 ms) for voltage reference that uses channel 1 of the timer array unit
- Setting up op amp
  - Setting P20/AMP0−, P21/AMP0O, P22/AMP0+, P23/AMP1−, P24/AMP1O, P25/AMP1+, P26/AMP2−, P27/AMP2O, and P150/AMP2+ as analog input pins
  - Enabling operation of op amp 0, op amp 1, and op amp 2
- Setting up D/A converter
  - Setting positive reference voltage as voltage reference output voltage
  - Setting P110/ANO0 and P111/ANO1 for analog output
  - Setting operating mode of channels 0 and 1 to normal mode
  - Setting resolution of channels 0 and 1 to 12 bits
  - Setting ANO0 output to 0.52 V
  - Setting ANO1 output to 0.5 V

(2) Processing after main loop

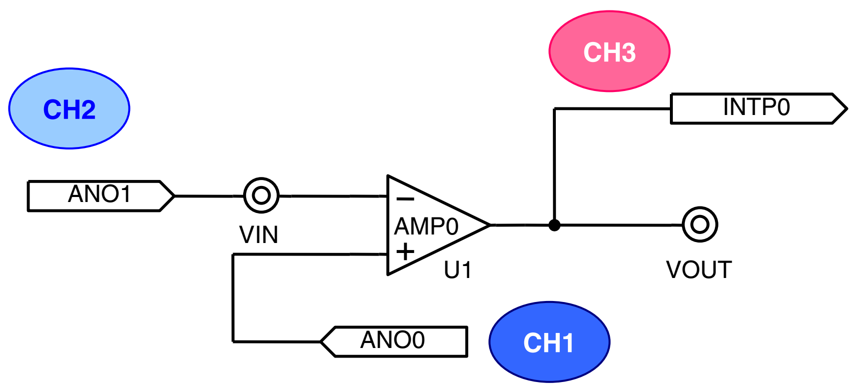
After initialization, operation of D/A converter channels 0 and 1 is enabled, as are the real-time counter operation and the INTRTC interrupt, and then the mode switches to HALT mode. Subsequently, each time an INTRTC interrupt occurs (1 second cycle), ANO0 output is changed alternately between 0.52 V and 0.54 V.

1.8 Comparator Circuit

[Overview]

The op amp is used as a comparator. Connect DAC0 output (ANIO1 pin) as the reference voltage to the AMP0– pin, and, in this case connect DAC0 output (ANIO0 pin) as the level being tested to the AMP0+ pin. When the level being tested goes above (or below) the reference voltage, AMP0 output (AMP0O pin) goes to high (or low) level. Since this connection is to the INTP0 pin, it can be used as an external interrupt.

[Circuit configuration]



	Pin	Center Potential	Amplitude	Frequency
Reference voltage	ANO1	1.0 V	–	–
Input voltage	ANO0	1.0 V	±0.1 V	1 kHz
Output voltage	AMP0O	1.5 V	±1.5 V	1 kHz

(1) Primary initial settings

<Option byte settings>

- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Settings during initialization immediately after a reset ends>

- Setting up I/O ports
  - Setting P73 for output of comparison results
- Setting CPU clock to X1 oscillator operation (20 MHz)
- Stopping the internal high-speed oscillator
- Setting up timer array unit 0
  - Setting channel 1 for interval operation at a 500 μs interval
  - Setting channel 4 for interval operation at a 10 μs interval
- Ensuring that the supply voltage is at least 2.7 V by using the low-voltage detector
- Setting up voltage reference
  - Setting the voltage reference output voltage to 2.0 V
  - Operation stabilization wait time (approximately 17 ms) for voltage reference that uses channel 1 of the timer array unit
- Setting up op amp
  - Setting P20/AMP0–, P21/AMP0O, and P22/AMP0+ as analog input pins
  - Enabling operation of op amp 0
- Setting up D/A converter
  - Setting positive reference voltage as voltage reference output voltage
  - Setting P110/ANO0 and P111/ANO1 for analog output
  - Setting operating mode of channel 0 to normal mode
  - Setting operating mode of channel 1 to normal mode
  - Setting resolution of channels 0 and 1 to 12 bits
- Setting up DMA controller
  - Setting DMA start trigger for INTTM04 interrupt
  - Setting transfer data size to 16 bits and number of transfers to 100
  - Setting transfer source to RAM area and transfer destination to DACSW0 register
  - Enabling operation of DMA channel 0
- Setting both the rising edge and the falling edge as valid edges of the INTP0 pin

(2) Processing after main loop

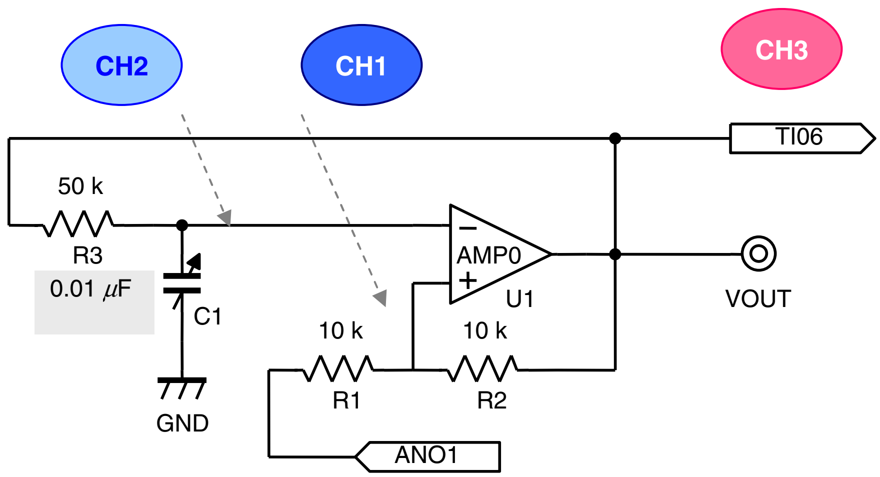
After initialization, operation of D/A converter channels 0 and 1 is enabled, as are the count operation of timer array unit 0 channel 4, the INTDMA0 interrupt, and the INTP0 interrupt, and then the mode switches to HALT mode. Subsequently, each time an INTP0 interrupt occurs, triggered by detection of the rising edge or falling edge of AMP0 output, the detected level of the INTP0 pin is output to P73. Also, each time an INTDMA0 interrupt occurs, the DMA controller is reset.

1.9 CR Oscillator 1

[Overview]

A D/A converter and op amp are used to configure this CR oscillator. The op amp is used as a comparator. DAC1 output (ANIO1 pin) is used as the reference voltage, a resistor is inserted between it and AMP0 output (AMP0O pin), and it is input to the AMP0+ pin. The input voltage to the AMP0+ pin varies according to the AMP output status. This variation causes the input level to the AMP0– pin to repeatedly charge and discharge. As a frequency measuring method, the input pulse interval measurement function of the timer array unit is used to measure the pulse width of AMP0 output. The measurement result is output to the LCD. This system is able to measure changes in the capacitance of C1 as frequency changes.

[Circuit configuration]



	Pin	Center Potential	Amplitude	Frequency
Reference voltage	ANO1	1.5 V	–	–
Reference voltage'	(AMP0+)	0.75 V/2.25 V	–	Approximately 0.83 kHz
Output voltage	AMP0O	0 V/3 V	–	Approximately 0.83 kHz

(1) Primary initial settings

<Option byte settings>

- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Settings during initialization immediately after a reset ends>

- Setting up I/O ports (all unused ports are set to low-level output)
- Setting CPU clock to X1 oscillator operation (20 MHz)
- Stopping the internal high-speed oscillator
- Setting up timer array unit 0
  - Setting channel 1 for interval operation at a 500 μs interval
  - Setting channel 4 for interval operation at a 1 ms interval
  - Setting channel 6 for input pulse interval measurement
- Ensuring that the supply voltage is at least 2.7 V by using the low-voltage detector
- Setting up op amp
  - Setting P20/AMP0–, P21/AMP0O, and P22/AMP0+ as analog input pins
  - Enabling operation of op amp 0
- Setting up D/A converter
  - Setting positive reference voltage source to AV_{DD1} pin
  - Setting P111/ANO1 for analog output
  - Setting operating mode of channel 1 to normal mode
  - Setting resolution of channel 1 to 12 bits
  - Setting ANO1 output to 1.5 V
- Setting up LCD controller/driver
  - Setting SEG4 to SEG53 for segment output
  - Selecting capacitance splitting method for LCD drive voltage generator
  - Setting LCD mode for eight time divisions and 1/4 bias
  - Setting data in A pattern area as display data
  - Setting LCD source clock (f_{LCD}) to f_{SUB}, and LCD clock to $f_{LCD}/2^7$
  - Setting LCD drive voltage (V_{LC0}) to 5.0 V
  - Enabling LCD display

(2) Processing after main loop

After initialization, operation of D/A converter channel 1 is enabled, and the count operation of timer array unit 0 channel 4 and the channel 6 input pulse interval measurement operation are started. Subsequently, each time an INTTM04 interrupt occurs (approximately every ms), the AMP0 output frequency is measured and the measurement results are output to the LCD.

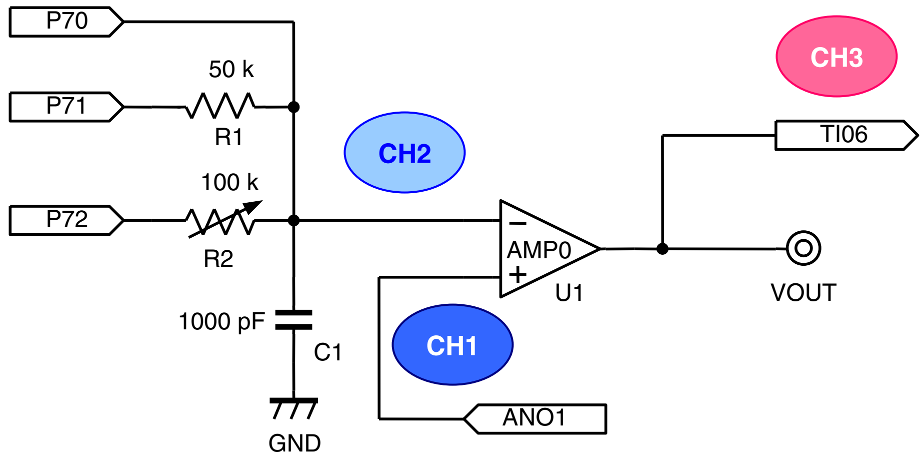
1.10 CR Oscillator 2

[Overview]

A D/A converter and op amp are used to configure this CR oscillator. The op amp is used as a comparator. DAC1 output (ANIO1 pin) is input to the AMP0+ pin as the reference voltage. Meanwhile, a microcontroller command sets P70 high-level output to the AMP0– pin, charging a load to C1. Afterward, P71 goes to low-level output and the C1 load is discharged via the known resistance R1. When this discharge causes the voltage to go below DAC1 output (ANIO1 pin), AMPO0 goes to high-level output. As a frequency measuring method, the input pulse interval measurement function of the timer array unit is used to measure the low-level period of AMP0 output. The same is implemented with regard to P72, which is connected to the unknown resistance R2. This R2 resistance is calculated from a comparison of these two and from the known resistance R1.

The time differential is calculated between two measurements made with reference voltages 1 V and 2 V. This eliminates differences due to time required for start processing, and provides a more accurate time measurement.

[Circuit configuration]



	Pin	Center Potential	Amplitude	Frequency
Reference voltage	ANO1	1.0 V/2.0 V	–	–
Output voltage	AMP0O	0 V/3 V	–	–

	Pin		DAC1 Output	Discharge Time
R1 measurement	P71 low-level output	P72 input mode	1.0 V	T_REF1
			2.0 V	T_REF2
R2 measurement	P71 input mode	P72 low-level output	1.0 V	T_TH1
			2.0 V	T_TH2

In the sample program, the count value differential between T_REF1 and T_REF2 and that between T_TH1 and T_TH2 are displayed on the LCD.

(1) Primary initial settings

<Option byte settings>

- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Settings during initialization immediately after a reset ends>

- Setting up I/O ports (all unused ports are set to low-level output)
- Setting CPU clock to X1 oscillator operation (20 MHz)
- Stopping the internal high-speed oscillator
- Setting up timer array unit 0
  - Setting channel 1 for interval operation at a 500 μs interval
  - Setting channel 6 for input pulse interval measurement
- Ensuring that the supply voltage is at least 2.7 V by using the low-voltage detector
- Setting up op amp
  - Setting P20/AMP0–, P21/AMP0O, and P22/AMP0+ as analog input pins
  - Enabling operation of op amp 0
- Setting up D/A converter
  - Setting positive reference voltage source to AV_{DD1} pin
  - Setting P111/ANO1 for analog output
  - Setting operating mode of channel 1 to normal mode
  - Setting resolution of channel 1 to 12 bits
  - Setting ANO1 output to 1.0 V
- Setting up LCD controller/driver
  - Setting SEG4 to SEG53 for segment output
  - Selecting capacitance splitting method for LCD drive voltage generator
  - Setting LCD mode for eight time divisions and 1/4 bias
  - Setting data in A pattern area as display data
  - Setting LCD source clock (f_{LCD}) to f_{SUB}, and LCD clock to $f_{LCD}/2^7$
  - Setting LCD drive voltage (V_{LC0}) to 5.0 V
  - Enabling LCD display

(2) Processing after main loop

After initialization, operation of D/A converter channel 1 is enabled. Subsequently, each time the port I/O setting is changed, the ANO1 output is changed (between 1.0 V and 2.0 V), which causes the capacitor to charge and discharge. The discharge time is measured by channel 6 of timer array unit 0, and the measurement results are displayed on the LCD.

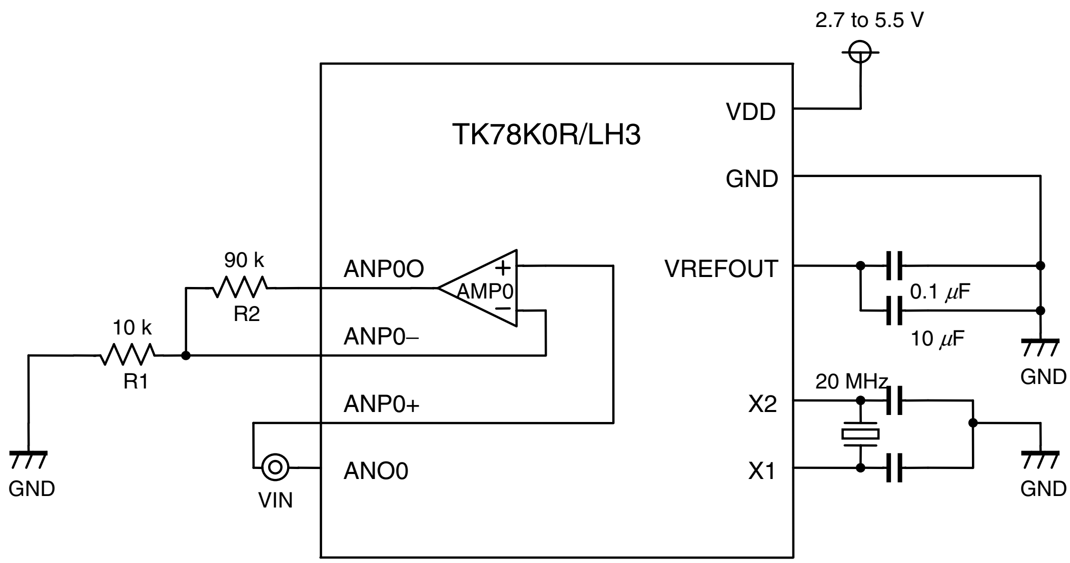
CHAPTER 2 CIRCUIT DIAGRAM

This chapter provides a circuit diagram and describes the devices used in this sample program other than the microcontroller.

2.1 Circuit Diagram

A circuit diagram is shown below.

2.1.1 Non-inverting amplifier circuit



Cautions 1. Use the microcontroller at a voltage in the range of 2.7 V ≤ V_{DD} ≤ 5.5 V.

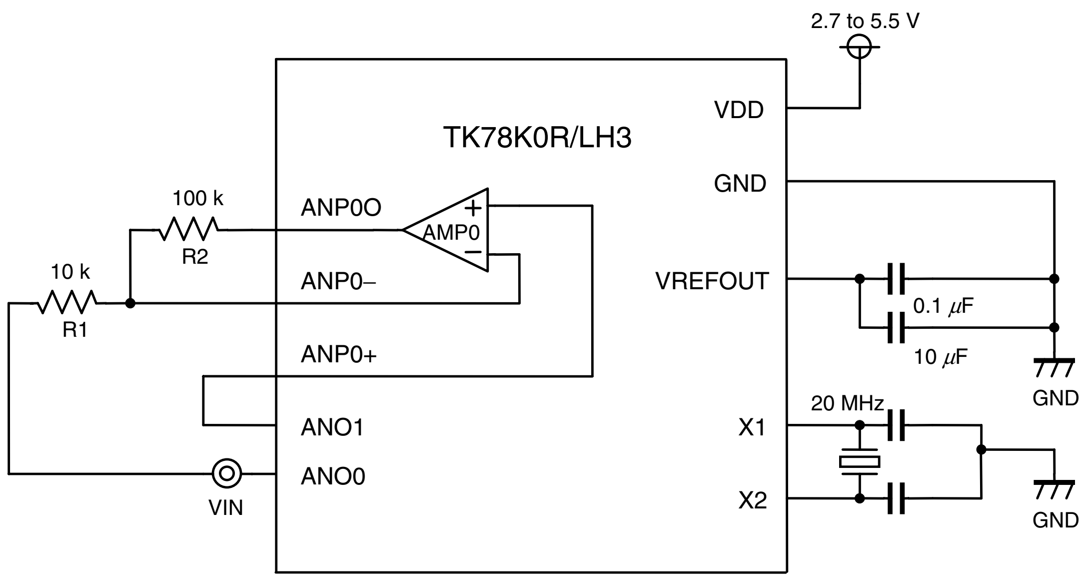
2. Make EV_{DD}, AV_{DD0}, and AV_{DD1} have the same potential as V_{DD}.

3. Make AV_{SS} have the same potential as EV_{SS} and V_{SS}, and connect it directly to GND.

4. During the voltage reference operation, always connect a tantalum capacitor (capacitance: 10 μF ±30%, ESR: 2 Ω (Max.), ESL: 10 nH (Max.)) and a ceramic capacitor (capacitance: 0.1 μF ±30%, ESR: 2 Ω (Max.), ESL: 10 nH (Max.)) to the V_{REFOUT}/AV_{REFP} pin to ensure stabilization of the reference voltage. Also, during the voltage reference operation, do not apply voltage from the V_{REFOUT}/AV_{REFP} pin.

5. For details of these settings, see the TK-78K0R/LH3+LCD User's Manual.

2.1.2 Inverting amplifier circuit



Cautions 1. Use the microcontroller at a voltage in the range of 2.7 V ≤ V_{DD} ≤ 5.5 V.

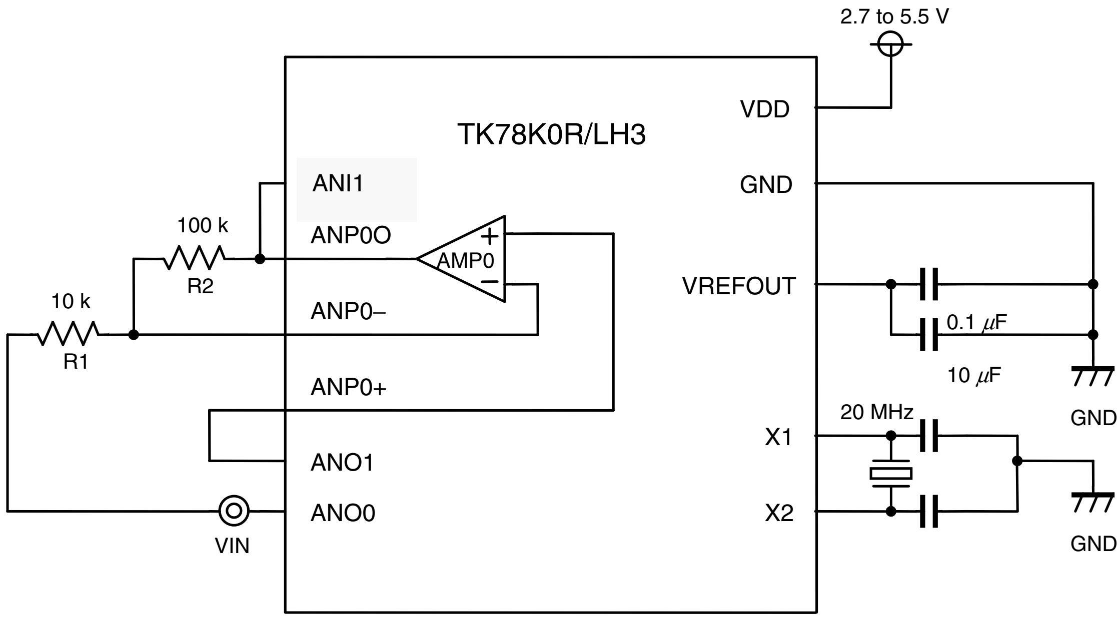
2. Make EV_{DD}, AV_{DD0}, and AV_{DD1} have the same potential as V_{DD}.

3. Make AV_{SS} have the same potential as EV_{SS} and V_{SS}, and connect it directly to GND.

4. During the voltage reference operation, always connect a tantalum capacitor (capacitance: 10 μF ±30%, ESR: 2 Ω (Max.), ESL: 10 nH (Max.)) and a ceramic capacitor (capacitance: 0.1 μF ±30%, ESR: 2 Ω (Max.), ESL: 10 nH (Max.)) to the V_{REFOUT}/AV_{REFP} pin to ensure stabilization of the reference voltage. Also, during the voltage reference operation, do not apply voltage from the V_{REFOUT}/AV_{REFP} pin.

5. For details of these settings, see the TK-78K0R/LH3+LCD User's Manual.

2.1.3 Offset adjustment of inverting amplifier circuit



Cautions 1. Use the microcontroller at a voltage in the range of 2.7 V ≤ V_{DD} ≤ 5.5 V.

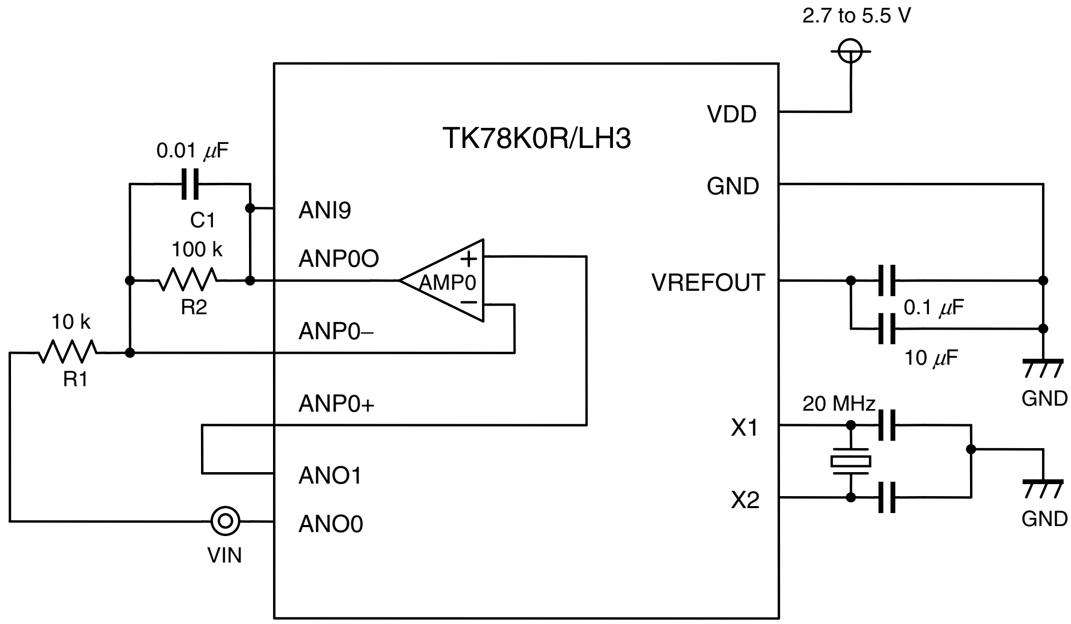
2. Make EV_{DD}, AV_{DD0}, and AV_{DD1} have the same potential as V_{DD}.

3. Make AV_{SS} have the same potential as EV_{SS} and V_{SS}, and connect it directly to GND.

4. During the voltage reference operation, always connect a tantalum capacitor (capacitance: 10 μF ±30%, ESR: 2 Ω (Max.), ESL: 10 nH (Max.)) and a ceramic capacitor (capacitance: 0.1 μF ±30%, ESR: 2 Ω (Max.), ESL: 10 nH (Max.)) to the V_{REFOUT}/AV_{REFP} pin to ensure stabilization of the reference voltage. Also, during the voltage reference operation, do not apply voltage from the V_{REFOUT}/AV_{REFP} pin.

5. For details of these settings, see the TK-78K0R/LH3+LCD User's Manual.

2.1.4 Offset adjustment of inverting amplifier circuit + low-pass filter



Cautions 1. Use the microcontroller at a voltage in the range of 2.7 V ≤ V_{DD} ≤ 5.5 V.

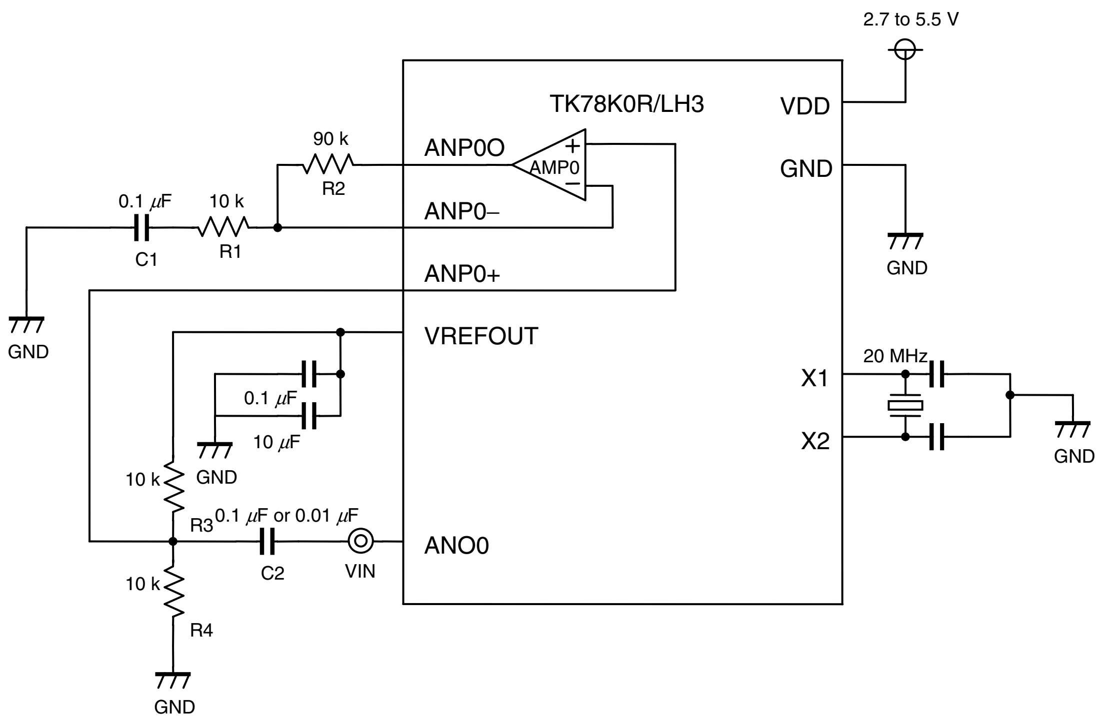
2. Make EV_{DD}, AV_{DD0}, and AV_{DD1} have the same potential as V_{DD}.

3. Make AV_{SS} have the same potential as EV_{SS} and V_{SS}, and connect it directly to GND.

4. During the voltage reference operation, always connect a tantalum capacitor (capacitance: 10 µF ±30%, ESR: 2 Ω (Max.), ESL: 10 nH (Max.)) and a ceramic capacitor (capacitance: 0.1 µF ±30%, ESR: 2 Ω (Max.), ESL: 10 nH (Max.)) to the V_{REFOUT}/AV_{REFP} pin to ensure stabilization of the reference voltage. Also, during the voltage reference operation, do not apply voltage from the V_{REFOUT}/AV_{REFP} pin.

5. For details of these settings, see the TK-78K0R/LH3+LCD User's Manual.

2.1.5 Non-inverting amplifier circuit (AC)



Cautions 1. Use the microcontroller at a voltage in the range of 2.7 V ≤ V_{DD} ≤ 5.5 V.

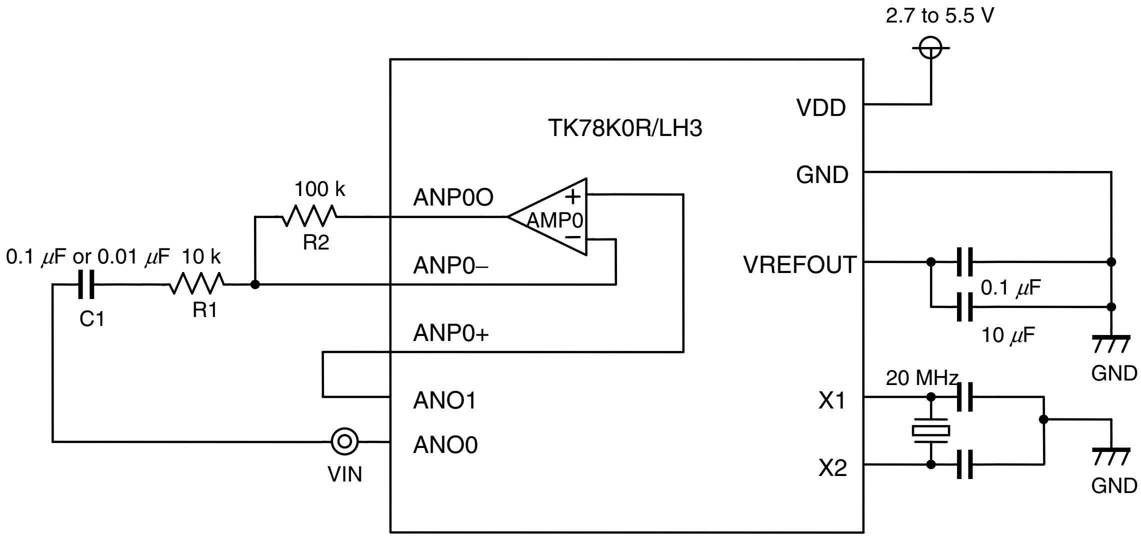
2. Make EV_{DD}, AV_{DD0}, and AV_{DD1} have the same potential as V_{DD}.

3. Make AV_{SS} have the same potential as EV_{SS} and V_{SS}, and connect it directly to GND.

4. During the voltage reference operation, always connect a tantalum capacitor (capacitance: 10 μF ±30%, ESR: 2 Ω (Max.), ESL: 10 nH (Max.)) and a ceramic capacitor (capacitance: 0.1 μF ±30%, ESR: 2 Ω (Max.), ESL: 10 nH (Max.)) to the V_{REFOUT}/AV_{REFP} pin to ensure stabilization of the reference voltage. Also, during the voltage reference operation, do not apply voltage from the V_{REFOUT}/AV_{REFP} pin.

5. For details of these settings, see the TK-78K0R/LH3+LCD User's Manual.

2.1.6 Inverting amplifier circuit (AC)



Cautions 1. Use the microcontroller at a voltage in the range of 2.7 V ≤ V_{DD} ≤ 5.5 V.

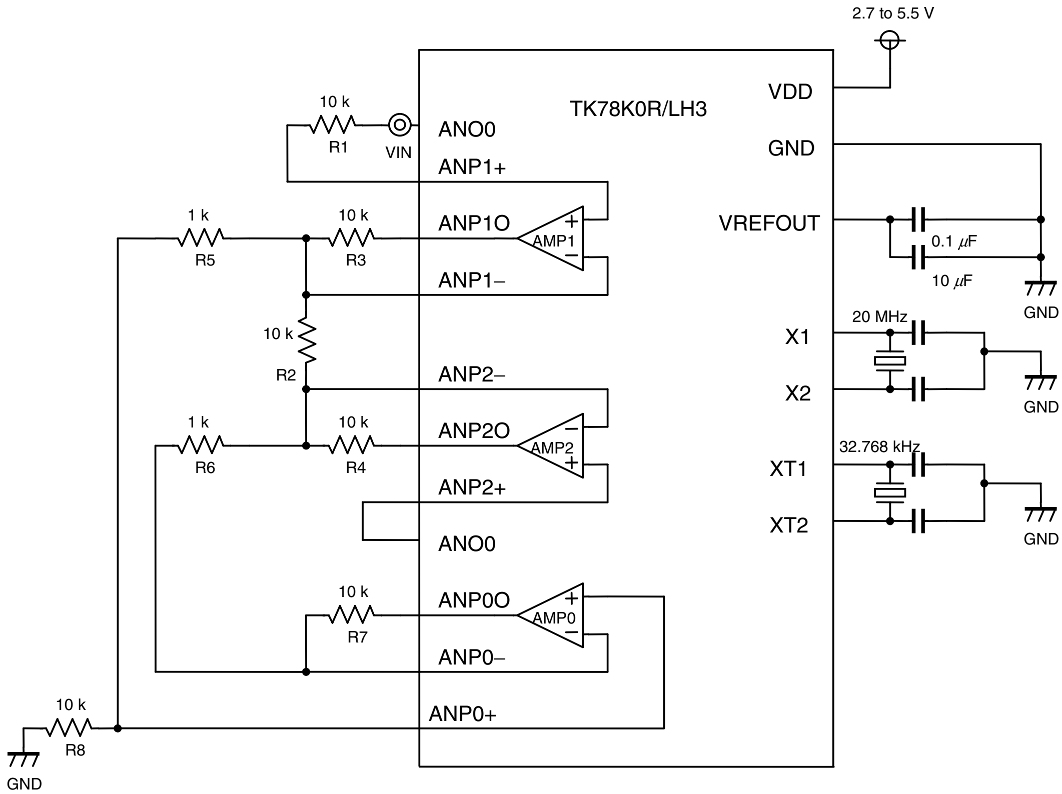
2. Make EV_{DD}, AV_{DD0}, and AV_{DD1} have the same potential as V_{DD}.

3. Make AV_{SS} have the same potential as EV_{SS} and V_{SS}, and connect it directly to GND.

4. During the voltage reference operation, always connect a tantalum capacitor (capacitance: 10 μF ±30%, ESR: 2 Ω (Max.), ESL: 10 nH (Max.)) and a ceramic capacitor (capacitance: 0.1 μF ±30%, ESR: 2 Ω (Max.), ESL: 10 nH (Max.)) to the V_{REFOUT}/AV_{REFP} pin to ensure stabilization of the reference voltage. Also, during the voltage reference operation, do not apply voltage from the V_{REFOUT}/AV_{REFP} pin.

5. For details of these settings, see the TK-78K0R/LH3+LCD User's Manual.

2.1.7 Instrumentation amplifier



Cautions 1. Use the microcontroller at a voltage in the range of 2.7 V ≤ V_{DD} ≤ 5.5 V.

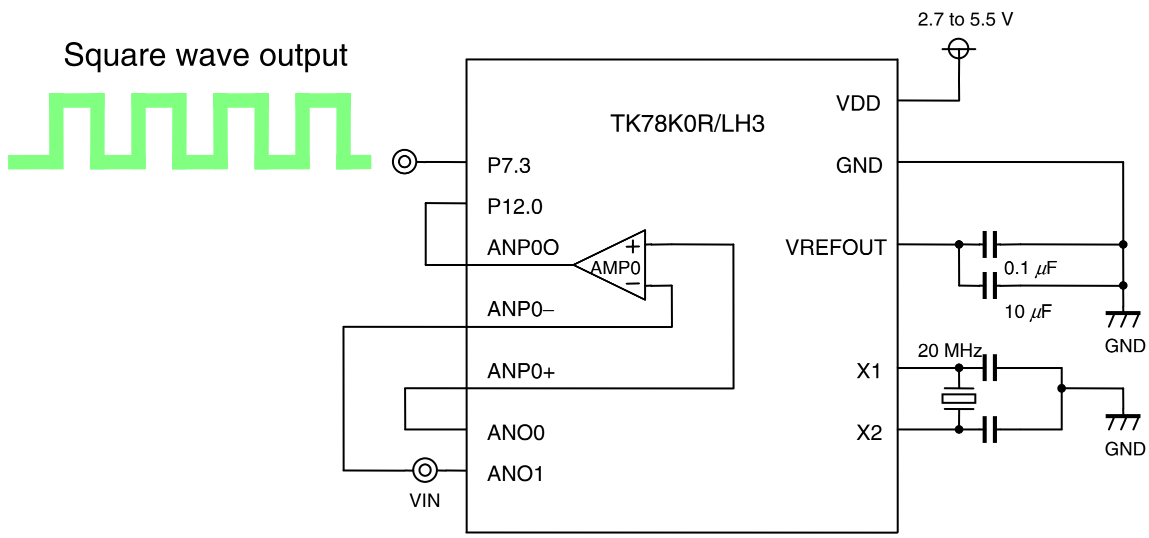
2. Make EV_{DD}, AV_{DD0}, and AV_{DD1} have the same potential as V_{DD}.

3. Make AV_{SS} have the same potential as EV_{SS} and V_{SS}, and connect it directly to GND.

4. During the voltage reference operation, always connect a tantalum capacitor (capacitance: 10 μF ±30%, ESR: 2 Ω (Max.), ESL: 10 nH (Max.)) and a ceramic capacitor (capacitance: 0.1 μF ±30%, ESR: 2 Ω (Max.), ESL: 10 nH (Max.)) to the V_{REFOUT}/AV_{REFP} pin to ensure stabilization of the reference voltage. Also, during the voltage reference operation, do not apply voltage from the V_{REFOUT}/AV_{REFP} pin.

5. For details of these settings, see the TK-78K0R/LH3+LCD User's Manual.

2.1.8 Comparator circuit



Cautions 1. Use the microcontroller at a voltage in the range of 2.7 V ≤ V_{DD} ≤ 5.5 V.

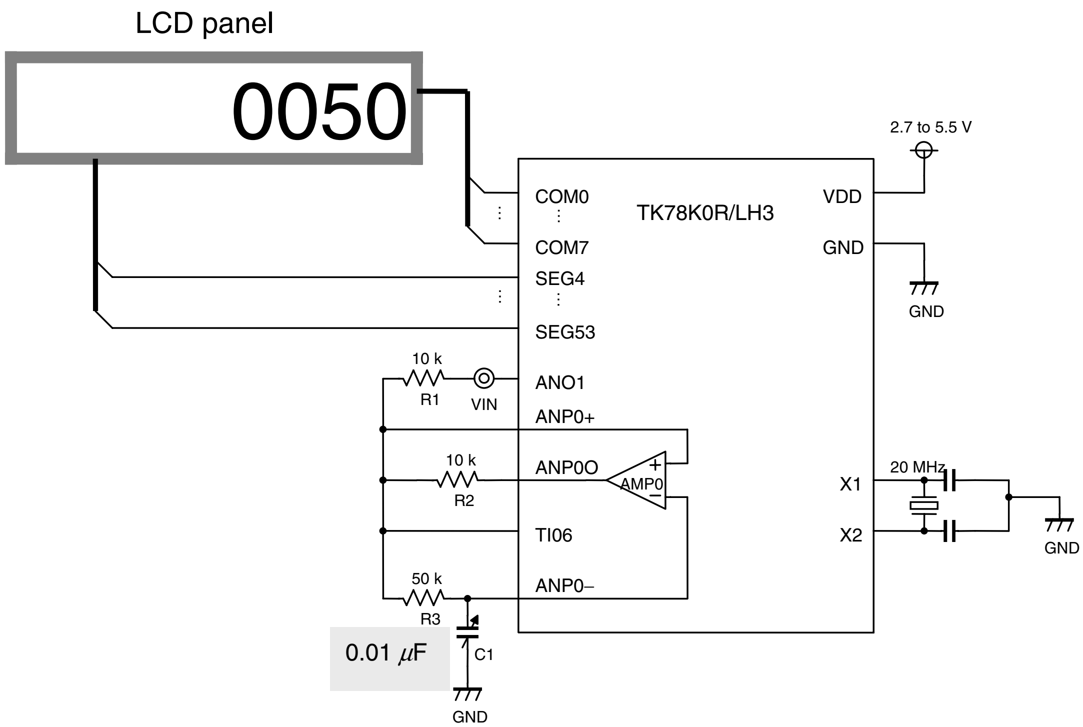
2. Make EV_{DD}, AV_{DD0}, and AV_{DD1} have the same potential as V_{DD}.

3. Make AV_{SS} have the same potential as EV_{SS} and V_{SS}, and connect it directly to GND.

4. During the voltage reference operation, always connect a tantalum capacitor (capacitance: 10 μF ±30%, ESR: 2 Ω (Max.), ESL: 10 nH (Max.)) and a ceramic capacitor (capacitance: 0.1 μF ±30%, ESR: 2 Ω (Max.), ESL: 10 nH (Max.)) to the V_{REFOUT}/AV_{REFP} pin to ensure stabilization of the reference voltage. Also, during the voltage reference operation, do not apply voltage from the V_{REFOUT}/AV_{REFP} pin.

5. For details of these settings, see the TK-78K0R/LH3+LCD User's Manual.

2.1.9 CR oscillator 1



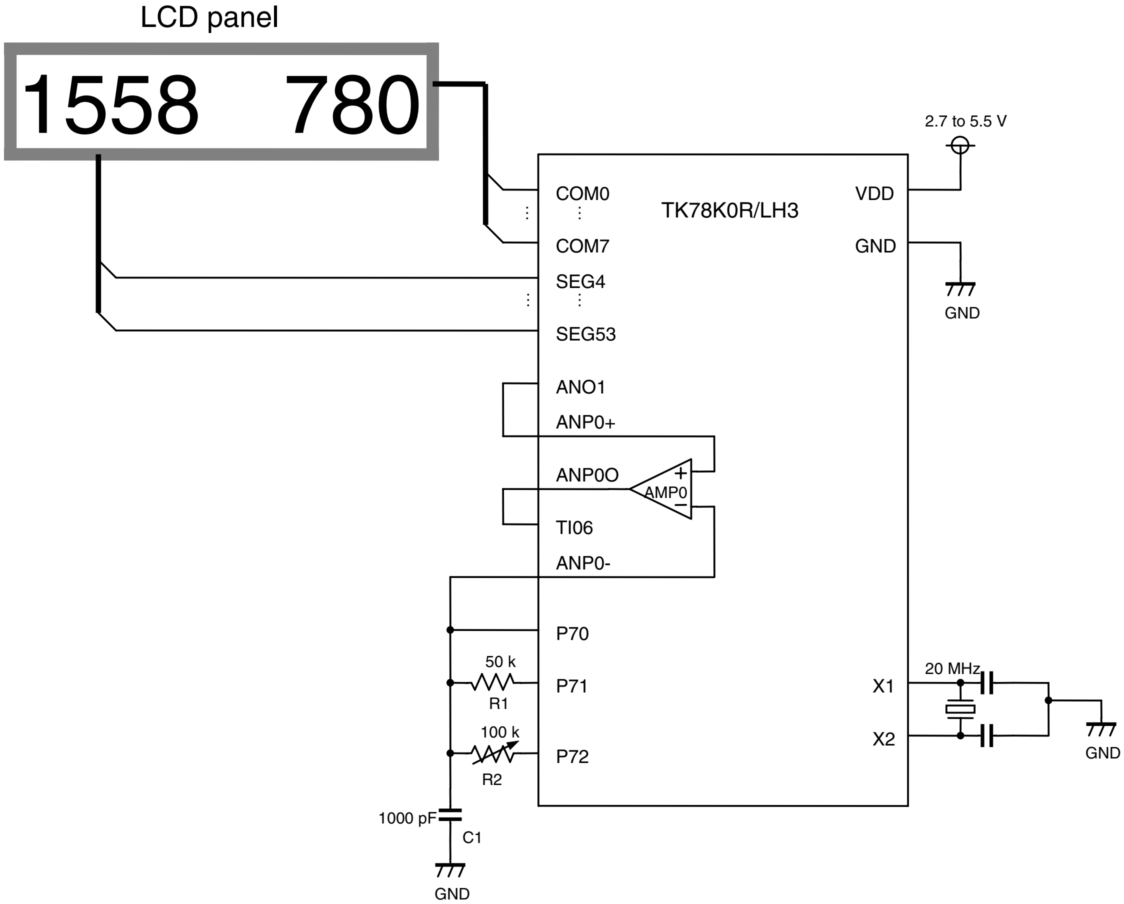
Cautions 1. Use the microcontroller at a voltage in the range of 2.7 V ≤ V_{DD} ≤ 5.5 V.

2. Make EV_{DD}, AV_{DD0}, and AV_{DD1} have the same potential as V_{DD}.

3. Make AV_{SS} have the same potential as EV_{SS} and V_{SS}, and connect it directly to GND.

4. For details of these settings, see the TK-78K0R/LH3+LCD User's Manual.

2.1.10 CR oscillator 2



Cautions 1. Use the microcontroller at a voltage in the range of 2.7 V ≤ V_{DD} ≤ 5.5 V.

2. Make EV_{DD}, AV_{DD0}, and AV_{DD1} have the same potential as V_{DD}.

3. Make AV_{SS} have the same potential as EV_{SS} and V_{SS}, and connect it directly to GND.

4. For details of these settings, see the TK-78K0R/LH3+LCD User's Manual.

2.2 Used Device Other than Microcontroller

The following device is used in addition to the microcontroller:

(1) LCD

Use the LCD provided with TK-78K0R/LH3+LCD.

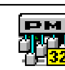
CHAPTER 3 SOFTWARE

This chapter describes the files included in the compressed file to be downloaded, internal peripheral functions of the microcontroller to be used, and initial settings and provides an operation overview of the sample program.

3.1 Included Files

The following table shows the files included in the compressed file to be downloaded.

3.1.1 Non-inverting amplifier circuit

File Name	Description	Compressed (*.zip) File Included	
			
main.c	C language source file for hardware initialization processing and main processing of microcontroller	●	●
op.asm	Assembler source file for setting the option byte (This file is used for setting up the watchdog timer and internal low-speed oscillator.)	●	●
78K0RLx3_sample_program.prw	Work space file for integrated development environment PM plus		●
78K0RLx3_sample_program.prj	Project file for integrated development environment PM plus		●

Remark  : Only the source file is included.

 : The files to be used with integrated development environment PM plus are included.

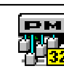
3.1.2 Inverting amplifier circuit

File Name	Description	Compressed (*.zip) File Included	
			
main.c	C language source file for hardware initialization processing and main processing of microcontroller	●	●
op.asm	Assembler source file for setting the option byte (This file is used for setting up the watchdog timer and internal low-speed oscillator.)	●	●
78K0RLx3_sample_program.prw	Work space file for integrated development environment PM plus		●
78K0RLx3_sample_program.prj	Project file for integrated development environment PM plus		●

Remark  : Only the source file is included.

 : The files to be used with integrated development environment PM plus are included.

3.1.3 Offset adjustment of inverting amplifier circuit

File Name	Description	Compressed (*.zip) File Included	
			
main.c	C language source file for hardware initialization processing and main processing of microcontroller	●	●
op.asm	Assembler source file for setting the option byte (This file is used for setting up the watchdog timer and internal low-speed oscillator.)	●	●
78K0RLx3_sample_program.prw	Work space file for integrated development environment PM plus		●
78K0RLx3_sample_program.prj	Project file for integrated development environment PM plus		●

Remark  : Only the source file is included.

 : The files to be used with integrated development environment PM plus are included.

3.1.4 Offset adjustment of inverting amplifier circuit + low-pass filter

File Name	Description	Compressed (*.zip) File Included	
			
main.c	C language source file for hardware initialization processing and main processing of microcontroller	●	●
op.asm	Assembler source file for setting the option byte (This file is used for setting up the watchdog timer and internal low-speed oscillator.)	●	●
78K0RLx3_sample_program.prw	Work space file for integrated development environment PM plus		●
78K0RLx3_sample_program.prj	Project file for integrated development environment PM plus		●

Remark  : Only the source file is included.

 : The files to be used with integrated development environment PM plus are included.

3.1.5 Non-inverting amplifier circuit (AC)

File Name	Description	Compressed (*.zip) File Included	
			
main.c	C language source file for hardware initialization processing and main processing of microcontroller	●	●
op.asm	Assembler source file for setting the option byte (This file is used for setting up the watchdog timer and internal low-speed oscillator.)	●	●
78K0RLx3_sample_program.prw	Work space file for integrated development environment PM plus		●
78K0RLx3_sample_program.prj	Project file for integrated development environment PM plus		●

Remark  : Only the source file is included.

 : The files to be used with integrated development environment PM plus are included.

3.1.6 Inverting amplifier circuit (AC)

File Name	Description	Compressed (*.zip) File Included	
main.c	C language source file for hardware initialization processing and main processing of microcontroller	●	●
op.asm	Assembler source file for setting the option byte (This file is used for setting up the watchdog timer and internal low-speed oscillator.)	●	●
78K0RLx3_sample_program.prw	Work space file for integrated development environment PM plus		●
78K0RLx3_sample_program.prj	Project file for integrated development environment PM plus		●

Remark : Only the source file is included.

: The files to be used with integrated development environment PM plus are included.

3.1.7 Instrumentation amplifier

File Name	Description	Compressed (*.zip) File Included	
main.c	C language source file for hardware initialization processing and main processing of microcontroller	●	●
op.asm	Assembler source file for setting the option byte (This file is used for setting up the watchdog timer and internal low-speed oscillator.)	●	●
78K0RLx3_sample_program.prw	Work space file for integrated development environment PM plus		●
78K0RLx3_sample_program.prj	Project file for integrated development environment PM plus		●

Remark : Only the source file is included.

: The files to be used with integrated development environment PM plus are included.

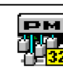
3.1.8 Comparator circuit

File Name	Description	Compressed (*.zip) File Included	
			
main.c	C language source file for hardware initialization processing and main processing of microcontroller	●	●
op.asm	Assembler source file for setting the option byte (This file is used for setting up the watchdog timer and internal low-speed oscillator.)	●	●
78K0RLx3_sample_program.prw	Work space file for integrated development environment PM plus		●
78K0RLx3_sample_program.prj	Project file for integrated development environment PM plus		●

Remark  : Only the source file is included.

 : The files to be used with integrated development environment PM plus are included.

3.1.9 CR oscillator 1

File Name	Description	Compressed (*.zip) File Included	
			
main.c	C language source file for hardware initialization processing and main processing of microcontroller	●	●
displayl.c	C language source file for LCD driver initialization and LCD display processing		
op.asm	Assembler source file for setting the option byte (This file is used for setting up the watchdog timer and internal low-speed oscillator.)	●	●
78K0RLx3_sample_program.prw	Work space file for integrated development environment PM plus		●
78K0RLx3_sample_program.prj	Project file for integrated development environment PM plus		●

Remark  : Only the source file is included.

 : The files to be used with integrated development environment PM plus are included.

3.1.10 CR oscillator 2

File Name	Description	Compressed (*.zip) File Included	
			
main.c	C language source file for hardware initialization processing and main processing of microcontroller	●	●
displayl.c	C language source file for LCD driver initialization and LCD display processing		
op.asm	Assembler source file for setting the option byte (This file is used for setting up the watchdog timer and internal low-speed oscillator.)	●	●
78K0RLx3_sample_program.prw	Work space file for integrated development environment PM plus		●
78K0RLx3_sample_program.prj	Project file for integrated development environment PM plus		●

Remark  : Only the source file is included.

 : The files to be used with integrated development environment PM plus are included.

3.2 Internal Peripheral Functions to Be Used

The following describes peripheral functions that are built into the microcontroller that is used by the sample program.

3.2.1 Non-inverting amplifier circuit

- High-speed system clock oscillator
  For CPU clock and peripheral hardware clock
- TAU0 channel 1
  This is used as an interval timer for the hardware operation stabilization wait.
- TAU0 channel 4
  This is used as an interval timer for DMA timing when generating signal waveforms input to AMP.
- DMA controller channel 4
  This transfers signal waveform data input to AMP to a D/A converter.
- ANO0
  This is used as the signal output that is input to AMP.
- AMP0
  This is used as a non-inverting amplifier circuit.

3.2.2 Inverting amplifier circuit

- High-speed system clock oscillator
  For CPU clock and peripheral hardware clock
- TAU0 channel 1
  This is used as an interval timer for the hardware operation stabilization wait.
- TAU0 channel 4
  This is used as an interval timer for DMA timing when generating signal waveforms input to AMP.
- DMA controller channel 4
  This transfers signal waveform data input to AMP to a D/A converter.
- ANO0
  This is used as the signal output that is input to AMP.
- ANO1
  This is used to output the bias voltage to be input to a non-inverting input pin of AMP.
- AMP0
  This is used as an inverting amplifier circuit.

3.2.3 Offset adjustment of inverting amplifier circuit

- High-speed system clock oscillator
  For CPU clock and peripheral hardware clock
- TAU0 channel 1
  This is used as an interval timer for the hardware operation stabilization wait.
- TAU0 channel 4
  This is used as an interval timer for DMA timing when generating signal waveforms input to AMP.
- DMA controller channel 4
  This transfers signal waveform data input to AMP to a D/A converter.
- ANO0
  This is used as the signal output that is input to AMP.
- ANO1
  This is used to output the bias voltage to be input to a non-inverting input pin of AMP.
- AMP0
  This is used as an inverting amplifier circuit.
- ANI1
  This measures AMP output voltage.

3.2.4 Offset adjustment of inverting amplifier circuit + low-pass filter

- High-speed system clock oscillator
  For CPU clock and peripheral hardware clock
- TAU0 channel 1
  This is used as an interval timer for the hardware operation stabilization wait.
- TAU0 channel 4
  This is used as an interval timer for DMA timing when generating signal waveforms input to AMP.
- DMA controller channel 4
  This transfers signal waveform data input to AMP to a D/A converter.
- ANO0
  This is used as the signal output that is input to AMP.
- ANO1
  This is used to output the bias voltage to be input to a non-inverting input pin of AMP.
- AMP0
  This is used as an inverting amplifier circuit.

3.2.5 Non-inverting amplifier circuit (AC)

- High-speed system clock oscillator
  For CPU clock and peripheral hardware clock
- TAU0 channel 1
  This is used as an interval timer for the hardware operation stabilization wait.
- TAU0 channel 4
  This is used as an interval timer for DMA timing when generating signal waveforms input to AMP.
- DMA controller channel 4
  This transfers signal waveform data input to AMP to a D/A converter.
- ANO0
  This is used as the signal output that is input to AMP.
- AMP0
  This is used as a non-inverting amplifier circuit.

3.2.6 Inverting amplifier circuit (AC)

- High-speed system clock oscillator
  For CPU clock and peripheral hardware clock
- TAU0 channel 1
  This is used as an interval timer for the hardware operation stabilization wait.
- TAU0 channel 4
  This is used as an interval timer for DMA timing when generating signal waveforms input to AMP.
- DMA controller channel 4
  This transfers signal waveform data input to AMP to a D/A converter.
- ANO0
  This is used as the signal output that is input to AMP.
- ANO1
  This is used to output the bias voltage to be input to a non-inverting input pin of AMP.
- AMP0
  This is used as an inverting amplifier circuit.

3.2.7 Instrumentation amplifier

- High-speed system clock oscillator
  For CPU clock and peripheral hardware clock
- Subsystem clock oscillator
  For CPU clock
- TAU0 channel 1
  This is used as an interval timer for the hardware operation stabilization wait.
- ANO0
  This is used to output the signal to be input to a non-inverting input pin of AMP1.
- ANO1
  This is used to output the bias voltage to be input to a non-inverting input pin of AMP2.
- AMP0
  This is used as a non-inverting amplifier circuit.
- AMP1
  This is used as a non-inverting amplifier circuit.
- AMP2
  This is used as a non-inverting amplifier circuit.

3.2.8 Comparator circuit

- High-speed system clock oscillator
  For CPU clock and peripheral hardware clock
- TAU0 channel 1
  This is used as an interval timer for the hardware operation stabilization wait.
- TAU0 channel 4
  This is used as an interval timer for DMA timing when generating signal waveforms input to AMP.
- DMA controller channel 0
  This transfers signal waveform data input to AMP to a D/A converter.
- ANO0
  This is used as the signal output that is input to AMP.
- ANO1
  This is used to output the bias voltage to be input to a non-inverting input pin of AMP.
- AMP0
  This is used as a comparator.
- INTP0
  This is used to input the comparator output and as a dual edge-triggered interrupt.
- P73
  This is used for the software output of the high/low status of comparator output.

3.2.9 CR oscillator 1

- High-speed system clock oscillator
  For CPU clock and peripheral hardware clock
- TAU0 channel 1
  This is used as an interval timer for the hardware operation stabilization wait.
- TAU0 channel 4
  This is used as an interval timer for the interval used to count oscillation pulses.
- TAU0 channel 6
  This is used as an event timer that counts oscillation pulses.
- ANO1
  This is used to output the bias voltage to be input to a non-inverting input pin of AMP.
- AMP0
  This is used as an inverting amplifier circuit.
- TI06
  This is used to input AMP0 output and as event input.
- LCD controller/driver
  This displays measured count values on the LCD on the board.

3.2.10 CR oscillator 2

- High-speed system clock oscillator
  For CPU clock and peripheral hardware clock
- TAU0 channel 1
  This is used as an interval timer for the hardware operation stabilization wait.
- TAU0 channel 4
  This is used as an interval timer for the interval used to count oscillation pulses.
- TAU0 channel 6
  This is used in capture mode as a timer for measuring the charge time of a capacitor.
- ANO1
  This is used to output a comparison reference voltage that is input to a non-inverting input pin of AMP.
- AMP0
  This is used as an inverting amplifier circuit.
- TI06
  This is used to input AMP0 output and as event input.
- AMP0
  This is used as a comparator.
- P70
  This charges a capacitor.
- P71
  This connects to a reference resistance and discharges a capacitor.
- P72
  This connects to the resistance to be measured and discharges a capacitor.
- LCD controller/driver
  This displays measured count values on the LCD on the board.

3.3 Initial Settings and Operation Overview

3.3.1 Non-inverting amplifier circuit

In this sample program, initialization sets the port settings, clock frequency selection, and the settings for timer array unit 0, the D/A converter, DMA controller, etc. After initialization is completed, sine waves will continue to be sent to the on-chip op amp from channel 4 of timer array unit 0, from the DMA controller, and from the D/A converter. The DMA controller is reset by the DMA transfer end interrupt, and sine wave output continues thereafter.

The details are described in the state transition diagram shown below.

Initial settings

<Option byte settings>
- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Settings during initialization immediately after a reset ends>
- Setting up I/O ports (all unused ports are set to low-level output)
- Setting CPU clock to X1 oscillator operation (20 MHz)
- Stopping the internal high-speed oscillator
- Setting up timer array unit 0
  - Setting channel 1 for interval operation at a 500 μs interval
  - Setting channel 4 for interval operation at a 10 μs interval and enabling counting
- Ensuring that the supply voltage is at least 2.7 V by using the low-voltage detector
- Setting up voltage reference
  - Setting the voltage reference output voltage to 2.0 V
  - Operation stabilization wait time (approximately 17 ms) for voltage reference that uses channel 1 of timer array unit 0
- Setting up op amp
  - Setting P20/AMP0–, P21/AMP0O, and P22/AMP0+ as analog input pins
  - Enabling operation of op amp 0
- Setting up D/A converter
  - Setting positive reference voltage as voltage reference output voltage
  - Setting P110/ANO0 for analog output
  - Setting operating mode of channel 0 to real-time output mode
  - Setting resolution of channel 0 to 12 bits
  - Enabling operation of channel 0
- Setting up DMA controller
  - Setting DMA start trigger for INTTM04 interrupt
  - Setting transfer data size to 16 bits and number of transfers to 100
  - Setting transfer source to RAM area and transfer destination to DACSW0 register
  - Enabling operation of DMA channel 0
- Enabling the INTDMA0 interrupt

↓

HALT mode

↑ ↓ INTDMA0 interrupt

Reset the DMA controller.

3.3.2 Inverting amplifier circuit

In this sample program, initialization sets the port settings, clock frequency selection, and the settings for timer array unit 0, the D/A converter, DMA controller, etc. After initialization is completed, sine waves will continue to be sent to the on-chip op amp from channel 4 of timer array unit 0, from the DMA controller, and from the D/A converter. The DMA controller is reset by the DMA transfer end interrupt, and sine wave output continues thereafter.

The details are described in the state transition diagram shown below.

Initial settings

<Option byte settings>
- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Settings during initialization immediately after a reset ends>
- Setting up I/O ports (all unused ports are set to low-level output)
- Setting CPU clock to X1 oscillator operation (20 MHz)
- Stopping the internal high-speed oscillator
- Setting up timer array unit 0
  - Setting channel 1 for interval operation at a 500 μs interval
  - Setting channel 4 for interval operation at a 10 μs interval and enabling counting
- Ensuring that the supply voltage is at least 2.7 V by using the low-voltage detector
- Setting up voltage reference
  - Setting the voltage reference output voltage to 2.0 V
  - Operation stabilization wait time (approximately 17 ms) for voltage reference that uses channel 1 of timer array unit 0
- Setting up op amp
  - Setting P20/AMP0–, P21/AMP0O, and P22/AMP0+ as analog input pins
  - Enabling operation of op amp 0
- Setting up D/A converter
  - Setting positive reference voltage as voltage reference output voltage
  - Setting P110/ANO0 and P111/ANO1 for analog output
  - Setting operating mode of channels 0 and 1 to normal mode
  - Setting resolution of channels 0 and 1 to 12 bits
  - Enabling operation of channels 0 and 1
- Setting up DMA controller
  - Setting DMA start trigger for INTTM04 interrupt
  - Setting transfer data size to 16 bits and number of transfers to 100
  - Setting transfer source to RAM area and transfer destination to DACSW0 register
  - Enabling operation of DMA channel 0
- Enabling the INTDMA0 interrupt

⬇

HALT mode

⬆ ⬇ INTDMA0 interrupt

Reset the DMA controller.

3.3.3 Offset adjustment of inverting amplifier circuit

In this sample program, initialization sets the port settings, clock frequency selection, and the settings for timer array unit 0, the D/A converter, the A/D converter, DMA controller, etc. After initialization is completed, sine waves will continue to be sent to the on-chip op amp from channel 4 of timer array unit 0, from the DMA controller, and from the D/A converter. When a DMA transfer end interrupt occurs, the ANI1 measurement value is checked as being 1 V, and the ANO1 output setting is adjusted to more closely approximate 1 V. The DMA controller is reset and sine wave output continues thereafter.

The details are described in the state transition diagram shown below.

Initial settings

<Option byte settings>
- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Settings during initialization immediately after a reset ends>
- Setting up I/O ports (all unused ports are set to low-level output)
- Setting CPU clock to X1 oscillator operation (20 MHz)
- Stopping the internal high-speed oscillator
- Setting up timer array unit 0
  - Setting channel 1 for interval operation at a 500 μs interval
  - Setting channel 4 for interval operation at a 10 μs interval and enabling counting
- Ensuring that the supply voltage is at least 2.7 V by using the low-voltage detector
- Setting up voltage reference
  - Setting the voltage reference output voltage to 2.0 V
  - Operation stabilization wait time (approximately 17 ms) for voltage reference that uses channel 1 of timer array unit 0
- Setting up op amp
  - Setting P20/AMP0–, P21/AMP0O, and P22/AMP0+ as analog input pins
  - Enabling operation of op amp 0
- Setting up D/A converter
  - Setting positive reference voltage as voltage reference output voltage
  - Setting P110/ANO0 and P111/ANO1 for analog output
  - Setting operating mode of channel 0 to real-time output mode
  - Setting operating mode of channel 1 to normal mode
  - Setting resolution of channels 0 and 1 to 12 bits
  - Enabling operation of channels 0 and 1
- Setting up DMA controller
  - Setting DMA start trigger for INTTM04 interrupt
  - Setting transfer data size to 16 bits and number of transfers to 100
  - Setting transfer source to RAM area and transfer destination to DACSW0 register
  - Enabling operation of DMA channel 0
- Enabling the INTDMA0 interrupt

↓

HALT mode

↑ ↓ INTDMA0 interrupt

- Use the A/D converter to adjust the ANO1 output.
- Reset the DMA controller.

3.3.4 Offset adjustment of inverting amplifier circuit + low-pass filter

In this sample program, initialization sets the port settings, clock frequency selection, and the settings for timer array unit 0, the D/A converter, the A/D converter, DMA controller, etc. After initialization is completed, sine waves will continue to be sent to the on-chip op amp from channel 4 of timer array unit 0, from the DMA controller, and from the D/A converter. When a DMA transfer end interrupt occurs, the ANI1 measurement value is checked as being 1 V, and the ANO1 output setting is adjusted to more closely approximate 1 V. The D/A converter output signal frequency is changed by counting the number of interrupts and periodically updating the channel 4 interval time of timer array unit 0. The DMA controller is reset and sine wave output continues thereafter.

The details are described in the state transition diagram shown below.

Initial settings

<Option byte settings>
- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Settings during initialization immediately after a reset ends>
- Setting up I/O ports (all unused ports are set to low-level output)
- Setting CPU clock to X1 oscillator operation (20 MHz)
- Stopping the internal high-speed oscillator
- Setting up timer array unit 0
  - Setting channel 1 for interval operation at a 500 μs interval
  - Setting channel 4 for interval operation at a 10 μs interval and enabling counting
- Ensuring that the supply voltage is at least 2.7 V by using the low-voltage detector
- Setting up voltage reference
  - Setting the voltage reference output voltage to 2.0 V
  - Operation stabilization wait time (approximately 17 ms) for voltage reference that uses channel 1 of timer array unit 0
- Setting up op amp
  - Setting P20/AMP0−, P21/AMP0O, and P22/AMP0+ as analog input pins
  - Enabling operation of op amp 0
- Setting up D/A converter
  - Setting positive reference voltage as voltage reference output voltage
  - Setting P110/ANO0 and P111/ANO1 for analog output
  - Setting operating mode of channel 0 to real-time output mode
  - Setting operating mode of channel 1 to normal mode
  - Setting resolution of channels 0 and 1 to 12 bits
  - Enabling operation of channels 0 and 1
- Setting up DMA controller
  - Setting DMA start trigger for INTTM04 interrupt
  - Setting transfer data size to 16 bits and number of transfers to 100
  - Setting transfer source to RAM area and transfer destination to DACSW0 register
  - Enabling operation of DMA channel 0
- Enabling the INTDMA0 interrupt

↓

HALT mode

↑ ↓ INTDMA0 interrupt

- Use the A/D converter to adjust the ANO1 output.
- Reset the DMA controller.
- Periodically, update the channel 4 count value of timer array unit 0 and change the frequency of the ANO0 output.

3.3.5 Non-inverting amplifier circuit (AC)

In this sample program, initialization sets the port settings, clock frequency selection, and the settings for timer array unit 0, the D/A converter, DMA controller, etc. After initialization is completed, alternating waveforms will continue to be sent to the on-chip op amp from channel 4 of timer array unit 0, from the DMA controller, and from the D/A converter. The number of DMA transfers set by DMA transfer end interrupts is adjusted and the DMA controller is reset so that the output waveform "(sin 0 ~ sin 1/2 • π) → (constant voltage) → (sin 1/2 • π ~ sin 1.5 π) → (constant voltage) → (sin 1.5 π ~ sin 2 π) → (slow variation voltage)" is repeated.

The details are described in the state transition diagram shown below.

Initial settings

<Option byte settings>

- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Settings during initialization immediately after a reset ends>

- Setting up I/O ports (all unused ports are set to low-level output)
- Setting CPU clock to X1 oscillator operation (20 MHz)
- Stopping the internal high-speed oscillator
- Setting up timer array unit 0
  - Setting channel 1 for interval operation at a 500 μs interval
  - Setting channel 4 for interval operation at a 100 μs interval and enabling counting
- Ensuring that the supply voltage is at least 2.7 V by using the low-voltage detector
- Setting up voltage reference
  - Setting the voltage reference output voltage to 2.0 V
  - Operation stabilization wait time (approximately 17 ms) for voltage reference that uses channel 1 of timer array unit 0
- Setting up op amp
  - Setting P20/AMP0–, P21/AMP0O, and P22/AMP0+ as analog input pins
  - Enabling operation of op amp 0
- Setting up D/A converter
  - Setting positive reference voltage as voltage reference output voltage
  - Setting P110/ANO0 for analog output
  - Setting operating mode of channel 0 to real-time output mode
  - Setting resolution of channel 0 to 12 bits
  - Enabling operation of channel 0
- Setting up DMA controller
  - Setting DMA start trigger for INTTM04 interrupt
  - Setting transfer data size to 16 bits and number of transfers to 25
  - Setting transfer source to RAM area and transfer destination to DACSW0 register
  - Enabling operation of DMA channel 0
- Enabling the INTDMA0 interrupt

↓

HALT mode

↑ ↓ INTDMA0 interrupt

- Update the DMA controller transfer count and the channel 4 count of timer array unit 0 and change the frequency of the ANO0 output.
- Reset the DMA controller.

3.3.6 Inverting amplifier circuit (AC)

In this sample program, initialization sets the port settings, clock frequency selection, and the settings for timer array unit 0, the D/A converter, DMA controller, etc. After initialization is completed, alternating waveforms will continue to be sent to the on-chip op amp from channel 4 of timer array unit 0, from the DMA controller, and from the D/A converter. The number of DMA transfers set by DMA transfer end interrupts is adjusted and the DMA controller is reset so that the output waveform "(sin 0 ~ sin 1/2 • π) → (constant voltage) → (sin 1/2 • π ~ sin π) → (slow variation voltage)" is repeated. D/A converter 1 outputs the bias voltage.

The details are described in the state transition diagram shown below.

Initial settings

<Option byte settings>
- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Settings during initialization immediately after a reset ends>
- Setting up I/O ports (all unused ports are set to low-level output)
- Setting CPU clock to X1 oscillator operation (20 MHz)
- Stopping the internal high-speed oscillator
- Setting up timer array unit 0
  - Setting channel 1 for interval operation at a 500 μs interval
  - Setting channel 4 for interval operation at a 100 μs interval and enabling counting
- Ensuring that the supply voltage is at least 2.7 V by using the low-voltage detector
- Setting up voltage reference
  - Setting the voltage reference output voltage to 2.0 V
  - Operation stabilization wait time (approximately 17 ms) for voltage reference that uses channel 1 of timer array unit 0
- Setting up op amp
  - Setting P20/AMP0–, P21/AMP0O, and P22/AMP0+ as analog input pins
  - Enabling operation of op amp 0
- Setting up D/A converter
  - Setting positive reference voltage as voltage reference output voltage
  - Setting P110/ANO0 and P111/ANO1 for analog output
  - Setting operating mode of channel 0 to real-time output mode
  - Setting operating mode of channel 1 to normal mode
  - Setting resolution of channels 0 and 1 to 12 bits
  - Enabling operation of channels 0 and 1
- Setting up DMA controller
  - Setting DMA start trigger for INTTM04 interrupt
  - Setting transfer data size to 16 bits and number of transfers to 25
  - Setting transfer source to RAM area and transfer destination to DACSW0 register
  - Enabling operation of DMA channel 0
- Enabling the INTDMA0 interrupt

↓

HALT mode

↑ ↓ INTDMA0 interrupt

- Update the DMA controller transfer count and the channel 4 count of timer array unit 0 and change the frequency of the ANO0 output.
- Reset the DMA controller.

3.3.7 Instrumentation amplifier

In this sample program, initialization sets the port settings, clock frequency selection, and the settings for the D/A converter, etc. After initialization is completed, the CPU clock is switched to the subsystem clock. A constant voltage is sent to the on-chip op amp from the D/A converter. Once per second, an RTC periodic interrupt causes the ANO0 output voltage to change. ANO1 outputs the bias voltage.

The details are described in the state transition diagram shown below.

Initial settings

<Option byte settings>

- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Settings during initialization immediately after a reset ends>

- Setting up I/O ports (all unused ports are set to low-level output)
- Setting CPU clock to XT1 oscillator operation (32.768 kHz)
- Stopping the internal high-speed oscillator
- Ensuring that the supply voltage is at least 2.7 V by using the low-voltage detector
- Setting up real-time counter
  - Oscillation stabilization wait time for subsystem clock that uses channel 1 of timer array unit 0
  - Setting of periodic interrupts (1 second cycle)
  - Enabling operation of real-time counter
- Setting up voltage reference
  - Setting the voltage reference output voltage to 2.0 V
  - Operation stabilization wait time (approximately 17 ms) for voltage reference that uses channel 1 of the timer array unit
- Setting up op amp
  - Setting P20/AMP0–, P21/AMP0O, P22/AMP0+, P23/AMP1–, P24/AMP1O, P25/AMP1+, P26/AMP2–, P27/AMP2O, and P150/AMP2+ as analog input pins
  - Enabling operation of op amp 0, op amp 1, and op amp 2
- Setting up D/A converter
  - Setting positive reference voltage as voltage reference output voltage
  - Setting P110/ANO0 and P111/ANO1 for analog output
  - Setting operating mode of channels 0 and 1 to normal mode
  - Setting resolution of channels 0 and 1 to 12 bits
  - Setting ANO0 output to 0.52 V
  - Setting ANO1 output to 0.50 V
  - Enabling operation of channels 0 and 1
- Enabling the INTRTC interrupt

↓

HALT mode

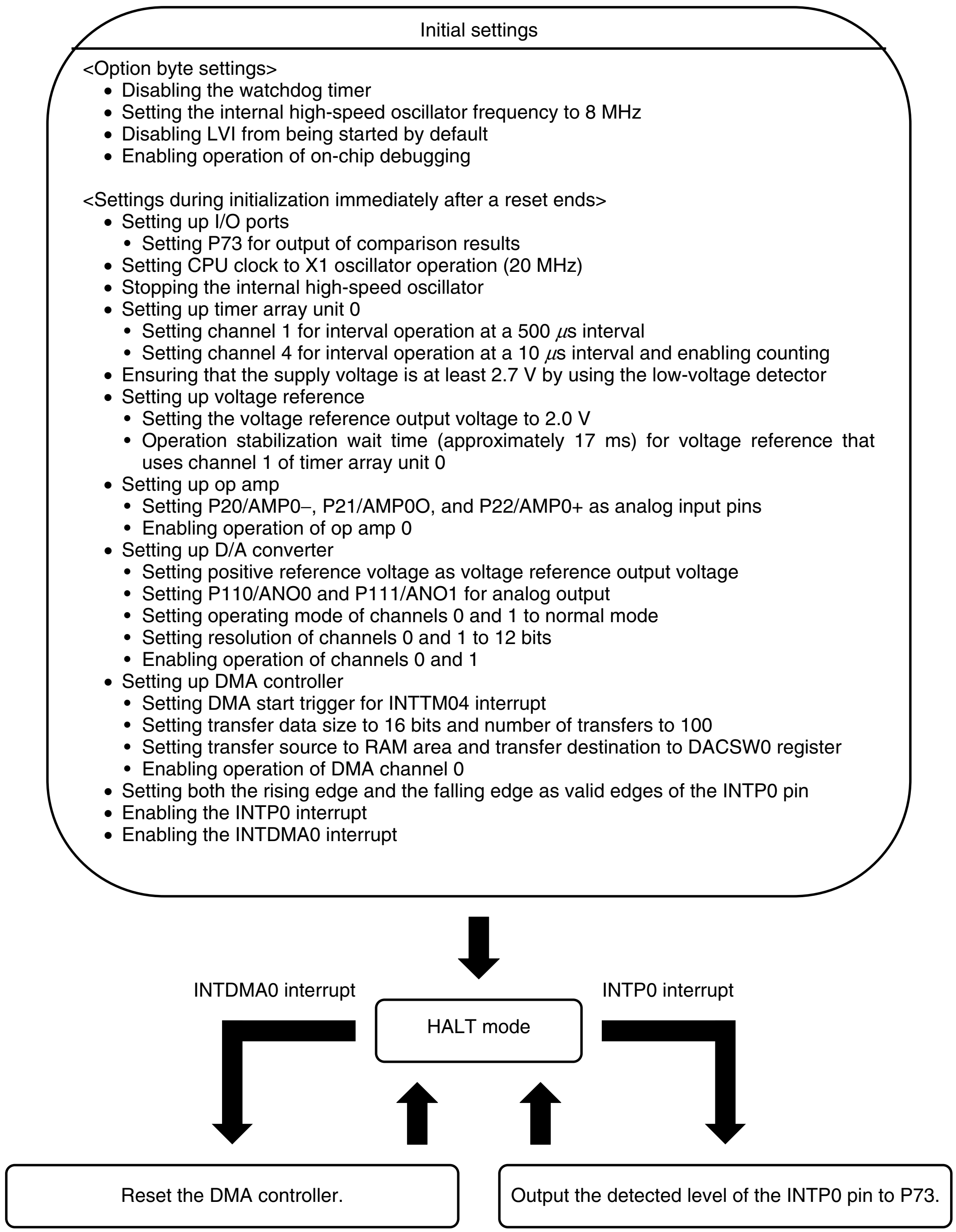
↑ ↓ INTRTC interrupt (1 second cycle)

Change the ANO0 output alternately between 0.52 V and 0.54 V.

3.3.8 Comparator circuit

In this sample program, initialization sets the port settings, clock frequency selection, and the settings for timer array unit 0, the D/A converter, DMA controller, etc. After initialization is completed, sine waves will continue to be sent to the on-chip op amp inverting input from channel 4 of timer array unit 0, from the DMA controller, and from the D/A converter. The DMA controller is reset by the DMA transfer end interrupt, and sine wave output continues thereafter. Output from the op amp is input to INTP0. INTP0 is used for dual-edge triggered interrupts, and when servicing an INTP0 interrupt, the output level of the op amp is output to PORT7.3 without modification. ANO1 outputs the comparison reference voltage that is input to a non-inverting input pin of the op amp.

The details are described in the state transition diagram shown below.



3.3.9 CR oscillator 1

In this sample program, initialization sets the port settings, clock frequency selection, and the settings for timer array unit 0, the LCD controller/driver, etc. After initialization is completed, channel 4 of timer array unit 0 is used to time the period (1 ms) for counting pulses output to TI06 in interval timer mode. Pulses input to TI06 by channel 6 of timer array unit 0 are counted in event count mode. ANO1 outputs the CR oscillation central voltage that is input to a non-inverting input pin of the op amp. Each time a timer array unit 0 channel 4 interrupt request occurs, the timer array unit 0 channel 6 count value is read and shown as a decimal display on the LCD.

The details are described in the state transition diagram shown below.

Initial settings

<Option byte settings>

- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Settings during initialization immediately after a reset ends>

- Setting up I/O ports (all unused ports are set to low-level output)
- Setting CPU clock to X1 oscillator operation (20 MHz)
- Stopping the internal high-speed oscillator
- Setting up timer array unit 0
  - Setting channel 1 for interval operation at a 500 μs interval
  - Setting channel 4 for interval operation at a 1 ms interval and enabling operation start
  - Setting channel 6 for input pulse interval measurement and enabling operation start
- Ensuring that the supply voltage is at least 2.7 V by using the low-voltage detector
- Setting up op amp
  - Setting P20/AMP0–, P21/AMP0O, and P22/AMP0+ as analog input pins
  - Enabling operation of op amp 0
- Setting up D/A converter
  - Setting positive reference voltage source to AV_{DD1} pin
  - Setting P111/ANO1 for analog output
  - Setting operating mode of channel 1 to normal mode
  - Setting resolution of channel 1 to 12 bits
  - Setting ANO1 output to 1.5 V
  - Enabling operation of channel 1
- Setting up LCD controller/driver
  - Setting SEG4 to SEG53 for segment output
  - Selecting capacitance splitting method for LCD drive voltage generator
  - Setting LCD mode for eight time divisions and 1/4 bias
  - Setting data in A pattern area as display data
  - Setting LCD source clock (f_{LCD}) to f_{SUB}, and LCD clock to $f_{LCD}/2^7$
  - Setting LCD drive voltage (V_{LC0}) to 5.0 V
  - Enabling LCD display

↓

Wait for interrupt.

↑ ↓ INTTM04 interrupt (approximately 1 ms cycle)

Measure the AMP0 output frequency and output the measurement results to the LCD.

3.3.10 CR oscillator 2

In this sample program, initialization sets the port settings, clock frequency selection, and the settings for timer array unit 0, the LCD controller/driver, etc. The op amp is used as a comparator. After initialization is completed, the comparison reference voltage 1 V is output to DAO1, which is connected to a non-inverting input pin of the op amp. P70 is set to high-level output and the capacitor connected to the inverting input pin of the op amp is charged. After waiting for a specified period (wait for charge completion), P70 is switched to input mode so that P70 becomes electrically detached. P71 connected to a reference voltage is set for low output (to start discharging the capacitor), while counting is enabled for channel 6 of timer array unit 0, which has been set to capture mode with the falling edge of TI06 as the trigger. The channel 6 count value (discharge time) of timer array unit 0 is captured each time a capture interrupt (discharge completion) occurs for timer channel 6. Next, comparison reference voltage 2 V is output to DAO1. The discharge time is captured via the same steps as after 1 V is output to DAO1. The discharge time is measured while 1 V and 2 V are output to DAO1 with the resistance to be measured, which is connected to P72 via the same steps as with P71. The count value differential between 1 V and 2 V output to DAO1 of P71 and the count value differential between 1 V and 2 V output to DAO1 of P72 are shown as a decimal display on the LCD.

The details are described in the state transition diagram shown below.

Initial settings

<Option byte settings>

- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Settings during initialization immediately after a reset ends>

- Setting up I/O ports (all unused ports are set to low-level output)
- Setting CPU clock to X1 oscillator operation (20 MHz)
- Stopping the internal high-speed oscillator
- Setting up timer array unit 0
  - Setting channel 1 for interval operation at a 500 μs interval
  - Setting channel 6 for input pulse interval measurement and enabling operation start
- Ensuring that the supply voltage is at least 2.7 V by using the low-voltage detector
- Setting up op amp
  - Setting P20/AMP0–, P21/AMP0O, and P22/AMP0+ as analog input pins
  - Enabling operation of op amp 0
- Setting up D/A converter
  - Setting positive reference voltage source to AV_{DD1} pin
  - Setting P111/ANO1 for analog output
  - Setting operating mode of channel 1 to normal mode
  - Setting resolution of channel 1 to 12 bits
  - Setting ANO1 output to 1.0 V
  - Enabling operation of channel 1
- Setting up LCD controller/driver
  - Setting SEG4 to SEG53 for segment output
  - Selecting capacitance splitting method for LCD drive voltage generator
  - Setting LCD mode for eight time divisions and 1/4 bias
  - Setting data in A pattern area as display data
  - Setting LCD source clock (f_{LCD}) to f_{SUB}, and LCD clock to $f_{LCD}/2^7$
  - Setting LCD drive voltage (V_{LC0}) to 5.0 V
  - Enabling LCD display

↓

Charge the capacitor by high-level output of port.

→

P71, P72, and DAC1 output changes

	P71	P72	DAC1 Output
1	Output	Input	1.0 V
2	Output	Input	2.0 V
3	Input	Output	1.0 V
4	Input	Output	2.0 V

←

- Discharge the capacitor.
- Measure the discharge time by channel 6 of timer array unit 0.
- Output the discharge time to the LCD.

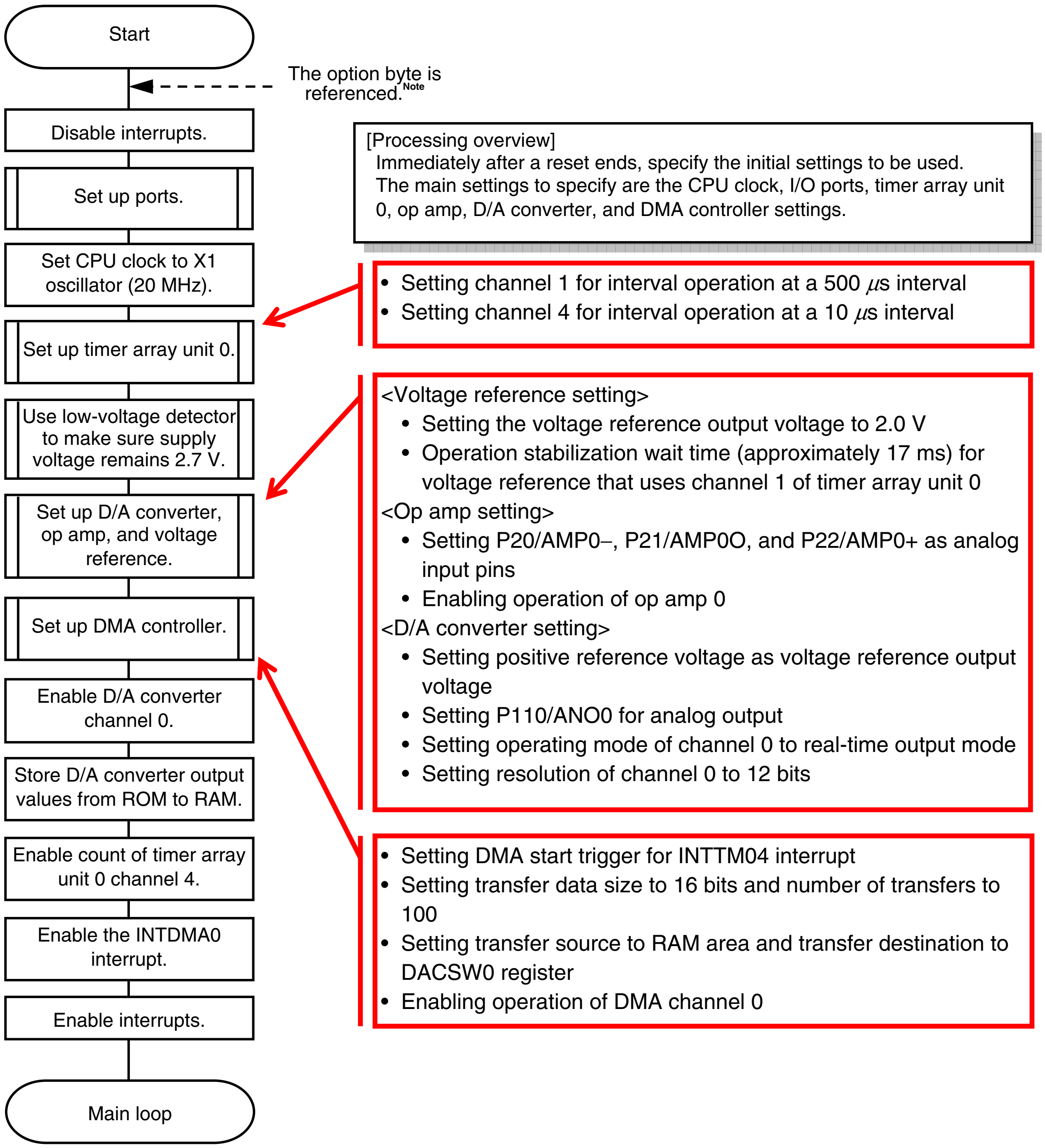
↑ (back to "Charge the capacitor by high-level output of port.")

3.4 Flow Charts

The flow charts for the sample program are shown below.

3.4.1 Non-inverting amplifier circuit

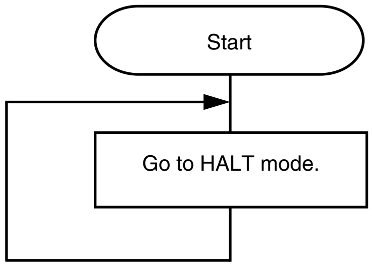
<Settings during initialization immediately after a reset ends>



Note The option byte is automatically referenced by the microcontroller immediately after a reset ends. In this sample program, the following settings are specified using the option byte:

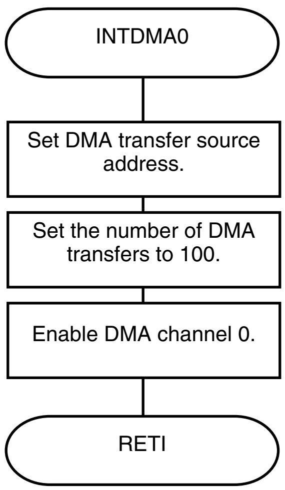
- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Main loop>



[Processing overview]
Use HALT mode and wait for INTDMA0 interrupts.

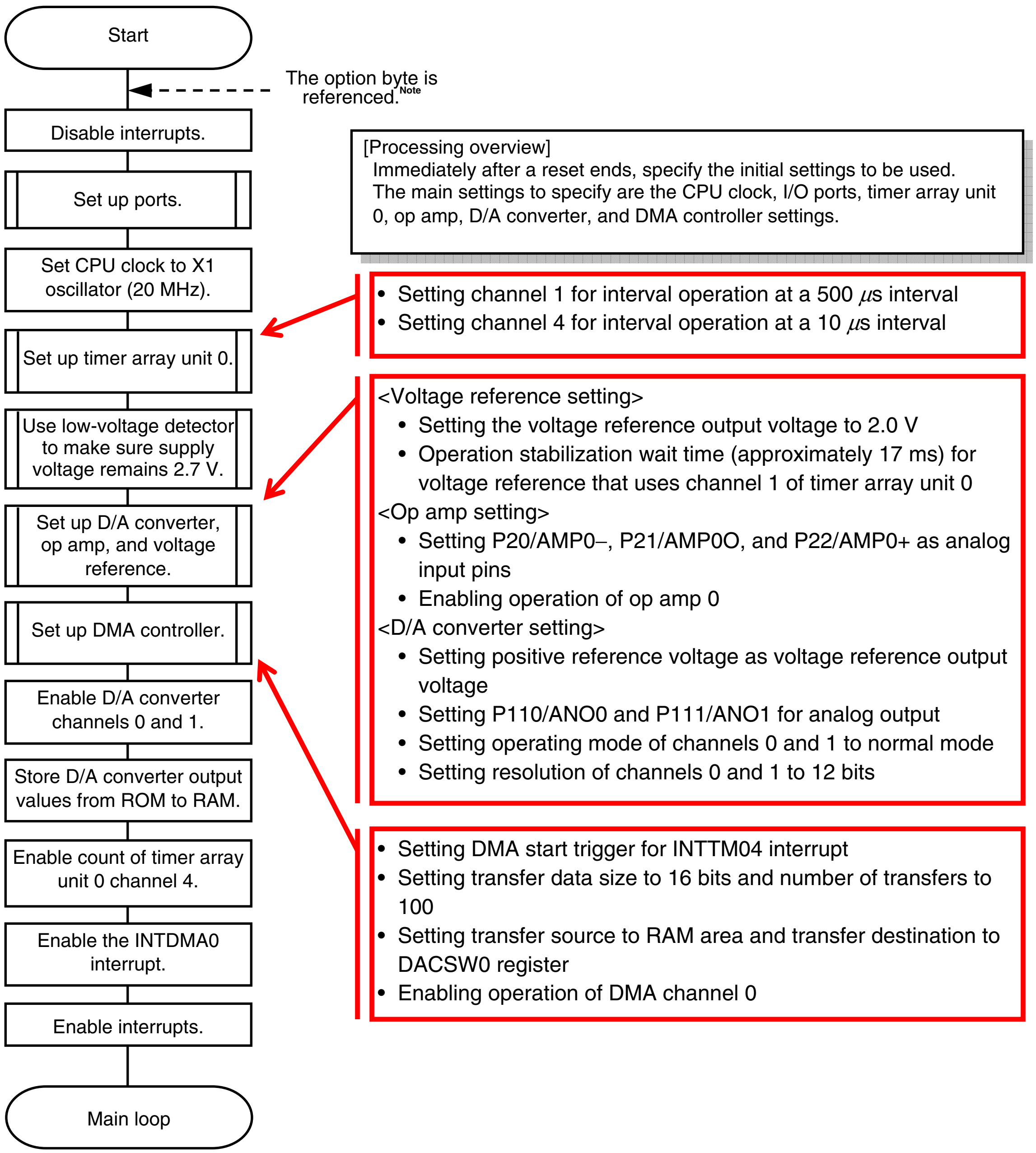
<INTDMA0 interrupt servicing>



[Processing overview]
Reset DMA channel 0.

3.4.2 Inverting amplifier circuit

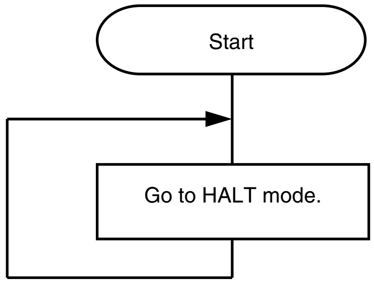
<Settings during initialization immediately after a reset ends>



Note The option byte is automatically referenced by the microcontroller immediately after a reset ends. In this sample program, the following settings are specified using the option byte:

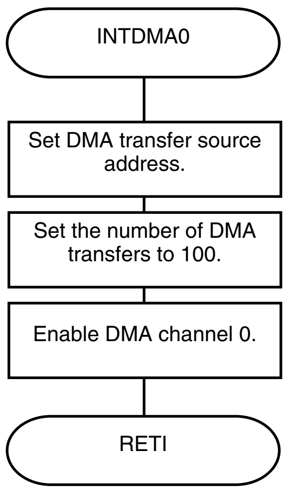
- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Main loop>



[Processing overview]
Use HALT mode and wait for INTDMA0 interrupts.

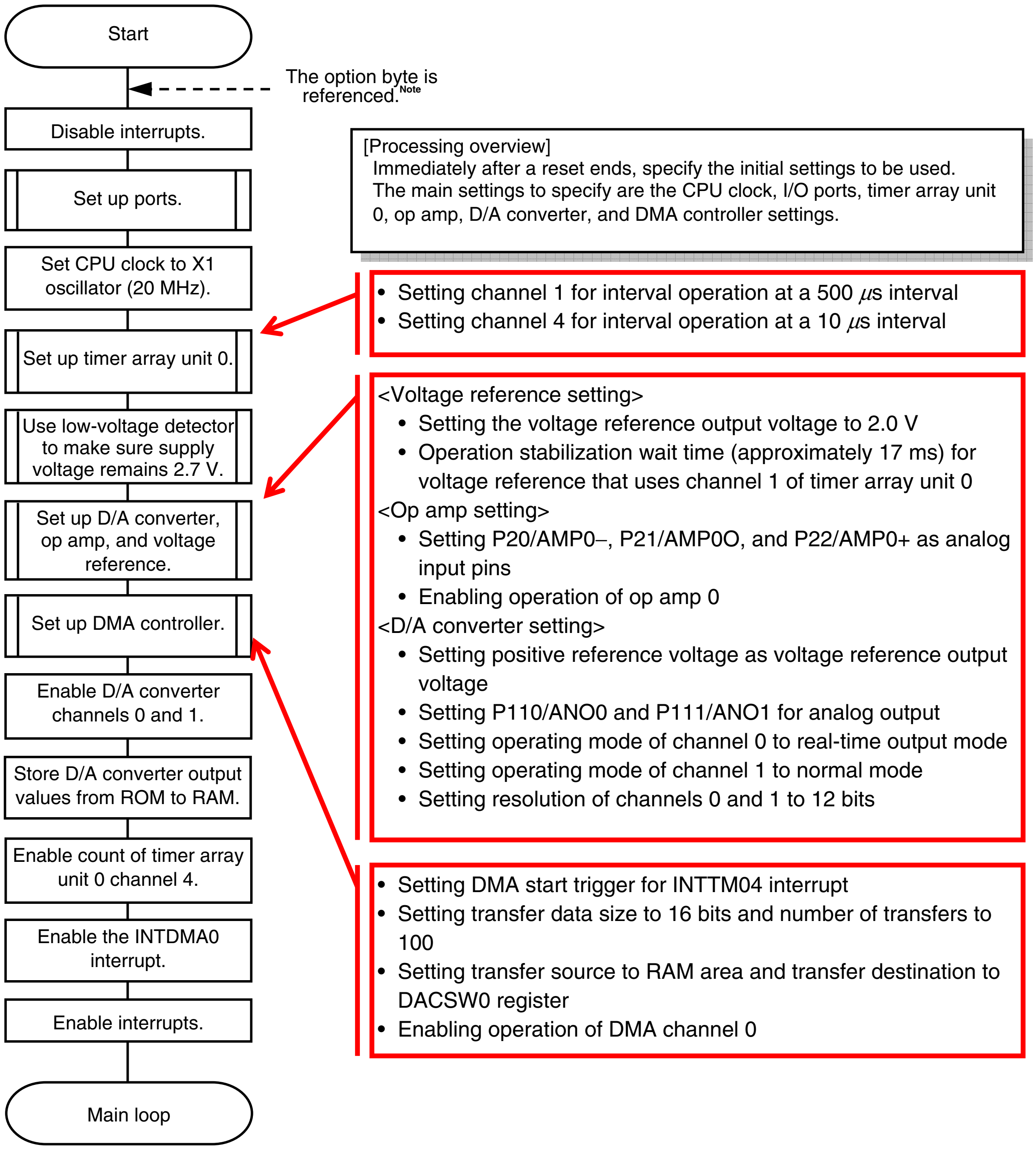
<INTDMA0 interrupt servicing>



[Processing overview]
Reset DMA channel 0.

3.4.3 Offset adjustment of inverting amplifier circuit

<Settings during initialization immediately after a reset ends>

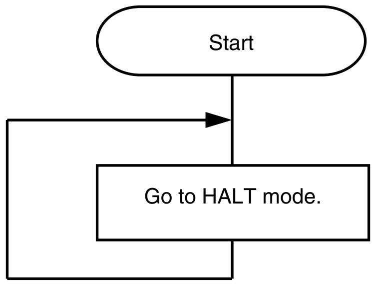


Note The option byte is automatically referenced by the microcontroller immediately after a reset ends. In this sample program, the following settings are specified using the option byte:

- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

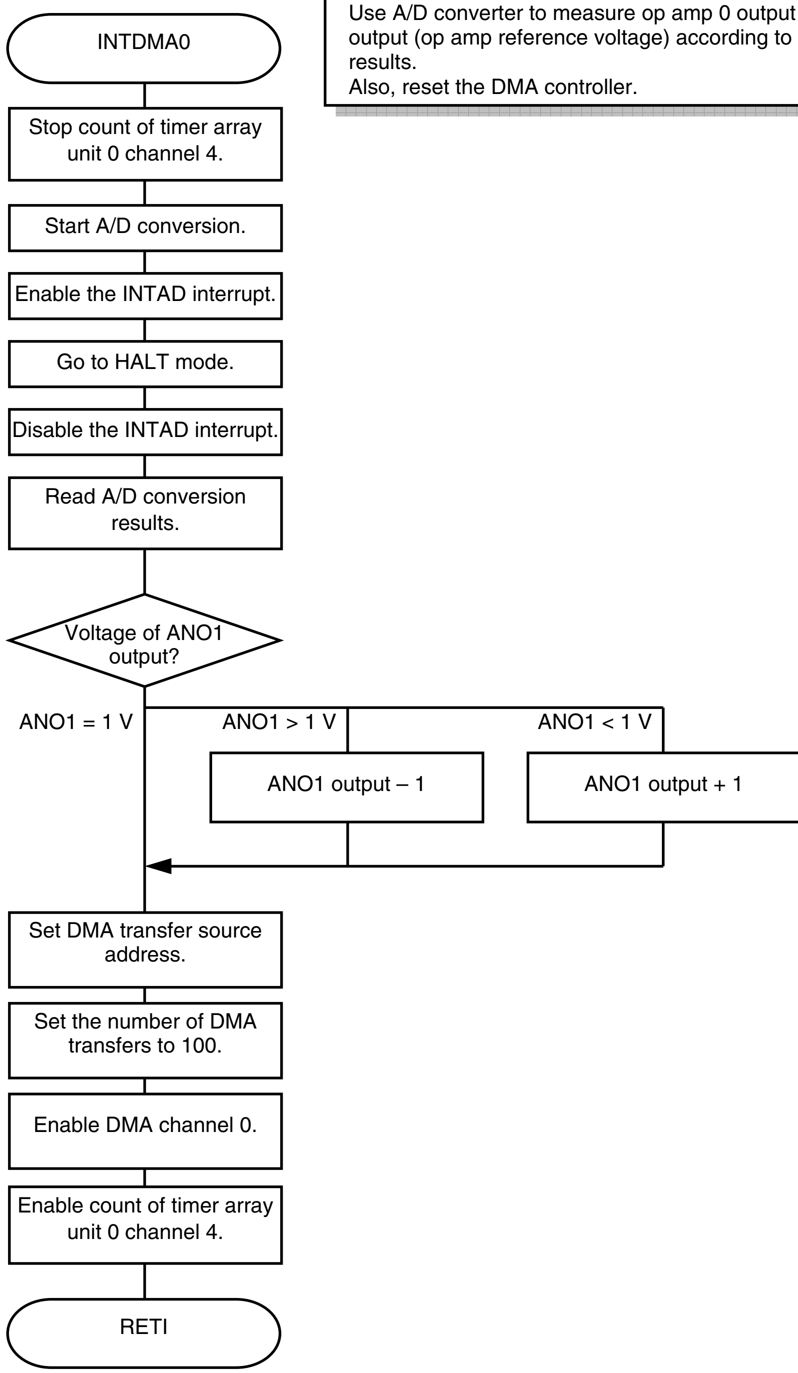
<Main loop>

[Processing overview]
Use HALT mode and wait for INTDMA0 interrupts.



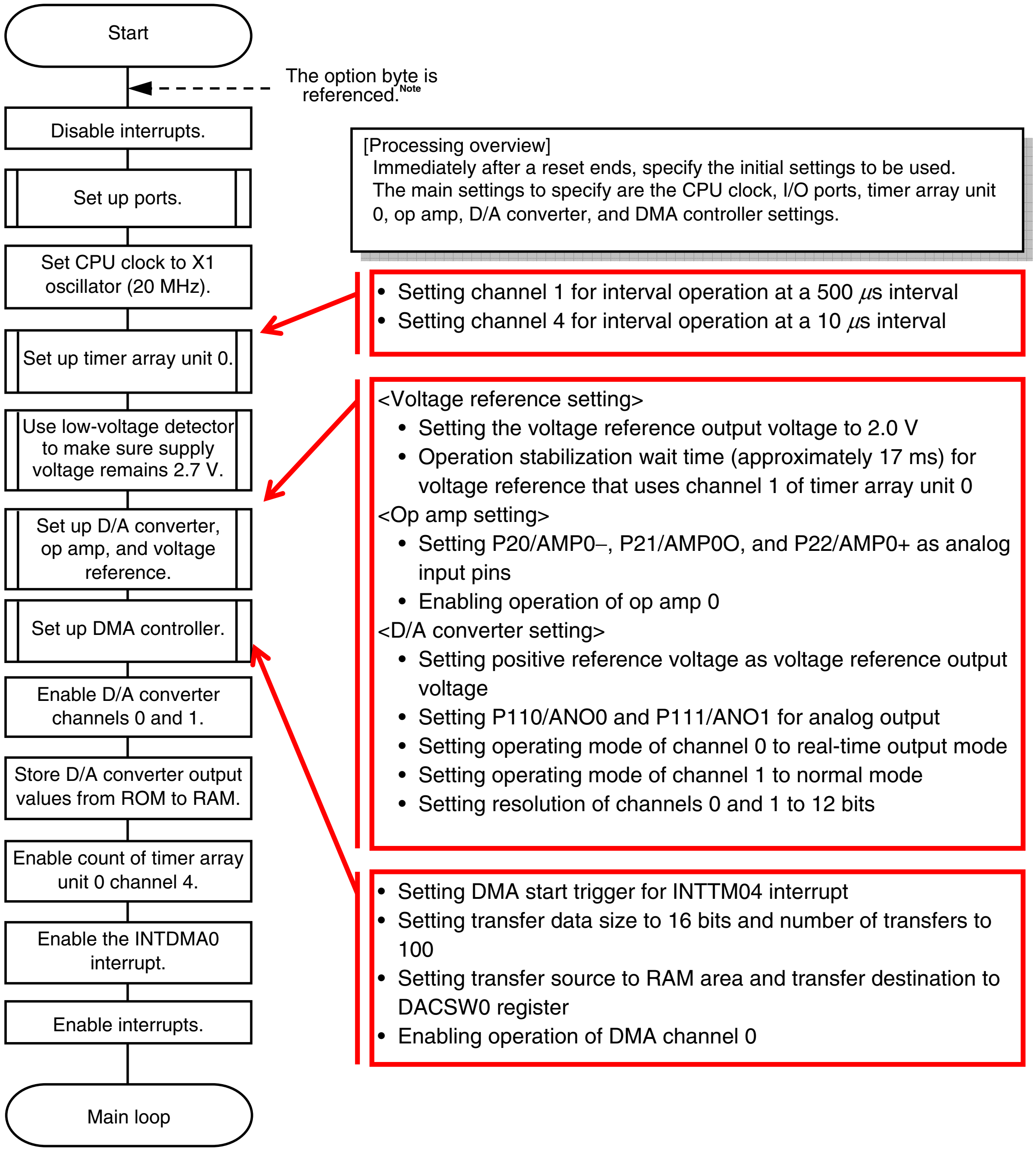
<INTDMA0 interrupt servicing>

[Processing overview]
Use A/D converter to measure op amp 0 output and adjust ANO1 output (op amp reference voltage) according to the measurement results.
Also, reset the DMA controller.



3.4.4 Offset adjustment of inverting amplifier circuit + low-pass filter

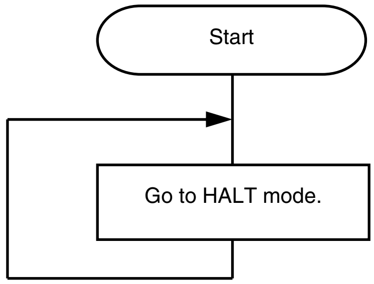
<Settings during initialization immediately after a reset ends>



Note The option byte is automatically referenced by the microcontroller immediately after a reset ends. In this sample program, the following settings are specified using the option byte:

- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

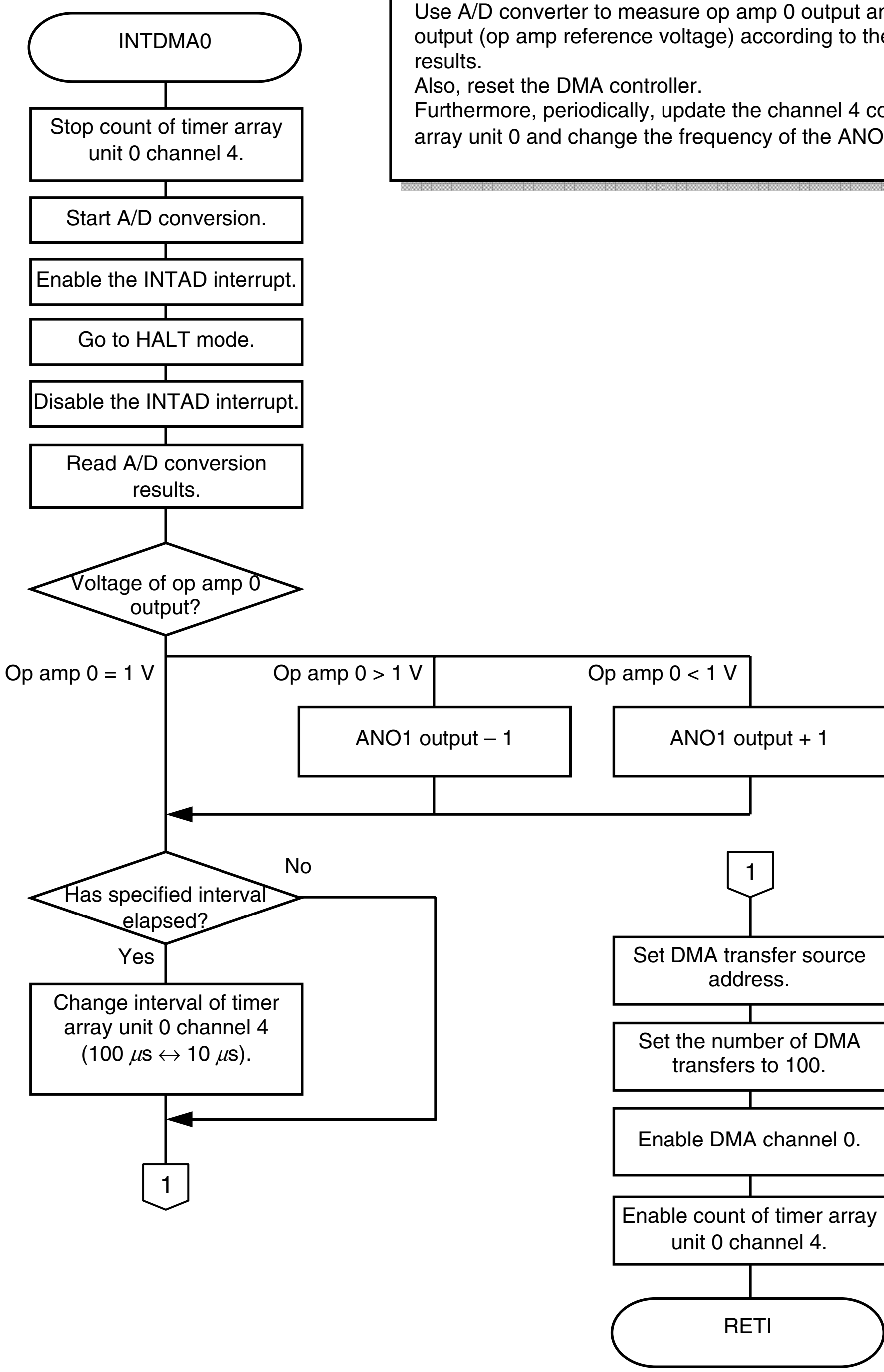
<Main loop>



[Processing overview]
Use HALT mode and wait for INTDMA0 interrupts.

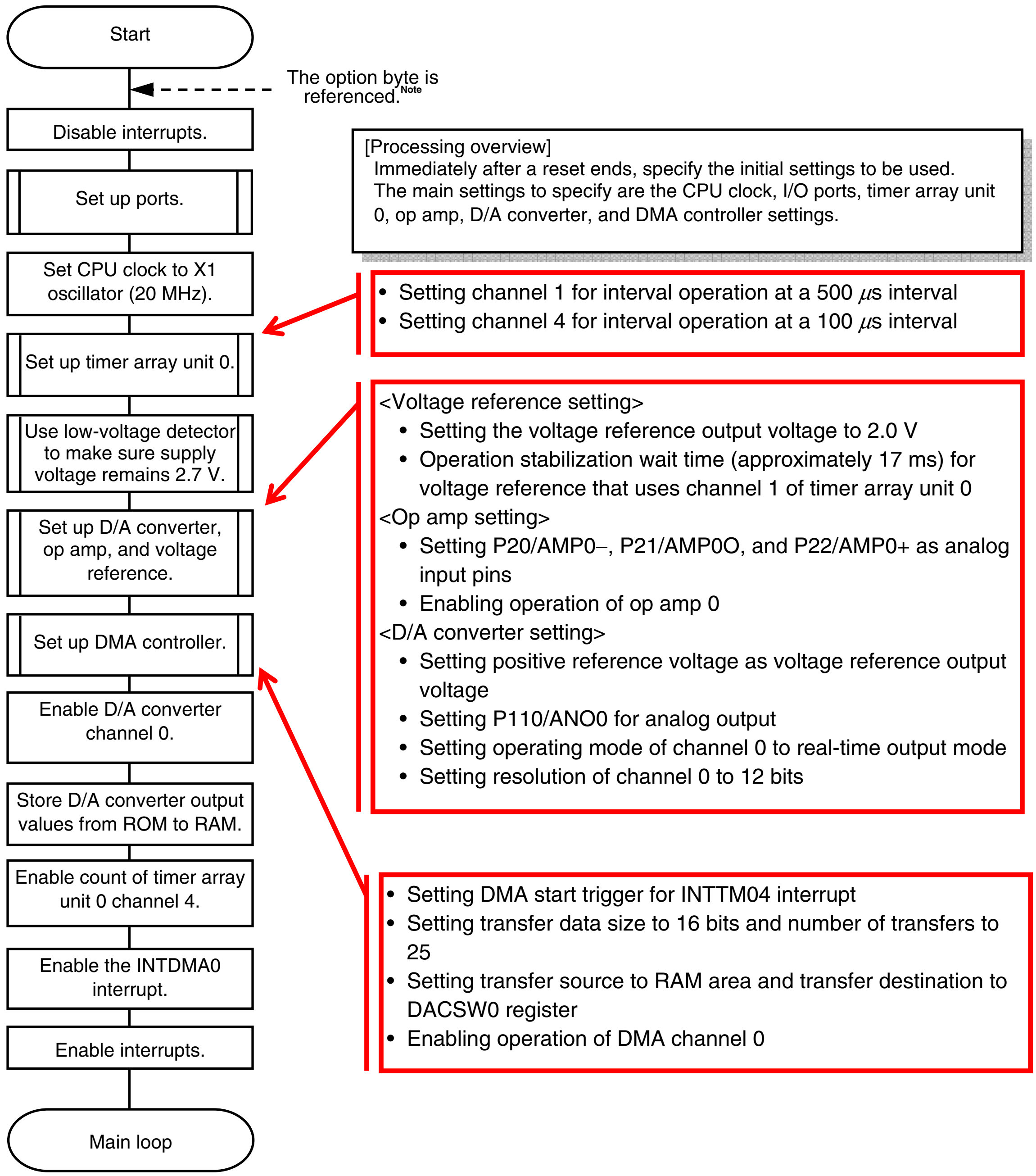
<INTDMA0 interrupt servicing>

[Processing overview]
Use A/D converter to measure op amp 0 output and adjust ANO1 output (op amp reference voltage) according to the measurement results.
Also, reset the DMA controller.
Furthermore, periodically, update the channel 4 count value of timer array unit 0 and change the frequency of the ANO0 output.



3.4.5 Non-inverting amplifier circuit (AC)

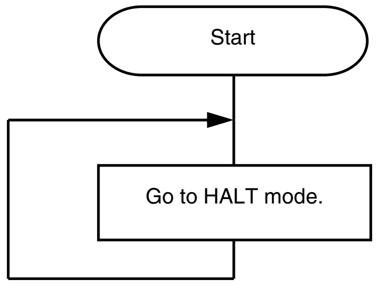
<Settings during initialization immediately after a reset ends>



Note The option byte is automatically referenced by the microcontroller immediately after a reset ends. In this sample program, the following settings are specified using the option byte:

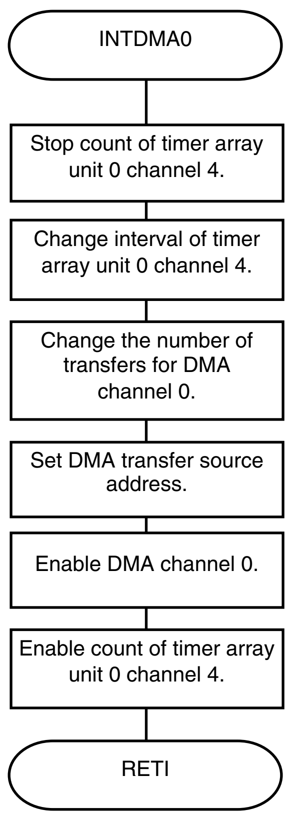
- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Main loop>



[Processing overview]
Use HALT mode and wait for INTDMA0 interrupts.

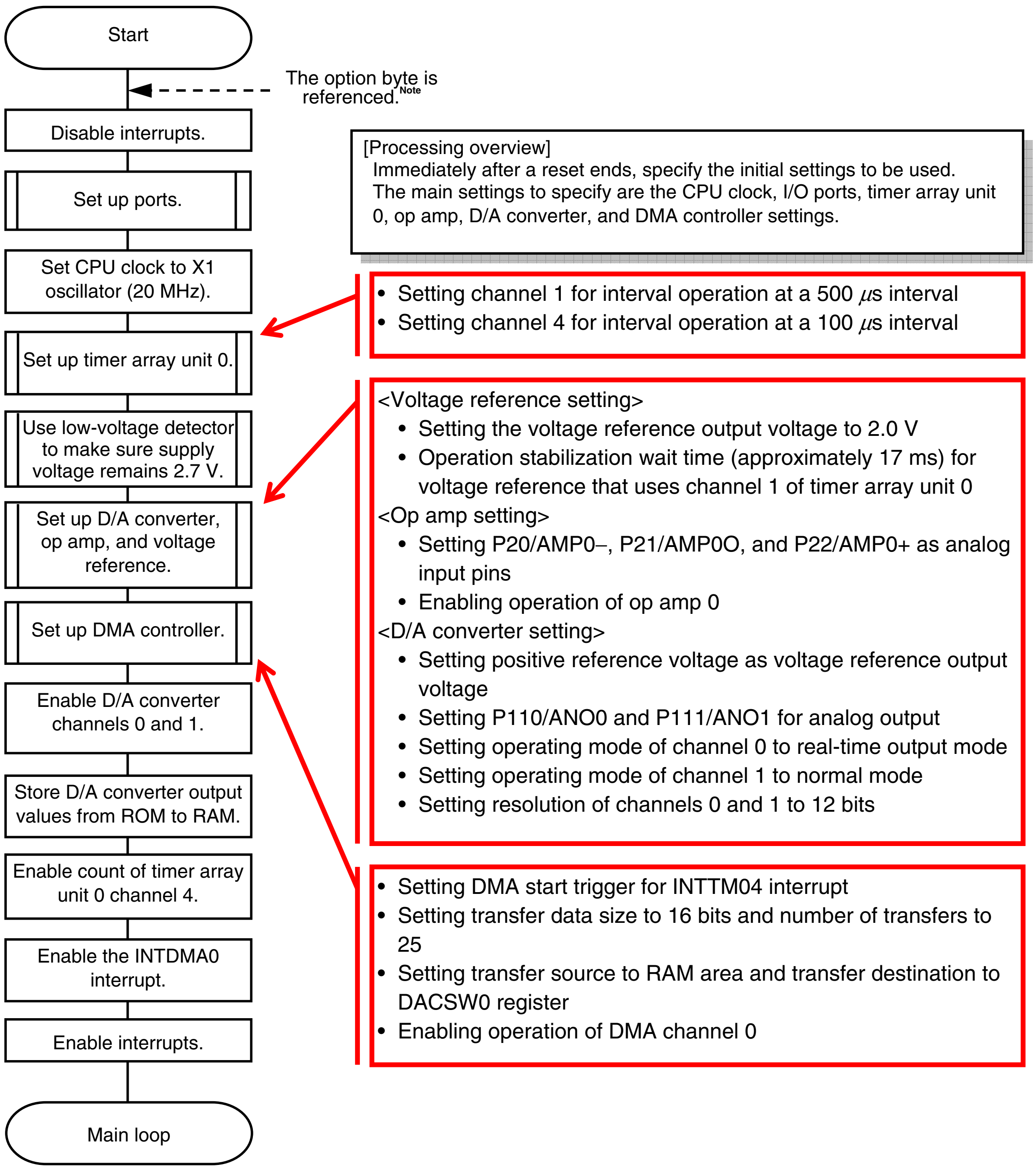
<INTDMA0 interrupt servicing>



[Processing overview]
This changes the interval of timer array unit 0 channel 4 and the number of transfers for DMA channel 0, which switches the ANO0 output transition timing (switching between short and long transition intervals).
Also, reset the DMA controller.

3.4.6 Inverting amplifier circuit (AC)

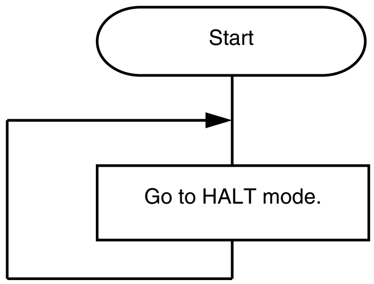
<Settings during initialization immediately after a reset ends>



Note The option byte is automatically referenced by the microcontroller immediately after a reset ends. In this sample program, the following settings are specified using the option byte:

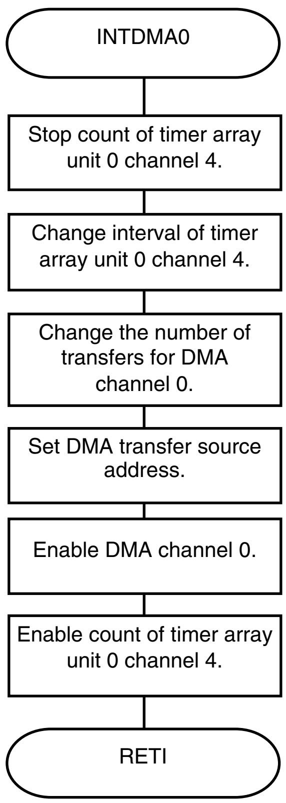
- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Main loop>



[Processing overview]
Use HALT mode and wait for INTDMA0 interrupts.

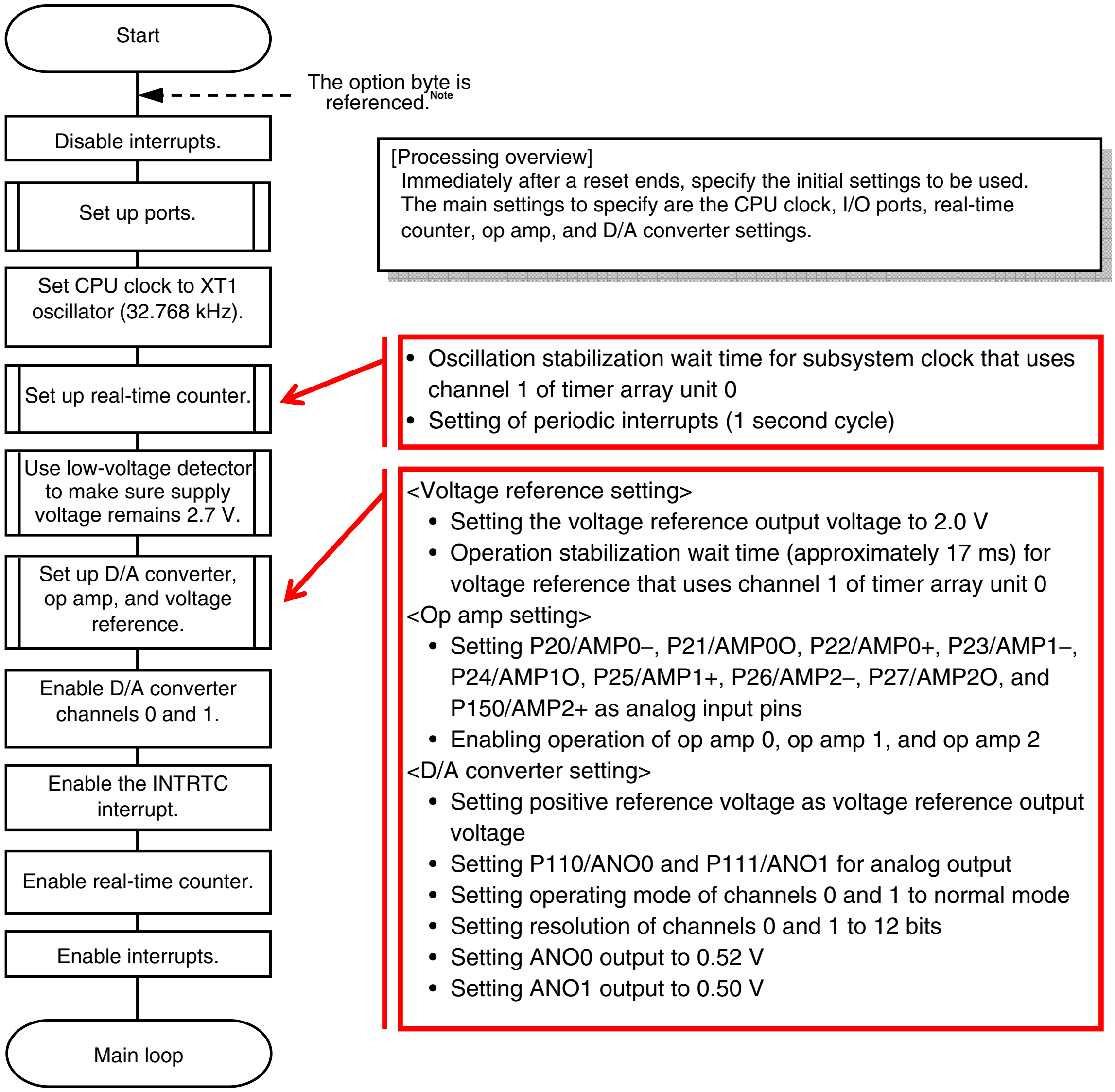
<INTDMA0 interrupt servicing>



[Processing overview]
This changes the interval of timer array unit 0 channel 4 and the number of transfers for DMA channel 0, which switches the ANO0 output transition timing (switching between short and long transition intervals).
Also, reset the DMA controller.

3.4.7 Instrumentation amplifier

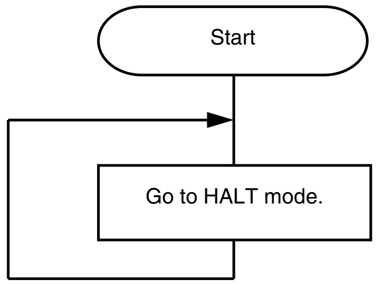
<Settings during initialization immediately after a reset ends>



Note The option byte is automatically referenced by the microcontroller immediately after a reset ends. In this sample program, the following settings are specified using the option byte:

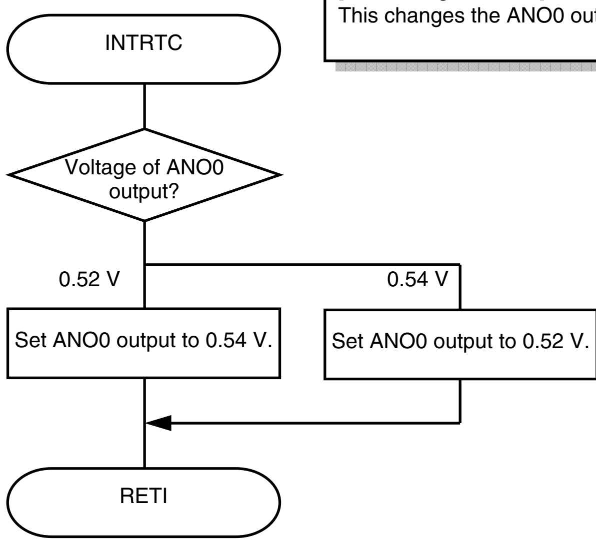
- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Main loop>



[Processing overview]
Use HALT mode and wait for INTRTC interrupts.

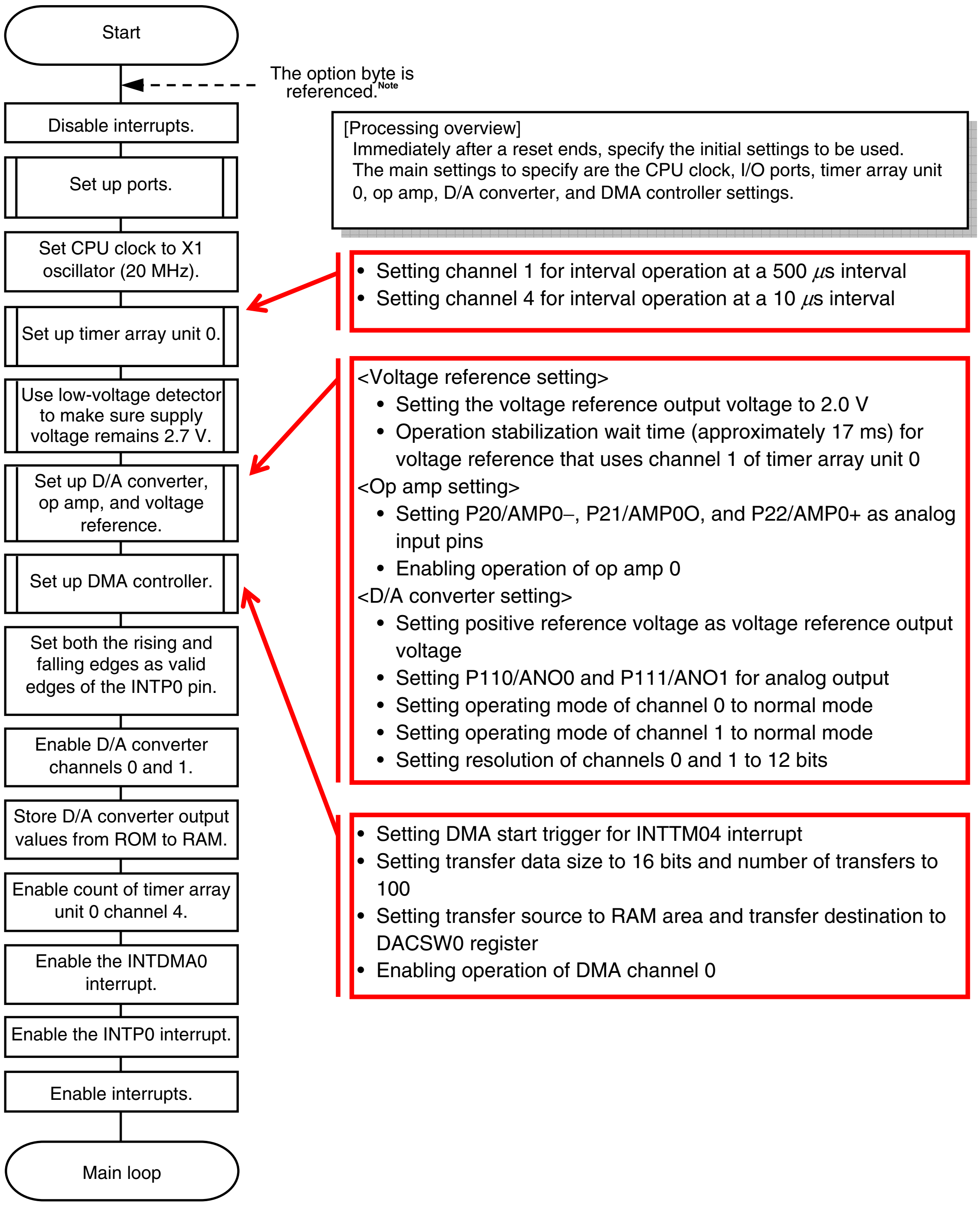
<INTRTC interrupt servicing>



[Processing overview]
This changes the ANO0 output (0.52 V ↔ 0.54 V).

3.4.8 Comparator circuit

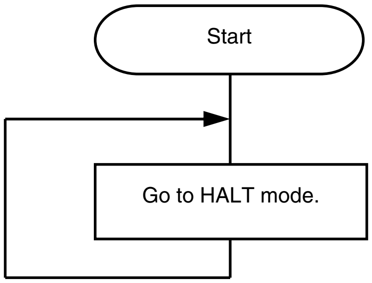
<Settings during initialization immediately after a reset ends>



Note The option byte is automatically referenced by the microcontroller immediately after a reset ends. In this sample program, the following settings are specified using the option byte:

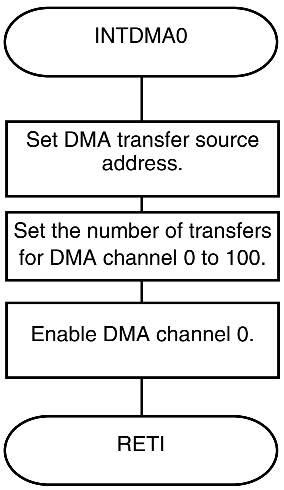
- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Main loop>



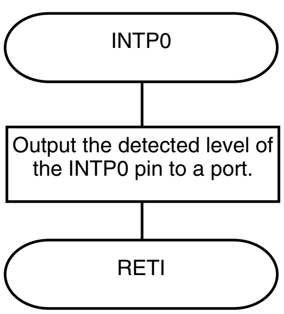
[Processing overview]
Use HALT mode and wait for INTDMA0 interrupts.

<INTDMA0 interrupt servicing>



[Processing overview]
Reset the DMA controller.

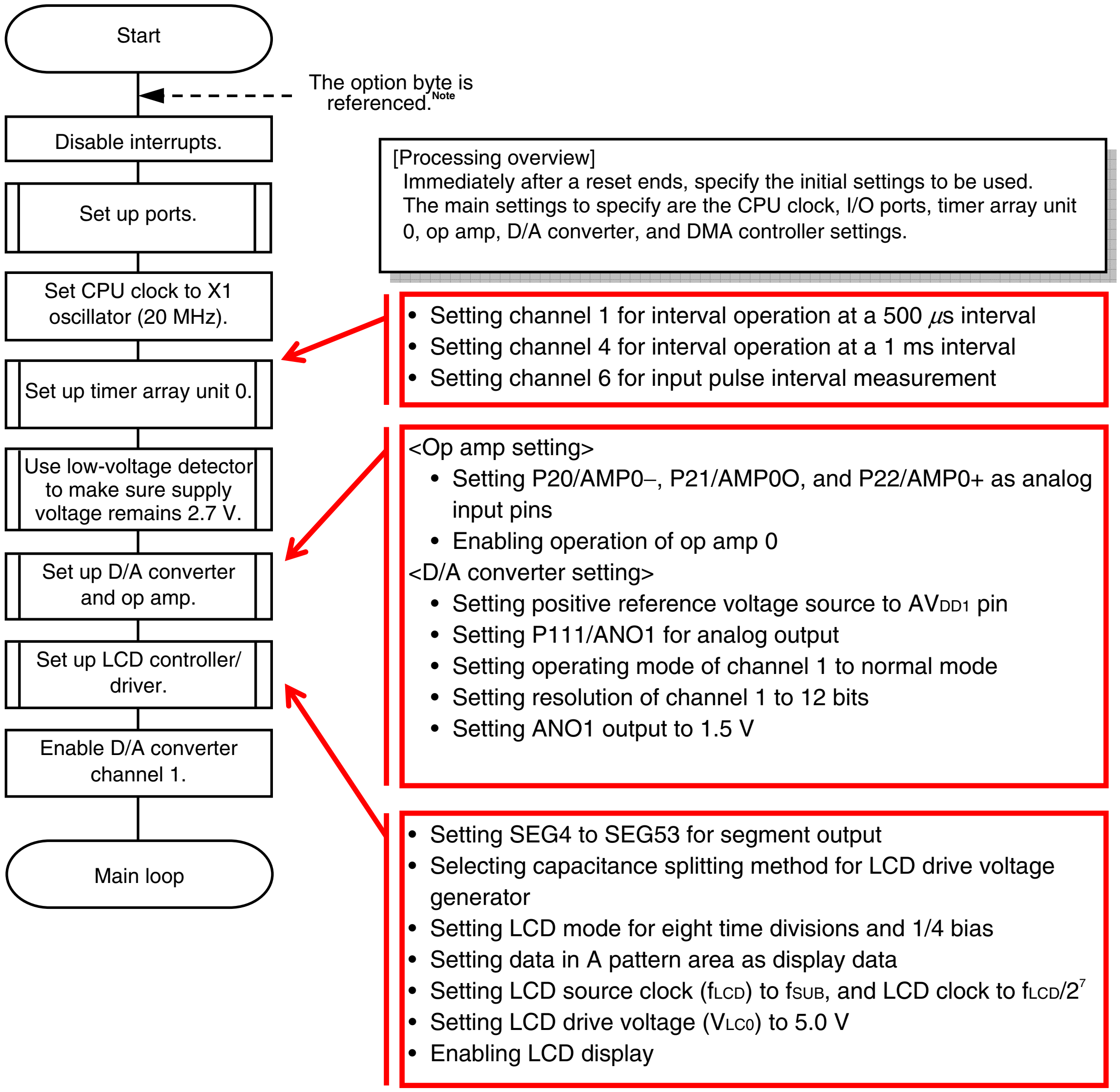
<INTP0 interrupt servicing>



[Processing overview]
Output the detected level of the INTP0 pin to a port.

3.4.9 CR oscillator 1

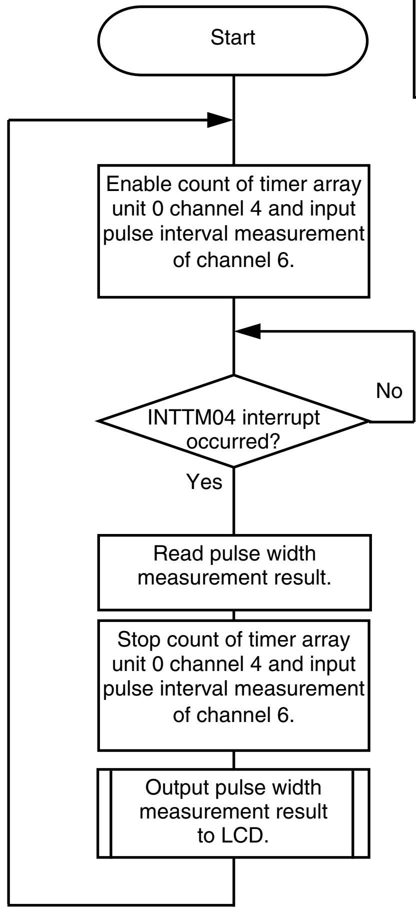
<Settings during initialization immediately after a reset ends>



Note The option byte is automatically referenced by the microcontroller immediately after a reset ends. In this sample program, the following settings are specified using the option byte:

- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

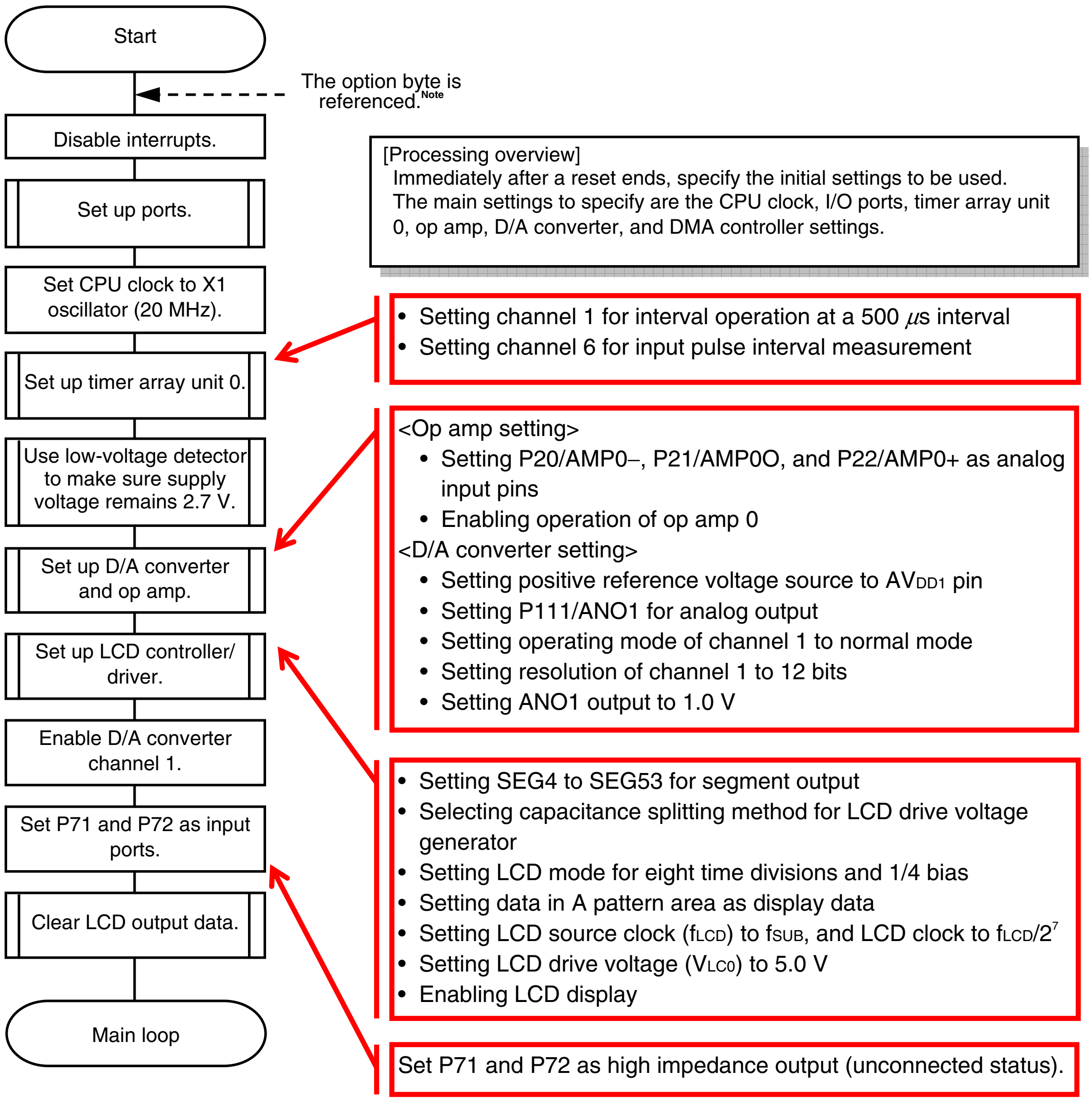
<Main loop>



[Processing overview]
The input pulse interval measurement function of timer array unit 0 channel 6 is used to measure the frequency of op amp 0 output.
Also, the measured frequency is output to the LCD.

3.4.10 CR oscillator 2

<Settings during initialization immediately after a reset ends>

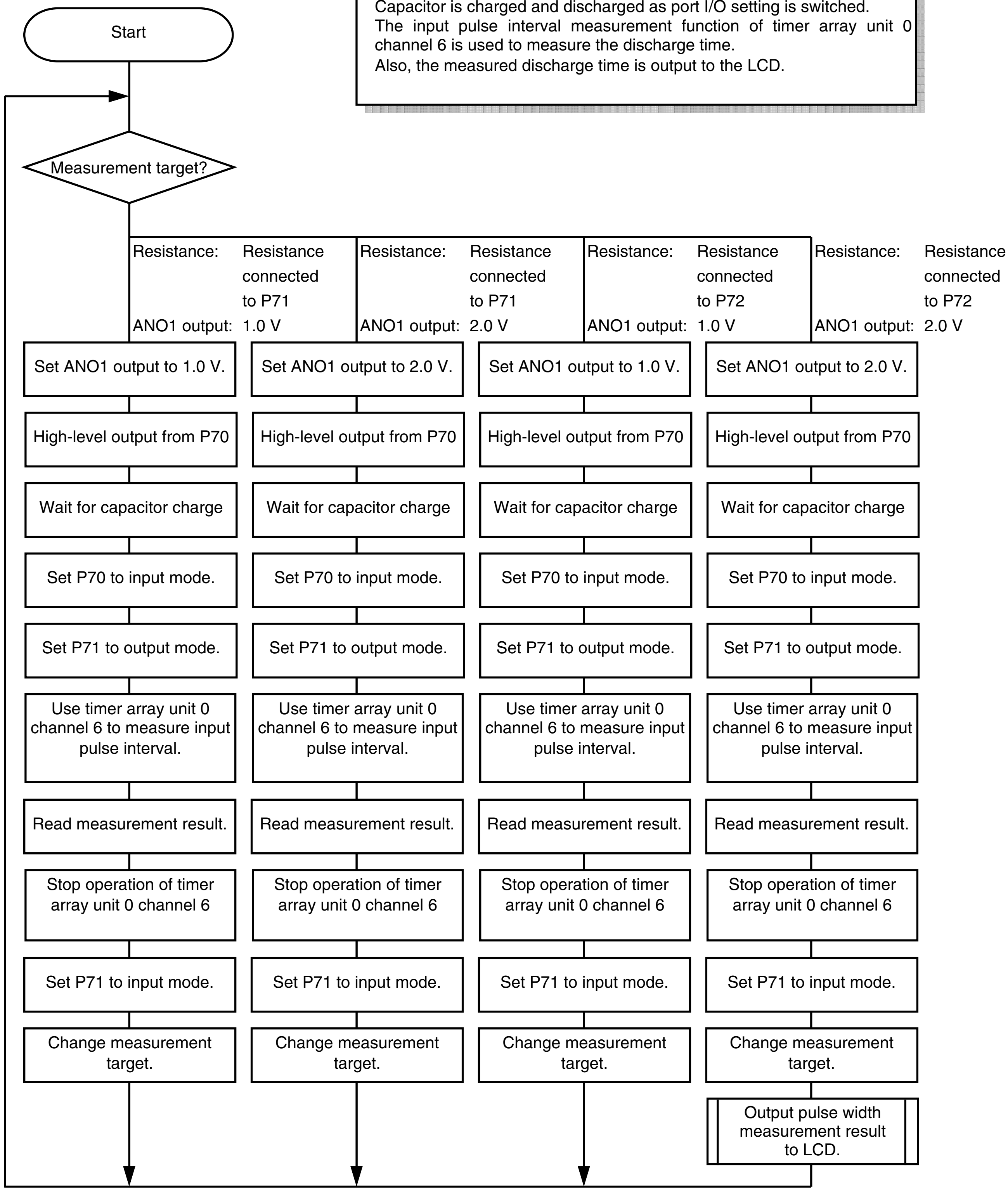


Note The option byte is automatically referenced by the microcontroller immediately after a reset ends. In this sample program, the following settings are specified using the option byte:

- Disabling the watchdog timer
- Setting the internal high-speed oscillator frequency to 8 MHz
- Disabling LVI from being started by default
- Enabling operation of on-chip debugging

<Main loop>

[Processing overview]
Capacitor is charged and discharged as port I/O setting is switched.
The input pulse interval measurement function of timer array unit 0 channel 6 is used to measure the discharge time.
Also, the measured discharge time is output to the LCD.



CHAPTER 4 SETTING METHODS

This chapter describes initialization of peripherals to be used, and processing by the sample program.

For details of option bytes, clock frequencies, and register setting methods, see the user's manual for this product (78K0R/Lx3) and the sample program.

4.1 Non-Inverting Amplifier Circuit

4.1.1 Initialization of peripherals to be used

(1) DMA initialization

The following operations are performed during DMA initialization.

<1> Operation of DMA channel 0 is enabled.

<2> SFR address setting for DMA transfer is specified.

<3> RAM address setting for DMA transfer is specified.

<4> The number of DMA transfers is set.

<5> The DMA transfer mode is set as follows:

- DMA transfer start software trigger: Software trigger does not operate
- DMA transfer direction: Internal RAM → SFR
- Transfer data size: 16 bits
- DMA transfer hold: No hold
- DMA start trigger: INTTM04

<6> DMA trigger wait mode is set.

```
        void    fn_DmaInit(void)
        {

<1>-----        DEN0 = 1;                               /* Enables operation of DMA channel 0 */

<2>-----        DSA0 = (unsigned char)&DACSW0;          /* DMA SFR Address Register 0 */
<3>-----        DRA0 = (unsigned short)&ushDacData[0];  /* DMA RAM Address Register 0 */
<4>-----        DBC0 = 100;                             /* DMA Byte Count Register 0 */
<5>-----        DMC0 = 0b01100100;                      /* DMA Mode Control Register 0 */
                        /*|||||||| */
                        /*||||++++------- IFC3-IFC0 */
                        /*||||           [Selection of DMA start source] */
                        /*||||            0000:Disables DMA transfer by interrupt */
                        /*||||                  (Only software trigger is enabled.) */
                        /*||||            0010:INTTM00 */
                        /*||||            0011:INTTM01 */
                        /*||||            0100:INTTM04 */
                        /*||||            0101:INTTM05 */
                        /*||||            0110:INTST0/INTCSI00 */
                        /*||||            0111:INTSR0/INTCSI01 */
                        /*||||            1000:INTST1/INTCSI10/INTIIC10 */
                        /*||||            1001:INTSR1 */
                        /*||||            1010:INTST3 */
                        /*||||            1011:INTSR3 */
                        /*||||            1100:INTAD */
                        /*|||+----------- DWAIT0 */
                        /*|||            [Pending of DMA transfer] */
                        /*|||             0:Executes DMA transfer upon DMA start request (not held
        pending). */
                        /*|||             1:Holds DMA start request pending if any. */
                        /*||+------------ DS0 */
                        /*||             [Specification of transfer data size for DMA transfer] */
                        /*||              0:8bits */
                        /*||              1:16bits */
                        /*|+------------- DRS0 */
                        /*|              [Selection of DMA transfer direction] */
                        /*|               0:SFR to internal RAM */
                        /*|               1:Internal RAM to SFR */
                        /*+------------- STG0 */
                        /*               [DMA transfer start software trigger] */
                        /*                0:No trigger operation */
                        /*                1:DMA transfer is started when DMA operation is enabled (DEN0
        = 1). */

<6>-----        DST0 = 1;               /* DMA transfer is started when DMA operation is enabled (DEN0
        = 1). */

        }
```

(2) Op amp and D/A converter initialization

The following operations are performed during op amp and D/A converter initialization.

<1> An input clock is supplied to the A/D converter.

<2> The voltage reference is set as follows:

- Positive reference voltage source of D/A converter: V_{REFOUT}
- Voltage reference operation control: Operation is enabled
- Voltage reference output voltage: 2.0 V

<3> Port 2 and port 15 are set for analog input.

<4> Wait mode is in effect until settings completed.

<5> Ports to be used for analog input are set to input mode.

<6> Operation of op amp 0 is enabled.

<7> An input clock is supplied to the D/A converter.

<8> D/A converter channel 0 is set as follows:

- Operating mode of D/A converter: Real-time output mode
- Resolution of D/A converter: 12 bits
- Positive reference voltage source of D/A converter: V_{REFOUT}

<9> Operation of D/A converter channel 0 is enabled.

<10> D/A converter is initialized.

```
     void    fn_AdcDacInit(void)
     {
<1>          ADCEN = 1;              /* supplies input clock to A/D converter, operational amplifier,
     and voltage reference */

             /*-------------------------------------------*/
             /*    Initialization of Voltage reference    */
             /*-------------------------------------------*/
             ADVRC = 0b00001000;     /* Analog reference voltage control register */
                     /*||||+|++--- : [1]Positive reference voltage supplies selection of A/D and D/A
     converters */
                     /*|||| |        [2]Operationcontrol of voltage reference */
                     /*|||| |        [3]Output voltage selection of voltage reference */
                     /*|||| |        [4]Operation control of input gate voltage boost circuit for
     A/D converter */
                     /*|||| |        [5]Relationship with the conversion mode used */
                     /*|||| | */
                     /*|||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
                     /*|||| |           [2]Stops operation (Hi-Z) */
                     /*|||| |           [3]2.5 V */
                     /*|||| |           [4]Stops operation */
                     /*|||| |           [5]Can be set in conversion mode 1 */
                     /*|||| | */
                     /*|||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
                     /*|||| |           [2]Stops operation (Hi-Z) */
                     /*|||| |           [3]2.0 V */
                     /*|||| |           [4]Enables operation */
<2>                  /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                     /*|||| | */
                     /*|||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
                     /*|||| |           [2]Stops operation (pull-down output) */
                     /*|||| |           [3]2.5 V */
                     /*|||| |           [4]Stops operation */
                     /*|||| |           [5] - */
                     /*|||| | */
                     /*|||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
                     /*|||| |           [2]Enables operation */
                     /*|||| |           [3]2.5 V */
                     /*|||| |           [4]Enables operation */
                     /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                     /*|||| | */
                     /*|||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
                     /*|||| |           [2]Stops operation (pull-down output) */
                     /*|||| |           [3]2.0 V */
                     /*|||| |           [4]Enables operation */
                     /*|||| |           [5] - */
                     /*|||| | */
                     /*|||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
                     /*|||| |           [2]Enables operation */
                     /*|||| |           [3]2.0 V */
                     /*|||| |           [4]Enables operation */
                     /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                     /*|||| | */
                     /*|||| | : Other than the above : Setting prohibited */
                     /*|||| | */
                     /*|+++-+----- : Be sure to set 0000 */
                     /*| */
                     /*+---------- : Reference voltage supply (negative side) of A/D converter
     selection */
                     /* : 0 : AVSS */
                     /* : 1 : AVREFM (external voltage reference input) */

             ADVRC.0 = 1;            /* Enables operation */
             ADVRC.1 = 1;            /* Output 2.0 V */
```

```
<3>     ADPC =  0b00000000;     /* A/D Port Configuration Register */
                /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                /*||| :          +------------------------- ANI15/AVREFM/P157 */
                /*||| :          | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                /*||| :          | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20
*/
                /*||| : 0 0 0 0 0 : A A A A A A A A A A A A A */
                /*||| : 0 0 0 0 1 : A A A A A A A A A A A A D */
                /*||| : 0 0 0 1 0 : A A A A A A A A A A A D D */
                /*||| : 0 0 0 1 1 : A A A A A A A A A A D D D */
                /*||| : 0 0 1 0 0 : A A A A A A A A A D D D D */
                /*||| : 0 0 1 0 1 : A A A A A A A A D D D D D */
                /*||| : 0 0 1 1 0 : A A A A A A A D D D D D D */
                /*||| : 0 0 1 1 1 : A A A A A A D D D D D D D */
                /*||| : 0 1 0 0 0 : A A A A D D D D D D D D D */
                /*||| : 0 1 0 0 1 : A A A D D D D D D D D D D */
                /*||| : 0 1 0 1 0 : A A D D D D D D D D D D D */
                /*||| : 0 1 1 1 1 : A D D D D D D D D D D D D */
                /*||| : 1 0 0 0 0 : D D D D D D D D D D D D D */
                /*||| */
                /*+++-------- : Be sure to set 000 */

        /* wait for setting time to 17msec */
<4>     fn_Wait500usBase(17000/500);

        /*------------------------------------------*/
        /*  Initialization of operational amplifier   */
        /*------------------------------------------*/
<5>     PM2.1 = 1;                      /* input port mode (AMP0O) */
        PM2.0 = 1;                      /* input port mode (AMP0-) */
        PM2.2 = 1;                      /* input port mode (AMP0+) */

        PM2.4 = 1;                      /* input port mode (AMP1O) */
        PM2.3 = 1;                      /* input port mode (AMP1-) */
        PM2.5 = 1;                      /* input port mode (AMP1+) */

        PM2.7 = 1;                      /* input port mode (AMP2O) */
        PM2.6 = 1;                      /* input port mode (AMP2-) */
        PM15.0 = 1;                     /* input port mode (AMP2+) */

        PM11.0 = 1;                     /* input port mode (ANO0) */

<6>     ADCEN = 1;                      /* supplies input clock to OPAmp A/D converter */
        OAC.0 = 1;                      /* operational amplifier (AMP0) enable */
        OAC.1 = 0;                      /* operational amplifier (AMP1) disable */
        OAC.2 = 0;                      /* operational amplifier (AMP2) disable */

        /*------------------------------------------*/
        /*      Initialization of D/A converter       */
        /*------------------------------------------*/
<7>     DACEN = 1;                      /* supplies input clock to D/A converter */
<8>     DAM.0 = 1;                      /* D/A converter operation is Real-time output mode */
        DAM.2 = 1;                      /* D/A conversion resolution is 12-bit */
        DAM.6 = 1;                      /* D/A converter positive reference voltage is VREFOUT
*/
<9>     DACE0 = 0;                      /* D/A converter CH0 disable */
<10>    DACSW0 = aushDacData[99];       /* D/A conversion value setting reigster 0 */

}
```

4.1.2 Main processing

The following operations are performed during main processing.

<1> Operation of D/A converter channel 0 is enabled.
<2> Sine wave output table is copied from ROM to RAM.
<3> TM04 interrupt is enabled.
<4> DMA transfer channel 0 interrupt is enabled.
<5> Interrupts are enabled.
<6> The mode switches to wait mode until an interrupt occurs.
<7> DMA transfer is reset and processing goes to <6>.

```
      void main(void)
      {
              unsigned short  count;          /* counter for initializing D/A convert data */

<1>           DACE0 = 1;                      /* D/A converter CH0 enable */

              /*==================================*/
              /*----------------------------------*/
              /*             Main Loop            */
              /*----------------------------------*/
              /*==================================*/

<2>           /*  initialize D/Aconverter data */
              for(count = 0; count<100; count++){
                      ushDacData[count] = aushDacData[count];
              }

<3>           TS0L.4 = 1;                     /* start TAU0 CH4 */
              TMIF04 = 0;                     /* reset INTTM04 interrupt request */

<4>           DMAMK0 = 0;                     /* enable DMA CH0 interrupt */

<5>           EI();                           /* enable all interrupts */

              /*============================================================*/
              /* if system have nothing to do, go to standby for power-saving */
              /*============================================================*/


<6>           while (1){
                      HALT();
                      NOP();
              }
      }


      /*--------------------------------------------------------------*/
      /* Module:      fn_intdma0                                      */
      /* Description: DMA ch0 interruput process                      */
      /*      parameter: --                                           */
      /*      return   : --                                           */
      /*--------------------------------------------------------------*/
      __interrupt void fn_intdma0(void)
      {
              DRA0 = (unsigned short)&ushDacData[0];  /* DMA RAM Address Register 0 */
              DBC0 = 100;                             /* DMA Byte Count Register 0 */
<7>
              DST0 = 1;       /* DMA transfer is started when DMA operation is enabled (DEN0 = 1).
      */
      }
```

4.2 Inverting Amplifier Circuit

4.2.1 Initialization of peripherals to be used

(1) DMA initialization

The following operations are performed during DMA initialization.

<1> Operation of DMA channel 0 is enabled.

<2> SFR address setting for DMA transfer is specified.

<3> RAM address setting for DMA transfer is specified.

<4> The number of DMA transfers is set.

<5> The DMA transfer mode is set as follows:

- DMA transfer start software trigger: Software trigger does not operate
- DMA transfer direction: Internal RAM → SFR
- Transfer data size: 16 bits
- DMA transfer hold: No hold
- DMA start trigger: INTTM04

<6> DMA trigger wait mode is set.

```
        void    fn_DmaInit(void)
        {

<1>-----        DEN0 = 1;                               /* Enables operation of DMA channel 0 */

<2>-----        DSA0 = (unsigned char)&DACSW0;          /* DMA SFR Address Register 0 */
<3>-----        DRA0 = (unsigned short)&ushDacData[0];  /* DMA RAM Address Register 0 */
<4>-----        DBC0 = 100;                             /* DMA Byte Count Register 0 */
<5>-----        DMC0 =  0b01100100;                     /* DMA Mode Control Register 0 */
                        /*|||||||| */
                        /*||||++++------- IFC3-IFC0 */
                        /*||||           [Selection of DMA start source] */
                        /*||||            0000:Disables DMA transfer by interrupt */
                        /*||||                  (Only software trigger is enabled.) */
                        /*||||            0010:INTTM00 */
                        /*||||            0011:INTTM01 */
                        /*||||            0100:INTTM04 */
                        /*||||            0101:INTTM05 */
                        /*||||            0110:INTST0/INTCSI00 */
                        /*||||            0111:INTSR0/INTCSI01 */
                        /*||||            1000:INTST1/INTCSI10/INTIIC10 */
                        /*||||            1001:INTSR1 */
                        /*||||            1010:INTST3 */
                        /*||||            1011:INTSR3 */
                        /*||||            1100:INTAD */
                        /*|||+----------- DWAIT0 */
                        /*|||            [Pending of DMA transfer] */
                        /*|||             0:Executes DMA transfer upon DMA start request (not held
        pending). */
                        /*|||             1:Holds DMA start request pending if any. */
                        /*||+------------ DS0 */
                        /*||             [Specification of transfer data size for DMA transfer] */
                        /*||              0:8bits */
                        /*||              1:16bits */
                        /*|+------------- DRS0 */
                        /*|              [Selection of DMA transfer direction] */
                        /*|               0:SFR to internal RAM */
                        /*|               1:Internal RAM to SFR */
                        /*+------------- STG0 */
                        /*               [DMA transfer start software trigger] */
                        /*                0:No trigger operation */
                        /*                1:DMA transfer is started when DMA operation is enabled (DEN0
        = 1). */

<6>-----        DST0 = 1;               /* DMA transfer is started when DMA operation is enabled (DEN0
        = 1). */

        }
```

(2) Op amp and D/A converter initialization

The following operations are performed during op amp and D/A converter initialization.

<1> An input clock is supplied to the A/D converter.

<2> The voltage reference is set as follows:

- Positive reference voltage source of D/A converter: V_{REFOUT}
- Voltage reference operation control: Operation is enabled
- Voltage reference output voltage: 2.0 V

<3> Port 2 and port 15 are set for analog input.

<4> Wait mode is in effect until settings completed.

<5> Ports to be used for analog input are set to input mode.

<6> Operation of op amp 0 is enabled.

<7> An input clock is supplied to the D/A converter.

<8> D/A converter channel 0 is set as follows:

- Operating mode of D/A converter: Normal output mode
- Resolution of D/A converter: 12 bits
- Positive reference voltage source of D/A converter: V_{REFOUT}

<9> D/A converter channel 0 is initialized.

<10> D/A converter channel 1 is set as follows:

- Operating mode of D/A converter: Normal output mode
- Resolution of D/A converter: 12 bits

<11> D/A converter channel 1 is initialized.

```
void    fn_AdcDacInit(void)
{
<1>     ADCEN = 1;              /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

        /*-------------------------------------------*/
        /*    Initialization of Voltage reference    */
        /*-------------------------------------------*/
<2>     ADVRC = 0b00001000;     /* Analog reference voltage control register */
                /*||||+|++--- : [1]Positive reference voltage supplies selection of A/D and D/A
converters */
                /*|||| |        [2]Operationcontrol of voltage reference */
                /*|||| |        [3]Output voltage selection of voltage reference */
                /*|||| |        [4]Operation control of input gate voltage boost circuit for
A/D converter */
                /*|||| |        [5]Relationship with the conversion mode used */
                /*|||| | */
                /*|||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
                /*|||| |           [2]Stops operation (Hi-Z) */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Stops operation */
                /*|||| |           [5]Can be set in conversion mode 1 */
                /*|||| | */
                /*|||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
                /*|||| |           [2]Stops operation (Hi-Z) */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Stops operation (pull-down output) */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Stops operation */
                /*|||| |           [5] - */
                /*|||| | */
                /*|||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Enables operation */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Stops operation (pull-down output) */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5] - */
                /*|||| | */
                /*|||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Enables operation */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : Other than the above : Setting prohibited */
                /*|||| | */
                /*|+++-+----- : Be sure to set 0000 */
                /*| */
                /*+---------- : Reference voltage supply (negative side) of A/D converter
selection */
                /* : 0 : AVSS */
                /* : 1 : AVREFM (external voltage reference input) */

        ADVRC.0 = 1;            /* Enables operation */
        ADVRC.1 = 1;            /* Output 2.0 V */
```

```
<3>     ADPC =  0b00000000;     /* A/D Port Configuration Register */
                /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                /*||| :           +------------------------ ANI15/AVREFM/P157 */
                /*||| :           | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                /*||| :           | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20
*/
                /*||| : 0 0 0 0 0 : A A A A A A A A A A A A A */
                /*||| : 0 0 0 0 1 : A A A A A A A A A A A A D */
                /*||| : 0 0 0 1 0 : A A A A A A A A A A A D D */
                /*||| : 0 0 0 1 1 : A A A A A A A A A A D D D */
                /*||| : 0 0 1 0 0 : A A A A A A A A A D D D D */
                /*||| : 0 0 1 0 1 : A A A A A A A A D D D D D */
                /*||| : 0 0 1 1 0 : A A A A A A A D D D D D D */
                /*||| : 0 0 1 1 1 : A A A A A A D D D D D D D */
                /*||| : 0 1 0 0 0 : A A A A D D D D D D D D D */
                /*||| : 0 1 0 0 1 : A A A D D D D D D D D D D */
                /*||| : 0 1 0 1 0 : A A D D D D D D D D D D D */
                /*||| : 0 1 1 1 1 : A D D D D D D D D D D D D */
                /*||| : 1 0 0 0 0 : D D D D D D D D D D D D D */
                /*||| */
                /*+++-------- : Be sure to set 000 */

        /* wait for setting time to 17msec */
<4>     fn_Wait500usBase(17000/500);

        /*------------------------------------------*/
        /*  Initialization of operational amplifier   */
        /*------------------------------------------*/
        PM2.1 = 1;                      /* input port mode (AMP0O) */
        PM2.0 = 1;                      /* input port mode (AMP0-) */
        PM2.2 = 1;                      /* input port mode (AMP0+) */

<5>     PM2.4 = 1;                      /* input port mode (AMP1O) */
        PM2.3 = 1;                      /* input port mode (AMP1-) */
        PM2.5 = 1;                      /* input port mode (AMP1+) */

        PM2.7 = 1;                      /* input port mode (AMP2O) */
        PM2.6 = 1;                      /* input port mode (AMP2-) */
        PM15.0 = 1;                     /* input port mode (AMP2+) */

        PM11.0 = 1;                     /* input port mode (ANO0) */

        ADCEN = 1;                      /* supplies input clock to OPAmp A/D converter */
<6>     OAC.0 = 1;                      /* operational amplifier (AMP0) enable */
        OAC.1 = 0;                      /* operational amplifier (AMP1) disable */
        OAC.2 = 0;                      /* operational amplifier (AMP2) disable */

        /*------------------------------------------*/
        /*      Initialization of D/A converter      */
        /*------------------------------------------*/
<7>     DACEN = 1;                      /* supplies input clock to D/A converter */
        DAM.0 = 0;                      /* [DAMD0] D/A converter operation is Normal mode */
<8>     DAM.2 = 1;                      /* [DARES0]D/A conversion resolution is 12-bit */
        DAM.6 = 1;                      /* [DAREF] D/A converter positive reference voltage is
VREFOUT */
<9>     DACSW0 = aushDacData[99];       /* D/A conversion value setting reigster 0 */

        /* set D/A chanel1 */
<10>    DAM.1 = 0;                      /* [DAMD1] D/A converter operation is Normal mode */
        DAM.3 = 1;                      /* [DARES1]D/A conversion resolution is 12-bit */
<11>    DACSW1 = aushDacData[99];       /* D/A conversion value setting reigster 1 */

}
```

4.2.2 Main processing

The following operations are performed during main processing.

<1> Operation of D/A converter channels 0 and 1 is enabled.
<2> Sine wave output table is copied from ROM to RAM.
<3> TM04 interrupt is enabled.
<4> DMA transfer channel 0 interrupt is enabled.
<5> Interrupts are enabled.
<6> The mode switches to wait mode until an interrupt occurs.
<7> DMA transfer is reset and processing goes to <6>.

```
     void main(void)
     {
             unsigned short  count;          /* counter for initializing D/A convert data */

<1>          DACE1 = 1;                      /* D/A converter CH1 enable */
             DACE0 = 1;

             /*==================================*/
             /*----------------------------------*/
             /*             Main Loop            */
             /*----------------------------------*/
             /*==================================*/

             /*  initialize D/Aconverter data */
<2>          for(count = 0; count<100; count++){
                     ushDacData[count] = aushDacData[count];
             }

<3>          TS0L.4 = 1;                     /* start TAU0 CH4 */
             TMIF04 = 0;                     /* reset INTTM04 interrupt request */

<4>          DMAMK0 = 0;                     /* enable DMA CH0 interrupt */

<5>          EI();                           /* enable all interrupts */

             /*============================================================*/
             /* if system have nothing to do, go to standby for power-saving */
             /*============================================================*/


             while (1){
<6>                  HALT();
                     NOP();
             }
     }


     /*----------------------------------------------------------------*/
     /* Module:      fn_intdma0                                        */
     /* Description: DMA ch0 interrupt process                         */
     /*      parameter: --                                             */
     /*      return   : --                                             */
     /*----------------------------------------------------------------*/
     __interrupt void fn_intdma0(void)
     {
             DRA0 = (unsigned short)&ushDacData[0];  /* DMA RAM Address Register 0 */
<7>          DBC0 = 100;                             /* DMA Byte Count Register 0 */

             DST0 = 1;       /* DMA transfer is started when DMA operation is enabled (DEN0 = 1).
     */
     }
```

4.3 Offset Adjustment of Inverting Amplifier Circuit

4.3.1 Initialization of peripherals to be used

(1) DMA initialization

The following operations are performed during DMA initialization.

<1> Operation of DMA channel 0 is enabled.

<2> SFR address setting for DMA transfer is specified.

<3> RAM address setting for DMA transfer is specified.

<4> The number of DMA transfers is set.

<5> The DMA transfer mode is set as follows:

- DMA transfer start software trigger: Software trigger does not operate
- DMA transfer direction: Internal RAM → SFR
- Transfer data size: 16 bits
- DMA transfer hold: No hold
- DMA start trigger: INTTM04

<6> DMA trigger wait mode is set.

```
void    fn_DmaInit(void)
{

<1>-----        DEN0 = 1;                               /* Enables operation of DMA channel 0 */

<2>-----        DSA0 = (unsigned char)&DACSW0;          /* DMA SFR Address Register 0 */
<3>-----        DRA0 = (unsigned short)&ushDacData[0];  /* DMA RAM Address Register 0 */
<4>-----        DBC0 = 100;                             /* DMA Byte Count Register 0 */
<5>-----        DMC0 =  0b01100100;                     /* DMA Mode Control Register 0 */
                        /*|||||||| */
                        /*||||++++------- IFC3-IFC0 */
                        /*||||           [Selection of DMA start source] */
                        /*||||            0000:Disables DMA transfer by interrupt */
                        /*||||                  (Only software trigger is enabled.) */
                        /*||||            0010:INTTM00 */
                        /*||||            0011:INTTM01 */
                        /*||||            0100:INTTM04 */
                        /*||||            0101:INTTM05 */
                        /*||||            0110:INTST0/INTCSI00 */
                        /*||||            0111:INTSR0/INTCSI01 */
                        /*||||            1000:INTST1/INTCSI10/INTIIC10 */
                        /*||||            1001:INTSR1 */
                        /*||||            1010:INTST3 */
                        /*||||            1011:INTSR3 */
                        /*||||            1100:INTAD */
                        /*|||+----------- DWAIT0 */
                        /*|||            [Pending of DMA transfer] */
                        /*|||             0:Executes DMA transfer upon DMA start request (not held
pending). */
                        /*|||             1:Holds DMA start request pending if any. */
                        /*||+------------ DS0 */
                        /*||             [Specification of transfer data size for DMA transfer] */
                        /*||              0:8bits */
                        /*||              1:16bits */
                        /*|+------------- DRS0 */
                        /*|              [Selection of DMA transfer direction] */
                        /*|               0:SFR to internal RAM */
                        /*|               1:Internal RAM to SFR */
                        /*+------------- STG0 */
                        /*               [DMA transfer start software trigger] */
                        /*                0:No trigger operation */
                        /*                1:DMA transfer is started when DMA operation is enabled (DEN0
= 1). */

<6>-----        DST0 = 1;               /* DMA transfer is started when DMA operation is enabled (DEN0
= 1). */

}
```

(2) Op amp and D/A converter initialization

The following operations are performed during op amp and D/A converter initialization.

<1> An input clock is supplied to the A/D converter.

<2> The voltage reference is set as follows:

- Positive reference voltage source of D/A converter: V_{REFOUT}
- Voltage reference operation control: Operation is enabled
- Voltage reference output voltage: 2.0 V

<3> Port 2 and port 15 are set for analog input.

<4> Wait mode is in effect until settings completed.

<5> Ports to be used for analog input are set to input mode.

<6> Operation of op amp 0 is enabled.

<7> An input clock is supplied to the D/A converter.

<8> D/A converter channel 0 is set as follows:

- Operating mode of D/A converter: Normal output mode
- Resolution of D/A converter: 12 bits
- Positive reference voltage source of D/A converter: V_{REFOUT}

<9> D/A converter channel 0 is initialized.

<10> D/A converter channel 1 is set as follows:

- Operating mode of D/A converter: Normal output mode
- Resolution of D/A converter: 12 bits

<11> D/A converter channel 1 is initialized.

```
void    fn_AdcDacInit(void)
{
<1>     ADCEN = 1;              /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

        /*---------------------------------------------*/
        /*  Initialization of Voltage reference         */
        /*---------------------------------------------*/
        ADVRC = 0b00001000;     /* Analog reference voltage control register */
                /*||||+|++--- : [1]Positive reference voltage supplies selection of A/D and D/A
converters */
                /*|||| |        [2]Operationcontrol of voltage reference */
                /*|||| |        [3]Output voltage selection of voltage reference */
                /*|||| |        [4]Operation control of input gate voltage boost circuit for
A/D converter */
                /*|||| |        [5]Relationship with the conversion mode used */
                /*|||| | */
                /*|||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
                /*|||| |           [2]Stops operation (Hi-Z) */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Stops operation */
                /*|||| |           [5]Can be set in conversion mode 1 */
                /*|||| | */
                /*|||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
                /*|||| |           [2]Stops operation (Hi-Z) */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
<2>             /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Stops operation (pull-down output) */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Stops operation */
                /*|||| |           [5] - */
                /*|||| | */
                /*|||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Enables operation */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Stops operation (pull-down output) */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5] - */
                /*|||| | */
                /*|||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Enables operation */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : Other than the above : Setting prohibited */
                /*|||| | */
                /*|+++-+----- : Be sure to set 0000 */
                /*| */
                /*+---------- : Reference voltage supply (negative side) of A/D converter
selection */
                /* : 0 : AVSS */
                /* : 1 : AVREFM (external voltage reference input) */

        ADVRC.0 = 1;            /* Enables operation */
        ADVRC.1 = 1;            /* Output 2.0 V */
```

```
<3>         ADPC =  0b00000000;     /* A/D Port Configuration Register */
                    /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                    /*||| :         +------------------------ ANI15/AVREFM/P157 */
                    /*||| :         | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                    /*||| :         | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20
*/
                    /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                    /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                    /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                    /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                    /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                    /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                    /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                    /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                    /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                    /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                    /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                    /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                    /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                    /*||| */
                    /*+++-------- : Be sure to set 000 */

        /* wait for setting time to 17msec */
<4>     fn_Wait500usBase(17000/500);

        /*-------------------------------------------*/
        /*  Initialization of operational amplifier  */
        /*-------------------------------------------*/
        PM2.1 = 1;                      /* input port mode (AMP0O) */
        PM2.0 = 1;                      /* input port mode (AMP0-) */
        PM2.2 = 1;                      /* input port mode (AMP0+) */

<5>     PM2.4 = 1;                      /* input port mode (AMP1O) */
        PM2.3 = 1;                      /* input port mode (AMP1-) */
        PM2.5 = 1;                      /* input port mode (AMP1+) */

        PM2.7 = 1;                      /* input port mode (AMP2O) */
        PM2.6 = 1;                      /* input port mode (AMP2-) */
        PM15.0 = 1;                     /* input port mode (AMP2+) */

        PM11.0 = 1;                     /* input port mode (ANO0) */

        ADCEN = 1;                      /* supplies input clock to OPAmp A/D converter */
<6>     OAC.0 = 1;                      /* operational amplifier (AMP0) enable */
        OAC.1 = 0;                      /* operational amplifier (AMP1) disable */
        OAC.2 = 0;                      /* operational amplifier (AMP2) disable */

        /*-------------------------------------------*/
        /*     Initialization of D/A converter     */
        /*-------------------------------------------*/
<7>     DACEN = 1;                      /* supplies input clock to D/A converter */
        DAM.0 = 0;                      /* [DAMD0] D/A converter operation is Normal mode */
<8>     DAM.2 = 1;                      /* [DARES0]D/A conversion resolution is 12-bit */
        DAM.6 = 1;                      /* [DAREF] D/A converter positive reference voltage is
VREFOUT */
<9>     DACSW0 = aushDacData[99];       /* D/A conversion value setting reigster 0 */

        /* set D/A chanel1 */
<10>    DAM.1 = 0;                      /* [DAMD1] D/A converter operation is Normal mode */
        DAM.3 = 1;                      /* [DARES1]D/A conversion resolution is 12-bit */
<11>    DACSW1 = aushDacData[80];       /* D/A conversion value setting reigster 1 */

}
```

4.3.2 Main processing

The following operations are performed during main processing.

<1> Operation of D/A converter channels 0 and 1 is enabled.
<2> Sine wave output table is copied from ROM to RAM.
<3> TM04 interrupt is enabled.
<4> DMA transfer channel 0 interrupt is enabled.
<5> Interrupts are enabled.
<6> The mode switches to wait mode until an interrupt occurs.
<7> TM04 is stopped.
<8> A/D conversion of the output voltage is performed when one cycle is completed.
<9> The output voltage A/D converted value is compared to the ideal-state reference voltage A/D converted value, and the offset is adjusted.
<10> After the offset is adjusted, a shift is triggered in the reference voltage (for operation checking).
<11> DMA transfer is reset, TM04 operation is resumed and processing goes to <6>.

```
      void main(void)
      {
              unsigned short  count;          /* counter for initializing D/A convert data */
              ushCycleCount = 0;              /* Cycle counter */

<1>           DACE1 = 1;                      /* D/A converter CH1 enable */
              DACE0 = 1;

              /*================================*/
              /*--------------------------------*/
              /*            Main Loop           */
              /*--------------------------------*/
              /*================================*/

<2>           /*  initialize D/Aconverter data */
              for(count = 0; count<100; count++){
                      ushDacData[count] = aushDacData[count];
              }

<3>           TS0L.4 = 1;                     /* start TAU0 CH4 */
              TMIF04 = 0;                     /* reset INTTM04 interrupt request */

<4>           DMAMK0 = 0;                     /* enable DMA CH0 interrupt */

<5>           EI();                           /* enable all interrupts */

              /*=========================================================*/
              /* if system have nothing to do, go to standby for power-saving */
              /*=========================================================*/
<6>           while (1){
                      HALT();
                      NOP();
              }
      }


      /*------------------------------------------------------------*/
      /* Module:      fn_intdma0                                    */
      /* Description: DMA ch0 interrupt process                     */
      /*      parameter: --                                         */
      /*      return   : --                                         */
      /*------------------------------------------------------------*/
      __interrupt void fn_intdma0(void)
      {
<7>           TT0L.4 = 1;                     /* stop TAU0 CH4 */
```

```
<8>
        /* start A/D conversion */
        ADCS = 1;       /* A/D converter enable */
        ADIF = 0;       /* clear A/D interrupt request flag */
        ADMK = 0;       /* enable A/D interrupt */
        HALT();         /* set HALT mode and wait conversion finished */
        ADMK = 1;       /* disable A/D interrupt */
        ADIF = 0;       /* clear A/D interrupt request flag */

<9>
        shVoltage = (short)(ADCR & 0x0fff);     /* get A/D result */
        shVoltage -= (2048 - 1);
        if(shVoltage < 0){
                DACSW1 += 1;            /* D/A conversion value setting reigster 1 */
        }
        else if(shVoltage > 2){
                DACSW1 -= 1;            /* D/A conversion value setting reigster 1 */
        }

<10>
        ushCycleCount++;                        /* Cycle counter */
        if(ushCycleCount > 100){                /* Cycle counter */
                DACSW1 = ushDacData[95];        /* D/A conversion value setting reigster 1 */
                ushCycleCount = 0;      /* reset Cycle counter */
        }

<11>
        DRA0 = (unsigned short)&ushDacData[0];  /* DMA RAM Address Register 0 */
        DBC0 = 100;                             /* DMA Byte Count Register 0 */

        DST0 = 1;       /* DMA transfer is started when DMA operation is enabled (DEN0 = 1).
*/

        TS0L.4 = 1;     /* start TAU0 CH4 */
        TMIF04 = 0;     /* reset INTTM04 interrupt request */

}
```

4.4 Offset Adjustment of Inverting Amplifier Circuit + Low-Pass Filter

4.4.1 Initialization of peripherals to be used

(1) DMA initialization

The following operations are performed during DMA initialization.

<1> Operation of DMA channel 0 is enabled.

<2> SFR address setting for DMA transfer is specified.

<3> RAM address setting for DMA transfer is specified.

<4> The number of DMA transfers is set.

<5> The DMA transfer mode is set as follows:

- DMA transfer start software trigger: Software trigger does not operate
- DMA transfer direction: Internal RAM → SFR
- Transfer data size: 16 bits
- DMA transfer hold: No hold
- DMA start trigger: INTTM04

<6> DMA trigger wait mode is set.

```
void    fn_DmaInit(void)
{

<1>-----        DEN0 = 1;                               /* Enables operation of DMA channel 0 */

<2>-----        DSA0 = (unsigned char)&DACSW0;          /* DMA SFR Address Register 0 */
<3>-----        DRA0 = (unsigned short)&ushDacData[0];  /* DMA RAM Address Register 0 */
<4>-----        DBC0 = 100;                             /* DMA Byte Count Register 0 */
<5>-----        DMC0 =  0b01100100;                     /* DMA Mode Control Register 0 */
                        /*|||||||| */
                        /*||||++++------- IFC3-IFC0 */
                        /*||||           [Selection of DMA start source] */
                        /*||||            0000:Disables DMA transfer by interrupt */
                        /*||||                  (Only software trigger is enabled.) */
                        /*||||            0010:INTTM00 */
                        /*||||            0011:INTTM01 */
                        /*||||            0100:INTTM04 */
                        /*||||            0101:INTTM05 */
                        /*||||            0110:INTST0/INTCSI00 */
                        /*||||            0111:INTSR0/INTCSI01 */
                        /*||||            1000:INTST1/INTCSI10/INTIIC10 */
                        /*||||            1001:INTSR1 */
                        /*||||            1010:INTST3 */
                        /*||||            1011:INTSR3 */
                        /*||||            1100:INTAD */
                        /*|||+----------- DWAIT0 */
                        /*|||            [Pending of DMA transfer] */
                        /*|||             0:Executes DMA transfer upon DMA start request (not held
pending). */
                        /*|||             1:Holds DMA start request pending if any. */
                        /*||+------------ DS0 */
                        /*||             [Specification of transfer data size for DMA transfer] */
                        /*||              0:8bits */
                        /*||              1:16bits */
                        /*|+------------- DRS0 */
                        /*|              [Selection of DMA transfer direction] */
                        /*|               0:SFR to internal RAM */
                        /*|               1:Internal RAM to SFR */
                        /*+-------------- STG0 */
                        /*               [DMA transfer start software trigger] */
                        /*                0:No trigger operation */
                        /*                1:DMA transfer is started when DMA operation is enabled (DEN0
= 1). */

<6>-----        DST0 = 1;               /* DMA transfer is started when DMA operation is enabled (DEN0
= 1). */

}
```

(2) Op amp and D/A converter initialization

The following operations are performed during op amp and D/A converter initialization.

<1> An input clock is supplied to the A/D converter.

<2> The voltage reference is set as follows:

- Positive reference voltage source of D/A converter: V_{REFOUT}
- Voltage reference operation control: Operation is enabled
- Voltage reference output voltage: 2.0 V

<3> Port 2 and port 15 are set for analog input.

<4> Wait mode is in effect until settings completed.

<5> Ports to be used for analog input are set to input mode.

<6> Operation of op amp 0 is enabled.

<7> An input clock is supplied to the D/A converter.

<8> D/A converter channel 0 is set as follows:

- Operating mode of D/A converter: Normal output mode
- Resolution of D/A converter: 12 bits
- Positive reference voltage source of D/A converter: V_{REFOUT}

<9> D/A converter channel 0 is initialized.

<10> D/A converter channel 1 is set as follows:

- Operating mode of D/A converter: Normal output mode
- Resolution of D/A converter: 12 bits

<11> D/A converter channel 1 is initialized.

```
void    fn_AdcDacInit(void)
{
<1>     ADCEN = 1;              /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

        /*---------------------------------------------*/
        /*  Initialization of Voltage reference         */
        /*---------------------------------------------*/
        ADVRC = 0b00001000;     /* Analog reference voltage control register */
                /*||||+|++--- : [1]Positive reference voltage supplies selection of A/D and D/A
converters */
                /*|||| |        [2]Operationcontrol of voltage reference */
                /*|||| |        [3]Output voltage selection of voltage reference */
                /*|||| |        [4]Operation control of input gate voltage boost circuit for
A/D converter */
                /*|||| |        [5]Relationship with the conversion mode used */
                /*|||| | */
                /*|||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
                /*|||| |           [2]Stops operation (Hi-Z) */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Stops operation */
                /*|||| |           [5]Can be set in conversion mode 1 */
                /*|||| | */
                /*|||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
                /*|||| |           [2]Stops operation (Hi-Z) */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
<2>             /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Stops operation (pull-down output) */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Stops operation */
                /*|||| |           [5] - */
                /*|||| | */
                /*|||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Enables operation */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Stops operation (pull-down output) */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5] - */
                /*|||| | */
                /*|||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Enables operation */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : Other than the above : Setting prohibited */
                /*|||| | */
                /*|+++-+----- : Be sure to set 0000 */
                /*| */
                /*+---------- : Reference voltage supply (negative side) of A/D converter
selection */
                /* : 0 : AVSS */
                /* : 1 : AVREFM (external voltage reference input) */

        ADVRC.0 = 1;            /* Enables operation */
        ADVRC.1 = 1;            /* Output 2.0 V */
```

```
<3>       ADPC =  0b00000000;     /* A/D Port Configuration Register */
                  /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                  /*||| :           +------------------------ ANI15/AVREFM/P157 */
                  /*||| :           | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                  /*||| :           | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20
*/
                  /*||| : 0 0 0 0 0 : A A A A A A A A A A A A A */
                  /*||| : 0 0 0 0 1 : A A A A A A A A A A A A D */
                  /*||| : 0 0 0 1 0 : A A A A A A A A A A A D D */
                  /*||| : 0 0 0 1 1 : A A A A A A A A A A D D D */
                  /*||| : 0 0 1 0 0 : A A A A A A A A A D D D D */
                  /*||| : 0 0 1 0 1 : A A A A A A A A D D D D D */
                  /*||| : 0 0 1 1 0 : A A A A A A A D D D D D D */
                  /*||| : 0 0 1 1 1 : A A A A A A D D D D D D D */
                  /*||| : 0 1 0 0 0 : A A A A D D D D D D D D D */
                  /*||| : 0 1 0 0 1 : A A A D D D D D D D D D D */
                  /*||| : 0 1 0 1 0 : A A D D D D D D D D D D D */
                  /*||| : 0 1 1 1 1 : A D D D D D D D D D D D D */
                  /*||| : 1 0 0 0 0 : D D D D D D D D D D D D D */
                  /*||| */
                  /*+++-------- : Be sure to set 000 */

          /* wait for setting time to 17msec */
<4>       fn_Wait500usBase(17000/500);

          /*-------------------------------------------*/
          /*  Initialization of operational amplifier   */
          /*-------------------------------------------*/
          PM2.1 = 1;                      /* input port mode (AMP0O) */
          PM2.0 = 1;                      /* input port mode (AMP0-) */
          PM2.2 = 1;                      /* input port mode (AMP0+) */

<5>       PM2.4 = 1;                      /* input port mode (AMP1O) */
          PM2.3 = 1;                      /* input port mode (AMP1-) */
          PM2.5 = 1;                      /* input port mode (AMP1+) */

          PM2.7 = 1;                      /* input port mode (AMP2O) */
          PM2.6 = 1;                      /* input port mode (AMP2-) */
          PM15.0 = 1;                     /* input port mode (AMP2+) */

          PM11.0 = 1;                     /* input port mode (ANO0) */

          ADCEN = 1;                      /* supplies input clock to OPAmp A/D converter */
<6>       OAC.0 = 1;                      /* operational amplifier (AMP0) enable */
          OAC.1 = 0;                      /* operational amplifier (AMP1) disable */
          OAC.2 = 0;                      /* operational amplifier (AMP2) disable */

          /*-------------------------------------------*/
          /*      Initialization of D/A converter      */
          /*-------------------------------------------*/
<7>       DACEN = 1;                      /* supplies input clock to D/A converter */
          DAM.0 = 0;                      /* [DAMD0] D/A converter operation is Normal mode */
<8>       DAM.2 = 1;                      /* [DARES0]D/A conversion resolution is 12-bit */
          DAM.6 = 1;                      /* [DAREF] D/A converter positive reference voltage is
VREFOUT */
<9>       DACSW0 = aushDacData[99];       /* D/A conversion value setting reigster 0 */

          /* set D/A chanel1 */
<10>      DAM.1 = 0;                      /* [DAMD1] D/A converter operation is Normal mode */
          DAM.3 = 1;                      /* [DARES1]D/A conversion resolution is 12-bit */
<11>      DACSW1 = aushDacData[90];       /* D/A conversion value setting reigster 1 */

}
```

4.4.2 Main processing

The following operations are performed during main processing.

<1> Operation of D/A converter channels 0 and 1 is enabled.
<2> Sine wave output table is copied from ROM to RAM.
<3> TM04 interrupt is enabled.
<4> DMA transfer channel 0 interrupt is enabled.
<5> Interrupts are enabled.
<6> The mode switches to wait mode until an interrupt occurs.
<7> TM04 is stopped.
<8> A/D conversion of the output voltage is performed when one cycle is completed.
<9> The output voltage A/D converted value is compared to the ideal-state reference voltage A/D converted value, and the offset is adjusted.
<10> After the offset is adjusted, a shift is triggered in the reference voltage (for operation checking). The DMA transfer interval is altered so that the input signal is set to switch alternately between 100 Hz and 1 kHz.
<11> DMA transfer is reset, TM04 operation is resumed and processing goes to <6>.

```
      void main(void)
      {
              unsigned short  count;          /* counter for initializing D/A convert data */
              ushCycleCount = 0;              /* Cycle counter */
              ucCycleStatus = 0;              /* Cycle status */

<1>           DACE1 = 1;                      /* D/A converter CH1 enable */
              DACE0 = 1;

              /*==================================*/
              /*----------------------------------*/
              /*              Main Loop           */
              /*----------------------------------*/
              /*==================================*/

              /*  initialize D/Aconverter data */
<2>           for(count = 0; count<100; count++){
                      ushDacData[count] = aushDacData[count];
              }

              TS0L.4 = 1;                     /* start TAU0 CH4 */
<3>           TMIF04 = 0;                     /* reset INTTM04 interrupt request */

<4>           DMAMK0 = 0;                     /* enable DMA CH0 interrupt */

<5>           EI();                           /* enable all interrupts */

              /*=============================================================*/
              /* if system have nothing to do, go to standby for power-saving */
              /*=============================================================*/
              while (1){
                      HALT();
<6>                   NOP();
              }
      }


      /*-------------------------------------------------------------*/
      /* Module:      fn_intdma0                                      */
      /* Description: DMA ch0 interruput process                      */
      /*      parameter: --                                           */
      /*      return   : --                                           */
      /*-------------------------------------------------------------*/
      __interrupt void fn_intdma0(void)
      {
<7>           TT0L.4 = 1;                     /* stop TAU0 CH4 */
```

```
        /* start A/D conversion */
        ADCS = 1;       /* A/D converter enable */
        ADIF = 0;       /* clear A/D interrupt request flag */
<8>     ADMK = 0;       /* enable A/D interrupt */
        HALT();         /* set HALT mode and wait conversion finished */
        ADMK = 1;       /* disable A/D interrupt */
        ADIF = 0;       /* clear A/D interrupt request flag */

        shVoltage = (short)(ADCR & 0x0fff);     /* get A/D result */
        shVoltage -= (2048 - 1);
        if(shVoltage < 0){
                DACSW1 += 1;            /* D/A conversion value setting reigster 1 */
<9>     }
        else if(shVoltage > 2){
                DACSW1 -= 1;            /* D/A conversion value setting reigster 1 */
        }

        ushCycleCount++;                /* Cycle counter */
        if(ushCycleCount > 50){         /* Cycle counter */
                switch(ucCycleStatus){                  /* Cycle status */
                        case    0:
                                TDR04 = 2000;           /* set interval time to 100us(=0.05us
* 2000) */
                                ucCycleStatus = 1;      /* Cycle status (100Hz) */
                                ushCycleCount = 20;     /* reset Cycle counter */
                                break;

<10>                    case    1:
                                TDR04 = 200;            /* set interval time to 10us(=0.05us
* 200) */
                                ucCycleStatus = 0;      /* Cycle status (1kHz) */
                                ushCycleCount = 0;      /* reset Cycle counter */
                                break;

                }
                DACSW1 = ushDacData[90];

        DRA0 = (unsigned short)&ushDacData[0];  /* DMA RAM Address Register 0 */
        DBC0 = 100;                             /* DMA Byte Count Register 0 */

        DST0 = 1;       /* DMA transfer is started when DMA operation is enabled (DEN0 = 1).
<11> */

        TS0L.4 = 1;     /* start TAU0 CH4 */
        TMIF04 = 0;     /* reset INTTM04 interrupt request */

}
```

4.5 Non-Inverting Amplifier Circuit (AC)

4.5.1 Initialization of peripherals to be used

(1) DMA initialization

The following operations are performed during DMA initialization.

<1> Operation of DMA channel 0 is enabled.

<2> SFR address setting for DMA transfer is specified.

<3> RAM address setting for DMA transfer is specified.

<4> The number of DMA transfers is set.

<5> The DMA transfer mode is set as follows:

- DMA transfer start software trigger: Software trigger does not operate
- DMA transfer direction: Internal RAM → SFR
- Transfer data size: 16 bits
- DMA transfer hold: No hold
- DMA start trigger: INTTM04

<6> DMA trigger wait mode is set.

```
void    fn_DmaInit(void)
{

<1>-----        DEN0 = 1;                              /* Enables operation of DMA channel 0 */

<2>-----        DSA0 = (unsigned char)&DACSW0;          /* DMA SFR Address Register 0 */
<3>-----        DRA0 = (unsigned short)&ushDacData[0];  /* DMA RAM Address Register 0 */
<4>-----        DBC0 = 25;                              /* DMA Byte Count Register 0 */
<5>-----        DMC0 =  0b01100100;                     /* DMA Mode Control Register 0 */
                        /*|||||||| */
                        /*||||++++------- IFC3-IFC0 */
                        /*||||           [Selection of DMA start source] */
                        /*||||            0000:Disables DMA transfer by interrupt */
                        /*||||                  (Only software trigger is enabled.) */
                        /*||||            0010:INTTM00 */
                        /*||||            0011:INTTM01 */
                        /*||||            0100:INTTM04 */
                        /*||||            0101:INTTM05 */
                        /*||||            0110:INTST0/INTCSI00 */
                        /*||||            0111:INTSR0/INTCSI01 */
                        /*||||            1000:INTST1/INTCSI10/INTIIC10 */
                        /*||||            1001:INTSR1 */
                        /*||||            1010:INTST3 */
                        /*||||            1011:INTSR3 */
                        /*||||            1100:INTAD */
                        /*|||+----------- DWAIT0 */
                        /*|||            [Pending of DMA transfer] */
                        /*|||             0:Executes DMA transfer upon DMA start request (not held
pending). */
                        /*|||             1:Holds DMA start request pending if any. */
                        /*||+------------ DS0 */
                        /*||             [Specification of transfer data size for DMA transfer] */
                        /*||              0:8bits */
                        /*||              1:16bits */
                        /*|+------------- DRS0 */
                        /*|              [Selection of DMA transfer direction] */
                        /*|               0:SFR to internal RAM */
                        /*|               1:Internal RAM to SFR */
                        /*+------------- STG0 */
                        /*               [DMA transfer start software trigger] */
                        /*                0:No trigger operation */
                        /*                1:DMA transfer is started when DMA operation is enabled (DEN0
= 1). */

<6>-----        DST0 = 1;               /* DMA transfer is started when DMA operation is enabled (DEN0
= 1). */

}
```

(2) Op amp and D/A converter initialization

The following operations are performed during op amp and D/A converter initialization.

<1> An input clock is supplied to the A/D converter.

<2> The voltage reference is set as follows:

- Positive reference voltage source of D/A converter: V_{REFOUT}
- Voltage reference operation control: Operation is enabled
- Voltage reference output voltage: 2.0 V

<3> Port 2 and port 15 are set for analog input.

<4> Wait mode is in effect until settings completed.

<5> Ports to be used for analog input are set to input mode.

<6> Operation of op amp 0 is enabled.

<7> An input clock is supplied to the D/A converter.

<8> D/A converter channel 0 is set as follows:

- Operating mode of D/A converter: Real-time output mode
- Resolution of D/A converter: 12 bits
- Positive reference voltage source of D/A converter: V_{REFOUT}

```
void    fn_AdcDacInit(void)
{
<1>     ADCEN = 1;              /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

        /*----------------------------------------------*/
        /*  Initialization of Voltage reference         */
        /*----------------------------------------------*/
        ADVRC = 0b00001000;     /* Analog reference voltage control register */
                /*||||+|++--- : [1]Positive reference voltage supplies selection of A/D and D/A
converters */
                /*|||| |        [2]Operationcontrol of voltage reference */
                /*|||| |        [3]Output voltage selection of voltage reference */
                /*|||| |        [4]Operation control of input gate voltage boost circuit for
A/D converter */
                /*|||| |        [5]Relationship with the conversion mode used */
                /*|||| | */
                /*|||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
                /*|||| |           [2]Stops operation (Hi-Z) */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Stops operation */
                /*|||| |           [5]Can be set in conversion mode 1 */
                /*|||| | */
                /*|||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
                /*|||| |           [2]Stops operation (Hi-Z) */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
<2>             /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Stops operation (pull-down output) */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Stops operation */
                /*|||| |           [5] - */
                /*|||| | */
                /*|||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Enables operation */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Stops operation (pull-down output) */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5] - */
                /*|||| | */
                /*|||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Enables operation */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : Other than the above : Setting prohibited */
                /*|||| | */
                /*|+++-+----- : Be sure to set 0000 */
                /*| */
                /*+---------- : Reference voltage supply (negative side) of A/D converter
selection */
                /* : 0 : AVSS */
                /* : 1 : AVREFM (external voltage reference input) */

        ADVRC.0 = 1;            /* Enables operation */
        ADVRC.1 = 1;            /* Output 2.0 V */
```

```
<3>         ADPC =  0b00000000;     /* A/D Port Configuration Register */
                    /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                    /*||| :          +------------------------- ANI15/AVREFM/P157 */
                    /*||| :          | +-+-+-------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                    /*||| :          | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20
*/
                    /*||| : 0 0 0 0 0 : A A A A A A A A A A A A A */
                    /*||| : 0 0 0 0 1 : A A A A A A A A A A A A D */
                    /*||| : 0 0 0 1 0 : A A A A A A A A A A A D D */
                    /*||| : 0 0 0 1 1 : A A A A A A A A A A D D D */
                    /*||| : 0 0 1 0 0 : A A A A A A A A A D D D D */
                    /*||| : 0 0 1 0 1 : A A A A A A A A D D D D D */
                    /*||| : 0 0 1 1 0 : A A A A A A A D D D D D D */
                    /*||| : 0 0 1 1 1 : A A A A A A D D D D D D D */
                    /*||| : 0 1 0 0 0 : A A A A D D D D D D D D D */
                    /*||| : 0 1 0 0 1 : A A A D D D D D D D D D D */
                    /*||| : 0 1 0 1 0 : A A D D D D D D D D D D D */
                    /*||| : 0 1 1 1 1 : A D D D D D D D D D D D D */
                    /*||| : 1 0 0 0 0 : D D D D D D D D D D D D D */
                    /*||| */
                    /*+++-------- : Be sure to set 000 */

            /* wait for setting time to 17msec */
<4>         fn_Wait500usBase(17000/500);

            /*--------------------------------------------*/
            /*  Initialization of operational amplifier   */
            /*--------------------------------------------*/
            PM2.1 = 1;              /* input port mode (AMP0O) */
            PM2.0 = 1;              /* input port mode (AMP0-) */
            PM2.2 = 1;              /* input port mode (AMP0+) */

<5>         PM2.4 = 1;              /* input port mode (AMP1O) */
            PM2.3 = 1;              /* input port mode (AMP1-) */
            PM2.5 = 1;              /* input port mode (AMP1+) */

            PM2.7 = 1;              /* input port mode (AMP2O) */
            PM2.6 = 1;              /* input port mode (AMP2-) */
            PM15.0 = 1;             /* input port mode (AMP2+) */

            PM11.0 = 1;             /* input port mode (ANO0) */

            ADCEN = 1;              /* supplies input clock to OPAmp A/D converter */
<6>         OAC.0 = 1;              /* operational amplifier (AMP0) enable */
            OAC.1 = 0;              /* operational amplifier (AMP1) disable */
            OAC.2 = 0;              /* operational amplifier (AMP2) disable */

            /*--------------------------------------------*/
            /*     Initialization of D/A converter        */
            /*--------------------------------------------*/
<7>         DACEN = 1;              /* supplies input clock to D/A converter */

            DAM.0 = 1;              /* [DAMD0] D/A converter operation is Real-time output mode */
<8>         DAM.2 = 1;              /* [DARES0]D/A conversion resolution is 12-bit */
            DAM.6 = 1;              /* [DAREF] D/A converter positive reference voltage is VREFOUT
*/

}
```

4.5.2 Main processing

The following operations are performed during main processing.

<1> Operation of D/A converter channel 0 is enabled.
<2> Sine wave output table is copied from ROM to RAM.
<3> TM04 interrupt is enabled.
<4> DMA transfer channel 0 interrupt is enabled.
<5> Interrupts are enabled.
<6> The mode switches to wait mode until an interrupt occurs.
<7> TM04 is stopped.
<8> The number of DMA transfers is adjusted and the DMA controller is reset so that the D/A converter output waveform "(sin 0 ~ sin (1/2 • π)) → (constant voltage) → (sin (1/2 • π) ~ sin (3/2 • π)) → (constant voltage) → (sin (3/2 • π) ~ sin (2 π)) → (slow variation voltage)" is repeated.
<9> TM04 operation is resumed and processing goes to <6>.

```
      void main(void)
      {
              unsigned short  count;          /* counter for initializing D/A convert data */
              ucCycleStatus = 0;              /* Cycle status */

<1>           DACE0 = 1;                      /* D/A converter CH0 enable */

              /*==================================*/
              /*----------------------------------*/
              /*             Main Loop            */
              /*----------------------------------*/
              /*==================================*/

              /*  initialize D/Aconverter data */
<2>           for(count = 0; count<100; count++){
                      ushDacData[count] = aushDacData[count];
              }

              TS0L.4 = 1;                     /* start TAU0 CH4 */
<3>           TMIF04 = 0;                     /* reset INTTM04 interrupt request */

<4>           DMAMK0 = 0;                     /* enable DMA CH0 interrupt */

<5>           EI();                           /* enable all interrupts */

              /*============================================================*/
              /* if system have nothing to do, go to standby for power-saving */
              /*============================================================*/
              while (1){
                      HALT();
<6>                   NOP();
              }
      }


      /*----------------------------------------------------------------*/
      /* Module:      fn_intdma0                                       */
      /* Description: DMA ch0 interrupput process                      */
      /*      parameter: --                                            */
      /*      return   : --                                            */
      /*----------------------------------------------------------------*/
      __interrupt void fn_intdma0(void)
      {
<7>           TT0L.4 = 1;                     /* stop TAU0 CH4 */
```

```
        switch(ucCycleStatus){                  /* Cycle status */
                case    0:
                        TDR04 = 20000;          /* set interval time to 1000us(=0.05us * 20000)
*/
                        ucCycleStatus = 1;      /* Cycle status (Flat) */
                        DBC0 = 1;               /* DMA Byte Count Register 0 */
                        break;

                case    1:
                        TDR04 = 200;            /* set interval time to 10us(=0.05us * 200) */
                        ucCycleStatus = 2;      /* Cycle status (2/4,3/4 sign) */
                        DBC0 = 50 - 1;          /* DMA Byte Count Register 0 */
                                break;

                case    2:
                        TDR04 = 20000;          /* set interval time to 1000us(=0.05us * 20000)
*/
                        ucCycleStatus = 3;      /* Cycle status (Flat) */
                        DBC0 = 1;               /* DMA Byte Count Register 0 */
                        break;

<8>             case    3:
                        TDR04 = 200;            /* set interval time to 10us(=0.05us * 200) */
                        ucCycleStatus = 4;      /* Cycle status (1kHz) */
                        DBC0 = 25 - 1;          /* DMA Byte Count Register 0 */
                        break;

                case    4:
                        DRA0 = (unsigned short)&ushDacData[0];  /* DMA RAM Address Register 0
*/
                        DACSW0 = ushDacData[99]; /* D/A conversion value setting reigster 0 */
                        TMR04 = 0b0000000000000000;/* Timer Mode Register 04 */
                        TDR04 = 20000;          /* set interval time to 1000us(=0.05us * 20000)
*/
                        ucCycleStatus = 5;      /* Cycle status (10Hz) */
                        DBC0 = 50;              /* DMA Byte Count Register 0 */
                        break;

                case    5:
                        DRA0 = (unsigned short)&ushDacData[0];  /* DMA RAM Address Register 0
*/
                        DACSW0 = ushDacData[99]; /* D/A conversion value setting reigster 0 */
                        TMR04 = 0b1000000000000000;/* Timer Mode Register 04 */
                        TDR04 = 200;            /* set interval time to 10us(=0.05us * 200) */
                        ucCycleStatus = 0;      /* Cycle status (1kHz) */
                        DBC0 = 25;              /* DMA Byte Count Register 0 */
                        break;
        }

        DST0 = 1;               /* DMA transfer is started when DMA operation is enabled (DEN0
= 1). */

        TS0L.4 = 1;             /* start TAU0 CH4 */
<9>     TMIF04 = 0;             /* reset INTTM04 interrupt request */
}
```

4.6 Inverting Amplifier Circuit (AC)

4.6.1 Initialization of peripherals to be used

(1) DMA initialization

The following operations are performed during DMA initialization.

<1> Operation of DMA channel 0 is enabled.

<2> SFR address setting for DMA transfer is specified.

<3> RAM address setting for DMA transfer is specified.

<4> The number of DMA transfers is set.

<5> The DMA transfer mode is set as follows:

- DMA transfer start software trigger: Software trigger does not operate
- DMA transfer direction: Internal RAM → SFR
- Transfer data size: 16 bits
- DMA transfer hold: No hold
- DMA start trigger: INTTM04

<6> DMA trigger wait mode is set.

```
        void    fn_DmaInit(void)
        {

<1>-----        DEN0 = 1;                               /* Enables operation of DMA channel 0 */

<2>-----        DSA0 = (unsigned char)&DACSW0;          /* DMA SFR Address Register 0 */
<3>-----        DRA0 = (unsigned short)&ushDacData[0];  /* DMA RAM Address Register 0 */
<4>-----        DBC0 = 25;                              /* DMA Byte Count Register 0 */
<5>-----        DMC0 =  0b01100100;                     /* DMA Mode Control Register 0 */
                        /*|||||||| */
                        /*||||++++------- IFC3-IFC0 */
                        /*||||           [Selection of DMA start source] */
                        /*||||            0000:Disables DMA transfer by interrupt */
                        /*||||                  (Only software trigger is enabled.) */
                        /*||||            0010:INTTM00 */
                        /*||||            0011:INTTM01 */
                        /*||||            0100:INTTM04 */
                        /*||||            0101:INTTM05 */
                        /*||||            0110:INTST0/INTCSI00 */
                        /*||||            0111:INTSR0/INTCSI01 */
                        /*||||            1000:INTST1/INTCSI10/INTIIC10 */
                        /*||||            1001:INTSR1 */
                        /*||||            1010:INTST3 */
                        /*||||            1011:INTSR3 */
                        /*||||            1100:INTAD */
                        /*|||+----------- DWAIT0 */
                        /*|||            [Pending of DMA transfer] */
                        /*|||             0:Executes DMA transfer upon DMA start request (not held
        pending). */
                        /*|||             1:Holds DMA start request pending if any. */
                        /*||+------------ DS0 */
                        /*||             [Specification of transfer data size for DMA transfer] */
                        /*||              0:8bits */
                        /*||              1:16bits */
                        /*|+------------- DRS0 */
                        /*|              [Selection of DMA transfer direction] */
                        /*|               0:SFR to internal RAM */
                        /*|               1:Internal RAM to SFR */
                        /*+------------- STG0 */
                        /*               [DMA transfer start software trigger] */
                        /*                0:No trigger operation */
                        /*                1:DMA transfer is started when DMA operation is enabled (DEN0
        = 1). */

<6>-----        DST0 = 1;                /* DMA transfer is started when DMA operation is enabled (DEN0
        = 1). */

        }
```

(2) Op amp and D/A converter initialization

The following operations are performed during op amp and D/A converter initialization.

<1> An input clock is supplied to the A/D converter.

<2> The voltage reference is set as follows:

- Positive reference voltage source of D/A converter: V_{REFOUT}
- Voltage reference operation control: Operation is enabled
- Voltage reference output voltage: 2.0 V

<3> Port 2 and port 15 are set for analog input.

<4> Wait mode is in effect until settings completed.

<5> Ports to be used for analog input are set to input mode.

<6> Operation of op amp 0 is enabled.

<7> An input clock is supplied to the D/A converter.

<8> D/A converter channel 0 is set as follows:

- Operating mode of D/A converter: Real-time output mode
- Resolution of D/A converter: 12 bits
- Positive reference voltage source of D/A converter: V_{REFOUT}

<9> D/A converter channel 1 is set as follows:

- Operating mode of D/A converter: Normal output mode
- Resolution of D/A converter: 12 bits

<10> D/A converter channel 1 is initialized.

```
void    fn_AdcDacInit(void)
{
<1>     ADCEN = 1;              /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

        /*-------------------------------------------*/
        /*    Initialization of Voltage reference    */
        /*-------------------------------------------*/
<2>     ADVRC = 0b00001000;     /* Analog reference voltage control register */
                /*||||+|++--- : [1]Positive reference voltage supplies selection of A/D and D/A
converters */
                /*|||| |        [2]Operationcontrol of voltage reference */
                /*|||| |        [3]Output voltage selection of voltage reference */
                /*|||| |        [4]Operation control of input gate voltage boost circuit for
A/D converter */
                /*|||| |        [5]Relationship with the conversion mode used */
                /*|||| | */
                /*|||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
                /*|||| |           [2]Stops operation (Hi-Z) */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Stops operation */
                /*|||| |           [5]Can be set in conversion mode 1 */
                /*|||| | */
                /*|||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
                /*|||| |           [2]Stops operation (Hi-Z) */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Stops operation (pull-down output) */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Stops operation */
                /*|||| |           [5] - */
                /*|||| | */
                /*|||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Enables operation */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Stops operation (pull-down output) */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5] - */
                /*|||| | */
                /*|||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Enables operation */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : Other than the above : Setting prohibited */
                /*|||| | */
                /*|+++-+----- : Be sure to set 0000 */
                /*| */
                /*+---------- : Reference voltage supply (negative side) of A/D converter
selection */
                /* : 0 : AVSS */
                /* : 1 : AVREFM (external voltage reference input) */

        ADVRC.0 = 1;            /* Enables operation */
        ADVRC.1 = 1;            /* Output 2.0 V */
```

```
<3>     ADPC =  0b00000000;     /* A/D Port Configuration Register */
                /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                /*||| :           +------------------------- ANI15/AVREFM/P157 */
                /*||| :           | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                /*||| :           | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20
*/
                /*||| : 0 0 0 0 0 : A A A A A A A A A A A A A */
                /*||| : 0 0 0 0 1 : A A A A A A A A A A A A D */
                /*||| : 0 0 0 1 0 : A A A A A A A A A A A D D */
                /*||| : 0 0 0 1 1 : A A A A A A A A A A D D D */
                /*||| : 0 0 1 0 0 : A A A A A A A A A D D D D */
                /*||| : 0 0 1 0 1 : A A A A A A A A D D D D D */
                /*||| : 0 0 1 1 0 : A A A A A A A D D D D D D */
                /*||| : 0 0 1 1 1 : A A A A A A D D D D D D D */
                /*||| : 0 1 0 0 0 : A A A A D D D D D D D D D */
                /*||| : 0 1 0 0 1 : A A A D D D D D D D D D D */
                /*||| : 0 1 0 1 0 : A A D D D D D D D D D D D */
                /*||| : 0 1 1 1 1 : A D D D D D D D D D D D D */
                /*||| : 1 0 0 0 0 : D D D D D D D D D D D D D */
                /*||| */
                /*+++-------- : Be sure to set 000 */

        /* wait for setting time to 17msec */
<4>     fn_Wait500usBase(17000/500);

        /*-------------------------------------------*/
        /*  Initialization of operational amplifier   */
        /*-------------------------------------------*/
        PM2.1 = 1;              /* input port mode (AMP0O) */
        PM2.0 = 1;              /* input port mode (AMP0-) */
        PM2.2 = 1;              /* input port mode (AMP0+) */

<5>     PM2.4 = 1;              /* input port mode (AMP1O) */
        PM2.3 = 1;              /* input port mode (AMP1-) */
        PM2.5 = 1;              /* input port mode (AMP1+) */

        PM2.7 = 1;              /* input port mode (AMP2O) */
        PM2.6 = 1;              /* input port mode (AMP2-) */
        PM15.0 = 1;             /* input port mode (AMP2+) */

        PM11.0 = 1;             /* input port mode (ANO0) */

        ADCEN = 1;              /* supplies input clock to OPAmp A/D converter */
<6>     OAC.0 = 1;              /* operational amplifier (AMP0) enable */
        OAC.1 = 0;              /* operational amplifier (AMP1) disable */
        OAC.2 = 0;              /* operational amplifier (AMP2) disable */

        /*-------------------------------------------*/
        /*      Initialization of D/A converter      */
        /*-------------------------------------------*/
<7>     DACEN = 1;              /* supplies input clock to D/A converter */

        DAM.0 = 1;              /* [DAMD0] D/A converter operation is Real-time output mode */
<8>     DAM.2 = 1;              /* [DARES0]D/A conversion resolution is 12-bit */
        DAM.6 = 1;              /* [DAREF] D/A converter positive reference voltage is VREFOUT
*/

        /* set D/A chanel1 */
<9>     DAM.1 = 0;              /* [DAMD1] D/A converter operation is Normal mode */
        DAM.3 = 1;              /* [DARES1]D/A conversion resolution is 12-bit */
<10>    DACSW1 = aushDacData[99];       /* D/A conversion value setting reigster 1 */}

}
```

4.6.2 Main processing

The following operations are performed during main processing.

<1> Operation of D/A converter channels 0 and 1 is enabled.

<2> Sine wave output table is copied from ROM to RAM.

<3> TM04 interrupt is enabled.

<4> DMA transfer channel 0 interrupt is enabled.

<5> Interrupts are enabled.

<6> The mode switches to wait mode until an interrupt occurs.

<7> TM04 is stopped.

<8> The number of DMA transfers is adjusted and the DMA controller is reset so that the D/A converter output waveform "(sin 0 ~ sin (1/2 • π)) → (constant voltage) → (sin (1/2 • π) ~ sin π) → (slow variation voltage)" is repeated.

<9> TM04 operation is resumed and processing goes to <6>.

```
      void main(void)
      {
              unsigned short  count;          /* counter for initializing D/A convert data */
              ucCycleStatus = 0;              /* Cycle status */

<1>           DACE0 = 1;                      /* D/A converter CH0 enable */
              DACE1 = 1;                      /* D/A converter CH1 enable */

              /*==================================*/
              /*----------------------------------*/
              /*             Main Loop            */
              /*----------------------------------*/
              /*==================================*/

              /*  initialize D/Aconverter data */
<2>           for(count = 0; count<100; count++){
                      ushDacData[count] = aushDacData[count];
              }

<3>           TS0L.4 = 1;                     /* start TAU0 CH4 */
              TMIF04 = 0;                     /* reset INTTM04 interrupt request */

<4>           DMAMK0 = 0;                     /* enable DMA CH0 interrupt */

<5>           EI();                           /* enable all interrupts */

              /*============================================================*/
              /* if system have nothing to do, go to standby for power-saving */
              /*============================================================*/
              while (1){
                      HALT();
<6>                   NOP();
              }
      }


      /*-----------------------------------------------------------------*/
      /* Module:      fn_intdma0                                         */
      /* Description: DMA ch0 interrupt process                          */
      /*      parameter: --                                              */
      /*      return   : --                                              */
      /*-----------------------------------------------------------------*/
      __interrupt void fn_intdma0(void)
      {
<7>           TT0L.4 = 1;                     /* stop TAU0 CH4 */
```

<8>

```
                case    0:
                        TDR04 = 20000;          /* set interval time to 1000us(=0.05us * 20000)
*/
                        ucCycleStatus = 1;      /* Cycle status (Flat) */
                        DBC0 = 1;               /* DMA Byte Count Register 0 */
                        break;

                case    1:
                        TDR04 = 200;            /* set interval time to 10us(=0.05us * 200) */
                        ucCycleStatus = 2;      /* Cycle status (2/4 sign) */
                        DBC0 = 25;              /* DMA Byte Count Register 0 */
                        break;

                case    2:
                        DRA0 = (unsigned short)&ushDacData[0];  /* DMA RAM Address Register 0
*/
                        DACSW0 = ushDacData[99]; /* D/A conversion value setting reigster 0 */
                        TMR04 = 0b0000000000000000;/* Timer Mode Register 04 */
                        TDR04 = 20000;          /* set interval time to 1000us(=0.05us * 20000)
*/
                        ucCycleStatus = 3;      /* Cycle status (slow) */
                        DBC0 = 50;              /* DMA Byte Count Register 0 */
                        break;

                case    3:
                        DRA0 = (unsigned short)&ushDacData[0];  /* DMA RAM Address Register 0
*/
                        DACSW0 = ushDacData[99]; /* D/A conversion value setting reigster 0 */
                        TMR04 = 0b1000000000000000;/* Timer Mode Register 04 */
                        TDR04 = 200;            /* set interval time to 10us(=0.05us * 200) */
                        ucCycleStatus = 0;      /* Cycle status (1kHz) */
                        DBC0 = 25;              /* DMA Byte Count Register 0 */
                        break;

        }

        DST0 = 1;               /* DMA transfer is started when DMA operation is enabled (DEN0
= 1). */
```

<9>

```
        TS0L.4 = 1;             /* start TAU0 CH4 */
        TMIF04 = 0;             /* reset INTTM04 interrupt request */
}
```

4.7 Instrumentation Amplifier

4.7.1 Initialization of peripherals to be used

(1) Real-time counter initialization

The following operations are performed during real-time counter initialization.

<1> An input clock is supplied to the real-time counter.

<2> The real-time counter is set as follows:

- Real-time counter operation: Counter operation is stopped
- RTC1HZ pin output: RTC1HZ pin output (1 Hz) is prohibited
- RTCCL pin output: RTCCL pin output (32.768 kHz) is prohibited
- 12-/24-hour system: 24-hour system
- Periodic interrupt: Once per second (simultaneous with second count-up)

<3> Real-time counter interrupt is disabled.

```
      void    fn_InitRtc(void)
      {
<1>-----        RTCEN = 1;              /* supplies operational real-time counter (RTC) input clock.
      */

<2>-----        RTCC0 = 0b00001010;             /* Real-Time Counter Control Register 0 */
                        /*|||||+++--- : Constant-period interrupt (INTRTC) selection */
                        /*||||| : 0 0 0 : Does not use constant-period interrupt function. */
                        /*||||| : 0 0 1 : Once per 0.5 s */
                        /*||||| : 0 1 0 : Once per 1 s */
                        /*||||| : 0 1 1 : Once per 1 m */
                        /*||||| : 1 0 0 : Once per 1 hour */
                        /*||||| : 1 0 1 : Once per 1 day */
                        /*||||| : 1 1 x : Once per 1 month */
                        /*|||||  x = don't care */
                        /*||||| */
                        /*||||+------ : Selection of 12-/24-hour system */
                        /*|||| : 0 : 12-hour system */
                        /*|||| : 1 : 24-hour system */
                        /*|||| */
                        /*|||+------- : RTCCL pin output control */
                        /*||| : 0 : Disables output of RTCCL pin (32 kHz). */
                        /*||| : 1 : Enables output of RTCCL pin (32 kHz). */
                        /*||| */
                        /*||+-------- : RTC1HZ pin output control */
                        /*|| : 0 : Disables output of RTC1HZ pin (1 Hz). */
                        /*|| : 1 : Enables output of RTC1HZ pin (1 Hz). */
                        /*|| */
                        /*|+--------- : Be sure to set 0 */
                        /*| */
                        /*+---------- : Real-time counter operation control */
                        /* : 0 : Stops counter operation. */
                        /* : 1 : Starts counter operation. */

<3>             RTCMK = 1;                      /* disable RTC interrupt */
                RTCIF = 0;                      /* clear RTC interrupt request flag */

      }
```

(2) Op amp and D/A converter initialization

The following operations are performed during op amp and D/A converter initialization.

<1> An input clock is supplied to the A/D converter.

<2> The voltage reference is set as follows:

- Positive reference voltage source of D/A converter: V_{REFOUT}
- Voltage reference operation control: Operation is enabled
- Voltage reference output voltage: 2.0 V

<3> Port 2 and port 15 are set for analog input.

<4> Wait mode is in effect until settings completed.

<5> Ports to be used for analog input are set to input mode.

<6> Operation of op amps 0 to 2 is enabled.

<7> An input clock is supplied to the D/A converter.

<8> D/A converter channel 0 is set as follows:

- Operating mode of D/A converter: Normal output mode
- Resolution of D/A converter: 12 bits
- Positive reference voltage source of D/A converter: V_{REFOUT}

<9> D/A converter channel 0 is initialized.

<10> D/A converter channel 1 is set as follows:

- Operating mode of D/A converter: Normal output mode
- Resolution of D/A converter: 12 bits

<11> D/A converter channel 1 is initialized.

```
void    fn_AdcDacInit(void)
{
<1>     ADCEN = 1;            /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

        /*-------------------------------------------*/
        /*   Initialization of Voltage reference    */
        /*-------------------------------------------*/
        ADVRC = 0b00001000;      /* Analog reference voltage control register */
                /*||||+|++--- : [1]Positive reference voltage supplies selection of A/D and D/A
converters */
                /*|||| |        [2]Operationcontrol of voltage reference */
                /*|||| |        [3]Output voltage selection of voltage reference */
                /*|||| |        [4]Operation control of input gate voltage boost circuit for
A/D converter */
                /*|||| |        [5]Relationship with the conversion mode used */
                /*|||| | */
                /*|||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
                /*|||| |           [2]Stops operation (Hi-Z) */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Stops operation */
                /*|||| |           [5]Can be set in conversion mode 1 */
                /*|||| | */
                /*|||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
                /*|||| |           [2]Stops operation (Hi-Z) */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
<2>             /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Stops operation (pull-down output) */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Stops operation */
                /*|||| |           [5] - */
                /*|||| | */
                /*|||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Enables operation */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Stops operation (pull-down output) */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5] - */
                /*|||| | */
                /*|||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Enables operation */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : Other than the above : Setting prohibited */
                /*|||| | */
                /*|+++-+----- : Be sure to set 0000 */
                /*| */
                /*+---------- : Reference voltage supply (negative side) of A/D converter
selection */
                /* : 0 : AVSS */
                /* : 1 : AVREFM (external voltage reference input) */

        ADVRC.0 = 1;          /* Enables operation */
        ADVRC.1 = 1;          /* Output 2.0 V */
```

```
<3>      ADPC =  0b00000000;     /* A/D Port Configuration Register */
                 /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                 /*||| :          +------------------------- ANI15/AVREFM/P157 */
                 /*||| :          | +-+-+-------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                 /*||| :          | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20
*/
                 /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                 /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                 /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                 /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                 /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                 /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                 /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                 /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                 /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                 /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                 /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                 /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                 /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                 /*||| */
                 /*+++-------- : Be sure to set 000 */

         /* wait for setting time to 17msec */
<4>      fn_Wait500usBase(17000/500);

         /*--------------------------------------------*/
         /*  Initialization of operational amplifier   */
         /*--------------------------------------------*/
<5>      PM2.1 = 1;                      /* input port mode (AMP0O) */
         PM2.0 = 1;                      /* input port mode (AMP0-) */
         PM2.2 = 1;                      /* input port mode (AMP0+) */

         PM2.4 = 1;                      /* input port mode (AMP1O) */
         PM2.3 = 1;                      /* input port mode (AMP1-) */
         PM2.5 = 1;                      /* input port mode (AMP1+) */

         PM2.7 = 1;                      /* input port mode (AMP2O) */
         PM2.6 = 1;                      /* input port mode (AMP2-) */
         PM15.0 = 1;                     /* input port mode (AMP2+) */

         PM11.0 = 1;                     /* input port mode (ANO0) */

<6>      ADCEN = 1;                      /* supplies input clock to OPAmp A/D converter */
         OAC.0 = 1;                      /* operational amplifier (AMP0) enable */
         OAC.1 = 1;                      /* operational amplifier (AMP1) enable */
         OAC.2 = 1;                      /* operational amplifier (AMP2) enable */

         /*--------------------------------------------*/
         /*      Initialization of D/A converter       */
         /*--------------------------------------------*/
<7>      DACEN = 1;                      /* supplies input clock to D/A converter */
<8>      DAM.0 = 0;                      /* [DAMD0] D/A converter operation is Normal mode */
         DAM.2 = 1;                      /* [DARES0]D/A conversion resolution is 12-bit */
         DAM.6 = 1;                      /* [DAREF] D/A converter positive reference voltage is
VREFOUT */
<9>      DACSW0 = CDAC0_52V;             /* D/A conversion value setting reigster 0 */

         /* set D/A chanel1 */
<10>     DAM.1 = 0;                      /* [DAMD1] D/A converter operation is Normal mode */
         DAM.3 = 1;                      /* [DARES1]D/A conversion resolution is 12-bit */
<11>     DACSW1 = CDAC0_50V;             /* D/A conversion value setting reigster 1 */

}
```

4.7.2 Main processing

The following operations are performed during main processing.

<1> Operation of D/A converter channels 0 and 1 is enabled.

<2> Real-time counter interrupt is enabled. Also, real-time counter operation is started at the same time.

<3> Interrupts are enabled.

<4> The mode switches to wait mode until an interrupt occurs.

<5> Output voltage of D/A converter is reset and processing goes to <4>.

```
      void main(void)
      {
<1>           DACE1 = 1;              /* D/A converter CH1 enable */
              DACE0 = 1;              /* D/A converter CH0 enable */

              /*==================================*/
              /*----------------------------------*/
              /*             Main Loop            */
              /*----------------------------------*/
              /*==================================*/

<2>           RTCIF = 0;              /* reset RCT interrupt request */
              RTCMK = 0;              /* enable RTC interrupt */
              RTCE  = 1;              /* starts counter operation */

<3>           EI();                   /* enable all interrupts */

              /*==============================================================*/
              /* if system have nothing to do, go to standby for power-saving */
              /*==============================================================*/
<4>           while (1){
                      HALT();
                      NOP();
              }
      }


      /*------------------------------------------------------------------*/
      /* Module:      fn_intrtc                                           */
      /* Description: RTC interruput process                              */
      /* parameter: --                                                    */
      /* return   : --                                                    */
      /*------------------------------------------------------------------*/
      __interrupt void fn_intrtc(void)
      {
<5>           switch(DACSW0)
              {
                      case    CDAC0_52V:              /* DAC0 output voltage : 0.52V */
                              DACSW0 = CDAC0_54V;     /* DAC0 output voltage : 0.52V -> 0.54V */
                              break;
                      case    CDAC0_54V:              /* DAC0 output voltage : 0.54V */
                      default:
                              DACSW0 = CDAC0_52V;     /* DAC0 output voltage : 0.54V -> 0.52V */
                              break;
              }
      }
```

4.8 Comparator Circuit

4.8.1 Initialization of peripherals to be used

(1) DMA initialization

The following operations are performed during DMA initialization.

<1> Operation of DMA channel 0 is enabled.

<2> SFR address setting for DMA transfer is specified.

<3> RAM address setting for DMA transfer is specified.

<4> The number of DMA transfers is set.

<5> The DMA transfer mode is set as follows:

- DMA transfer start software trigger: Software trigger does not operate
- DMA transfer direction: Internal RAM → SFR
- Transfer data size: 16 bits
- DMA transfer hold: No hold
- DMA start trigger: INTTM04

<6> DMA trigger wait mode is set.

```
void    fn_DmaInit(void)
{

<1>-----        DEN0 = 1;                               /* Enables operation of DMA channel 0 */

<2>-----        DSA0 = (unsigned char)&DACSW0;          /* DMA SFR Address Register 0 */
<3>-----        DRA0 = (unsigned short)&ushDacData[0];  /* DMA RAM Address Register 0 */
<4>-----        DBC0 = 100;                             /* DMA Byte Count Register 0 */
<5>-----        DMC0 =  0b01100100;                     /* DMA Mode Control Register 0 */
                        /*|||||||| */
                        /*||||++++------- IFC3-IFC0 */
                        /*||||           [Selection of DMA start source] */
                        /*||||            0000:Disables DMA transfer by interrupt */
                        /*||||                  (Only software trigger is enabled.) */
                        /*||||            0010:INTTM00 */
                        /*||||            0011:INTTM01 */
                        /*||||            0100:INTTM04 */
                        /*||||            0101:INTTM05 */
                        /*||||            0110:INTST0/INTCSI00 */
                        /*||||            0111:INTSR0/INTCSI01 */
                        /*||||            1000:INTST1/INTCSI10/INTIIC10 */
                        /*||||            1001:INTSR1 */
                        /*||||            1010:INTST3 */
                        /*||||            1011:INTSR3 */
                        /*||||            1100:INTAD */
                        /*|||+----------- DWAIT0 */
                        /*|||            [Pending of DMA transfer] */
                        /*|||             0:Executes DMA transfer upon DMA start request (not held
pending). */
                        /*|||             1:Holds DMA start request pending if any. */
                        /*||+------------ DS0 */
                        /*||             [Specification of transfer data size for DMA transfer] */
                        /*||              0:8bits */
                        /*||              1:16bits */
                        /*|+------------- DRS0 */
                        /*|              [Selection of DMA transfer direction] */
                        /*|               0:SFR to internal RAM */
                        /*|               1:Internal RAM to SFR */
                        /*+------------- STG0 */
                        /*               [DMA transfer start software trigger] */
                        /*                0:No trigger operation */
                        /*                1:DMA transfer is started when DMA operation is enabled (DEN0
= 1). */

<6>-----        DST0 = 1;               /* DMA transfer is started when DMA operation is enabled (DEN0
= 1). */

}
```

(2) Op amp and D/A converter initialization

The following operations are performed during op amp and D/A converter initialization.

<1> An input clock is supplied to the A/D converter.

<2> The voltage reference is set as follows:

- Positive reference voltage source of D/A converter: V_{REFOUT}
- Voltage reference operation control: Operation is enabled
- Voltage reference output voltage: 2.0 V

<3> Port 2 and port 15 are set for analog input.

<4> Wait mode is in effect until settings completed.

<5> Ports to be used for analog input are set to input mode.

<6> Operation of op amp 0 is enabled.

<7> An input clock is supplied to the D/A converter.

<8> D/A converter channel 0 is set as follows:

- Operating mode of D/A converter: Normal output mode
- Resolution of D/A converter: 12 bits
- Positive reference voltage source of D/A converter: V_{REFOUT}

<9> D/A converter channel 0 is initialized.

<10> D/A converter channel 1 is set as follows:

- Operating mode of D/A converter: Normal output mode
- Resolution of D/A converter: 12 bits

<11> D/A converter channel 1 is initialized.

<1>

<2>

```
void    fn_AdcDacInit(void)
{
        ADCEN = 1;              /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

        /*------------------------------------------*/
        /*    Initialization of Voltage reference    */
        /*------------------------------------------*/
        ADVRC = 0b00001000;     /* Analog reference voltage control register */
                /*||||+|++--- : [1]Positive reference voltage supplies selection of A/D and D/A
converters */
                /*|||| |        [2]Operationcontrol of voltage reference */
                /*|||| |        [3]Output voltage selection of voltage reference */
                /*|||| |        [4]Operation control of input gate voltage boost circuit for
A/D converter */
                /*|||| |        [5]Relationship with the conversion mode used */
                /*|||| | */
                /*|||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
                /*|||| |           [2]Stops operation (Hi-Z) */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Stops operation */
                /*|||| |           [5]Can be set in conversion mode 1 */
                /*|||| | */
                /*|||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
                /*|||| |           [2]Stops operation (Hi-Z) */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Stops operation (pull-down output) */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Stops operation */
                /*|||| |           [5] - */
                /*|||| | */
                /*|||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Enables operation */
                /*|||| |           [3]2.5 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Stops operation (pull-down output) */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5] - */
                /*|||| | */
                /*|||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
                /*|||| |           [2]Enables operation */
                /*|||| |           [3]2.0 V */
                /*|||| |           [4]Enables operation */
                /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                /*|||| | */
                /*|||| | : Other than the above : Setting prohibited */
                /*|||| | */
                /*|+++-+----- : Be sure to set 0000 */
                /*| */
                /*+---------- : Reference voltage supply (negative side) of A/D converter
selection */
                /* : 0 : AVSS */
                /* : 1 : AVREFM (external voltage reference input) */

        ADVRC.0 = 1;            /* Enables operation */
        ADVRC.1 = 1;            /* Output 2.0 V */
```

```
<3>        ADPC =  0b00000000;      /* A/D Port Configuration Register */
                   /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                   /*||| :           +------------------------- ANI15/AVREFM/P157 */
                   /*||| :           | +-+-+-------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                   /*||| :           | | | | +-+-+-+-+-+-+-+---- ANI7/AMP2O/P27 - ANI0/AMP0-/P20
*/
                   /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                   /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                   /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                   /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                   /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                   /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                   /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                   /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                   /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                   /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                   /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                   /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                   /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                   /*||| */
                   /*+++-------- : Be sure to set 000 */

           /* wait for setting time to 17msec */
<4>        fn_Wait500usBase(17000/500);

           /*-------------------------------------------*/
           /*  Initialization of operational amplifier   */
           /*-------------------------------------------*/
<5>        PM2.1 = 1;                      /* input port mode (AMP0O) */
           PM2.0 = 1;                      /* input port mode (AMP0-) */
           PM2.2 = 1;                      /* input port mode (AMP0+) */

           PM2.4 = 1;                      /* input port mode (AMP1O) */
           PM2.3 = 1;                      /* input port mode (AMP1-) */
           PM2.5 = 1;                      /* input port mode (AMP1+) */

           PM2.7 = 1;                      /* input port mode (AMP2O) */
           PM2.6 = 1;                      /* input port mode (AMP2-) */
           PM15.0 = 1;                     /* input port mode (AMP2+) */

           PM11.0 = 1;                     /* input port mode (ANO0) */

<6>        ADCEN = 1;                      /* supplies input clock to OPAmp A/D converter */
           OAC.0 = 1;                      /* operational amplifier (AMP0) enable */
           OAC.1 = 0;                      /* operational amplifier (AMP1) disable */
           OAC.2 = 0;                      /* operational amplifier (AMP2) disable */

           /*-------------------------------------------*/
           /*      Initialization of D/A converter      */
           /*-------------------------------------------*/
<7>        DACEN = 1;                      /* supplies input clock to D/A converter */
<8>        DAM.0 = 0;                      /* [DAMD0] D/A converter operation is Normal mode */
           DAM.2 = 1;                      /* [DARES0]D/A conversion resolution is 12-bit */
           DAM.6 = 1;                      /* [DAREF] D/A converter positive reference voltage is
VREFOUT */
<9>        DACSW0 = aushDacData[99];       /* D/A conversion value setting reigster 0 */

<10>       /* set D/A chanel1 */
           DAM.1 = 0;                      /* [DAMD1] D/A converter operation is Normal mode */
           DAM.3 = 1;                      /* [DARES1]D/A conversion resolution is 12-bit */
<11>       DACSW1 = aushDacData[99];       /* D/A conversion value setting reigster 1 */

}
```

4.8.2 Main processing

The following operations are performed during main processing.

<1> P7.3 is set as an output port and P12.0 is set as an input port.
<2> P12.0 is set for dual edge detection.
<3> Operation of D/A converter channels 0 and 1 is enabled.
<4> Sine wave output table is copied from ROM to RAM.
<5> TM04 interrupt is enabled.
<6> DMA transfer channel 0 interrupt is enabled.
<7> INTP0 interrupt is enabled.
<8> Interrupts are enabled.
<9> The mode switches to wait mode until an interrupt occurs.
<10> DMA transfer is reset and processing goes to <9>.
<11> Output level of P7.3 is matched to P12.0, and processing goes to <9>.

```
      void main(void)
      {
              unsigned short  count;          /* counter for initializing D/A convert data */

              /* Port setting */
              P12.0 = 0;
<1>           PM12.0 = 1;
              P7.3 = 0;
              PM7.3 = 0;

              /* INTP0 pin valid edge selection */
<2>           EGP0.0 = 1;                     /* Both rising and falling edges */
              EGN0.0 = 1;

<3>           DACE1 = 1;                      /* D/A converter CH1 enable */
              DACE0 = 1;

              /*==================================*/
              /*----------------------------------*/
              /*             Main Loop            */
              /*----------------------------------*/
              /*==================================*/

              /*  initialize D/Aconverter data */
<4>           for(count = 0; count<100; count++){
                      ushDacData[count] = aushDacData[count];
              }

<5>           TS0L.4 = 1;                     /* start TAU0 CH4 */
              TMIF04 = 0;                     /* reset INTTM04 interrupt request */

<6>           DMAMK0 = 0;                     /* enable DMA CH0 interrupt */

<7>           PIF0 = 0;                       /* reset INTP0 interrupt request */
              PMK0 = 0;                       /* enable INTP0 interrupt */

<8>           EI();                           /* enable all interrupts */

              /*============================================================*/
              /* if system have nothing to do, go to standby for power-saving */
              /*============================================================*/
              while (1){
<9>                   HALT();
                      NOP();
              }
      }
```

```
/*--------------------------------------------------------------*/
/* Module:      fn_intdma0                                      */
/* Description: DMA ch0 interruput process                      */
/* parameter: --                                                */
/* return   : --                                                */
/*--------------------------------------------------------------*/
__interrupt void fn_intdma0(void)
```

<10>

```
{
        DRA0 = (unsigned short)&ushDacData[0];  /* DMA RAM Address Register 0 */
        DBC0 = 100;                             /* DMA Byte Count Register 0 */

        DST0 = 1;       /* DMA transfer is started when DMA operation is enabled (DEN0 = 1).
*/
}
```

```
/*--------------------------------------------------------------*/
/* Module:      fn_intp0                                        */
/* Description: INTP0 interruput process                        */
/* parameter: --                                                */
/* return   : --                                                */
/*--------------------------------------------------------------*/
```

<11>

```
__interrupt void fn_intp0(void)
{
        P7.3 = P12.0;
}
```

4.9 CR Oscillator 1

4.9.1 Initialization of peripherals to be used

(1) Op amp and D/A converter initialization

The following operations are performed during op amp and D/A converter initialization.

<1> An input clock is supplied to the A/D converter.

<2> Port 2 and port 15 are set for analog input.

<3> Ports to be used for analog input are set to input mode.

<4> Operation of op amp 0 is enabled.

<5> An input clock is supplied to the D/A converter.

<6> D/A converter channel 1 is set as follows:

- Operating mode of D/A converter: Normal output mode
- Resolution of D/A converter: 12 bits

<7> D/A converter channel 1 is initialized.

```
      void    fn_AdcDacInit(void)
      {
<1>           ADCEN = 1;              /* supplies input clock to A/D converter and operational */

<2>           ADPC =  0b00000000;     /* A/D Port Configuration Register */
                      /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                      /*||| :          +------------------------- ANI15/AVREFM/P157 */
                      /*||| :          | +-+-+-------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                      /*||| :          | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20
      */
                      /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                      /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                      /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                      /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                      /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                      /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                      /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                      /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                      /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                      /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                      /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                      /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                      /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                      /*||| */
                      /*+++-------- : Be sure to set 000 */

              /*-------------------------------------------*/
              /*  Initialization of operational amplifier   */
              /*-------------------------------------------*/
<3>           PM2.1 = 1;              /* input port mode (AMP0O) */
              PM2.0 = 1;              /* input port mode (AMP0-) */
              PM2.2 = 1;              /* input port mode (AMP0+) */

              PM2.4 = 1;              /* input port mode (AMP1O) */
              PM2.3 = 1;              /* input port mode (AMP1-) */
              PM2.5 = 1;              /* input port mode (AMP1+) */

              PM2.7 = 1;              /* input port mode (AMP2O) */
              PM2.6 = 1;              /* input port mode (AMP2-) */
              PM15.0 = 1;             /* input port mode (AMP2+) */

              PM11.0 = 1;             /* input port mode (ANO0) */

<4>           ADCEN = 1;              /* supplies input clock to OPAmp A/D converter */
              OAC.0 = 1;              /* operational amplifier (AMP0) enable */
              OAC.1 = 0;              /* operational amplifier (AMP1) disable */
              OAC.2 = 0;              /* operational amplifier (AMP2) disable */

              /*-------------------------------------------*/
              /*      Initialization of D/A converter       */
              /*-------------------------------------------*/
<5>           DACEN = 1;              /* supplies input clock to D/A converter */

<6>           /* set D/A chanel1 */
              DAM.1 = 0;              /* [DAMD1] D/A converter operation is Normal mode */
              DAM.3 = 1;              /* [DARES1]D/A conversion resolution is 12-bit */
<7>           DACSW1 = 2048;          /* D/A conversion value setting reigster 1 */

      }
```

(2) LCD controller/driver initialization

The following operations are performed during LCD controller/driver initialization.

<1> The LCD drive voltage generator is set to internal voltage boosting method.

<2> The SEG8 to SEG26 pins are set to segment output enable.

<3> The P50 to P57, P90 to P97, P100 to P102, and P140 to P147 pins are set to segment output.

<4> The input switch control register is set as follows:

- Schmitt trigger buffer control of TI04/SEG50/P53: Input disabled
- Schmitt trigger buffer control of TI02/SEG51/P52: Input disabled
- Schmitt trigger buffer control of RxD3/SEG53/P50: Input disabled

<5> The LCD RAM area is cleared.

<6> LCD clock control register 0 is set as follows:

- LCD source clock (f_{LCD}): $f_{CLK}/2^8$
- LCD clock (LCDCL) selection: $f_{LCD}/2^7$

<7> Voltage V_{LC0} is set to 5.0 V.

<8> A 2 ms wait is inserted.

<9> The LCD display mode register is set as follows:

- LCD display enable/disable: Ground level output to segment pin and common pin
- Enable/stop operation of booster circuit and capacitance splitting circuit: Operation enabled
- Display data area: Data in A pattern area is displayed
- Display mode of LCD controller/driver: Eight time divisions and 1/4 bias

<10> A 500 ms wait is inserted.

<11> LCD display is set to ON.

```
        void    fn_DisplayInit(void)
        {
                /*----------------------------------------*/
                /* Initialization of LCD controler/driver */
                /*----------------------------------------*/
<1>-----        LCDMD = 0b00010000;             /* LCD Mode Register */
                        /*++||++++--- : Be sure to set 000000 */
                        /*  || */
                        /*  ++------- : LCD drive voltage generator selection */
                        /*   : 0 0 : External resistance division method */
                        /*   : 0 1 : Internal voltage boosting method */
                        /*   : 1 0 : Capacitance split method */
                        /*   : 1 1 : Setting prohibited */
<2>-----        SEGEN = 0b00011111;             /* Segment Enable Register */
                        /*|||||||+--- : Control segment signal output from pins SEG8-SEG11 */
                        /*||||||+---- : Control segment signal output from pins SEG12-SEG15 */
                        /*|||||+----- : Control segment signal output from pins SEG16-SEG19 */
                        /*||||+------ : Control segment signal output from pins SEG20-SEG23 */
                        /*|||+------- : Control segment signal output from pins SEG24-SEG26 */
                        /*||| : 0 : segment signal output disable */
                        /*||| : 1 : segment signal output enable */
                        /*||| */
                        /*+++-------- : Be sure to set 000 */
<3>-----        PFALL = 0b01111111;             /* Port Function Register ALL */
                        /*|||||||+--- : Pins P50-P53 port/segment output specification */
                        /*||||||+---- : Pins P54-P57 port/segment output specification */
                        /*|||||+----- : Pins P90-P93 port/segment output specification */
                        /*||||+------ : Pins P94-P97 port/segment output specification */
                        /*|||+------- : Pins P100-P102 port/segment output specification */
                        /*||+-------- : Pins P140-P143 port/segment output specification */
                        /*|+--------- : Pins P144-P147 port/segment output specification */
                        /*| : 0 : Used the pins as port (other than segment output) */
                        /*| : 1 : Used the pins as segment output */
                        /*| */
                        /*+---------- : Be sure to set 0 */
<4>-----        ISC =   0b00000000;             /* Input Switch Control Register */
                        /*|||||||+--- : Switching external interrupt (INTP0) input */
                        /*||||||| : 0 : Uses the input signal of the INTP0 pin as an external interrupt
        (normal operation). */
                        /*||||||| : 1 : Uses the input signal of the RXD3 pin as an external interrupt
        */
                        /*|||||||     (to measure the pulse widths of the sync break field and sync field).
        */
                        /*||||||| */
                        /*||||||+---- : ISC1 Switching channel 7 input of timer array unit TAUS */
                        /*|||||| : 0 : Uses the input signal of the TI07 pin as a timer input (normal
        operation). */
                        /*|||||| : 1 : Input signal of RXD3 pin is used as timer input (wakeup signal
        detection). */
                        /*|||||| */
                        /*|||||+----- : RxD3/SEG53/P50 pin schmitt-triggered buffer control */
                        /*||||| : 0 : Disables input */
                        /*||||| : 1 : Enables input */
                        /*||||| */
                        /*||||+------ : TI02/SEG51/P52 pin schmitt-triggered buffer control */
                        /*|||| : 0 : Disables input */
                        /*|||| : 1 : Enables input */
                        /*|||| */
                        /*|||+------- : TI04/SEG50/P53 pin schmitt-triggered buffer control */
                        /*||| : 0 : Disables input */
                        /*||| : 1 : Enables input */
                        /*||| */
                        /*+++-------- : Be sure to set 000 */

<5>-----        fn_DisplayAllClear();           /* clear all LCD String area */
```

```
<6>-----    LCDC0 = 0b00110011;             /* LCD Clock Control Register */
                    /*|||||+++--- : LCD clock (LCDCL) selection */
                    /*||||| : 0 0 0 : fLCD/2^4 */
                    /*||||| : 0 0 1 : fLCD/2^5 */
                    /*||||| : 0 1 0 : fLCD/2^6 */
                    /*||||| : 0 1 1 : fLCD/2^7 */
                    /*||||| : 1 0 0 : fLCD/2^8 */
                    /*||||| : 1 0 1 : fLCD/2^9 */
                    /*||||| : Other than above : Setting prohibited */
                    /*||||| */
                    /*++||+------ : Be sure to set 000 */
                    /*  || */
                    /*  ++------- : LCD source clock (fLCD) selection */
                    /* : 0 0 : fSUB */
                    /* : 0 1 : fCLK/2^6 */
                    /* : 1 0 : fCLK/2^7 */
                    /* : 1 1 : fCLK/2^8 */

<7>-----    VLCD = 0x0A;                    /* set LCD boost level to 5V */
            /* wait for the reference voltage setup time (2ms(min.)) */
<8>-----    fn_Wait500usBase(2000/500);

<9>-----    LCDM =  0b00100111;             /* LCD Display Mode Register */
                    /*|||||+++--- : LCD controller/driver display mode selection */
                    /*|||||  *When the external resistance division method is used */
                    /*|||||  : 0 0 0 : Four-time-slice mode & 1/3 bias method */
                    /*|||||  : 0 0 1 : Three-time-slice mode & 1/3 bias method */
                    /*|||||  : 0 1 0 : Two-time-slice mode & 1/2 bias method */
                    /*|||||  : 0 1 1 : Three-time-slice mode & 1/2 bias method */
                    /*|||||  : 1 0 0 : Static */
                    /*|||||  : 1 1 1 : Eight-time-slice mode & 1/4 bias method */
                    /*||||| */
                    /*|||||  *When the internal voltage boosting method is used */
                    /*|||||  : 0 0 0 : Four-time-slice mode & 1/3 bias method */
                    /*|||||  : 0 0 1 : Three-time-slice mode & 1/3 bias method */
                    /*|||||  : 0 1 0 : Four-time-slice mode & 1/3 bias method */
                    /*|||||  : 0 1 1 : Four-time-slice mode & 1/3 bias method */
                    /*|||||  : 1 0 0 : Setting prohibited */
                    /*|||||  : 1 1 1 : Eight-time-slice mode & 1/4 bias method */
                    /*||||| */
                    /*|||||  *When the capacitor split method is used */
                    /*|||||  : 0 0 0 : Four-time-slice mode & 1/3 bias method */
                    /*|||||  : 0 0 1 : Three-time-slice mode & 1/3 bias method */
                    /*|||||  : 0 1 0 : Four-time-slice mode & 1/3 bias method */
                    /*|||||  : 0 1 1 : Four-time-slice mode & 1/3 bias method */
                    /*|||||  : 1 0 0 : Setting prohibited */
                    /*|||||  : 1 1 1 : Four-time-slice mode & 1/3 bias method */
                    /*||||| */
                    /*|||||  : Other than above : Setting prohibited
                    /*||||| */
                    /*|||++------ : LCD display data area control */
                    /*||| : 0 0 : Display the data of an A pattern area */
                    /*|||         (lower 4 bits of LCD display data memory) */
                    /*||| : 0 1 : Display the data of an A pattern area */
                    /*|||         (higher 4 bits of LCD display data memory) */
                    /*||| : 1 0 : Display the data of an A pattern area and the B pattern area in
turn. */
                    /*|||         (The on and off light indication which synchronized */
                    /*|||          in a constant-period interrupt timing of RTC) */
                    /*||| : 1 1 : Display the data of an A pattern area and the B pattern area in
turn. */
                    /*|||         (The on and off light indication which synchronized */
                    /*|||          in a constant-period interrupt timing of RTC) */
                    /*||| */
```

```
                /*||+-------- : Voltage boost circuit and capacitor split circuit operation
enable/disable */
                /*|| : 0 : Stops voltage boost circuit and capacitor split circuit operation
*/
                /*|| : 1 : Enables voltage boost circuit and capacitor split circuit operation
*/
                /*|| */
                /*++--------- : LCD display enable/disable */
                /* : 0 0 : Output ground level to segment/common pin */
                /* : 0 1 : Display off (all segment outputs are deselected.) */
                /* : 1 0 : Output ground level to segment/common pin */
                /* : 1 1 : Display on */

        /* software to wait for the operation stabilization time (over 500ms) */
<10>    fn_Wait500usBase(500000/500);

<11>    SCOC = 1;                       /* output deselect level to SEG and LCD waveform to COM
*/
        LCDON = 1;                      /* display on */
}
```

4.9.2 Main processing

The following operations are performed during main processing.

<1> Operation of D/A converter channel 1 is enabled.
<2> Operation of TM04 and TM06 is started.
<3> Wait continues until 1 ms has elapsed.
<4> Count value for 1 ms period is retrieved.
<5> Operation of TM04 and TM06 is stopped.
<6> Count value is displayed on LCD.

```
     void main(void)
     {
             unsigned short  ushCount;       /* Pulse counter */

<1>          DACE1 = 1;                      /* D/A converter CH1 enable */
             ushCount = 0;

             /*==================================*/
             /*----------------------------------*/
             /*            Main Loop             */
             /*----------------------------------*/
             /*==================================*/

             /*==============================================================*/
             /* if system have nothing to do, go to standby for power-saving */
             /*==============================================================*/

             while (1){
<2>                  TS0L.6 = 1;             /* start TAU0 CH6 (pulse counter) */
                     TS0L.4 = 1;             /* start TAU0 CH4 (interval timer) */
                     TMIF04 = 0;             /* clear interval timer request */

<3>                  while(TMIF04 == 0){     /* wait interval */
                             NOP();
                     }

<4>                  ushCount = TCR06;       /* get pulse count */

<5>                  TT0L.6 = 1;             /* stop TAU0 CH6  (pulse counter)*/
                     TT0L.4 = 1;             /* stop TAU0 CH4 (interval timer) */

<6>                  ushCount = 0xffff - ushCount;   /* get pulse count */
                     fn_Display(ushCount);           /* display pulse count */
             }
     }
```

4.10 CR Oscillator 2

4.10.1 Initialization of peripherals to be used

(1) Op amp and D/A converter initialization

The following operations are performed during op amp and D/A converter initialization.

<1> An input clock is supplied to the A/D converter.

<2> Port 2 and port 15 are set for analog input.

<3> Ports to be used for analog input are set to input mode.

<4> Operation of op amp 0 is enabled.

<5> An input clock is supplied to the D/A converter.

<6> D/A converter channel 1 is set as follows:

- Operating mode of D/A converter: Normal output mode
- Resolution of D/A converter: 12 bits

<7> D/A converter channel 1 is initialized.

```
        void     fn_AdcDacInit(void)
        {
<1>              ADCEN = 1;              /* supplies input clock to A/D converter and operational */

<2>              ADPC =  0b00000000;     /* A/D Port Configuration Register */
                         /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                         /*||| :           +------------------------- ANI15/AVREFM/P157 */
                         /*||| :           | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                         /*||| :           | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20
        */
                         /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                         /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                         /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                         /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                         /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                         /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                         /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                         /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                         /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                         /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                         /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                         /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                         /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                         /*||| */
                         /*+++--------- : Be sure to set 000 */

                 /*------------------------------------------*/
                 /*  Initialization of operational amplifier   */
                 /*------------------------------------------*/
                 PM2.1 = 1;              /* input port mode (AMP0O) */
                 PM2.0 = 1;              /* input port mode (AMP0-) */
                 PM2.2 = 1;              /* input port mode (AMP0+) */

                 PM2.4 = 1;              /* input port mode (AMP1O) */
<3>              PM2.3 = 1;              /* input port mode (AMP1-) */
                 PM2.5 = 1;              /* input port mode (AMP1+) */

                 PM2.7 = 1;              /* input port mode (AMP2O) */
                 PM2.6 = 1;              /* input port mode (AMP2-) */
                 PM15.0 = 1;             /* input port mode (AMP2+) */

                 PM11.0 = 1;             /* input port mode (ANO0) */

<4>              ADCEN = 1;              /* supplies input clock to OPAmp A/D converter */
                 OAC.0 = 1;              /* operational amplifier (AMP0) enable */
                 OAC.1 = 0;              /* operational amplifier (AMP1) disable */
                 OAC.2 = 0;              /* operational amplifier (AMP2) disable */

                 /*------------------------------------------*/
                 /*      Initialization of D/A converter      */
                 /*------------------------------------------*/
<5>              DACEN = 1;              /* supplies input clock to D/A converter */

<6>              /* set D/A chanel1 */
                 DAM.1 = 0;              /* [DAMD1] D/A converter operation is Normal mode */
                 DAM.3 = 1;              /* [DARES1]D/A conversion resolution is 12-bit */
<7>              DACSW1 = 1365;          /* D/A conversion value setting reigster 1 */
                                         /* ANO1 = 1V (1/3 * 2^12) */
        }
```

(2) LCD controller/driver initialization

The following operations are performed during LCD controller/driver initialization.

<1> The LCD drive voltage generator is set to internal voltage boosting method.

<2> The SEG8 to SEG26 pins are set to segment output enable.

<3> The P50 to P57, P90 to P97, P100 to P102, and P140 to P147 pins are set to segment output.

<4> The input switch control register is set as follows:

- Schmitt trigger buffer control of TI04/SEG50/P53: Input disabled
- Schmitt trigger buffer control of TI02/SEG51/P52: Input disabled
- Schmitt trigger buffer control of RxD3/SEG53/P50: Input disabled

<5> The LCD RAM area is cleared.

<6> LCD clock control register 0 is set as follows:

- LCD source clock (f_{LCD}): $f_{CLK}/2^8$
- LCD clock (LCDCL) selection: $f_{LCD}/2^7$

<7> Voltage V_{LC0} is set to 5.0 V.

<8> A 2 ms wait is inserted.

<9> The LCD display mode register is set as follows:

- LCD display enable/disable: Ground level output to segment pin and common pin
- Enable/stop operation of booster circuit and capacitance splitting circuit: Operation enabled
- Display data area: Data in A pattern area is displayed
- Display mode of LCD controller/driver: Eight time divisions and 1/4 bias

<10> A 500 ms wait is inserted.

<11> LCD display is set to ON.

```
     void    fn_DisplayInit(void)
     {
             /*---------------------------------------*/
             /* Initialization of LCD controler/driver */
             /*---------------------------------------*/
<1>-----     LCDMD = 0b00010000;             /* LCD Mode Register */
                     /*++||++++--- : Be sure to set 000000 */
                     /*  || */
                     /*  ++------- : LCD drive voltage generator selection */
                     /*   : 0 0 : External resistance division method */
                     /*   : 0 1 : Internal voltage boosting method */
                     /*   : 1 0 : Capacitance split method */
                     /*   : 1 1 : Setting prohibited */
<2>-----     SEGEN = 0b00011111;             /* Segment Enable Register */
                     /*|||||||+--- : Control segment signal output from pins SEG8-SEG11 */
                     /*||||||+---- : Control segment signal output from pins SEG12-SEG15 */
                     /*|||||+----- : Control segment signal output from pins SEG16-SEG19 */
                     /*||||+------ : Control segment signal output from pins SEG20-SEG23 */
                     /*|||+------- : Control segment signal output from pins SEG24-SEG26 */
                     /*||| : 0 : segment signal output disable */
                     /*||| : 1 : segment signal output enable */
                     /*||| */
                     /*+++-------- : Be sure to set 000 */
<3>-----     PFALL = 0b01111111;             /* Port Function Register ALL */
                     /*|||||||+--- : Pins P50-P53 port/segment output specification */
                     /*||||||+---- : Pins P54-P57 port/segment output specification */
                     /*|||||+----- : Pins P90-P93 port/segment output specification */
                     /*||||+------ : Pins P94-P97 port/segment output specification */
                     /*|||+------- : Pins P100-P102 port/segment output specification */
                     /*||+-------- : Pins P140-P143 port/segment output specification */
                     /*|+--------- : Pins P144-P147 port/segment output specification */
                     /*| : 0 : Used the pins as port (other than segment output) */
                     /*| : 1 : Used the pins as segment output */
                     /*| */
                     /*+---------- : Be sure to set 0 */
<4>-----     ISC =   0b00000000;             /* Input Switch Control Register */
                     /*|||||||+--- : Switching external interrupt (INTP0) input */
                     /*||||||| : 0 : Uses the input signal of the INTP0 pin as an external interrupt
     (normal operation). */
                     /*||||||| : 1 : Uses the input signal of the RXD3 pin as an external interrupt
     */
                     /*|||||||     (to measure the pulse widths of the sync break field and sync field).
     */
                     /*||||||| */
                     /*||||||+---- : ISC1 Switching channel 7 input of timer array unit TAUS */
                     /*|||||| : 0 : Uses the input signal of the TI07 pin as a timer input (normal
     operation). */
                     /*|||||| : 1 : Input signal of RXD3 pin is used as timer input (wakeup signal
     detection). */
                     /*|||||| */
                     /*|||||+----- : RxD3/SEG53/P50 pin schmitt-triggered buffer control */
                     /*||||| : 0 : Disables input */
                     /*||||| : 1 : Enables input */
                     /*||||| */
                     /*||||+------ : TI02/SEG51/P52 pin schmitt-triggered buffer control */
                     /*|||| : 0 : Disables input */
                     /*|||| : 1 : Enables input */
                     /*|||| */
                     /*|||+------- : TI04/SEG50/P53 pin schmitt-triggered buffer control */
                     /*||| : 0 : Disables input */
                     /*||| : 1 : Enables input */
                     /*||| */
                     /*+++-------- : Be sure to set 000 */

<5>-----     fn_DisplayAllClear();           /* clear all LCD String area */
```

```
<6>-----    LCDC0 = 0b00110011;             /* LCD Clock Control Register */
                    /*|||||+++--- : LCD clock (LCDCL) selection */
                    /*||||| : 0 0 0 : fLCD/2^4 */
                    /*||||| : 0 0 1 : fLCD/2^5 */
                    /*||||| : 0 1 0 : fLCD/2^6 */
                    /*||||| : 0 1 1 : fLCD/2^7 */
                    /*||||| : 1 0 0 : fLCD/2^8 */
                    /*||||| : 1 0 1 : fLCD/2^9 */
                    /*||||| : Other than above : Setting prohibited */
                    /*||||| */
                    /*++||+------ : Be sure to set 000 */
                    /*  || */
                    /*  ++------- : LCD source clock (fLCD) selection */
                    /* : 0 0 : fSUB */
                    /* : 0 1 : fCLK/2^6 */
                    /* : 1 0 : fCLK/2^7 */
                    /* : 1 1 : fCLK/2^8 */

<7>-----    VLCD = 0x0A;                    /* set LCD boost level to 5V */
            /* wait for the reference voltage setup time (2ms(min.)) */
<8>-----    fn_Wait500usBase(2000/500);

<9>-----    LCDM =  0b00100111;             /* LCD Display Mode Register */
                    /*|||||+++--- : LCD controller/driver display mode selection */
                    /*|||||  *When the external resistance division method is used */
                    /*|||||  : 0 0 0 : Four-time-slice mode & 1/3 bias method */
                    /*|||||  : 0 0 1 : Three-time-slice mode & 1/3 bias method */
                    /*|||||  : 0 1 0 : Two-time-slice mode & 1/2 bias method */
                    /*|||||  : 0 1 1 : Three-time-slice mode & 1/2 bias method */
                    /*|||||  : 1 0 0 : Static */
                    /*|||||  : 1 1 1 : Eight-time-slice mode & 1/4 bias method */
                    /*||||| */
                    /*|||||  *When the internal voltage boosting method is used */
                    /*|||||  : 0 0 0 : Four-time-slice mode & 1/3 bias method */
                    /*|||||  : 0 0 1 : Three-time-slice mode & 1/3 bias method */
                    /*|||||  : 0 1 0 : Four-time-slice mode & 1/3 bias method */
                    /*|||||  : 0 1 1 : Four-time-slice mode & 1/3 bias method */
                    /*|||||  : 1 0 0 : Setting prohibited */
                    /*|||||  : 1 1 1 : Eight-time-slice mode & 1/4 bias method */
                    /*||||| */
                    /*|||||  *When the capacitor split method is used */
                    /*|||||  : 0 0 0 : Four-time-slice mode & 1/3 bias method */
                    /*|||||  : 0 0 1 : Three-time-slice mode & 1/3 bias method */
                    /*|||||  : 0 1 0 : Four-time-slice mode & 1/3 bias method */
                    /*|||||  : 0 1 1 : Four-time-slice mode & 1/3 bias method */
                    /*|||||  : 1 0 0 : Setting prohibited */
                    /*|||||  : 1 1 1 : Four-time-slice mode & 1/3 bias method */
                    /*||||| */
                    /*|||||  : Other than above : Setting prohibited
                    /*||||| */
                    /*|||++------ : LCD display data area control */
                    /*||| : 0 0 : Display the data of an A pattern area */
                    /*|||        (lower 4 bits of LCD display data memory) */
                    /*||| : 0 1 : Display the data of an A pattern area */
                    /*|||        (higher 4 bits of LCD display data memory) */
                    /*||| : 1 0 : Display the data of an A pattern area and the B pattern area in
turn. */
                    /*|||        (The on and off light indication which synchronized */
                    /*|||         in a constant-period interrupt timing of RTC) */
                    /*||| : 1 1 : Display the data of an A pattern area and the B pattern area in
turn. */
                    /*|||        (The on and off light indication which synchronized */
                    /*|||         in a constant-period interrupt timing of RTC) */
                    /*||| */
```

```
                /*||+-------- : Voltage boost circuit and capacitor split circuit operation
enable/disable */
                /*|| : 0 : Stops voltage boost circuit and capacitor split circuit operation
*/
                /*|| : 1 : Enables voltage boost circuit and capacitor split circuit operation
*/
                /*|| */
                /*++--------- : LCD display enable/disable */
                /* : 0 0 : Output ground level to segment/common pin */
                /* : 0 1 : Display off (all segment outputs are deselected.) */
                /* : 1 0 : Output ground level to segment/common pin */
                /* : 1 1 : Display on */

        /* software to wait for the operation stabilization time (over 500ms) */
<10>    fn_Wait500usBase(500000/500);

<11>    SCOC = 1;                       /* output deselect level to SEG and LCD waveform to COM
*/
        LCDON = 1;                      /* display on */
}
```

4.10.2 Main processing

The following operations are performed during main processing.

<1> Operation of D/A converter channel 1 is enabled.
<2> P7.1 and P7.2 are set as input ports.
<3> Interrupts are enabled.
<4> LCD display is initialized.
<5> D/A converter channel 1 output voltage is set to 1 V.
<6> For six clock cycles, P7.1 is set to a high-level output port and then is set back to an input port.
<7> P7.1 is set to a low-level output port and the count value from that period is retrieved. Afterward, P7.1 is set back to an input port.
<8> D/A converter channel 1 output voltage is set to 2 V.
<9> For six clock cycles, P7.1 is set to a high-level output port and then is set back to an input port.
<10> P7.1 is set to a low-level output port and the count value from that period is retrieved. Afterward, P7.1 is set back to an input port.
<11> D/A converter channel 1 output voltage is set to 1 V.
<12> For six clock cycles, P7.2 is set to a high-level output port and then is set back to an input port.
<13> P7.2 is set to a low-level output port and the count value from that period is retrieved. Afterward, P7.2 is set back to an input port.
<14> D/A converter channel 1 output voltage is set to 2 V.
<15> For six clock cycles, P7.2 is set to a high-level output port and then is set back to an input port.
<16> P7.2 is set to a low-level output port and the count value from that period is retrieved. Afterward, P7.2 is set back to an input port.
<17> Using count values from the respective ports of P7.1 and P7.2, the count value during 1.0 V operation is subtracted from the count value during 2.0 V operation, and the remainder is displayed on the LCD.

```
      void main(void)
      {
              unsigned short  count;          /* counter for initializing D/A convert data */
              unsigned short  *pMeasure;      /* pointer for measuring data */


<1>           DACE1 = 1;                      /* D/A converter CH1 enable */

              ucMeasureStatus = 0;            /* measuring status */

              /*==================================*/
              /*----------------------------------*/
              /*            Main Loop             */
              /*----------------------------------*/
              /*==================================*/

<2>           PM7.1 = 1;                      /* set R_REF HiZ(disconnect) */
              PM7.2 = 1;                      /* set R_REF HiZ(disconnect) */

<3>           EI();                           /* enable all interrupts */

<4>           fn_Display(0, 0);               /* initial display */

              /*============================================================*/
              /* if system have nothing to do, go to standby for power-saving */
              /*============================================================*/

              while (1){
                      switch(ucMeasureStatus){        /* measuring status */
                              case    0:                                      /* R_REF1 */
<5>                                   DACSW1 = 1365;  /* D/A conversion value setting reigster 1 */
                                                      /* ANO1 = 1V (1/3 * 2^12) */
<6>                                   P7.0 = 1;
                                      PM7.0 = 0;      /* charge CAPACITOR */
                                      NOP();
                                      NOP();
                                      NOP();
                                      NOP();
                                      NOP();
                                      NOP();          /* wait charge complete */
                                      PM7.0 = 1;      /* end charge */

<7>                                   P7.1 = 0;
                                      PM7.1 = 0;      /* set R_REF low(start discharge) */
                                      TS0L.6 = 1;     /* start TIMER06 */
                                      TMIF06 = 0;     /* clear measure end */
                                      while(TMIF06 == 0);     /* wait measure end */
                                      ushR_REF1 = TDR06;      /* get measuring data */
                                      TT0L.6 = 1;             /* stop TAU0 CH6 */
                                      PM7.1 = 1;              /* set R_REF HiZ(stop discharge) */

                                      ucMeasureStatus = 1;    /* set R_REF2 measure */
                                      break;

                              case    1:              /* R_REF2 */
<8>                                   DACSW1 = 2730;  /* D/A conversion value setting reigster 1 */
                                                      /* ANO1 = 2V (2/3 * 2^12) */
<9>                                   P7.0 = 1;
                                      PM7.0 = 0;      /* charge CAPACITOR */
                                      NOP();
                                      NOP();
                                      NOP();
                                      NOP();
                                      NOP();
                                      NOP();          /* wait charge complete */
                                      PM7.0 = 1;      /* end charge */
```

```
                                P7.1 = 0;
                                PM7.1 = 0;      /* set R_REF low(start discharge) */
                                TS0L.6 = 1;     /* start TIMER06 */
<10>                            TMIF06 = 0;     /* clear measure end */
                                while(TMIF06 == 0);     /* wait measure end */
                                ushR_REF2 = TDR06;      /* get measuring data */
                                TT0L.6 = 1;             /* stop TAU0 CH6 */
                                PM7.1 = 1;              /* set R_REF HiZ(stop discharge) */

                                ucMeasureStatus = 2;    /* set R_TH1 measure */
                                break;

                        case    2:              /* R_TH1 */
<11>                            DACSW1 = 1365;  /* D/A conversion value setting reigster 1 */
                                                /* ANO1 = 1V (1/3 * 2^12) */
                                P7.0 = 1;
                                PM7.0 = 0;      /* charge CAPACITOR */
                                NOP();
<12>                            NOP();
                                NOP();
                                NOP();
                                NOP();
                                NOP();          /* wait charge complete */
                                PM7.0 = 1;      /* end charge */

                                P7.2 = 0;
                                PM7.2 = 0;      /* set R_TH low(start discharge) */
                                TS0L.6 = 1;     /* start TIMER06 */
<13>                            TMIF06 = 0;     /* clear measure end */
                                while(TMIF06 == 0);     /* wait measure end */
                                ushR_TH1 = TDR06;       /* get measuring data */
                                TT0L.6 = 1;             /* stop TAU0 CH6 */
                                PM7.2 = 1;              /* set R_TH HiZ(stop discharge) */

                                ucMeasureStatus = 3;    /* set R_TH2 measure */
                                break;

                        case    3:              /* R_TH2 */
<14>                            DACSW1 = 2730;  /* D/A conversion value setting reigster 1 */
                                                /* ANO1 = 2V (2/3 * 2^12) */
                                P7.0 = 1;
                                PM7.0 = 0;      /* charge CAPACITOR */
                                NOP();
<15>                            NOP();
                                NOP();
                                NOP();
                                NOP();
                                NOP();          /* wait charge complete */
                                PM7.0 = 1;      /* end charge */

                                P7.2 = 0;
                                PM7.2 = 0;      /* set R_TH low(start discharge) */
                                TS0L.6 = 1;     /* start TIMER06 */
                                TMIF06 = 0;     /* clear measure end */
<16>                            while(TMIF06 == 0);     /* wait measure end */
                                ushR_TH2 = TDR06;       /* get measuring data */
                                TT0L.6 = 1;             /* stop TAU0 CH6 */
                                PM7.2 = 1;              /* set R_TH HiZ(stop discharge) */

                                /* display R_TH,R_REF data to LCD */
<17>                            fn_Display((ushR_REF1 - ushR_REF2), (ushR_TH1 - ushR_TH2));

                                ucMeasureStatus = 0;    /* set R_REF1 measure */
                }
        }
}
```

This chapter describes the signal test results for the various circuits.

5.1 Non-Inverting Amplifier Circuit

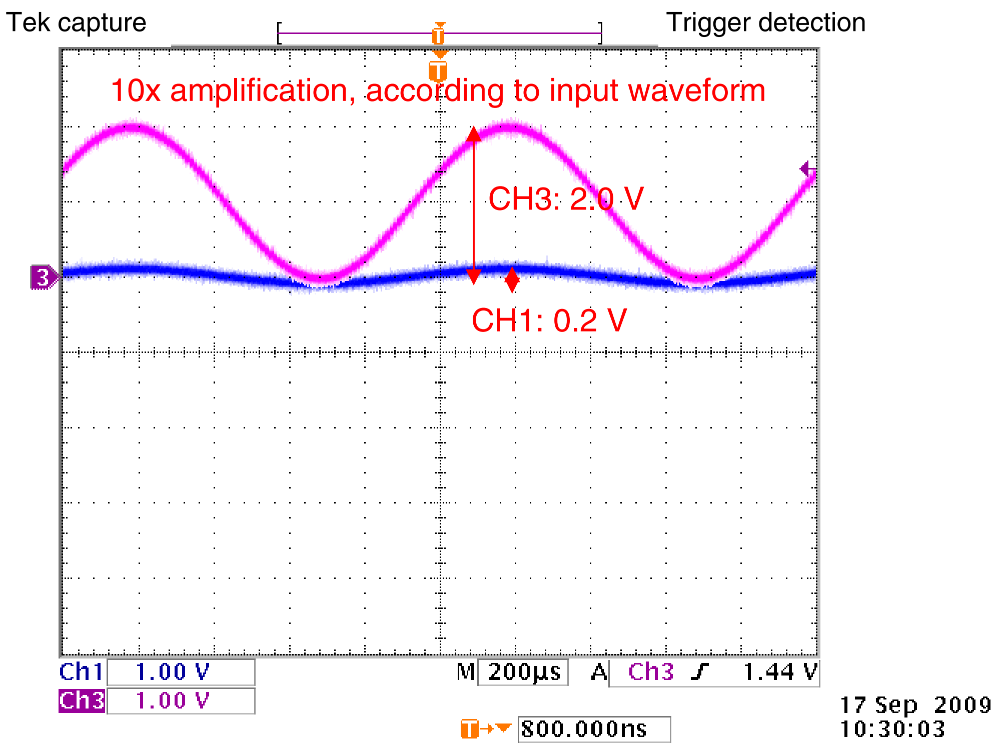


Figure 5-1. I/O Signal Waveforms of Non-Inverting Amplifier Circuit

5.2 Inverting Amplifier Circuit

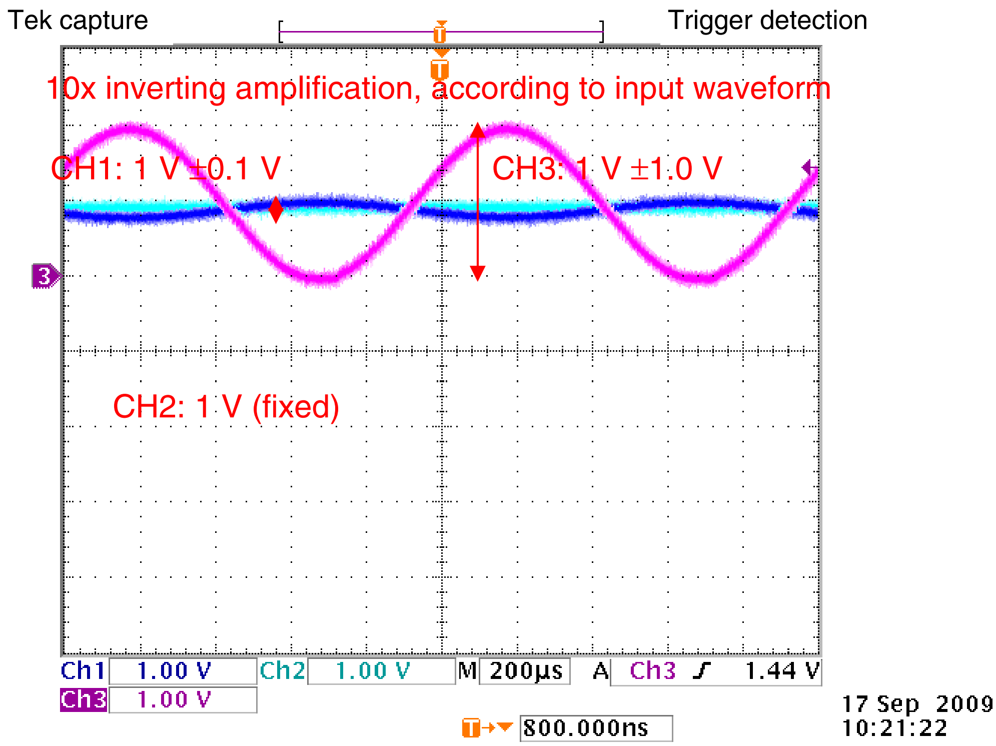


Figure 5-2. I/O Signal Waveforms of Inverting Amplifier Circuit

5.3 Offset Adjustment of Inverting Amplifier Circuit

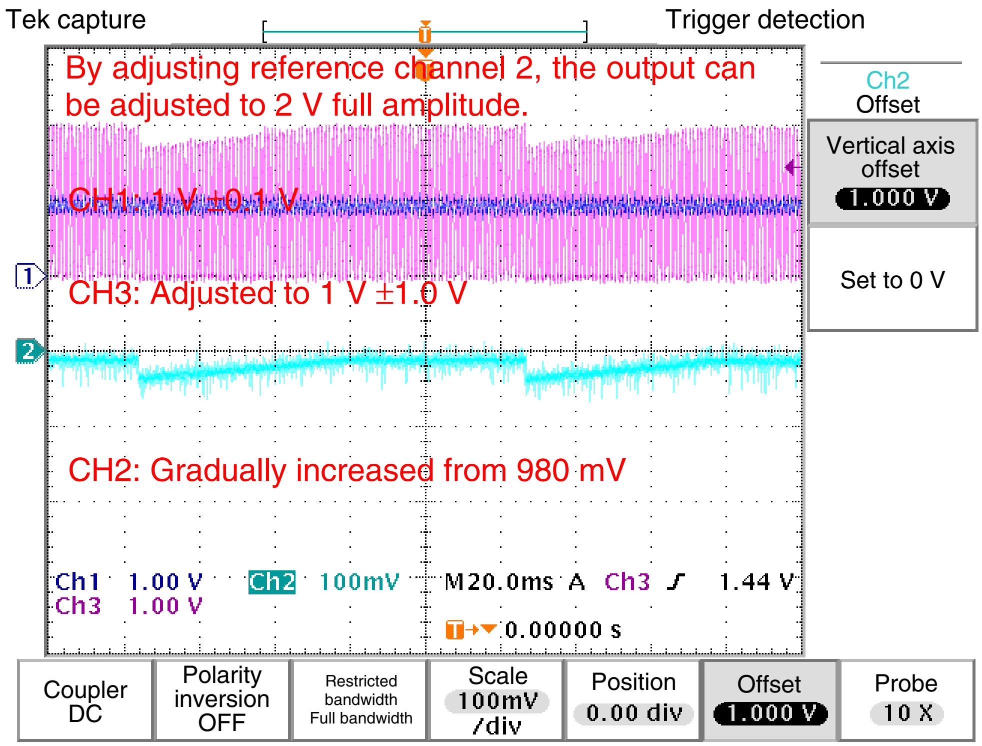


Figure 5-3. I/O Signal Waveforms of Offset Adjustment of Inverting Amplifier Circuit

5.4 Offset Adjustment of Inverting Amplifier Circuit + Low-Pass Filter

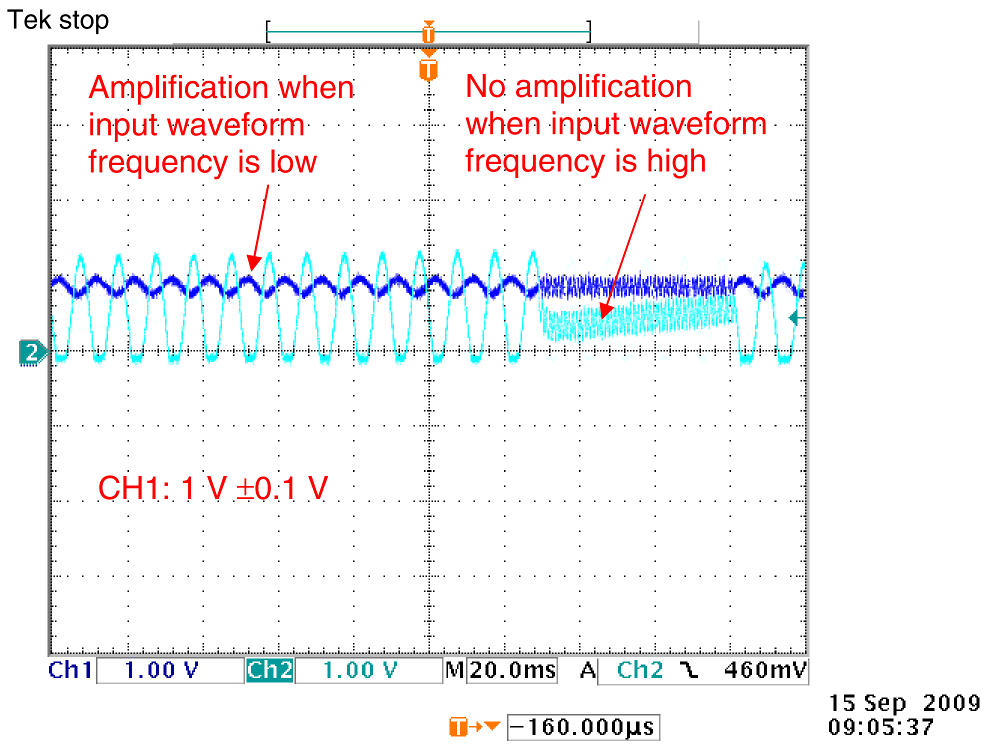
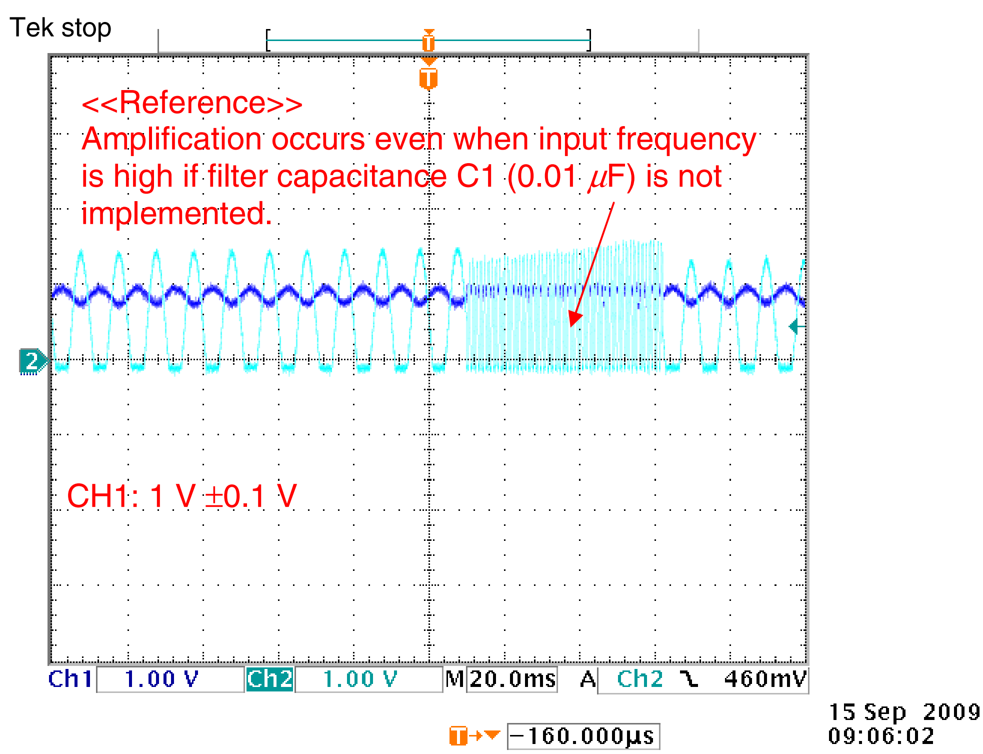


Figure 5-4-1. I/O Signal Waveforms of Offset Adjustment of Inverting Amplifier Circuit + Low-Pass Filter



<<Reference>>

Figure 5-4-2. I/O Signal Waveforms of Offset Adjustment of Inverting Amplifier Circuit + Low-Pass Filter (No Filter C)

5.5 Non-Inverting Amplifier Circuit (AC)

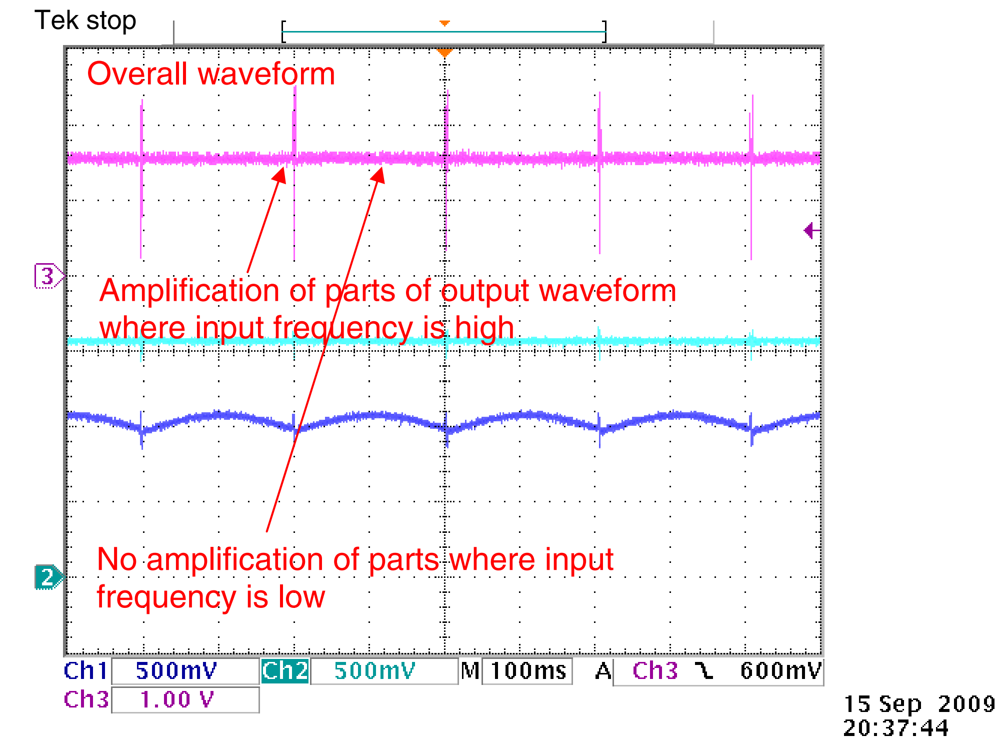


Figure 5-5-1. I/O Signal Waveforms of Non-Inverting Amplifier Circuit (AC)

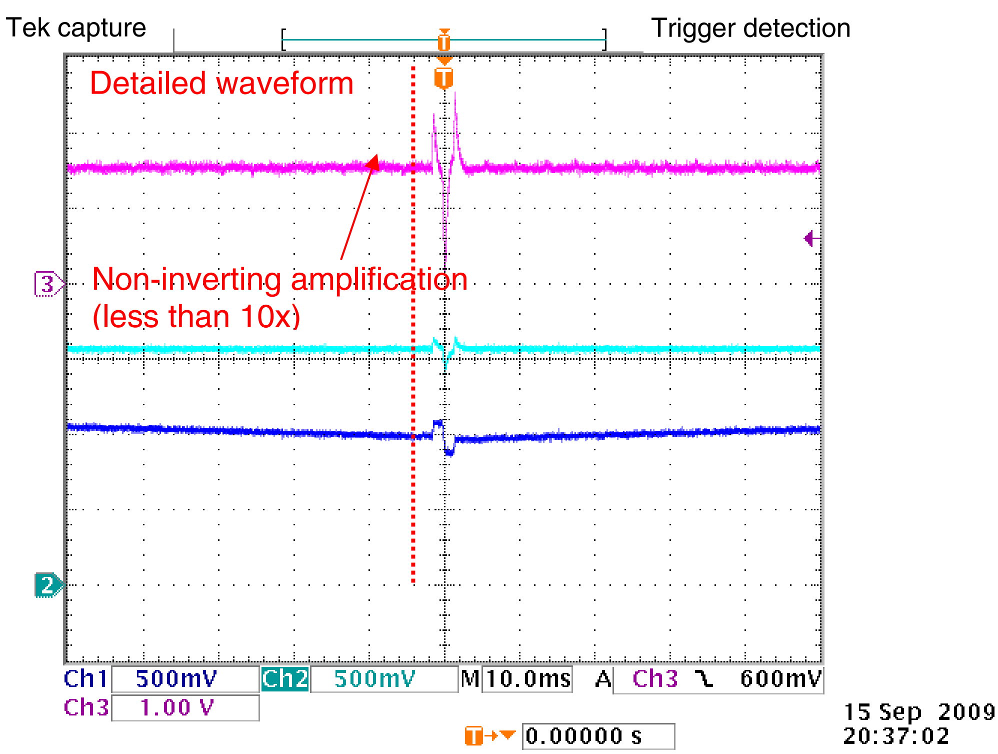


Figure 5-5-2. I/O Signal Waveforms of Non-Inverting Amplifier Circuit (AC)

5.6 Inverting Amplifier Circuit (AC)

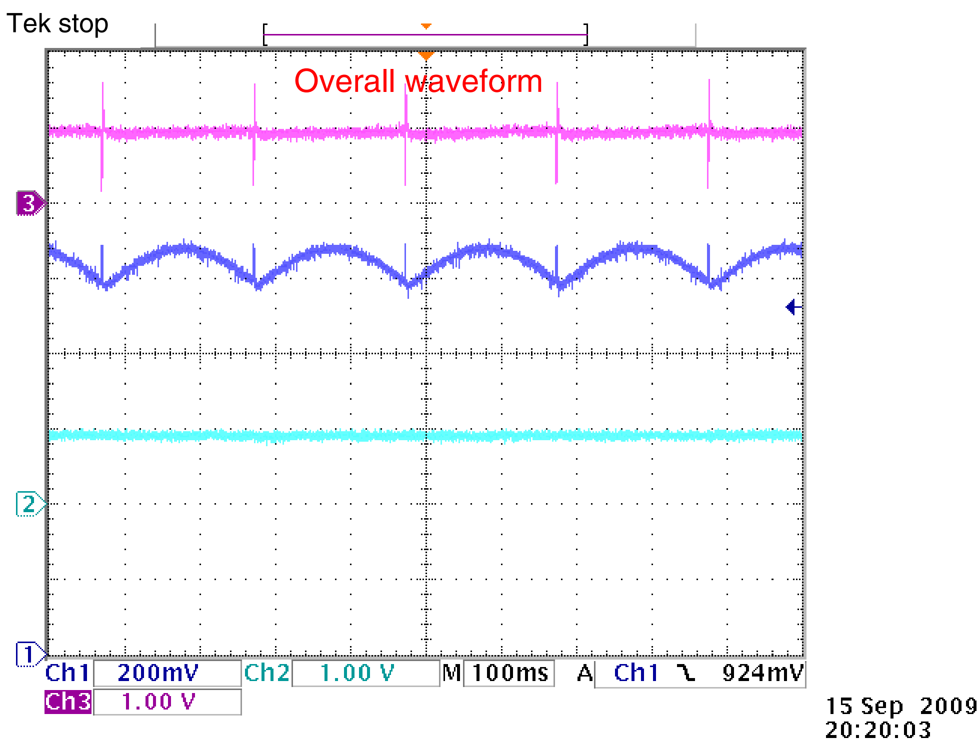


Figure 5-6-1. I/O Signal Waveforms of Inverting Amplifier Circuit (AC)

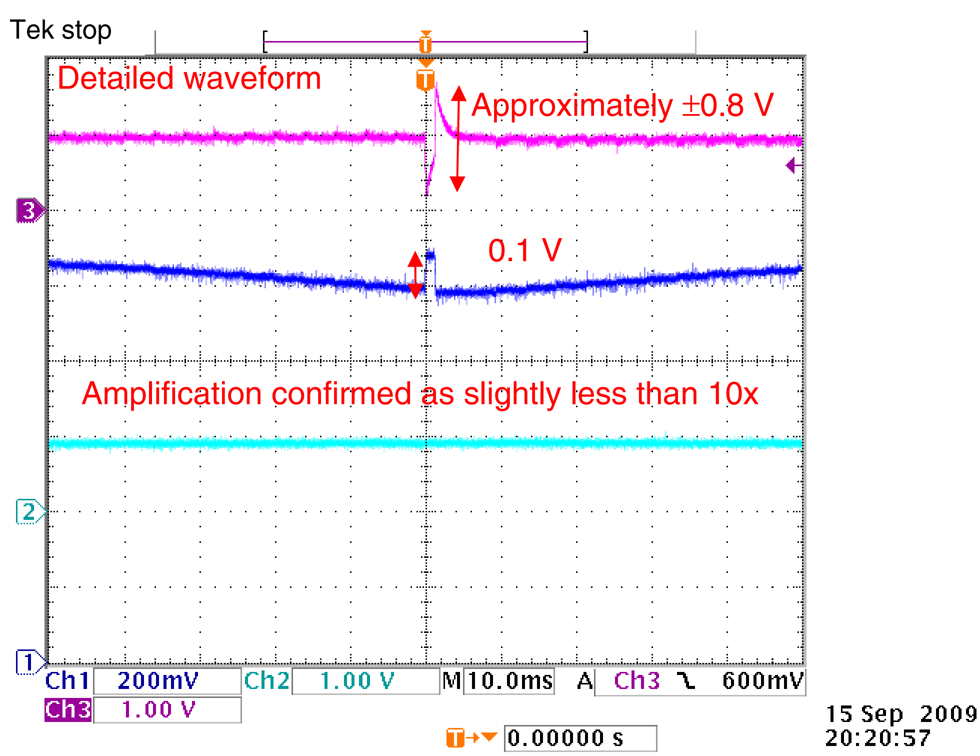


Figure 5-6-2. I/O Signal Waveforms of Inverting Amplifier Circuit (AC)

5.7 Instrumentation Amplifier

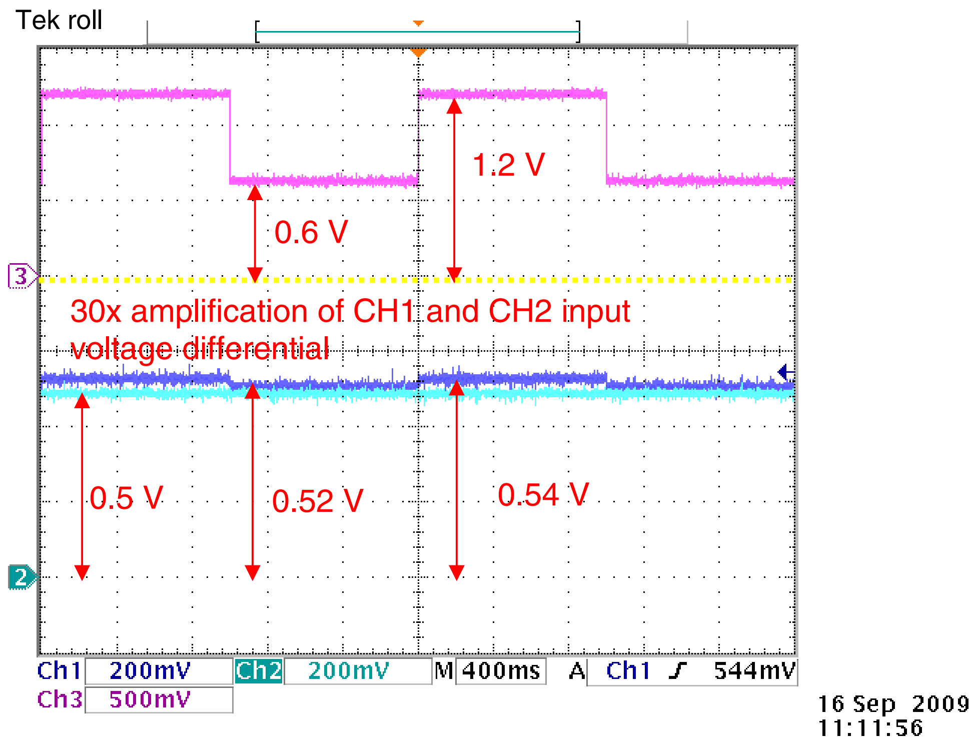


Figure 5-7. I/O Signal Waveforms of Instrumentation Amplifier

5.8 Comparator Circuit

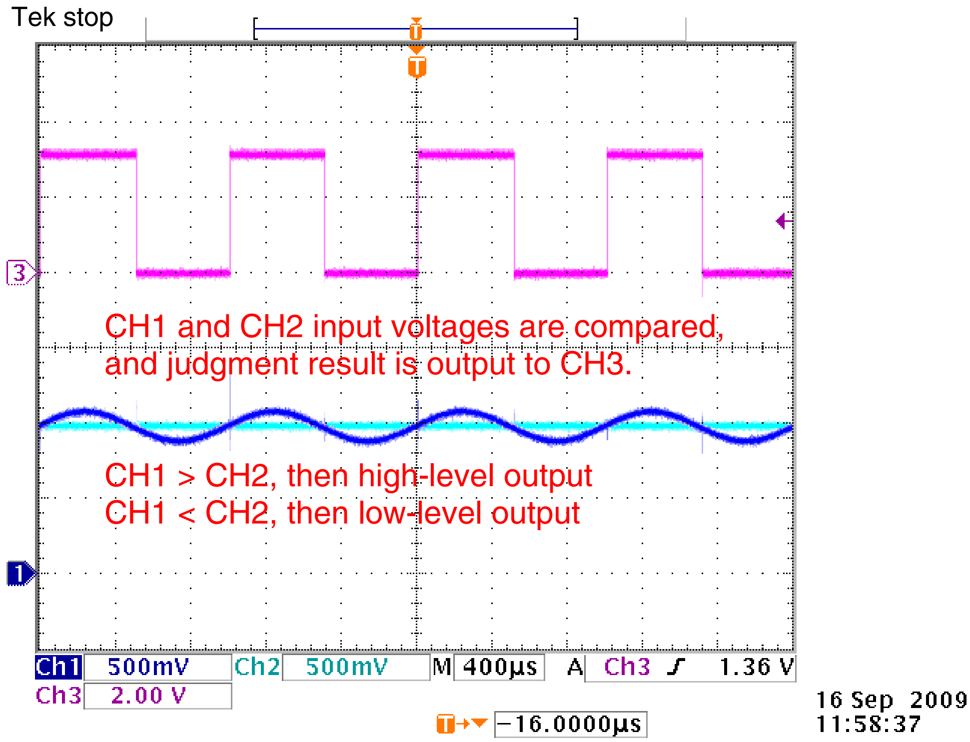


Figure 5-8. I/O Signal Waveforms of Comparator Circuit

5.9 CR Oscillator 1

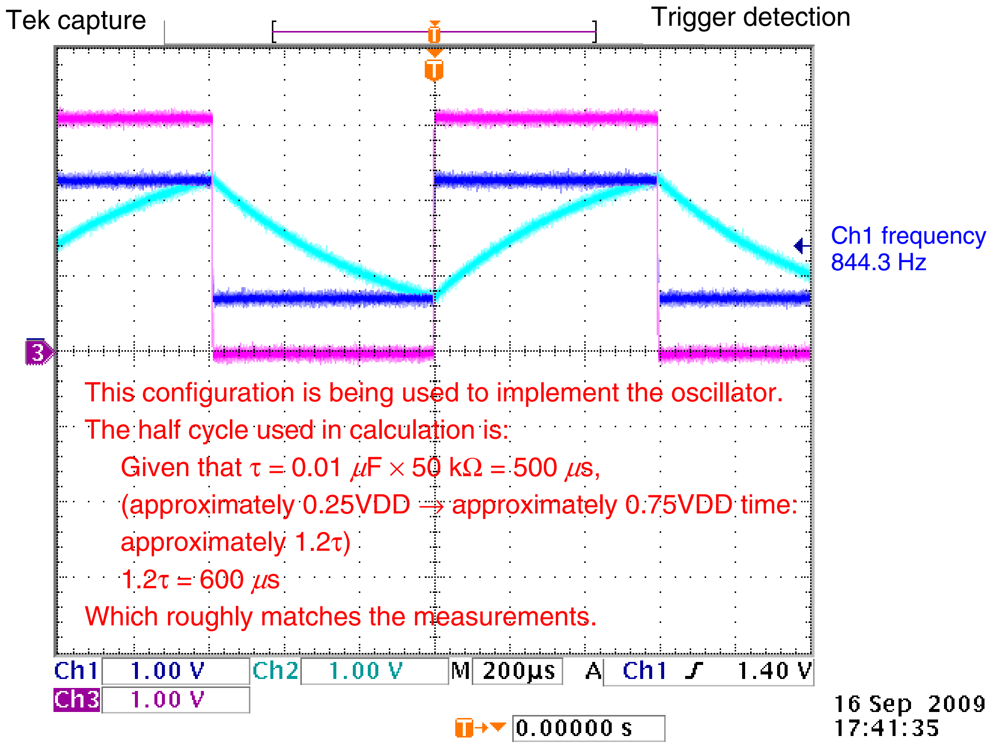


Figure 5-9. I/O Signal Waveforms of CR Oscillator 1

5.10 CR Oscillator 2

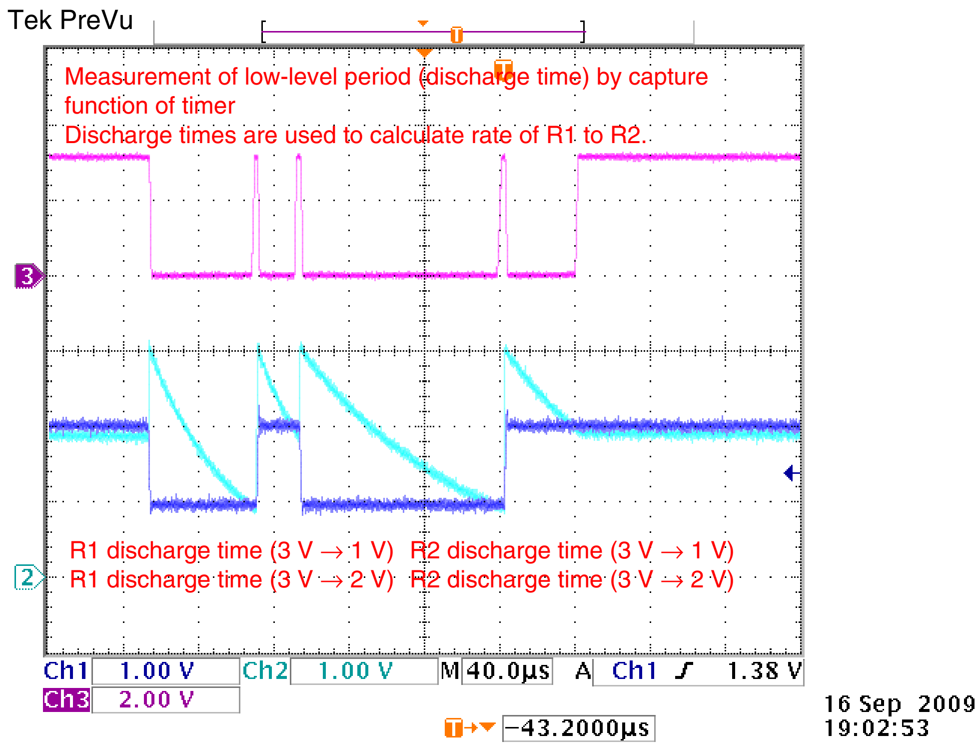


Figure 5-10. I/O Signal Waveforms of CR Oscillator 2

CHAPTER 6 RELATED DOCUMENTS

<table>
<tr><th colspan="2">Document Name</th><th>English</th></tr>
<tr><td colspan="2">78K0R/Lx3 User's Manual</td><td>PDF</td></tr>
<tr><td colspan="2">TK-78K0R/LH3+LCD User's Manual</td><td>PDF</td></tr>
<tr><td rowspan="2">CC78K0R C Compiler User's Manual</td><td>Language</td><td>PDF</td></tr>
<tr><td>Operation</td><td>PDF</td></tr>
<tr><td colspan="2">PM+ Project Manager User's Manual</td><td>PDF</td></tr>
</table>

Remark TK-78K0R/LH3+LCD is a product of TESSERA TECHNOLOGY INC.
Contact: TESSERA TECHNOLOGY INC. (http://www.tessera.co.jp)

APPENDIX A PROGRAM LIST

As a program list example, the source program is shown below.

(1) Non-inverting amplifier circuit

● main.c

```
/*
 * Copyright (C) NEC Electronics Corporation 2006
 * NEC ELECTRONICS CONFIDENTIAL AND PROPRIETARY
 * All rights reserved by NEC Electronics Corporation.
 * This program must be used solely for the purpose for which
 * it was furnished by NEC Electronics Corporation. No part of this
 * program may be reproduced or disclosed to others, in any
 * form, without the prior written permission of NEC Electronics
 * Corporation. Use of copyright notice dose not evidence
 * publication of the program.
 */

/*-----------------------------------------------------------------------*/
/* #pragma directive for CC78K0
                                        */
/*-----------------------------------------------------------------------*/
#pragma         SFR
#pragma         DI
#pragma         EI
#pragma         HALT
#pragma         NOP

#pragma interrupt INTDMA0 fn_intdma0 RB1

/*-----------------------------------------------------------------------*/
/* Include files
                                                  */
/*-----------------------------------------------------------------------*/
/* TAU:TDR0n value operation by CK00 (fCLK/2^3 = 5MHz) */
#define CCK00_500USEC    (2500 - 1)              /* 500us (0.2[us/clk] * 2500[count]) */
```

```
/*--------------------------------------------------------------------*/
/* Function prototyps
                                                    */
/*--------------------------------------------------------------------*/
                    void        fn_Wait500usBase(unsigned short);        /* Delays the program for (Time * 500us) */
                    void        fn_InitPort(void);                                    /* Setting of I/O ports
*/
                    void        fn_InitTau0(void);                                    /*  Setting  of  Timer
array unit 0 */
                    void        fn_InitLvi(void);                                     /*     Setting     of
Low-voltage detectior */
                    void        fn_InitVr(void);                                      /*  Setting  of  Voltage
reference */
                    void        fn_AdcDacInit(void);                         /* Setting of D/A converter */
                    void        fn_DmaInit(void);                                     /*  Setting  of  DMA
controler */


/*--------------------------------------------------------------------*/
/* Extern variables/constants
                    */
/*--------------------------------------------------------------------*/

/*--------------------------------------------------------------------*/
/* Local constants
                                                    */
/*--------------------------------------------------------------------*/
/*--------------------------------------------------------------------*/
/* Global variables
                                                    */
/*--------------------------------------------------------------------*/

/*--------------------------------------------------------------------*/
/* Local variables
                                                    */
/*--------------------------------------------------------------------*/

/*--------------------------------------------------------------------*/
```

```
/* Code
                                                        */
/*---------------------------------------------------------------------*/
/*==============================================================================

          D/A output data

==============================================================================*/
static unsigned short ushDacData[100];         /* D/A data for DMA */
const unsigned short aushDacData[100] = {      /* defined D/A data */
                                               /* n        sin(0.2pi*n/1024) */
/*                  204,                /* 0    0.100000 */
                    217,                /* 1    0.106279 */
                    230,                /* 2    0.112533 */
                    243,                /* 3    0.118738 */
                    255,                /* 4    0.124869 */
                    268,                /* 5    0.130902 */
                    280,                /* 6    0.136812 */
                    292,                /* 7    0.142578 */
                    303,                /* 8    0.148175 */
                    314,                /* 9    0.153583 */
                    325,                /* 10   0.158779 */
                    335,                /* 11   0.163742 */
                    345,                /* 12   0.168455 */
                    354,                /* 13   0.172897 */
                    362,                /* 14   0.177051 */
                    370,                /* 15   0.180902 */
                    377,                /* 16   0.184433 */
                    384,                /* 17   0.187631 */
                    390,                /* 18   0.190483 */
                    395,                /* 19   0.192978 */
                    399,                /* 20   0.195106 */
                    403,                /* 21   0.196858 */
                    406,                /* 22   0.198229 */
                    408,                /* 23   0.199211 */
                    409,                /* 24   0.199803 */
                    409,                /* 25   0.200000 */
                    409,                /* 26   0.199803 */
                    408,                /* 27   0.199211 */
```

```
406,        /* 28   0.198229 */
403,        /* 29   0.196858 */
399,        /* 30   0.195106 */
395,        /* 31   0.192978 */
390,        /* 32   0.190483 */
384,        /* 33   0.187631 */
377,        /* 34   0.184433 */
370,        /* 35   0.180902 */
362,        /* 36   0.177051 */
354,        /* 37   0.172897 */
345,        /* 38   0.168455 */
335,        /* 39   0.163742 */
325,        /* 40   0.158779 */
314,        /* 41   0.153583 */
303,        /* 42   0.148175 */
292,        /* 43   0.142578 */
280,        /* 44   0.136812 */
268,        /* 45   0.130902 */
255,        /* 46   0.124869 */
243,        /* 47   0.118738 */
230,        /* 48   0.112533 */
217,        /* 49   0.106279 */
204,        /* 50   0.100000 */
191,        /* 51   0.093721 */
179,        /* 52   0.087467 */
166,        /* 53   0.081262 */
153,        /* 54   0.075131 */
141,        /* 55   0.069098 */
129,        /* 56   0.063188 */
117,        /* 57   0.057422 */
106,        /* 58   0.051825 */
 95,        /* 59   0.046417 */
 84,        /* 60   0.041221 */
 74,        /* 61   0.036258 */
 64,        /* 62   0.031545 */
 55,        /* 63   0.027103 */
 47,        /* 64   0.022949 */
 39,        /* 65   0.019098 */
 31,        /* 66   0.015567 */
```

```
        25,             /* 67    0.012369 */
        19,             /* 68    0.009517 */
        14,             /* 69    0.007022 */
        10,             /* 70    0.004894 */
         6,             /* 71    0.003142 */
         3,             /* 72    0.001771 */
         1,             /* 73    0.000789 */
         0,             /* 74    0.000197 */
         0,             /* 75    0.000000 */
         0,             /* 76    0.000197 */
         1,             /* 77    0.000789 */
         3,             /* 78    0.001771 */
         6,             /* 79    0.003142 */
        10,             /* 80    0.004894 */
        14,             /* 81    0.007022 */
        19,             /* 82    0.009517 */
        25,             /* 83    0.012369 */
        31,             /* 84    0.015567 */
        39,             /* 85    0.019098 */
        47,             /* 86    0.022949 */
        55,             /* 87    0.027103 */
        64,             /* 88    0.031545 */
        74,             /* 89    0.036258 */
        84,             /* 90    0.041221 */
        95,             /* 91    0.046417 */
       106,             /* 92    0.051825 */
       117,             /* 93    0.057422 */
       129,             /* 94    0.063188 */
       141,             /* 95    0.069098 */
       153,             /* 96    0.075131 */
       166,             /* 97    0.081262 */
       179,             /* 98    0.087467 */
       191,             /* 99    0.093721 */
       204                       /* 100    0.100000 */

};

/*--------------------------------------------------------------------*/
/* Hardware initialization
```

```
                                  */
/*--------------------------------------------------------------------*/
void hdwinit(void)
{
        DI();                              /* disable all interrupts */

        /*-----------------------------------------*/
        /*          Initialization of port          */
        /*-----------------------------------------*/
        fn_InitPort();

        /*----------------------------------------*/
        /*         Initialization of clock         */
        /*----------------------------------------*/
        CMC =     0b01010011;                  /* Clock Operation Mode Control Register */
                        /*|||||||+--- : Control of high-speed system clock oscillation frequency */
                        /*||||||| : 0 : 2 MHz <= fMX <= 10 MHz */
                        /*||||||| : 1 : 10 MHz < fMX <= 20 MHz */
                        /*||||||| */
                        /*|||||++---- : XT1 oscillator oscillation mode selection */
                        /*||||| : 0 0 : Low-consumption oscillation */
                        /*||||| : 0 1 : Normal oscillation */
                        /*||||| : 1 x : Super-low-consumption oscillation */
                        /*|||||    x = don't care */
                        /*||||| */
                        /*||||+------ : Be sure to set 0 */
                        /*|||| */
                        /*|||+------- : [1] Subsystem clock pin operation mode */
                        /*|||              [2] XT1/P123 pin and XT2/P124 pin */
                        /*||| : 0 : [1]Input port mode */
                        /*|||          [2]Input port */
                        /*||| */
                        /*||| : 1 : [1]XT1 oscillation mode */
                        /*|||          [2]Crystal resonator connection */
                        /*||| */
                        /*||+-------- : Be sure to set 0 */
                        /*|| */
                        /*++--------- : [1]EXCLK OSCSEL High-speed system clock pin operation mode */
                        /*                 [2]X1/P121 pin */
```

```
                    /*                    [3]X2/EXCLK/P122 pin */
                    /* : 0 0 : [1]Input port mode */
                    /*             [2][3]Input port */
                    /* */
                    /* : 0 1 : [1]X1 oscillation mode */
                    /*             [2][3]Crystal/ceramic resonator connection */
                    /* */
                    /* : 1 0 : [1]Input port mode */
                    /*             [2][3]Input port */
                    /* */
                    /* : 1 1 : [1]External clock input mode */
                    /*             [2]Input port */
                    /*             [3]External clock input */

    MSTOP = 0;                          /* X1 oscillator operating */
    XTSTOP = 0;                         /* XT1 oscillator operating */

    OSMC =  0b00000001;                 /* Operation Speed Mode Control Register */
                    /*||||||++--- : fCLK frequency selection */
                    /*|||||| : 0 0 : Operates at a frequency of 10 MHz or less. */
                    /*|||||| : 0 1 : Operates at a frequency higher than 10 MHz. */
                    /*|||||| : 1 0 : Operates at a frequency of 1 MHz. */
                    /*|||||| */
                    /*|+++++----- : Be sure to set 00000 */
                    /*| */
                    /*+---------- : Setting in subsystem clock HALT mode */
                    /* : 0 : Enables subsystem clock supply to peripheral functions. */
                    /*         (See Table 21-1 Operating Statuses in HALT Mode (2/3) */
                    /*          for the peripheral functions whose operations are enabled.) */
                    /* : 1 : Stops subsystem clock supply to peripheral functions except real-time counter,
*/
                    /*          clock output/buzzer output, and LCD controller/driver. */

    while(OSTC.0 != 1){            /* wait X1 oscillation stabilization */
            NOP();
    }

    /*-- Caution --------------------------------------------*/
    /* To increase fCLK to 10 MHz or higher, set FSEL to '1',    */
```

```
/* then change fCLK after two or more clocks have elapsed. */
/*--------------------------------------------------------*/
NOP();
NOP();

CKC =     0b00010000;                 /* System Clock Control Register */
                    /*|+|+++++--- : Selection of CPU/peripheral hardware clock (fCLK) */
                    /*| | : 0 0 x 0 0 0 : fIH */
                    /*| | : 0 0 x 0 0 1 : fIH/2 (default) */
                    /*| | : 0 0 x 0 1 0 : fIH/2^2 */
                    /*| | : 0 0 x 0 1 1 : fIH/2^3 */
                    /*| | : 0 0 x 1 0 0 : fIH/2^4 */
                    /*| | : 0 0 x 1 0 1 : fIH/2^5 */
                    /*| | : 0 1 x 0 0 0 : fMX */
                    /*| | : 0 1 x 0 0 1 : fMX/2 */
                    /*| | : 0 1 x 0 1 0 : fMX/2^2 */
                    /*| | : 0 1 x 0 1 1 : fMX/2^3 */
                    /*| | : 0 1 x 1 0 0 : fMX/2^4 */
                    /*| | : 0 1 x 1 0 1 : fMX/2^5 */
                    /*| | : 1 x 0 x x x : fSUB */
                    /*| | : 1 x 1 x x x : fSUB/2 */
                    /*| | : Other than above : Setting prohibited */
                    /*| |    x = don't care */
                    /*| | */
                    /*| +-------- : Status of Main system clock (fMAIN) */
                    /*| : 0 : Internal high-speed oscillation clock (fIH) */
                    /*| : 1 : High-speed system clock (fMX) */
                    /*| */
                    /*+---------- : Status of CPU/peripheral hardware clock (fCLK) */
                    /* : 0 : Main system clock (fMAIN) */
                    /* : 1 : Subsystem clock (fSUB) */

/* Confirming the CPU clock status */
while((CLS != 0)||(MCS != 1)){
          NOP();
}
/* CPU is operating on a High-speed system clock */
HIOSTOP = 1;                          /* internal high-speed oscillation stopped */
```

```
OSTS =    0b00000111;                  /* Oscillation Stabilization Time Select Register */
                    /*||||++++--- : Oscillation stabilization time selection */
                    /*|||| : 0 0 0 : 2^8/fX */
                    /*|||| : 0 0 1 : 2^9/fX */
                    /*|||| : 0 1 0 : 2^10/fX */
                    /*|||| : 0 1 1 : 2^11/fX */
                    /*|||| : 1 0 0 : 2^13/fX */
                    /*|||| : 1 0 1 : 2^15/fX */
                    /*|||| : 1 1 0 : 2^17/fX */
                    /*|||| : 1 1 1 : 2^18/fX */
                    /*|||| */
                    /*++++------- : Be sure to set 000000 */

/*----------------------------------------*/
/*          Initialization of timer          */
/*----------------------------------------*/
fn_InitTau0();

/* software to wait for the operation stabilization time */
/* (over 200ms from when XT1 enable) */
fn_Wait500usBase(200000/500);

/*----------------------------------------*/
/* Initialization of low-voltage detector */
/*----------------------------------------*/
fn_InitLvi();

/*----------------------------------------*/
/*   Initialization of A/D,D/A,OPAmp                */
/*----------------------------------------*/
fn_AdcDacInit();

/*----------------------------------------*/
/*   Initialization of DMA                                    */
/*----------------------------------------*/
fn_DmaInit();

EI();                                  /* enable all interrupts */
```

```
}

/*-------------------------------------------------------------*/
/* Module:          fn_InitPort
                    */
/* Description:     Setting of I/O ports                                      */
/*       parameter: --
                                  */
/*       return     : --
                                  */
/*-------------------------------------------------------------*/
void     fn_InitPort(void)
{
/*---------------------------------------------------------*/
/*       Ports configuration for digital input and output                  */
/*---------------------------------------------------------*/
         ADPC =    0b00010000;          /* A/D Port Configuration Register */
                              /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                              /*||| :               +------------------------- ANI15/AVREFM/P157 */
                              /*||| :               | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                              /*||| :               | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
                              /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                              /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                              /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                              /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                              /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                              /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                              /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                              /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                              /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                              /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                              /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                              /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                              /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                              /*||| */
                              /*+++-------- : Be sure to set 000 */

/*---------------------------------------------------------*/
/*       Setting of Port 0
```

```
        */
/*---------------------------------------------------------*/
        P0 =      0b00000000;      /* Set P00-P02 Output latch to Low */
        PM0 =     0b11111000;      /* Set P00-P02 to output port */
                                              /* P00-P02:Unused */

/*---------------------------------------------------------*/
/*      Setting of Port 1
        */
/*---------------------------------------------------------*/
        P1 =      0b00000000;      /* Set P10-P17 Output latch to Low */
        PM1 =     0b00000000;      /* Set P10-P17 to output port */
                                              /* P10-P15:Unused */

/*---------------------------------------------------------*/
/*      Setting of Port 2
        */
/*---------------------------------------------------------*/
        P2 =      0b00000000;      /* Set P20-P27 Output latch to Low */
        PM2 =     0b11111111;      /* Set P20-P27 to input port */
                                              /* P20-P27:Unused */

/*---------------------------------------------------------*/
/*      Setting of Port 3
        */
/*---------------------------------------------------------*/
        P3 =      0b00001100;      /* Set P30-P31,P34 Output latch to Low */
                                              /* Set P33,P32 Output latch High */
        PM3 =     0b11100000;      /* Set P30-P34 to output port */
                                              /* P30-P34:Unused */

/*---------------------------------------------------------*/
/*      Setting of Port 4
        */
/*---------------------------------------------------------*/
        P4 =      0b00000000;      /* Set P40-P41 Output latch to Low */
        PM4 =     0b11111100;      /* Set P40-P41 to output port */
                                              /* P40-P41:Unused */
```

```
/*--------------------------------------------------------*/
/*      Setting of Port 5
        */
/*--------------------------------------------------------*/
        P5 =    0b00000000;     /* Set P50-P57 Output latch to Low */
        PM5 =   0b11110000;     /* Set P50-P57 to output port */
                                        /* P50-P57:Unused */

/*--------------------------------------------------------*/
/*      Setting of Port 6
        */
/*--------------------------------------------------------*/
        P6 =    0b00000000;     /* Set P60-P61 Output latch to Low */
        PM6 =   0b11111100;     /* Set P60-P61 to output port */
                                        /* P60-P61:Unused */

/*--------------------------------------------------------*/
/*      Setting of Port 7
        */
/*--------------------------------------------------------*/
        P7 =    0b00000000;     /* Set P70-P77 Output latch to Low */
        PM7 =   0b00000000;     /* Set P70-P77 to output port */
                                        /* P70-P77:Unused */

/*--------------------------------------------------------*/
/*      Setting of Port 8
        */
/*--------------------------------------------------------*/
        P8 =    0b00000000;     /* Set P80-P88 Output latch to Low */
        PM8 =   0b00000000;     /* Set P80-P88 to output port */
                                        /* P80-P88:Unused */

/*--------------------------------------------------------*/
/*      Setting of Port 9
        */
/*--------------------------------------------------------*/
        P9 =    0b00000000;     /* Set P90-P97 Output latch to Low */
        PM9 =   0b00000000;     /* Set P90-P97 to output port */
                                        /* P90-P97:Unused */
```

```
/*----------------------------------------------------------*/
/*      Setting of Port 10
        */
/*----------------------------------------------------------*/
        P10 =   0b00000000;         /* Set P100-P102 Output latch to Low */
        PM10 =  0b11111000;         /* Set P100-P102 to output port */
                                                /* P100-P102:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 11
        */
/*----------------------------------------------------------*/
        P11 =   0b00000000;         /* Set P110-P111 Output latch to Low */
        PM11 =  0b11111100;         /* Set P110-P111 to output port */
                                                /* P110-P111:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 12
        */
/*----------------------------------------------------------*/
        P12 =   0b00000000;         /* Set P120 Output latch to Low */
        PM12 =  0b11111110;         /* Set P120 to output port */
                                                /* P120-P124:Unused */
                                                /* *P121-P124:Input port */

/*----------------------------------------------------------*/
/*      Setting of Port 13
        */
/*----------------------------------------------------------*/
        P13 =   0b00000000;         /* Set P130 Output latch to Low */
                                                /* P130:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 14
        */
/*----------------------------------------------------------*/
        P14 =   0b00000000;         /* Set P140-P147 Output latch to Low */
        PM14 =  0b00000000;         /* Set P140-P147 to output port */
```

```
                                                        /* P140-P147:Unused */

/*--------------------------------------------------------*/
/*      Setting of Port 15
        */
/*--------------------------------------------------------*/
        P15 =   0b00000000;     /* Set P150-P152,P157 Output latch to Low */
        PM15 =  0b11111111;     /* Set P150-P152,P157 to input port */
                                                        /* P150-P152,P157:Unused */

}

/*------------------------------------------------------------*/
/* Module:          fn_InitTau0
                                */
/* Description:     Setting of Timer array unit 0                          */
/*      parameter: --
                                */
/*      return    : --
                                */
/*------------------------------------------------------------*/
void    fn_InitTau0(void)
{
        TAU0EN = 1;                     /* supplies input clock to timer array unit 0 */
        TPS0L = 0b00000010;     /* Timer Clock Select Register 0 */
                                /*||||++++--- : Selection of operation clock (CK00) */
                                /*++++------- : Selection of operation clock (CK01) */
                                /* : 0 0 0 0 : CK0m = fCLK */
                                /* : 0 0 0 1 : CK0m = fCLK/2 */
                                /* : 0 0 1 0 : CK0m = fCLK/2^2 */
                                /* : 0 0 1 1 : CK0m = fCLK/2^3 */
                                /* : 0 1 0 0 : CK0m = fCLK/2^4 */
                                /* : 0 1 0 1 : CK0m = fCLK/2^5 */
                                /* : 0 1 1 0 : CK0m = fCLK/2^6 */
                                /* : 0 1 1 1 : CK0m = fCLK/2^7 */
                                /* : 1 0 0 0 : CK0m = fCLK/2^8 */
                                /* : 1 0 0 1 : CK0m = fCLK/2^9 */
                                /* : 1 0 1 0 : CK0m = fCLK/2^10 */
                                /* : 1 0 1 1 : CK0m = fCLK/2^11 */
```

```
                    /* : 1 1 0 0 : CK0m = fCLK/2^12 */
                    /* : 1 1 0 1 : CK0m = fCLK/2^13 */
                    /* : 1 1 1 0 : CK0m = fCLK/2^14 */
                    /* : 1 1 1 1 : CK0m = fCLK/2^15 */
                    /*    m = 0, 1 */

/* CH1:for wait */
TMR01 = 0b0000000000000000;          /* Timer Mode Register 01 */
                    /*||||||||||||++++--- : [1]Operation mode of channel 1 */
                    /*||||||||||||          [2]Count operation of TCR */
                    /*||||||||||||          [3]Independent operation */
                    /*||||||||||||          [4]Setting of starting counting and interrupt */
                    /*|||||||||||| : 0 0 0 0 : [1]Interval timer mode */
                    /*||||||||||||             [2]Counting down */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                (timer output does not change, either). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 0 0 1 : [1]Interval timer mode */
                    /*||||||||||||             [2]Counting down */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is generated when counting is started */
                    /*||||||||||||                (timer output also changes). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 0 0 : [1]Capture mode */
                    /*||||||||||||             [2]Counting up */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                (timer output does not change, either). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 0 1 : [1]Capture mode */
                    /*||||||||||||             [2]Counting up */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is generated when counting is started */
                    /*||||||||||||                (timer output also changes). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 1 0 : [1]Event counter mode */
                    /*||||||||||||             [2]Counting down */
                    /*||||||||||||             [3]Possible */
```

```
/*||||||||||||                  [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                     (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 0 : [1]One-count mode */
/*||||||||||||                  [2]Counting down */
/*||||||||||||                  [3]Impossible */
/*||||||||||||                  [4]Start trigger is invalid during counting operation. */
/*||||||||||||                     At that time, interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 1 : [1]One-count mode */
/*||||||||||||                  [2]Counting down */
/*||||||||||||                  [3]Impossible */
/*||||||||||||                  [4]Start trigger is valid during counting operation. */
/*||||||||||||                     At that time, interrupt is also generated. */
/*|||||||||||| */
/*|||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*||||||||||||                  [2]Counting up */
/*||||||||||||                  [3]Possible */
/*||||||||||||                  [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                     (timer output does not change, either). */
/*||||||||||||                     Start trigger is invalid during counting operation. */
/*||||||||||||                     At that time interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : Other than above : Setting prohibited */
/*|||||||||||| */
/*||||||||||++------- : Be sure to set 00 */
/*|||||||||| */
/*||||||||++--------- : Selection of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 valid
edge */
/*||||||||                (the timer input used with channel 1 is selected by using TIS0
register). */
/*|||||||| : 0 0 : Falling edge */
/*|||||||| : 0 1 : Rising edge */
/*|||||||| : 1 0 : Both edges (when low-level width is measured) */
/*||||||||           Start trigger: Falling edge, Capture trigger: Rising edge */
/*|||||||| : 1 1 : Both edges (when high-level width is measured) */
/*||||||||           Start trigger: Rising edge, Capture trigger: Falling edge */
/*|||||||| */
/*|||||+++----------- : Setting of start trigger or capture trigger of channel 1 */
```

```
                              /*|||||  : 0 0 0 : Only software trigger start is valid (other trigger sources are unselected).
*/
                              /*|||||  : 0 0 1 : Valid edge of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 is used
as both the start trigger and capture trigger. */
                              /*|||||  : 0 1 0 : Both the edges of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 are
used as a start trigger and a capture trigger. */
                              /*|||||  : 1 0 0 : Interrupt signal of the master channel is used (when the channel is used
as a slave channel with the combination operation function). */
                              /*|||||  : Other than above : Setting prohibited */
                              /*|||||  */
                              /*||||+-------------- : Selection of slave/master of channel 1 */
                              /*||||  : 0 : Operates as slave channel with combination operation function. */
                              /*||||  : 1 : Operates as master channel with combination operation function. */
                              /*||||  */
                              /*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
                              /*|||  : 0 : Operation clock MCK specified by CKS01 bit */
                              /*|||  : 1 : Valid edge of input signal input from TI01 pin, fSUB/2, fSUB/4, or INTRTC1 */
                              /*|||        (the timer input used with channel 1 is selected by using TIS0 register). */
                              /*|||  */
                              /*|++---------------- : Be sure to set 00 */
                              /*|  */
                              /*+------------------ : Selection of operation clock (MCK) of channel 1 */
                              /*   : 0 : Operation clock CK00 set by TPS0 register */
                              /*   : 1 : Operation clock CK01 set by TPS0 register */
        TDR01 = CCK00_500USEC;          /* set interval time to 500us */
        TMMK01 = 1;                                       /* disable interrupt */


        /* CH4:for DMA */
        TMR04 = 0b1000000000000000;     /* Timer Mode Register 04 */
                              /*||||||||||||++++--- : [1]Operation mode of channel 4 */
                              /*||||||||||||          [2]Count operation of TCR */
                              /*||||||||||||          [3]Independent operation */
                              /*||||||||||||          [4]Setting of starting counting and interrupt */
                              /*|||||||||||| : 0 0 0 0 : [1]Interval timer mode */
                              /*||||||||||||              [2]Counting down */
                              /*||||||||||||              [3]Possible */
                              /*||||||||||||              [4]Timer interrupt is not generated when counting is started */
                              /*||||||||||||                 (timer output does not change, either). */
```

```
/*|||||||||||| */
/*|||||||||||| : 0 0 0 1 : [1]Interval timer mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is generated when counting is started */
/*||||||||||||                (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 0 0 : [1]Capture mode */
/*||||||||||||             [2]Counting up */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 0 1 0 1 : [1]Capture mode */
/*||||||||||||             [2]Counting up */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is generated when counting is started */
/*||||||||||||                (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 1 0 : [1]Event counter mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 0 : [1]One-count mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Impossible */
/*||||||||||||             [4]Start trigger is invalid during counting operation. */
/*||||||||||||                At that time, interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 1 : [1]One-count mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Impossible */
/*||||||||||||             [4]Start trigger is valid during counting operation. */
/*||||||||||||                At that time, interrupt is also generated. */
/*|||||||||||| */
/*|||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*||||||||||||             [2]Counting up */
```

```
/*||||||||||||                 [3]Possible */
/*||||||||||||                 [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                    (timer output does not change, either). */
/*||||||||||||                    Start trigger is invalid during counting operation. */
/*||||||||||||                    At that time interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : Other than above : Setting prohibited */
/*|||||||||||| */
/*||||||||||++------- : Be sure to set 00 */
/*|||||||||| */
/*||||||||++--------- : Selection of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 valid
edge */
/*||||||||                 (the timer input used with channel 4 is selected by using TIS0
register). */
/*|||||||| : 0 0 : Falling edge */
/*|||||||| : 0 1 : Rising edge */
/*|||||||| : 1 0 : Both edges (when low-level width is measured) */
/*||||||||             Start trigger: Falling edge, Capture trigger: Rising edge */
/*|||||||| : 1 1 : Both edges (when high-level width is measured) */
/*||||||||             Start trigger: Rising edge, Capture trigger: Falling edge */
/*|||||||| */
/*|||||+++----------- : Setting of start trigger or capture trigger of channel 4 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are unselected).
*/
/*||||| : 0 0 1 : Valid edge of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 is used
as both the start trigger and capture trigger. */
/*||||| : 0 1 0 : Both the edges of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 are
used as a start trigger and a capture trigger. */
/*||||| : 1 0 0 : Interrupt signal of the master channel is used (when the channel is used
as a slave channel with the combination operation function). */
/*||||| : Other than above : Setting prohibited */
/*||||| */
/*||||+-------------- : Selection of slave/master of channel 4 */
/*|||| : 0 : Operates as slave channel with combination operation function. */
/*|||| : 1 : Operates as master channel with combination operation function. */
/*|||| */
/*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
/*||| : 0 : Operation clock MCK specified by CKS04 bit */
/*||| : 1 : Valid edge of input signal input from TI04 pin, fSUB/2, fSUB/4, or INTRTC1 */
```

```
                        /*|||           (the timer input used with channel 4 is selected by using TIS0 register). */
                        /*||| */
                        /*|++---------------- : Be sure to set 00 */
                        /*| */
                        /*+------------------ : Selection of operation clock (MCK) of channel 4 */
                        /* : 0 : Operation clock CK00 set by TPS0 register */
                        /* : 1 : Operation clock CK01 set by TPS0 register */
    TDR04 = 200;        /* set interval time to 10us(=0.05us * 200) */
    TMMK04 = 1;                 /* disable interrupt */



}

/*--------------------------------------------------------------*/
/* Module:        fn_InitLvi
                  */
/* Description:   Setting of Low-voltage detectior                              */
/*    parameter: --
                        */
/*    return    : --
                        */
/*--------------------------------------------------------------*/
void    fn_InitLvi(void)
{
    unsigned short      loop;       /* waiting counter */

    LVIMK = 1;                          /* disable LVI interrupt */

    LVIS =    0b00001001;               /* Low-Voltage Detection Level Select Register */
                        /*||||++++--- : Detection level */
                        /*|||| : 0 0 0 0 : VLVI0 (4.22V) */
                        /*|||| : 0 0 0 1 : VLVI1 (4.07V) */
                        /*|||| : 0 0 1 0 : VLVI2 (3.92V) */
                        /*|||| : 0 0 1 1 : VLVI3 (3.76V) */
                        /*|||| : 0 1 0 0 : VLVI4 (3.61V) */
                        /*|||| : 0 1 0 1 : VLVI5 (3.45V) */
                        /*|||| : 0 1 1 0 : VLVI6 (3.30V) */
                        /*|||| : 0 1 1 1 : VLVI7 (3.15V) */
                        /*|||| : 1 0 0 0 : VLVI8 (2.99V) */
```

```
                    /*|||| : 1 0 0 1 : VLVI9 (2.84V) */
                    /*|||| : 1 0 1 0 : VLVI10 (2.68V) */
                    /*|||| : 1 0 1 1 : VLVI11 (2.53V) */
                    /*|||| : 1 1 0 0 : VLVI12 (2.38V) */
                    /*|||| : 1 1 0 1 : VLVI13 (2.22V) */
                    /*|||| : 1 1 1 0 : VLVI14 (2.07V) */
                    /*|||| : 1 1 1 1 : VLVI15 (1.91V) */
                    /*|||| */
                    /*++++------- : Be sure to set 0000 */

    LVIM =    0b10000000;              /* Low-Voltage Detection Register */
                    /*|||||||+--- : LVIF Low-voltage detection flag */
                    /*||||||| : 0 : * LVISEL = 0: VDD >= VLVI, or when LVI operation is disabled */
                    /*|||||||        * LVISEL = 1: EXLVI >= VEXLVI, or when LVI operation is disabled */
                    /*||||||| : 1 : * LVISEL = 0: VDD < VLVI */
                    /*|||||||        * LVISEL = 1: EXLVI < VEXLVI */
                    /*||||||| */
                    /*||||||+---- : Low-voltage detection operation mode (interrupt/reset) selection(LVIMD) */
                    /*|||||| : 0 : * LVISEL = 0: Generates an internal interrupt signal */
                    /*||||||                     when VDD drops lower than VLVI (VDD < VLVI) */
                    /*||||||                     or when VDD becomes VLVI or higher (VDD >= VLVI).
*/
                    /*||||||        * LVISEL = 1: Generates an interrupt signal */
                    /*||||||                     when EXLVI drops lower than VEXLVI (EXLVI <
VEXLVI) */
                    /*||||||                     or when EXLVI becomes VEXLVI or higher (EXLVI >=
VEXLVI). */
                    /*|||||| : 1 : * LVISEL = 0: Generates an internal reset signal when VDD < VLVI */
                    /*||||||                     and releases the reset signal when VDD >= VLVI. */
                    /*||||||        * LVISEL = 1: Generates an internal reset signal when EXLVI < VEXLVI
*/
                    /*||||||                     and releases the reset signal when EXLVI >= VEXLVI.
*/
                    /*|||||| */
                    /*|||||+----- : Voltage detection selection(LVISEL) */
                    /*||||| : 0 : Detects level of supply voltage (VDD) */
                    /*||||| : 1 : Detects level of input voltage from external input pin (EXLVI) */
                    /*||||| */
                    /*|++++------ : Be sure to set 0000 */
```

```
                    /*| */
                    /*+---------- : Enables low-voltage detection operation */
                    /* : 0 : Disables operation */
                    /* : 1 : Enables operation */

        /* software to wait for the operation stabilization time (210us) */
        for(loop = 500; loop > 0; loop--){
                NOP();
        }
        /* wait until VLVI≤VDD */
        while( LVIF ){
                NOP();
        }

        LVIIF = 0;                  /* clear LVI interrupt request flag */
}



/*-------------------------------------------------------------*/
/* Module:         fn_DmaInit
                    */
/* Description:    Setting of DMA contoroler                                  */
/*      parameter: --
                    */
/*      return    : --
                    */
/*-------------------------------------------------------------*/
void    fn_DmaInit(void)
{

        DEN0 = 1;                                   /* Enables operation of DMA channel 0 */

        DSA0 = (unsigned char)&DACSW0;                          /* DMA SFR Address Register 0
*/
        DRA0 = (unsigned short)&ushDacData[0];      /* DMA RAM Address Register 0 */
        DBC0 = 100;                                         /* DMA Byte Count Register 0 */
        DMC0 = 0b01100100;                          /* DMA Mode Control Register 0 */
                /*|||||||| */
```

```
                    /*||||++++------- IFC3-IFC0 */
                    /*||||            [Selection of DMA start source] */
                    /*||||              0000:Disables DMA transfer by interrupt */
                    /*||||                   (Only software trigger is enabled.) */
                    /*||||              0010:INTTM00 */
                    /*||||              0011:INTTM01 */
                    /*||||              0100:INTTM04 */
                    /*||||              0101:INTTM05 */
                    /*||||              0110:INTST0/INTCSI00 */
                    /*||||              0111:INTSR0/INTCSI01 */
                    /*||||              1000:INTST1/INTCSI10/INTIIC10 */
                    /*||||              1001:INTSR1 */
                    /*||||              1010:INTST3 */
                    /*||||              1011:INTSR3 */
                    /*||||              1100:INTAD */
                    /*|||+----------- DWAIT0 */
                    /*|||             [Pending of DMA transfer] */
                    /*|||               0:Executes DMA transfer upon DMA start request (not held pending). */
                    /*|||               1:Holds DMA start request pending if any. */
                    /*||+------------ DS0 */
                    /*||              [Specification of transfer data size for DMA transfer] */
                    /*||                0:8bits */
                    /*||                1:16bits */
                    /*|+------------- DRS0 */
                    /*|               [Selection of DMA transfer direction] */
                    /*|                 0:SFR to internal RAM */
                    /*|                 1:Internal RAM to SFR */
                    /*+------------- STG0 */
                    /*                [DMA transfer start software trigger] */
                    /*                  0:No trigger operation */
                    /*                  1:DMA transfer is started when DMA operation is enabled (DEN0 = 1). */

        DST0 = 1;               /* DMA transfer is started when DMA operation is enabled (DEN0 = 1). */

}


/*------------------------------------------------------------*/
/* Module:      fn_Wait500usBase
```

```
        */
/* Description:    Delays the program for (Time * 500us)                  */
/*      parameter: wait time(Time)
        */
/*      return     : --
                          */
/*-------------------------------------------------------------*/
void    fn_Wait500usBase(unsigned short Time)
{
        TS0L.1 = 1;                        /* start TAU0 CH1 */
        TMIF01 = 0;

        for(; Time > 0; Time--){           /* wait for (parameter * 500)us */
                while(!TMIF01){
                        NOP();
                }
                TMIF01 = 0;
        }

        TT0L.1 = 1;                        /* stop TAU0 CH1 */
}


/*-------------------------------------------------------------*/
/* Module:         fn_AdcDacInit
                */
/* Description:    Initialization of A/D D/A OPAmp module                 */
/*      parameter: --
                          */
/*      return     : --
                          */
/*-------------------------------------------------------------*/
void    fn_AdcDacInit(void)
{
        ADCEN = 1;                         /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

        /*------------------------------------------*/
        /*   Initialization of Voltage reference              */
```

```
/*-------------------------------------------*/
ADVRC = 0b00001000;        /* Analog reference voltage control register */
                    /*||||+|++--- : [1]Positive reference voltage supplies selection of A/D and D/A
converters */
                    /*|||| |        [2]Operationcontrol of voltage reference */
                    /*|||| |        [3]Output voltage selection of voltage reference */
                    /*|||| |        [4]Operation control of input gate voltage boost circuit for A/D converter
*/
                    /*|||| |        [5]Relationship with the conversion mode used */
                    /*|||| | */
                    /*|||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
                    /*|||| |            [2]Stops operation (Hi-Z) */
                    /*|||| |            [3]2.5 V */
                    /*|||| |            [4]Stops operation */
                    /*|||| |            [5]Can be set in conversion mode 1 */
                    /*|||| | */
                    /*|||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
                    /*|||| |            [2]Stops operation (Hi-Z) */
                    /*|||| |            [3]2.0 V */
                    /*|||| |            [4]Enables operation */
                    /*|||| |            [5]Can be set in conversion mode 2 or 3 */
                    /*|||| | */
                    /*|||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
                    /*|||| |            [2]Stops operation (pull-down output) */
                    /*|||| |            [3]2.5 V */
                    /*|||| |            [4]Stops operation */
                    /*|||| |            [5] - */
                    /*|||| | */
                    /*|||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
                    /*|||| |            [2]Enables operation */
                    /*|||| |            [3]2.5 V */
                    /*|||| |            [4]Enables operation */
                    /*|||| |            [5]Can be set in conversion mode 2 or 3 */
                    /*|||| | */
                    /*|||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
                    /*|||| |            [2]Stops operation (pull-down output) */
                    /*|||| |            [3]2.0 V */
                    /*|||| |            [4]Enables operation */
                    /*|||| |            [5] - */
```

```
                    /*|||| | */
                    /*|||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
                    /*|||| |           [2]Enables operation */
                    /*|||| |           [3]2.0 V */
                    /*|||| |           [4]Enables operation */
                    /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                    /*|||| | */
                    /*|||| | : Other than the above : Setting prohibited */
                    /*|||| | */
                    /*|+++-+----- : Be sure to set 0000 */
                    /*| */
                    /*+---------- : Reference voltage supply (negative side) of A/D converter selection */
                    /* : 0 : AVSS */
                    /* : 1 : AVREFM (external voltage reference input) */

ADVRC.0 = 1;            /* Enables operation */
ADVRC.1 = 1;            /* Output 2.0 V */

ADPC =  0b00000000;     /* A/D Port Configuration Register */
                    /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                    /*||| :           +------------------------ ANI15/AVREFM/P157 */
                    /*||| :           | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                    /*||| :           | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
                    /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                    /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                    /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                    /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                    /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                    /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                    /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                    /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                    /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                    /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                    /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                    /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                    /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                    /*||| */
                    /*+++-------- : Be sure to set 000 */
```

```
        /* wait for settling time to 17msec */
        fn_Wait500usBase(17000/500);

        /*-------------------------------------------*/
        /*    Initialization of operational amplifier    */
        /*-------------------------------------------*/
        PM2.1 = 1;                    /* input port mode (AMP0O) */
        PM2.0 = 1;                    /* input port mode (AMP0-) */
        PM2.2 = 1;                    /* input port mode (AMP0+) */

        PM2.4 = 1;                    /* input port mode (AMP1O) */
        PM2.3 = 1;                    /* input port mode (AMP1-) */
        PM2.5 = 1;                    /* input port mode (AMP1+) */

        PM2.7 = 1;                    /* input port mode (AMP2O) */
        PM2.6 = 1;                    /* input port mode (AMP2-) */
        PM15.0 = 1;                   /* input port mode (AMP2+) */

        PM11.0 = 1;                   /* input port mode (ANO0) */

        ADCEN = 1;                    /* supplies input clock to OPAmp A/D converter */
        OAC.0 = 1;                    /* operational amplifier (AMP0) enable */
        OAC.1 = 0;                    /* operational amplifier (AMP1) disable */
        OAC.2 = 0;                    /* operational amplifier (AMP2) disable */

        /*-------------------------------------------*/
        /*          Initialization of D/A converter          */
        /*-------------------------------------------*/
        DACEN = 1;                            /* supplies input clock to D/A converter */
        DAM.0 = 1;                            /* D/A converter operation is Real-time output mode */
        DAM.2 = 1;                            /* D/A conversion resolution is 12-bit */
        DAM.6 = 1;                            /* D/A converter positive reference voltage is
VREFOUT */
        DACE0 = 0;                            /* D/A converter CH0 disable */
        DACSW0 = aushDacData[99];     /* D/A conversion value setting reigster 0 */

    }
    /*--------------------------------------------------------------*/
    /* Module:          main
```

```
                              */
/* Description:   Main process
        */
/*      parameter: --
                              */
/*      return    : --
                              */
/*-----------------------------------------------------------*/
void main(void)
{
        unsigned short      count;            /* counter for initializing D/A convert data */
        DACE0 = 1;                                            /* D/A converter CH0 enable */

        /*---------------------------------*/
        /*      Initialization of module      */
        /*---------------------------------*/
/*      fn_SensorInit();              /* Initialization of Sensor module */

        /*---------------------------------*/
        /*     Initialization of variable     */
        /*---------------------------------*/
/*      uc100msec = C100MSEC;                 /* initialize 100msec counter */

        /*=================================*/
        /*---------------------------------*/
        /*              Main Loop             */
        /*---------------------------------*/
        /*=================================*/


        /*   initialize D/Aconverter data */
        for(count = 0; count<100; count++){
                ushDacData[count] = aushDacData[count];
        }

        TS0L.4 = 1;                           /* start TAU0 CH4 */
        TMIF04 = 0;                           /* reset INTTM04 interrupt request */

        DMAMK0 = 0;                           /* enable DMA CH0 interrupt */
```

```
        EI();                                   /* enable all interrupts */

        /*============================================================*/
        /* if system have nothing to do, go to standby for power-saving */
        /*============================================================*/


        while (1){
                HALT();
                NOP();
        }
}


/*-------------------------------------------------------------*/
/* Module:         fn_intdma0
                                */
/* Description:    DMA ch0 interruput process                                    */
/*      parameter: --
                                */
/*      return     : --
                                */
/*-------------------------------------------------------------*/
__interrupt void fn_intdma0(void)
{
        DRA0 = (unsigned short)&ushDacData[0];          /* DMA RAM Address Register 0 */
        DBC0 = 100;                                     /* DMA Byte Count Register 0 */

        DST0 = 1;                               /* DMA transfer is started when DMA operation is enabled
(DEN0 = 1). */
}
```

(2) Inverting amplifier circuit

● main.c

```
/*
 * Copyright (C) NEC Electronics Corporation 2006
 * NEC ELECTRONICS CONFIDENTIAL AND PROPRIETARY
 * All rights reserved by NEC Electronics Corporation.
 * This program must be used solely for the purpose for which
 * it was furnished by NEC Electronics Corporation. No part of this
 * program may be reproduced or disclosed to others, in any
 * form, without the prior written permission of NEC Electronics
 * Corporation. Use of copyright notice dose not evidence
 * publication of the program.
 */

/*------------------------------------------------------------------------*/
/* #pragma directive for CC78K0
                                        */
/*------------------------------------------------------------------------*/
#pragma         SFR
#pragma         DI
#pragma         EI
#pragma         HALT
#pragma         NOP

#pragma interrupt INTDMA0 fn_intdma0 RB1

/*------------------------------------------------------------------------*/
/* Include files
                                                      */
/*------------------------------------------------------------------------*/
/* TAU:TDR0n value operation by CK00 (fCLK/2^3 = 5MHz) */
#define CCK00_500USEC    (2500 - 1)          /* 500us (0.2[us/clk] * 2500[count]) */

/* TAU:TDR0n value operation by CK01 (fCLK = 20MHz) */
#define CCK01_PLAY                (2500 - 1) /* 0.125ms = about 8kHz for play */
                                                                                /*
(0.05[us/clk] * 2500[count]) */
```

```
/*-----------------------------------------------------------------------*/
/* Function prototyps
                                                    */
/*-----------------------------------------------------------------------*/
                void     fn_Wait500usBase(unsigned short);     /* Delays the program for (Time * 500us) */
                void     fn_InitPort(void);                                  /* Setting of I/O ports
*/
                void     fn_InitTau0(void);                                  /* Setting of Timer
array unit 0 */
                void     fn_InitLvi(void);                                   /*    Setting    of
Low-voltage detectior */
                void     fn_InitVr(void);                                    /* Setting of Voltage
reference */
                void     fn_AdcDacInit(void);                        /* Setting of D/A converter */
                void     fn_DmaInit(void);                                   /* Setting of DMA
controler */


/*-----------------------------------------------------------------------*/
/* Extern variables/constants
                */
/*-----------------------------------------------------------------------*/

/*-----------------------------------------------------------------------*/
/* Local constants
                                                    */
/*-----------------------------------------------------------------------*/
/*-----------------------------------------------------------------------*/
/* Global variables
                                                    */
/*-----------------------------------------------------------------------*/

/*-----------------------------------------------------------------------*/
/* Local variables
                                                    */
/*-----------------------------------------------------------------------*/

/*-----------------------------------------------------------------------*/
/* Code
```

```
                                                              */
/*----------------------------------------------------------------------*/
/*==================================================================================

        D/A output data

==================================================================================*/
static unsigned short ushDacData[100];          /* D/A data for DMA */
const unsigned short aushDacData[100] = {       /* defined D/A data */
                                                /* n        sin(0.2pi*n/100) */
/*              2048,              /* 0         0.100000 */
                2061,              /* 1         0.106279 */
                2074,              /* 2         0.112533 */
                2086,              /* 3         0.118738 */
                2099,              /* 4         0.124869 */
                2111,              /* 5         0.130902 */
                2123,              /* 6         0.136812 */
                2135,              /* 7         0.142578 */
                2147,              /* 8         0.148175 */
                2158,              /* 9         0.153583 */
                2168,              /* 10        0.158779 */
                2179,              /* 11        0.163742 */
                2188,              /* 12        0.168455 */
                2197,              /* 13        0.172897 */
                2206,              /* 14        0.177051 */
                2214,              /* 15        0.180902 */
                2221,              /* 16        0.184433 */
                2227,              /* 17        0.187631 */
                2233,              /* 18        0.190483 */
                2238,              /* 19        0.192978 */
                2243,              /* 20        0.195106 */
                2246,              /* 21        0.196858 */
                2249,              /* 22        0.198229 */
                2251,              /* 23        0.199211 */
                2252,              /* 24        0.199803 */
                2253,              /* 25        0.200000 */
                2252,              /* 26        0.199803 */
                2251,              /* 27        0.199211 */
                2249,              /* 28        0.198229 */
```

```
2246,                   /* 29      0.196858 */
2243,                   /* 30      0.195106 */
2238,                   /* 31      0.192978 */
2233,                   /* 32      0.190483 */
2227,                   /* 33      0.187631 */
2221,                   /* 34      0.184433 */
2214,                   /* 35      0.180902 */
2206,                   /* 36      0.177051 */
2197,                   /* 37      0.172897 */
2188,                   /* 38      0.168455 */
2179,                   /* 39      0.163742 */
2168,                   /* 40      0.158779 */
2158,                   /* 41      0.153583 */
2147,                   /* 42      0.148175 */
2135,                   /* 43      0.142578 */
2123,                   /* 44      0.136812 */
2111,                   /* 45      0.130902 */
2099,                   /* 46      0.124869 */
2086,                   /* 47      0.118738 */
2074,                   /* 48      0.112533 */
2061,                   /* 49      0.106279 */
2048,                   /* 50      0.100000 */
2035,                   /* 51      0.093721 */
2022,                   /* 52      0.087467 */
2010,                   /* 53      0.081262 */
1997,                   /* 54      0.075131 */
1985,                   /* 55      0.069098 */
1973,                   /* 56      0.063188 */
1961,                   /* 57      0.057422 */
1949,                   /* 58      0.051825 */
1938,                   /* 59      0.046417 */
1928,                   /* 60      0.041221 */
1917,                   /* 61      0.036258 */
1908,                   /* 62      0.031545 */
1899,                   /* 63      0.027103 */
1890,                   /* 64      0.022949 */
1882,                   /* 65      0.019098 */
1875,                   /* 66      0.015567 */
1869,                   /* 67      0.012369 */
```

```
        1863,           /* 68     0.009517 */
        1858,           /* 69     0.007022 */
        1853,           /* 70     0.004894 */
        1850,           /* 71     0.003142 */
        1847,           /* 72     0.001771 */
        1845,           /* 73     0.000789 */
        1844,           /* 74     0.000197 */
        1843,           /* 75     0.000000 */
        1844,           /* 76     0.000197 */
        1845,           /* 77     0.000789 */
        1847,           /* 78     0.001771 */
        1850,           /* 79     0.003142 */
        1853,           /* 80     0.004894 */
        1858,           /* 81     0.007022 */
        1863,           /* 82     0.009517 */
        1869,           /* 83     0.012369 */
        1875,           /* 84     0.015567 */
        1882,           /* 85     0.019098 */
        1890,           /* 86     0.022949 */
        1899,           /* 87     0.027103 */
        1908,           /* 88     0.031545 */
        1917,           /* 89     0.036258 */
        1928,           /* 90     0.041221 */
        1938,           /* 91     0.046417 */
        1949,           /* 92     0.051825 */
        1961,           /* 93     0.057422 */
        1973,           /* 94     0.063188 */
        1985,           /* 95     0.069098 */
        1997,           /* 96     0.075131 */
        2010,           /* 97     0.081262 */
        2022,           /* 98     0.087467 */
        2035,           /* 99     0.093721 */
        2048                      /* 100    0.100000 */

};

/*--------------------------------------------------------------------*/
/* Hardware initialization
                          */
```

```
/*-----------------------------------------------------------------------*/
void hdwinit(void)
{
        DI();                               /* disable all interrupts */

        /*-----------------------------------------*/
        /*              Initialization of port              */
        /*-----------------------------------------*/
        fn_InitPort();

        /*----------------------------------------*/
        /*          Initialization of clock             */
        /*----------------------------------------*/
        CMC =     0b01010011;                     /* Clock Operation Mode Control Register */
                            /*|||||||+--- : Control of high-speed system clock oscillation frequency */
                            /*||||||| : 0 : 2 MHz <= fMX <= 10 MHz */
                            /*||||||| : 1 : 10 MHz < fMX <= 20 MHz */
                            /*||||||| */
                            /*|||||++---- : XT1 oscillator oscillation mode selection */
                            /*||||| : 0 0 : Low-consumption oscillation */
                            /*||||| : 0 1 : Normal oscillation */
                            /*||||| : 1 x : Super-low-consumption oscillation */
                            /*|||||    x = don't care */
                            /*||||| */
                            /*||||+------ : Be sure to set 0 */
                            /*|||| */
                            /*|||+------- : [1] Subsystem clock pin operation mode */
                            /*|||                  [2] XT1/P123 pin and XT2/P124 pin */
                            /*||| : 0 : [1]Input port mode */
                            /*|||            [2]Input port */
                            /*||| */
                            /*||| : 1 : [1]XT1 oscillation mode */
                            /*|||            [2]Crystal resonator connection */
                            /*||| */
                            /*||+-------- : Be sure to set 0 */
                            /*|| */
                            /*++--------- : [1]EXCLK OSCSEL High-speed system clock pin operation mode */
                            /*                       [2]X1/P121 pin */
                            /*                       [3]X2/EXCLK/P122 pin */
```

```
                        /* : 0 0 : [1]Input port mode */
                        /*         [2][3]Input port */
                        /* */
                        /* : 0 1 : [1]X1 oscillation mode */
                        /*         [2][3]Crystal/ceramic resonator connection */
                        /* */
                        /* : 1 0 : [1]Input port mode */
                        /*         [2][3]Input port */
                        /* */
                        /* : 1 1 : [1]External clock input mode */
                        /*         [2]Input port */
                        /*         [3]External clock input */

    MSTOP = 0;                        /* X1 oscillator operating */
    XTSTOP = 0;                       /* XT1 oscillator operating */

    OSMC =   0b00000001;              /* Operation Speed Mode Control Register */
                        /*||||||++--- : fCLK frequency selection */
                        /*|||||| : 0 0 : Operates at a frequency of 10 MHz or less. */
                        /*|||||| : 0 1 : Operates at a frequency higher than 10 MHz. */
                        /*|||||| : 1 0 : Operates at a frequency of 1 MHz. */
                        /*|||||| */
                        /*|+++++----- : Be sure to set 00000 */
                        /*| */
                        /*+---------- : Setting in subsystem clock HALT mode */
                        /* : 0 : Enables subsystem clock supply to peripheral functions. */
                        /*      (See Table 21-1 Operating Statuses in HALT Mode (2/3) */
                        /*       for the peripheral functions whose operations are enabled.) */
                        /* : 1 : Stops subsystem clock supply to peripheral functions except real-time counter,
*/
                        /*       clock output/buzzer output, and LCD controller/driver. */

    while(OSTC.0 != 1){           /* wait X1 oscillation stabilization */
            NOP();
    }

    /*-- Caution ---------------------------------------------*/
    /* To increase fCLK to 10 MHz or higher, set FSEL to '1',    */
    /* then change fCLK after two or more clocks have elapsed. */
```

```
/*-------------------------------------------------------------*/
NOP();
NOP();

CKC =	0b00010000;			/* System Clock Control Register */
			/*|+|+++++--- : Selection of CPU/peripheral hardware clock (fCLK) */
			/*| | : 0 0 x 0 0 0 : fIH */
			/*| | : 0 0 x 0 0 1 : fIH/2 (default) */
			/*| | : 0 0 x 0 1 0 : fIH/2^2 */
			/*| | : 0 0 x 0 1 1 : fIH/2^3 */
			/*| | : 0 0 x 1 0 0 : fIH/2^4 */
			/*| | : 0 0 x 1 0 1 : fIH/2^5 */
			/*| | : 0 1 x 0 0 0 : fMX */
			/*| | : 0 1 x 0 0 1 : fMX/2 */
			/*| | : 0 1 x 0 1 0 : fMX/2^2 */
			/*| | : 0 1 x 0 1 1 : fMX/2^3 */
			/*| | : 0 1 x 1 0 0 : fMX/2^4 */
			/*| | : 0 1 x 1 0 1 : fMX/2^5 */
			/*| | : 1 x 0 x x x : fSUB */
			/*| | : 1 x 1 x x x : fSUB/2 */
			/*| | : Other than above : Setting prohibited */
			/*| |    x = don't care */
			/*| | */
			/*| +-------- : Status of Main system clock (fMAIN) */
			/*| : 0 : Internal high-speed oscillation clock (fIH) */
			/*| : 1 : High-speed system clock (fMX) */
			/*| */
			/*+---------- : Status of CPU/peripheral hardware clock (fCLK) */
			/* : 0 : Main system clock (fMAIN) */
			/* : 1 : Subsystem clock (fSUB) */

/* Confirming the CPU clock status */
while((CLS != 0)||(MCS != 1)){
	NOP();
}
/* CPU is operating on a High-speed system clock */
HIOSTOP = 1;				/* internal high-speed oscillation stopped */

OSTS =	0b00000111;			/* Oscillation Stabilization Time Select Register */
```

```
}
                              /*||||++++--- : Oscillation stabilization time selection */
                              /*|||| : 0 0 0 : 2^8/fX */
                              /*|||| : 0 0 1 : 2^9/fX */
                              /*|||| : 0 1 0 : 2^10/fX */
                              /*|||| : 0 1 1 : 2^11/fX */
                              /*|||| : 1 0 0 : 2^13/fX */
                              /*|||| : 1 0 1 : 2^15/fX */
                              /*|||| : 1 1 0 : 2^17/fX */
                              /*|||| : 1 1 1 : 2^18/fX */
                              /*|||| */
                              /*++++------- : Be sure to set 000000 */

        /*----------------------------------------*/
        /*          Initialization of timer          */
        /*----------------------------------------*/
        fn_InitTau0();

        /* software to wait for the operation stabilization time */
        /* (over 200ms from when XT1 enable) */
        fn_Wait500usBase(200000/500);

        /*----------------------------------------*/
        /* Initialization of low-voltage detector */
        /*----------------------------------------*/
        fn_InitLvi();

        /*----------------------------------------*/
        /*   Initialization of A/D,D/A,OPAmp                 */
        /*----------------------------------------*/
        fn_AdcDacInit();

        /*----------------------------------------*/
        /*   Initialization of DMA                                  */
        /*----------------------------------------*/
        fn_DmaInit();

        EI();                              /* enable all interrupts */

}
```

```
/*------------------------------------------------------------*/
/* Module:          fn_InitPort
                    */
/* Description:     Setting of I/O ports                                          */
/*      parameter: --
                              */
/*      return    : --
                              */
/*------------------------------------------------------------*/
void    fn_InitPort(void)
{
/*--------------------------------------------------------*/
/*      Ports configuration for digital input and output                 */
/*--------------------------------------------------------*/
        ADPC =  0b00010000;         /* A/D Port Configuration Register */
                              /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                              /*||| :               +------------------------ ANI15/AVREFM/P157 */
                              /*||| :               | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                              /*||| :               | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
                              /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                              /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                              /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                              /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                              /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                              /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                              /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                              /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                              /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                              /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                              /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                              /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                              /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                              /*||| */
                              /*+++-------- : Be sure to set 000 */

/*--------------------------------------------------------*/
/*      Setting of Port 0
        */
```

```
/*------------------------------------------------------------*/
        P0 =        0b00000000;        /* Set P00-P02 Output latch to Low */
        PM0 =       0b11111000;        /* Set P00-P02 to output port */
                                                     /* P00-P02:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 1
        */
/*------------------------------------------------------------*/
        P1 =        0b00000000;        /* Set P10-P17 Output latch to Low */
        PM1 =       0b00000000;        /* Set P10-P17 to output port */
                                                     /* P10-P15:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 2
        */
/*------------------------------------------------------------*/
        P2 =        0b00000000;        /* Set P20-P27 Output latch to Low */
        PM2 =       0b11111111;        /* Set P20-P27 to input port */
                                                     /* P20-P27:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 3
        */
/*------------------------------------------------------------*/
        P3 =        0b00001100;        /* Set P30-P31,P34 Output latch to Low */
                                                     /* Set P33,P32 Output latch High */
        PM3 =       0b11100000;        /* Set P30-P34 to output port */
                                                     /* P30-P34:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 4
        */
/*------------------------------------------------------------*/
        P4 =        0b00000000;        /* Set P40-P41 Output latch to Low */
        PM4 =       0b11111100;        /* Set P40-P41 to output port */
                                                     /* P40-P41:Unused */

/*------------------------------------------------------------*/
```

```
/*      Setting of Port 5
        */
/*-----------------------------------------------------------*/
        P5 =    0b00000000;     /* Set P50-P57 Output latch to Low */
        PM5 =   0b11110000;     /* Set P50-P57 to output port */
                                        /* P50-P57:Unused */


/*-----------------------------------------------------------*/
/*      Setting of Port 6
        */
/*-----------------------------------------------------------*/
        P6 =    0b00000000;     /* Set P60-P61 Output latch to Low */
        PM6 =   0b11111100;     /* Set P60-P61 to output port */
                                        /* P60-P61:Unused */


/*-----------------------------------------------------------*/
/*      Setting of Port 7
        */
/*-----------------------------------------------------------*/
        P7 =    0b00000000;     /* Set P70-P77 Output latch to Low */
        PM7 =   0b00000000;     /* Set P70-P77 to output port */
                                        /* P70-P77:Unused */


/*-----------------------------------------------------------*/
/*      Setting of Port 8
        */
/*-----------------------------------------------------------*/
        P8 =    0b00000000;     /* Set P80-P88 Output latch to Low */
        PM8 =   0b00000000;     /* Set P80-P88 to output port */
                                        /* P80-P88:Unused */


/*-----------------------------------------------------------*/
/*      Setting of Port 9
        */
/*-----------------------------------------------------------*/
        P9 =    0b00000000;     /* Set P90-P97 Output latch to Low */
        PM9 =   0b00000000;     /* Set P90-P97 to output port */
                                        /* P90-P97:Unused */
```

```
/*------------------------------------------------------------*/
/*      Setting of Port 10
        */
/*------------------------------------------------------------*/
        P10 =   0b00000000;     /* Set P100-P102 Output latch to Low */
        PM10 =  0b11111000;     /* Set P100-P102 to output port */
                                            /* P100-P102:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 11
        */
/*------------------------------------------------------------*/
        P11 =   0b00000000;     /* Set P110-P111 Output latch to Low */
        PM11 =  0b11111100;     /* Set P110-P111 to output port */
                                            /* P110-P111:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 12
        */
/*------------------------------------------------------------*/
        P12 =   0b00000000;     /* Set P120 Output latch to Low */
        PM12 =  0b11111110;     /* Set P120 to output port */
                                            /* P120-P124:Unused */
                                            /* *P121-P124:Input port */

/*------------------------------------------------------------*/
/*      Setting of Port 13
        */
/*------------------------------------------------------------*/
        P13 =   0b00000000;     /* Set P130 Output latch to Low */
                                            /* P130:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 14
        */
/*------------------------------------------------------------*/
        P14 =   0b00000000;     /* Set P140-P147 Output latch to Low */
        PM14 =  0b00000000;     /* Set P140-P147 to output port */
                                            /* P140-P147:Unused */
```

```
/*----------------------------------------------------------*/
/*      Setting of Port 15
        */
/*----------------------------------------------------------*/
        P15 =   0b00000000;         /* Set P150-P152,P157 Output latch to Low */
        PM15 =  0b11111111;         /* Set P150-P152,P157 to input port */
                                                    /* P150-P152,P157:Unused */

}

/*--------------------------------------------------------------*/
/* Module:          fn_InitTau0
                            */
/* Description:     Setting of Timer array unit 0                                   */
/*      parameter: --
                            */
/*      return    : --
                            */
/*--------------------------------------------------------------*/
void    fn_InitTau0(void)
{
        TAU0EN = 1;                             /* supplies input clock to timer array unit 0 */
        TPS0L =  0b00000010;        /* Timer Clock Select Register 0 */
                            /*||||++++--- : Selection of operation clock (CK00) */
                            /*++++------- : Selection of operation clock (CK01) */
                            /* : 0 0 0 0 : CK0m = fCLK */
                            /* : 0 0 0 1 : CK0m = fCLK/2 */
                            /* : 0 0 1 0 : CK0m = fCLK/2^2 */
                            /* : 0 0 1 1 : CK0m = fCLK/2^3 */
                            /* : 0 1 0 0 : CK0m = fCLK/2^4 */
                            /* : 0 1 0 1 : CK0m = fCLK/2^5 */
                            /* : 0 1 1 0 : CK0m = fCLK/2^6 */
                            /* : 0 1 1 1 : CK0m = fCLK/2^7 */
                            /* : 1 0 0 0 : CK0m = fCLK/2^8 */
                            /* : 1 0 0 1 : CK0m = fCLK/2^9 */
                            /* : 1 0 1 0 : CK0m = fCLK/2^10 */
                            /* : 1 0 1 1 : CK0m = fCLK/2^11 */
                            /* : 1 1 0 0 : CK0m = fCLK/2^12 */
```

```
                        /* : 1 1 0 1 : CK0m = fCLK/2^13 */
                        /* : 1 1 1 0 : CK0m = fCLK/2^14 */
                        /* : 1 1 1 1 : CK0m = fCLK/2^15 */
                        /*    m = 0, 1 */

/* CH1:for wait */
TMR01 = 0b0000000000000000;             /* Timer Mode Register 01 */
                        /*||||||||||||++++--- : [1]Operation mode of channel 1 */
                        /*||||||||||||          [2]Count operation of TCR */
                        /*||||||||||||          [3]Independent operation */
                        /*||||||||||||          [4]Setting of starting counting and interrupt */
                        /*|||||||||||| : 0 0 0 0 : [1]Interval timer mode */
                        /*||||||||||||             [2]Counting down */
                        /*||||||||||||             [3]Possible */
                        /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
                        /*||||||||||||                (timer output does not change, either). */
                        /*|||||||||||| */
                        /*|||||||||||| : 0 0 0 1 : [1]Interval timer mode */
                        /*||||||||||||             [2]Counting down */
                        /*||||||||||||             [3]Possible */
                        /*||||||||||||             [4]Timer interrupt is generated when counting is started */
                        /*||||||||||||                (timer output also changes). */
                        /*|||||||||||| */
                        /*|||||||||||| : 0 1 0 0 : [1]Capture mode */
                        /*||||||||||||             [2]Counting up */
                        /*||||||||||||             [3]Possible */
                        /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
                        /*||||||||||||                (timer output does not change, either). */
                        /*|||||||||||| */
                        /*|||||||||||| : 0 1 0 1 : [1]Capture mode */
                        /*||||||||||||             [2]Counting up */
                        /*||||||||||||             [3]Possible */
                        /*||||||||||||             [4]Timer interrupt is generated when counting is started */
                        /*||||||||||||                (timer output also changes). */
                        /*|||||||||||| */
                        /*|||||||||||| : 0 1 1 0 : [1]Event counter mode */
                        /*||||||||||||             [2]Counting down */
                        /*||||||||||||             [3]Possible */
                        /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
```

```
/*||||||||||||                        (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 0 : [1]One-count mode */
/*||||||||||||                 [2]Counting down */
/*||||||||||||                 [3]Impossible */
/*||||||||||||                 [4]Start trigger is invalid during counting operation. */
/*||||||||||||                    At that time, interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 1 : [1]One-count mode */
/*||||||||||||                 [2]Counting down */
/*||||||||||||                 [3]Impossible */
/*||||||||||||                 [4]Start trigger is valid during counting operation. */
/*||||||||||||                    At that time, interrupt is also generated. */
/*|||||||||||| */
/*|||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*||||||||||||                 [2]Counting up */
/*||||||||||||                 [3]Possible */
/*||||||||||||                 [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                    (timer output does not change, either). */
/*||||||||||||                    Start trigger is invalid during counting operation. */
/*||||||||||||                    At that time interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : Other than above : Setting prohibited */
/*|||||||||||| */
/*||||||||||++------- : Be sure to set 00 */
/*|||||||||| */
/*||||||||++--------- : Selection of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 valid
edge */
/*||||||||                    (the timer input used with channel 1 is selected by using TIS0
register). */
/*|||||||| : 0 0 : Falling edge */
/*|||||||| : 0 1 : Rising edge */
/*|||||||| : 1 0 : Both edges (when low-level width is measured) */
/*||||||||              Start trigger: Falling edge, Capture trigger: Rising edge */
/*|||||||| : 1 1 : Both edges (when high-level width is measured) */
/*||||||||              Start trigger: Rising edge, Capture trigger: Falling edge */
/*|||||||| */
/*|||||+++----------- : Setting of start trigger or capture trigger of channel 1 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are unselected).
```

```
*/
                        /*|||||  : 0 0 1 : Valid edge of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 is used
as both the start trigger and capture trigger. */
                        /*|||||  : 0 1 0 : Both the edges of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 are
used as a start trigger and a capture trigger. */
                        /*|||||  : 1 0 0 : Interrupt signal of the master channel is used (when the channel is used
as a slave channel with the combination operation function). */
                        /*|||||  : Other than above : Setting prohibited */
                        /*||||| */
                        /*||||+-------------- : Selection of slave/master of channel 1 */
                        /*||||  : 0 : Operates as slave channel with combination operation function. */
                        /*||||  : 1 : Operates as master channel with combination operation function. */
                        /*|||| */
                        /*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
                        /*|||  : 0 : Operation clock MCK specified by CKS01 bit */
                        /*|||  : 1 : Valid edge of input signal input from TI01 pin, fSUB/2, fSUB/4, or INTRTC1 */
                        /*|||        (the timer input used with channel 1 is selected by using TIS0 register). */
                        /*||| */
                        /*|++---------------- : Be sure to set 00 */
                        /*| */
                        /*+------------------ : Selection of operation clock (MCK) of channel 1 */
                        /*   : 0 : Operation clock CK00 set by TPS0 register */
                        /*   : 1 : Operation clock CK01 set by TPS0 register */
      TDR01 = CCK00_500USEC;          /* set interval time to 500us */
      TMMK01 = 1;                                /* disable interrupt */


      /* CH4:for DMA */
      TMR04 = 0b1000000000000000;      /* Timer Mode Register 04 */
                        /*||||||||||||++++--- : [1]Operation mode of channel 4 */
                        /*||||||||||||          [2]Count operation of TCR */
                        /*||||||||||||          [3]Independent operation */
                        /*||||||||||||          [4]Setting of starting counting and interrupt */
                        /*|||||||||||| : 0 0 0 0 : [1]Interval timer mode */
                        /*||||||||||||              [2]Counting down */
                        /*||||||||||||              [3]Possible */
                        /*||||||||||||              [4]Timer interrupt is not generated when counting is started */
                        /*||||||||||||                 (timer output does not change, either). */
                        /*|||||||||||| */
```

```
/*|||||||||||| : 0 0 0 1 : [1]Interval timer mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is generated when counting is started */
/*||||||||||||                (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 0 0 : [1]Capture mode */
/*||||||||||||             [2]Counting up */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 0 1 0 1 : [1]Capture mode */
/*||||||||||||             [2]Counting up */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is generated when counting is started */
/*||||||||||||                (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 1 0 : [1]Event counter mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 0 : [1]One-count mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Impossible */
/*||||||||||||             [4]Start trigger is invalid during counting operation. */
/*||||||||||||                At that time, interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 1 : [1]One-count mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Impossible */
/*||||||||||||             [4]Start trigger is valid during counting operation. */
/*||||||||||||                At that time, interrupt is also generated. */
/*|||||||||||| */
/*|||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*||||||||||||             [2]Counting up */
/*||||||||||||             [3]Possible */
```

```
/*||||||||||||                    [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                       (timer output does not change, either). */
/*||||||||||||                       Start trigger is invalid during counting operation. */
/*||||||||||||                       At that time interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : Other than above : Setting prohibited */
/*|||||||||||| */
/*||||||||||++------- : Be sure to set 00 */
/*|||||||||| */
/*||||||||++--------- : Selection of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 valid
edge */
/*||||||||                    (the timer input used with channel 4 is selected by using TIS0
register). */
/*|||||||| : 0 0 : Falling edge */
/*|||||||| : 0 1 : Rising edge */
/*|||||||| : 1 0 : Both edges (when low-level width is measured) */
/*||||||||              Start trigger: Falling edge, Capture trigger: Rising edge */
/*|||||||| : 1 1 : Both edges (when high-level width is measured) */
/*||||||||              Start trigger: Rising edge, Capture trigger: Falling edge */
/*|||||||| */
/*|||||+++----------- : Setting of start trigger or capture trigger of channel 4 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are unselected).
*/
/*||||| : 0 0 1 : Valid edge of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 is used
as both the start trigger and capture trigger. */
/*||||| : 0 1 0 : Both the edges of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 are
used as a start trigger and a capture trigger. */
/*||||| : 1 0 0 : Interrupt signal of the master channel is used (when the channel is used
as a slave channel with the combination operation function). */
/*||||| : Other than above : Setting prohibited */
/*||||| */
/*||||+-------------- : Selection of slave/master of channel 4 */
/*|||| : 0 : Operates as slave channel with combination operation function. */
/*|||| : 1 : Operates as master channel with combination operation function. */
/*|||| */
/*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
/*||| : 0 : Operation clock MCK specified by CKS04 bit */
/*||| : 1 : Valid edge of input signal input from TI04 pin, fSUB/2, fSUB/4, or INTRTC1 */
/*|||          (the timer input used with channel 4 is selected by using TIS0 register). */
```

```
                        /*||| */
                        /*|++---------------- : Be sure to set 00 */
                        /*| */
                        /*+------------------ : Selection of operation clock (MCK) of channel 4 */
                        /* : 0 : Operation clock CK00 set by TPS0 register */
                        /* : 1 : Operation clock CK01 set by TPS0 register */
    TDR04 = 200;        /* set interval time to 10us(=0.05us * 200) */
    TMMK04 = 1;                 /* disable interrupt */

}

/*---------------------------------------------------------------*/
/* Module:        fn_InitLvi
                  */
/* Description:   Setting of Low-voltage detectior                                  */
/*    parameter: --
                        */
/*    return    : --
                        */
/*---------------------------------------------------------------*/
void    fn_InitLvi(void)
{
    unsigned short      loop;       /* waiting counter */

    LVIMK = 1;                              /* disable LVI interrupt */

    LVIS =    0b00001001;                   /* Low-Voltage Detection Level Select Register */
                        /*||||++++--- : Detection level */
                        /*|||| : 0 0 0 0 : VLVI0 (4.22V) */
                        /*|||| : 0 0 0 1 : VLVI1 (4.07V) */
                        /*|||| : 0 0 1 0 : VLVI2 (3.92V) */
                        /*|||| : 0 0 1 1 : VLVI3 (3.76V) */
                        /*|||| : 0 1 0 0 : VLVI4 (3.61V) */
                        /*|||| : 0 1 0 1 : VLVI5 (3.45V) */
                        /*|||| : 0 1 1 0 : VLVI6 (3.30V) */
                        /*|||| : 0 1 1 1 : VLVI7 (3.15V) */
                        /*|||| : 1 0 0 0 : VLVI8 (2.99V) */
                        /*|||| : 1 0 0 1 : VLVI9 (2.84V) */
                        /*|||| : 1 0 1 0 : VLVI10 (2.68V) */
```

```
                    /*|||| : 1 0 1 1 : VLVI11 (2.53V) */
                    /*|||| : 1 1 0 0 : VLVI12 (2.38V) */
                    /*|||| : 1 1 0 1 : VLVI13 (2.22V) */
                    /*|||| : 1 1 1 0 : VLVI14 (2.07V) */
                    /*|||| : 1 1 1 1 : VLVI15 (1.91V) */
                    /*|||| */
                    /*++++------- : Be sure to set 0000 */

    LVIM =  0b10000000;              /* Low-Voltage Detection Register */
                    /*|||||||+--- : LVIF Low-voltage detection flag */
                    /*||||||| : 0 : * LVISEL = 0: VDD >= VLVI, or when LVI operation is disabled */
                    /*|||||||       * LVISEL = 1: EXLVI >= VEXLVI, or when LVI operation is disabled */
                    /*||||||| : 1 : * LVISEL = 0: VDD < VLVI */
                    /*|||||||       * LVISEL = 1: EXLVI < VEXLVI */
                    /*||||||| */
                    /*||||||+---- : Low-voltage detection operation mode (interrupt/reset) selection(LVIMD) */
                    /*|||||| : 0 : * LVISEL = 0: Generates an internal interrupt signal */
                    /*||||||                     when VDD drops lower than VLVI (VDD < VLVI) */
                    /*||||||                     or when VDD becomes VLVI or higher (VDD >= VLVI).
*/
                    /*||||||      * LVISEL = 1: Generates an interrupt signal */
                    /*||||||                     when EXLVI drops lower than VEXLVI (EXLVI <
VEXLVI) */
                    /*||||||                     or when EXLVI becomes VEXLVI or higher (EXLVI >=
VEXLVI). */
                    /*|||||| : 1 : * LVISEL = 0: Generates an internal reset signal when VDD < VLVI */
                    /*||||||                     and releases the reset signal when VDD >= VLVI. */
                    /*||||||      * LVISEL = 1: Generates an internal reset signal when EXLVI < VEXLVI
*/
                    /*||||||                     and releases the reset signal when EXLVI >= VEXLVI.
*/
                    /*|||||| */
                    /*|||||+----- : Voltage detection selection(LVISEL) */
                    /*||||| : 0 : Detects level of supply voltage (VDD) */
                    /*||||| : 1 : Detects level of input voltage from external input pin (EXLVI) */
                    /*||||| */
                    /*|++++------ : Be sure to set 0000 */
                    /*| */
                    /*+---------- : Enables low-voltage detection operation */
```

```
                            /* : 0 : Disables operation */
                            /* : 1 : Enables operation */

        /* software to wait for the operation stabilization time (210us) */
        for(loop = 500; loop > 0; loop--){
                NOP();
        }
        /* wait until VLVI≤VDD */
        while( LVIF ){
                NOP();
        }

        LVIIF = 0;                      /* clear LVI interrupt request flag */
}



/*-----------------------------------------------------------------*/
/* Module:          fn_DmaInit
                            */
/* Description:     Setting of DMA contoroler                                       */
/*      parameter: --
                            */
/*      return    : --
                            */
/*-----------------------------------------------------------------*/
void    fn_DmaInit(void)
{

        DEN0 = 1;                                       /* Enables operation of DMA channel 0 */

        DSA0 = (unsigned char)&DACSW0;                                  /* DMA SFR Address Register 0
*/
        DRA0 = (unsigned short)&ushDacData[0];          /* DMA RAM Address Register 0 */
        DBC0 = 100;                                             /* DMA Byte Count Register 0 */
        DMC0 = 0b01100100;                              /* DMA Mode Control Register 0 */
                        /*|||||||| */
                        /*||||++++------- IFC3-IFC0 */
                        /*||||              [Selection of DMA start source] */
                        /*||||               0000:Disables DMA transfer by interrupt */
```

```
                    /*||||                           (Only software trigger is enabled.) */
                    /*||||            0010:INTTM00 */
                    /*||||            0011:INTTM01 */
                    /*||||            0100:INTTM04 */
                    /*||||            0101:INTTM05 */
                    /*||||            0110:INTST0/INTCSI00 */
                    /*||||            0111:INTSR0/INTCSI01 */
                    /*||||            1000:INTST1/INTCSI10/INTIIC10 */
                    /*||||            1001:INTSR1 */
                    /*||||            1010:INTST3 */
                    /*||||            1011:INTSR3 */
                    /*||||            1100:INTAD */
                    /*|||+----------- DWAIT0 */
                    /*|||            [Pending of DMA transfer] */
                    /*|||             0:Executes DMA transfer upon DMA start request (not held pending). */
                    /*|||             1:Holds DMA start request pending if any. */
                    /*||+------------ DS0 */
                    /*||             [Specification of transfer data size for DMA transfer] */
                    /*||              0:8bits */
                    /*||              1:16bits */
                    /*|+------------- DRS0 */
                    /*|              [Selection of DMA transfer direction] */
                    /*|               0:SFR to internal RAM */
                    /*|               1:Internal RAM to SFR */
                    /*+------------- STG0 */
                    /*               [DMA transfer start software trigger] */
                    /*                0:No trigger operation */
                    /*                1:DMA transfer is started when DMA operation is enabled (DEN0 = 1). */

        DST0 = 1;                   /* DMA transfer is started when DMA operation is enabled (DEN0 = 1). */

}


/*--------------------------------------------------------------*/
/* Module:          fn_Wait500usBase
        */
/* Description:     Delays the program for (Time * 500us)                    */
/*      parameter: wait time(Time)
```

```
            */
/*          return      : --
                                    */
/*-------------------------------------------------------------*/
void    fn_Wait500usBase(unsigned short Time)
{
        TS0L.1 = 1;                             /* start TAU0 CH1 */
        TMIF01 = 0;

        for(; Time > 0; Time--){            /* wait for (parameter * 500)us */
                while(!TMIF01){
                        NOP();
                }
                TMIF01 = 0;
        }

        TT0L.1 = 1;                             /* stop TAU0 CH1 */
}


/*-------------------------------------------------------------*/
/* Module:          fn_AdcDacInit
                    */
/* Description:     Initialization of A/D D/A OPAmp module                  */
/*          parameter: --
                                    */
/*          return      : --
                                    */
/*-------------------------------------------------------------*/
void    fn_AdcDacInit(void)
{
        ADCEN = 1;                              /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

        /*-------------------------------------------*/
        /*    Initialization of Voltage reference                 */
        /*-------------------------------------------*/
        ADVRC = 0b00001000;             /* Analog reference voltage control register */
                            /*||||+|++---  :  [1]Positive reference voltage supplies selection of A/D and D/A
```

```
converters */
                          /*|||| |         [2]Operationcontrol of voltage reference */
                          /*|||| |         [3]Output voltage selection of voltage reference */
                          /*|||| |         [4]Operation control of input gate voltage boost circuit for A/D converter
*/
                          /*|||| |         [5]Relationship with the conversion mode used */
                          /*|||| | */
                          /*|||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
                          /*|||| |           [2]Stops operation (Hi-Z) */
                          /*|||| |           [3]2.5 V */
                          /*|||| |           [4]Stops operation */
                          /*|||| |           [5]Can be set in conversion mode 1 */
                          /*|||| | */
                          /*|||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
                          /*|||| |           [2]Stops operation (Hi-Z) */
                          /*|||| |           [3]2.0 V */
                          /*|||| |           [4]Enables operation */
                          /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                          /*|||| | */
                          /*|||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
                          /*|||| |           [2]Stops operation (pull-down output) */
                          /*|||| |           [3]2.5 V */
                          /*|||| |           [4]Stops operation */
                          /*|||| |           [5] - */
                          /*|||| | */
                          /*|||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
                          /*|||| |           [2]Enables operation */
                          /*|||| |           [3]2.5 V */
                          /*|||| |           [4]Enables operation */
                          /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                          /*|||| | */
                          /*|||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
                          /*|||| |           [2]Stops operation (pull-down output) */
                          /*|||| |           [3]2.0 V */
                          /*|||| |           [4]Enables operation */
                          /*|||| |           [5] - */
                          /*|||| | */
                          /*|||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
                          /*|||| |           [2]Enables operation */
```

```
                    /*|||| |                [3]2.0 V */
                    /*|||| |                [4]Enables operation */
                    /*|||| |                [5]Can be set in conversion mode 2 or 3 */
                    /*|||| | */
                    /*|||| | : Other than the above : Setting prohibited */
                    /*|||| | */
                    /*|+++-+----- : Be sure to set 0000 */
                    /*| */
                    /*+---------- : Reference voltage supply (negative side) of A/D converter selection */
                    /* : 0 : AVSS */
                    /* : 1 : AVREFM (external voltage reference input) */

    ADVRC.0 = 1;                /* Enables operation */
    ADVRC.1 = 1;                /* Output 2.0 V */

    ADPC =   0b00000000;        /* A/D Port Configuration Register */
                    /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                    /*||| :                 +------------------------ ANI15/AVREFM/P157 */
                    /*||| :                 | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                    /*||| :                 | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
                    /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                    /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                    /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                    /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                    /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                    /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                    /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                    /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                    /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                    /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                    /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                    /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                    /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                    /*||| */
                    /*+++-------- : Be sure to set 000 */

    /* wait for settling time to 17msec */
    fn_Wait500usBase(17000/500);
```

```
        /*-------------------------------------------*/
        /*    Initialization of operational amplifier    */
        /*-------------------------------------------*/
        PM2.1 = 1;                  /* input port mode (AMP0O) */
        PM2.0 = 1;                  /* input port mode (AMP0-) */
        PM2.2 = 1;                  /* input port mode (AMP0+) */

        PM2.4 = 1;                  /* input port mode (AMP1O) */
        PM2.3 = 1;                  /* input port mode (AMP1-) */
        PM2.5 = 1;                  /* input port mode (AMP1+) */

        PM2.7 = 1;                  /* input port mode (AMP2O) */
        PM2.6 = 1;                  /* input port mode (AMP2-) */
        PM15.0 = 1;                 /* input port mode (AMP2+) */

        PM11.0 = 1;                 /* input port mode (ANO0) */

        ADCEN = 1;                  /* supplies input clock to OPAmp A/D converter */
        OAC.0 = 1;                  /* operational amplifier (AMP0) enable */
        OAC.1 = 0;                  /* operational amplifier (AMP1) disable */
        OAC.2 = 0;                  /* operational amplifier (AMP2) disable */

        /*-------------------------------------------*/
        /*          Initialization of D/A converter          */
        /*-------------------------------------------*/
        DACEN = 1;                          /* supplies input clock to D/A converter */

    /*  DAMD0 = 1;                          /* [DAMD0] D/A converter operation is Real-time
output mode */
        DAM.0 = 0;                          /* [DAMD0] D/A converter operation is Normal mode
*/
        DAM.2 = 1;                          /* [DARES0]D/A conversion resolution is 12-bit */
        DAM.6 = 1;                          /* [DAREF] D/A converter positive reference voltage
is VREFOUT */
        DACSW0 = aushDacData[99];   /* D/A conversion value setting reigster 0 */

        /* set D/A chanel1 */
        DAM.1 = 0;                          /* [DAMD1] D/A converter operation is Normal mode
*/
```

```
        DAM.3 = 1;                                              /* [DARES1]D/A conversion resolution is 12-bit */
        DACSW1 = aushDacData[99];                 /* D/A conversion value setting reigster 1 */

}
/*-------------------------------------------------------------*/
/* Module:          main
                                */
/* Description:     Main process
        */
/*      parameter: --
                                */
/*      return    : --
                                */
/*-------------------------------------------------------------*/
void main(void)
{
        unsigned short        count;                /* counter for initializing D/A convert data */
        DACE1 = 1;                                              /* D/A converter CH1 enable */
        DACE0 = 1;                                              /* D/A converter CH0 enable */


        /*==================================*/
        /*---------------------------------*/
        /*                Main Loop                  */
        /*---------------------------------*/
        /*==================================*/

        /*    initialize D/Aconverter data */
        for(count = 0; count<100; count++){
                ushDacData[count] = aushDacData[count];
        }

        TS0L.4 = 1;                               /* start TAU0 CH4 */
        TMIF04 = 0;                               /* reset INTTM04 interrupt request */

        DMAMK0 = 0;                               /* enable DMA CH0 interrupt */

        EI();                                     /* enable all interrupts */
```

```
        /*==============================================================*/
        /* if system have nothing to do, go to standby for power-saving */
        /*==============================================================*/


        while (1){
                HALT();
                NOP();
        }
}


/*----------------------------------------------------------------*/
/* Module:          fn_intdma0
                            */
/* Description:     DMA ch0 interruput process                                     */
/*      parameter: --
                            */
/*      return     : --
                            */
/*----------------------------------------------------------------*/
__interrupt void fn_intdma0(void)
{
        DRA0 = (unsigned short)&ushDacData[0];         /* DMA RAM Address Register 0 */
        DBC0 = 100;                                             /* DMA Byte Count Register 0 */

        DST0 = 1;                              /*  DMA  transfer  is  started  when  DMA  operation  is  enabled
(DEN0 = 1). */
}
```

(3) Offset adjustment of inverting amplifier circuit

● main.c

```
/*
 * Copyright (C) NEC Electronics Corporation 2006
 * NEC ELECTRONICS CONFIDENTIAL AND PROPRIETARY
 * All rights reserved by NEC Electronics Corporation.
 * This program must be used solely for the purpose for which
 * it was furnished by NEC Electronics Corporation. No part of this
 * program may be reproduced or disclosed to others, in any
 * form, without the prior written permission of NEC Electronics
 * Corporation. Use of copyright notice dose not evidence
 * publication of the program.
 */

/*----------------------------------------------------------------------*/
/* #pragma directive for CC78K0
                                        */
/*----------------------------------------------------------------------*/
#pragma         SFR
#pragma         DI
#pragma         EI
#pragma         HALT
#pragma         NOP

#pragma interrupt INTDMA0 fn_intdma0 RB1

/*----------------------------------------------------------------------*/
/* Include files
                                                    */
/*----------------------------------------------------------------------*/
/* TAU:TDR0n value operation by CK00 (fCLK/2^3 = 5MHz) */
#define CCK00_500USEC    (2500 - 1)        /* 500us (0.2[us/clk] * 2500[count]) */


/*----------------------------------------------------------------------*/
/* Function prototyps
                                                    */
/*----------------------------------------------------------------------*/
```

```
                void    fn_Wait500usBase(unsigned short);     /* Delays the program for (Time * 500us) */
                void    fn_InitPort(void);                              /* Setting of I/O ports
*/
                void    fn_InitTau0(void);                              /* Setting of Timer
array unit 0 */
                void    fn_InitLvi(void);                               /* Setting of
Low-voltage detectior */
                void    fn_InitVr(void);                                /* Setting of Voltage
reference */
                void    fn_AdcDacInit(void);                       /* Setting of D/A converter */
                void    fn_DmaInit(void);                               /* Setting of DMA
controler */


  /*--------------------------------------------------------------------*/
  /* Extern variables/constants
                */
  /*--------------------------------------------------------------------*/

  /*--------------------------------------------------------------------*/
  /* Local constants
                          */
  /*--------------------------------------------------------------------*/
  /*--------------------------------------------------------------------*/
  /* Global variables
                          */
  /*--------------------------------------------------------------------*/

  /*--------------------------------------------------------------------*/
  /* Local variables
                          */
  /*--------------------------------------------------------------------*/
  static   short        shVoltage;                        /* voltage (Hex) */
  static   unsigned short     ushCycleCount;       /* Cycle counter */

  /*--------------------------------------------------------------------*/
  /* Code
                                  */
  /*--------------------------------------------------------------------*/
```

```
/*===============================================================================

        D/A output data

===============================================================================*/
static unsigned short ushDacData[100];          /* D/A data for DMA */
const unsigned short aushDacData[100] = {       /* defined D/A data */
                                                /* n        sin(0.2pi*n/100) */
/*              2048,               /* 0        0.100000 */
                2061,               /* 1        0.106279 */
                2074,               /* 2        0.112533 */
                2086,               /* 3        0.118738 */
                2099,               /* 4        0.124869 */
                2111,               /* 5        0.130902 */
                2123,               /* 6        0.136812 */
                2135,               /* 7        0.142578 */
                2147,               /* 8        0.148175 */
                2158,               /* 9        0.153583 */
                2168,               /* 10       0.158779 */
                2179,               /* 11       0.163742 */
                2188,               /* 12       0.168455 */
                2197,               /* 13       0.172897 */
                2206,               /* 14       0.177051 */
                2214,               /* 15       0.180902 */
                2221,               /* 16       0.184433 */
                2227,               /* 17       0.187631 */
                2233,               /* 18       0.190483 */
                2238,               /* 19       0.192978 */
                2243,               /* 20       0.195106 */
                2246,               /* 21       0.196858 */
                2249,               /* 22       0.198229 */
                2251,               /* 23       0.199211 */
                2252,               /* 24       0.199803 */
                2253,               /* 25       0.200000 */
                2252,               /* 26       0.199803 */
                2251,               /* 27       0.199211 */
                2249,               /* 28       0.198229 */
                2246,               /* 29       0.196858 */
                2243,               /* 30       0.195106 */
```

```
2238,           /* 31     0.192978 */
2233,           /* 32     0.190483 */
2227,           /* 33     0.187631 */
2221,           /* 34     0.184433 */
2214,           /* 35     0.180902 */
2206,           /* 36     0.177051 */
2197,           /* 37     0.172897 */
2188,           /* 38     0.168455 */
2179,           /* 39     0.163742 */
2168,           /* 40     0.158779 */
2158,           /* 41     0.153583 */
2147,           /* 42     0.148175 */
2135,           /* 43     0.142578 */
2123,           /* 44     0.136812 */
2111,           /* 45     0.130902 */
2099,           /* 46     0.124869 */
2086,           /* 47     0.118738 */
2074,           /* 48     0.112533 */
2061,           /* 49     0.106279 */
2048,           /* 50     0.100000 */
2035,           /* 51     0.093721 */
2022,           /* 52     0.087467 */
2010,           /* 53     0.081262 */
1997,           /* 54     0.075131 */
1985,           /* 55     0.069098 */
1973,           /* 56     0.063188 */
1961,           /* 57     0.057422 */
1949,           /* 58     0.051825 */
1938,           /* 59     0.046417 */
1928,           /* 60     0.041221 */
1917,           /* 61     0.036258 */
1908,           /* 62     0.031545 */
1899,           /* 63     0.027103 */
1890,           /* 64     0.022949 */
1882,           /* 65     0.019098 */
1875,           /* 66     0.015567 */
1869,           /* 67     0.012369 */
1863,           /* 68     0.009517 */
1858,           /* 69     0.007022 */
```

```
        1853,           /* 70   0.004894 */
        1850,           /* 71   0.003142 */
        1847,           /* 72   0.001771 */
        1845,           /* 73   0.000789 */
        1844,           /* 74   0.000197 */
        1843,           /* 75   0.000000 */
        1844,           /* 76   0.000197 */
        1845,           /* 77   0.000789 */
        1847,           /* 78   0.001771 */
        1850,           /* 79   0.003142 */
        1853,           /* 80   0.004894 */
        1858,           /* 81   0.007022 */
        1863,           /* 82   0.009517 */
        1869,           /* 83   0.012369 */
        1875,           /* 84   0.015567 */
        1882,           /* 85   0.019098 */
        1890,           /* 86   0.022949 */
        1899,           /* 87   0.027103 */
        1908,           /* 88   0.031545 */
        1917,           /* 89   0.036258 */
        1928,           /* 90   0.041221 */
        1938,           /* 91   0.046417 */
        1949,           /* 92   0.051825 */
        1961,           /* 93   0.057422 */
        1973,           /* 94   0.063188 */
        1985,           /* 95   0.069098 */
        1997,           /* 96   0.075131 */
        2010,           /* 97   0.081262 */
        2022,           /* 98   0.087467 */
        2035,           /* 99   0.093721 */
        2048            /* 100  0.100000 */

};

/*----------------------------------------------------------------------*/
/* Hardware initialization
                                    */
/*----------------------------------------------------------------------*/
void hdwinit(void)
```

```
{
        DI();                                   /* disable all interrupts */

        /*----------------------------------------*/
        /*          Initialization of port          */
        /*----------------------------------------*/
        fn_InitPort();

        /*---------------------------------------*/
        /*          Initialization of clock         */
        /*---------------------------------------*/
        CMC =    0b01010011;                    /* Clock Operation Mode Control Register */
                          /*|||||||+--- : Control of high-speed system clock oscillation frequency */
                          /*||||||| : 0 : 2 MHz <= fMX <= 10 MHz */
                          /*||||||| : 1 : 10 MHz < fMX <= 20 MHz */
                          /*||||||| */
                          /*|||||++---- : XT1 oscillator oscillation mode selection */
                          /*||||| : 0 0 : Low-consumption oscillation */
                          /*||||| : 0 1 : Normal oscillation */
                          /*||||| : 1 x : Super-low-consumption oscillation */
                          /*|||||   x = don't care */
                          /*||||| */
                          /*||||+------ : Be sure to set 0 */
                          /*|||| */
                          /*|||+------- : [1] Subsystem clock pin operation mode */
                          /*|||                [2] XT1/P123 pin and XT2/P124 pin */
                          /*||| : 0 : [1]Input port mode */
                          /*|||           [2]Input port */
                          /*||| */
                          /*||| : 1 : [1]XT1 oscillation mode */
                          /*|||           [2]Crystal resonator connection */
                          /*||| */
                          /*||+-------- : Be sure to set 0 */
                          /*|| */
                          /*++--------- : [1]EXCLK OSCSEL High-speed system clock pin operation mode */
                          /*                    [2]X1/P121 pin */
                          /*                    [3]X2/EXCLK/P122 pin */
                          /* : 0 0 : [1]Input port mode */
                          /*             [2][3]Input port */
```

```
                    /* */
                    /* : 0 1 : [1]X1 oscillation mode */
                    /*         [2][3]Crystal/ceramic resonator connection */
                    /* */
                    /* : 1 0 : [1]Input port mode */
                    /*         [2][3]Input port */
                    /* */
                    /* : 1 1 : [1]External clock input mode */
                    /*         [2]Input port */
                    /*         [3]External clock input */

    MSTOP = 0;                      /* X1 oscillator operating */
    XTSTOP = 0;                     /* XT1 oscillator operating */

    OSMC =  0b00000001;             /* Operation Speed Mode Control Register */
                    /*||||||++--- : fCLK frequency selection */
                    /*|||||| : 0 0 : Operates at a frequency of 10 MHz or less. */
                    /*|||||| : 0 1 : Operates at a frequency higher than 10 MHz. */
                    /*|||||| : 1 0 : Operates at a frequency of 1 MHz. */
                    /*|||||| */
                    /*|+++++----- : Be sure to set 00000 */
                    /*| */
                    /*+---------- : Setting in subsystem clock HALT mode */
                    /* : 0 : Enables subsystem clock supply to peripheral functions. */
                    /*      (See Table 21-1 Operating Statuses in HALT Mode (2/3) */
                    /*       for the peripheral functions whose operations are enabled.) */
                    /* : 1 : Stops subsystem clock supply to peripheral functions except real-time counter,
*/
                    /*       clock output/buzzer output, and LCD controller/driver. */

    while(OSTC.0 != 1){             /* wait X1 oscillation stabilization */
        NOP();
    }

    /*-- Caution ---------------------------------------------*/
    /* To increase fCLK to 10 MHz or higher, set FSEL to '1',    */
    /* then change fCLK after two or more clocks have elapsed. */
    /*--------------------------------------------------------*/
    NOP();
```

```
NOP();

CKC =      0b00010000;                  /* System Clock Control Register */
                    /*|+|+++++--- : Selection of CPU/peripheral hardware clock (fCLK) */
                    /*| | : 0 0 x 0 0 0 : fIH */
                    /*| | : 0 0 x 0 0 1 : fIH/2 (default) */
                    /*| | : 0 0 x 0 1 0 : fIH/2^2 */
                    /*| | : 0 0 x 0 1 1 : fIH/2^3 */
                    /*| | : 0 0 x 1 0 0 : fIH/2^4 */
                    /*| | : 0 0 x 1 0 1 : fIH/2^5 */
                    /*| | : 0 1 x 0 0 0 : fMX */
                    /*| | : 0 1 x 0 0 1 : fMX/2 */
                    /*| | : 0 1 x 0 1 0 : fMX/2^2 */
                    /*| | : 0 1 x 0 1 1 : fMX/2^3 */
                    /*| | : 0 1 x 1 0 0 : fMX/2^4 */
                    /*| | : 0 1 x 1 0 1 : fMX/2^5 */
                    /*| | : 1 x 0 x x x : fSUB */
                    /*| | : 1 x 1 x x x : fSUB/2 */
                    /*| | : Other than above : Setting prohibited */
                    /*| |   x = don't care */
                    /*| | */
                    /*| +-------- : Status of Main system clock (fMAIN) */
                    /*| : 0 : Internal high-speed oscillation clock (fIH) */
                    /*| : 1 : High-speed system clock (fMX) */
                    /*| */
                    /*+---------- : Status of CPU/peripheral hardware clock (fCLK) */
                    /* : 0 : Main system clock (fMAIN) */
                    /* : 1 : Subsystem clock (fSUB) */

/* Confirming the CPU clock status */
while((CLS != 0)||(MCS != 1)){
        NOP();
}
/* CPU is operating on a High-speed system clock */
HIOSTOP = 1;                            /* internal high-speed oscillation stopped */

OSTS =     0b00000111;                  /* Oscillation Stabilization Time Select Register */
                    /*||||++++--- : Oscillation stabilization time selection */
                    /*|||| : 0 0 0 : 2^8/fX */
```

```
                                /*|||| : 0 0 1 : 2^9/fX */
                                /*|||| : 0 1 0 : 2^10/fX */
                                /*|||| : 0 1 1 : 2^11/fX */
                                /*|||| : 1 0 0 : 2^13/fX */
                                /*|||| : 1 0 1 : 2^15/fX */
                                /*|||| : 1 1 0 : 2^17/fX */
                                /*|||| : 1 1 1 : 2^18/fX */
                                /*|||| */
                                /*++++------- : Be sure to set 000000 */

        /*----------------------------------------*/
        /*          Initialization of timer        */
        /*----------------------------------------*/
        fn_InitTau0();

        /* software to wait for the operation stabilization time */
        /* (over 200ms from when XT1 enable) */
        fn_Wait500usBase(200000/500);

        /*----------------------------------------*/
        /* Initialization of low-voltage detector */
        /*----------------------------------------*/
        fn_InitLvi();

        /*----------------------------------------*/
        /*   Initialization of A/D,D/A,OPAmp                 */
        /*----------------------------------------*/
        fn_AdcDacInit();

        /*----------------------------------------*/
        /*   Initialization of DMA                                        */
        /*----------------------------------------*/
        fn_DmaInit();

        EI();                                   /* enable all interrupts */

}

/*--------------------------------------------------------------*/
```

```
/* Module:          fn_InitPort
                    */
/* Description:     Setting of I/O ports                                        */
/*       parameter: --
                              */
/*       return    : --
                              */
/*------------------------------------------------------------*/
void     fn_InitPort(void)
{
/*---------------------------------------------------------*/
/*       Ports configuration for digital input and output                  */
/*---------------------------------------------------------*/
         ADPC =   0b00010000;         /* A/D Port Configuration Register */
                              /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                              /*||| :                 +------------------------ ANI15/AVREFM/P157 */
                              /*||| :                 | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                              /*||| :                 | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
                              /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                              /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                              /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                              /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                              /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                              /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                              /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                              /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                              /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                              /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                              /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                              /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                              /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                              /*||| */
                              /*+++-------- : Be sure to set 000 */

/*---------------------------------------------------------*/
/*       Setting of Port 0
         */
/*---------------------------------------------------------*/
         P0 =     0b00000000;         /* Set P00-P02 Output latch to Low */
```

```
        PM0 =   0b11111000;     /* Set P00-P02 to output port */
                                        /* P00-P02:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 1
        */
/*------------------------------------------------------------*/
        P1 =    0b00000000;     /* Set P10-P17 Output latch to Low */
        PM1 =   0b00000000;     /* Set P10-P17 to output port */
                                        /* P10-P15:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 2
        */
/*------------------------------------------------------------*/
        P2 =    0b00000000;     /* Set P20-P27 Output latch to Low */
        PM2 =   0b11111111;     /* Set P20-P27 to input port */
                                        /* P20-P27:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 3
        */
/*------------------------------------------------------------*/
        P3 =    0b00001100;     /* Set P30-P31,P34 Output latch to Low */
                                        /* Set P33,P32 Output latch High */
        PM3 =   0b11100000;     /* Set P30-P34 to output port */
                                        /* P30-P34:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 4
        */
/*------------------------------------------------------------*/
        P4 =    0b00000000;     /* Set P40-P41 Output latch to Low */
        PM4 =   0b11111100;     /* Set P40-P41 to output port */
                                        /* P40-P41:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 5
        */
```

```
/*--------------------------------------------------------*/
        P5 =      0b00000000;       /* Set P50-P57 Output latch to Low */
        PM5 =     0b11110000;       /* Set P50-P57 to output port */
                                                      /* P50-P57:Unused */

/*--------------------------------------------------------*/
/*      Setting of Port 6
        */
/*--------------------------------------------------------*/
        P6 =      0b00000000;       /* Set P60-P61 Output latch to Low */
        PM6 =     0b11111100;       /* Set P60-P61 to output port */
                                                      /* P60-P61:Unused */

/*--------------------------------------------------------*/
/*      Setting of Port 7
        */
/*--------------------------------------------------------*/
        P7 =      0b00000000;       /* Set P70-P77 Output latch to Low */
        PM7 =     0b00000000;       /* Set P70-P77 to output port */
                                                      /* P70-P77:Unused */

/*--------------------------------------------------------*/
/*      Setting of Port 8
        */
/*--------------------------------------------------------*/
        P8 =      0b00000000;       /* Set P80-P88 Output latch to Low */
        PM8 =     0b00000000;       /* Set P80-P88 to output port */
                                                      /* P80-P88:Unused */

/*--------------------------------------------------------*/
/*      Setting of Port 9
        */
/*--------------------------------------------------------*/
        P9 =      0b00000000;       /* Set P90-P97 Output latch to Low */
        PM9 =     0b00000000;       /* Set P90-P97 to output port */
                                                      /* P90-P97:Unused */

/*--------------------------------------------------------*/
/*      Setting of Port 10
```

```
        */
/*----------------------------------------------------------*/
        P10 =     0b00000000;        /* Set P100-P102 Output latch to Low */
        PM10 =    0b11111000;        /* Set P100-P102 to output port */
                                                          /* P100-P102:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 11
        */
/*----------------------------------------------------------*/
        P11 =     0b00000000;        /* Set P110-P111 Output latch to Low */
        PM11 =    0b11111100;        /* Set P110-P111 to output port */
                                                          /* P110-P111:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 12
        */
/*----------------------------------------------------------*/
        P12 =     0b00000000;        /* Set P120 Output latch to Low */
        PM12 =    0b11111110;        /* Set P120 to output port */
                                                          /* P120-P124:Unused */
                                                          /* *P121-P124:Input port */

/*----------------------------------------------------------*/
/*      Setting of Port 13
        */
/*----------------------------------------------------------*/
        P13 =     0b00000000;        /* Set P130 Output latch to Low */
                                                          /* P130:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 14
        */
/*----------------------------------------------------------*/
        P14 =     0b00000000;        /* Set P140-P147 Output latch to Low */
        PM14 =    0b00000000;        /* Set P140-P147 to output port */
                                                          /* P140-P147:Unused */

/*----------------------------------------------------------*/
```

```
/*       Setting of Port 15
         */
/*-----------------------------------------------------------*/
         P15 =    0b00000000;       /* Set P150-P152,P157 Output latch to Low */
         PM15 =   0b11111111;       /* Set P150-P152,P157 to input port */
                                                  /* P150-P152,P157:Unused */

}

/*-------------------------------------------------------------*/
/* Module:          fn_InitTau0
                          */
/* Description:     Setting of Timer array unit 0                                        */
/*       parameter: --
                          */
/*       return     : --
                          */
/*-------------------------------------------------------------*/
void     fn_InitTau0(void)
{
         TAU0EN = 1;                          /* supplies input clock to timer array unit 0 */
         TPS0L =  0b00000010;       /* Timer Clock Select Register 0 */
                          /*||||++++--- : Selection of operation clock (CK00) */
                          /*++++------- : Selection of operation clock (CK01) */
                          /* : 0 0 0 0 : CK0m = fCLK */
                          /* : 0 0 0 1 : CK0m = fCLK/2 */
                          /* : 0 0 1 0 : CK0m = fCLK/2^2 */
                          /* : 0 0 1 1 : CK0m = fCLK/2^3 */
                          /* : 0 1 0 0 : CK0m = fCLK/2^4 */
                          /* : 0 1 0 1 : CK0m = fCLK/2^5 */
                          /* : 0 1 1 0 : CK0m = fCLK/2^6 */
                          /* : 0 1 1 1 : CK0m = fCLK/2^7 */
                          /* : 1 0 0 0 : CK0m = fCLK/2^8 */
                          /* : 1 0 0 1 : CK0m = fCLK/2^9 */
                          /* : 1 0 1 0 : CK0m = fCLK/2^10 */
                          /* : 1 0 1 1 : CK0m = fCLK/2^11 */
                          /* : 1 1 0 0 : CK0m = fCLK/2^12 */
                          /* : 1 1 0 1 : CK0m = fCLK/2^13 */
                          /* : 1 1 1 0 : CK0m = fCLK/2^14 */
```

```
                    /* : 1 1 1 1 : CK0m = fCLK/2^15 */
                    /*    m = 0, 1 */

/* CH1:for wait */
TMR01 = 0b0000000000000000;          /* Timer Mode Register 01 */
                    /*||||||||||||++++--- : [1]Operation mode of channel 1 */
                    /*||||||||||||          [2]Count operation of TCR */
                    /*||||||||||||          [3]Independent operation */
                    /*||||||||||||          [4]Setting of starting counting and interrupt */
                    /*|||||||||||| : 0 0 0 0 : [1]Interval timer mode */
                    /*||||||||||||             [2]Counting down */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                (timer output does not change, either). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 0 0 1 : [1]Interval timer mode */
                    /*||||||||||||             [2]Counting down */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is generated when counting is started */
                    /*||||||||||||                (timer output also changes). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 0 0 : [1]Capture mode */
                    /*||||||||||||             [2]Counting up */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                (timer output does not change, either). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 0 1 : [1]Capture mode */
                    /*||||||||||||             [2]Counting up */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is generated when counting is started */
                    /*||||||||||||                (timer output also changes). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 1 0 : [1]Event counter mode */
                    /*||||||||||||             [2]Counting down */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                (timer output does not change, either). */
                    /*|||||||||||| */
```

```
/*||||||||||||| : 1 0 0 0 : [1]One-count mode */
/*|||||||||||||             [2]Counting down */
/*|||||||||||||             [3]Impossible */
/*|||||||||||||             [4]Start trigger is invalid during counting operation. */
/*|||||||||||||                At that time, interrupt is not generated, either. */
/*||||||||||||| */
/*||||||||||||| : 1 0 0 1 : [1]One-count mode */
/*|||||||||||||             [2]Counting down */
/*|||||||||||||             [3]Impossible */
/*|||||||||||||             [4]Start trigger is valid during counting operation. */
/*|||||||||||||                At that time, interrupt is also generated. */
/*||||||||||||| */
/*||||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*|||||||||||||             [2]Counting up */
/*|||||||||||||             [3]Possible */
/*|||||||||||||             [4]Timer interrupt is not generated when counting is started */
/*|||||||||||||                (timer output does not change, either). */
/*|||||||||||||                Start trigger is invalid during counting operation. */
/*|||||||||||||                At that time interrupt is not generated, either. */
/*||||||||||||| */
/*||||||||||||| : Other than above : Setting prohibited */
/*||||||||||||| */
/*|||||||||||++------- : Be sure to set 00 */
/*||||||||||| */
/*|||||||||++--------- : Selection of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 valid 
edge */
/*|||||||||                (the timer input used with channel 1 is selected by using TIS0 
register). */
/*||||||||| : 0 0 : Falling edge */
/*||||||||| : 0 1 : Rising edge */
/*||||||||| : 1 0 : Both edges (when low-level width is measured) */
/*|||||||||         Start trigger: Falling edge, Capture trigger: Rising edge */
/*||||||||| : 1 1 : Both edges (when high-level width is measured) */
/*|||||||||         Start trigger: Rising edge, Capture trigger: Falling edge */
/*||||||||| */
/*||||||+++----------- : Setting of start trigger or capture trigger of channel 1 */
/*|||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are unselected). 
*/
/*|||||| : 0 0 1 : Valid edge of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 is used
```

```
as both the start trigger and capture trigger. */
                                /*||||| : 0 1 0 : Both the edges of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 are
used as a start trigger and a capture trigger. */
                                /*||||| : 1 0 0 : Interrupt signal of the master channel is used (when the channel is used
as a slave channel with the combination operation function). */
                                /*||||| : Other than above : Setting prohibited */
                                /*||||| */
                                /*||||+-------------- : Selection of slave/master of channel 1 */
                                /*|||| : 0 : Operates as slave channel with combination operation function. */
                                /*|||| : 1 : Operates as master channel with combination operation function. */
                                /*|||| */
                                /*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
                                /*||| : 0 : Operation clock MCK specified by CKS01 bit */
                                /*||| : 1 : Valid edge of input signal input from TI01 pin, fSUB/2, fSUB/4, or INTRTC1 */
                                /*|||        (the timer input used with channel 1 is selected by using TIS0 register). */
                                /*||| */
                                /*|++---------------- : Be sure to set 00 */
                                /*| */
                                /*+------------------ : Selection of operation clock (MCK) of channel 1 */
                                /* : 0 : Operation clock CK00 set by TPS0 register */
                                /* : 1 : Operation clock CK01 set by TPS0 register */
        TDR01 = CCK00_500USEC;          /* set interval time to 500us */
        TMMK01 = 1;                                     /* disable interrupt */



        /* CH4:for DMA */
        TMR04 = 0b1000000000000000;     /* Timer Mode Register 04 */
                                /*||||||||||||++++--- : [1]Operation mode of channel 4 */
                                /*||||||||||||         [2]Count operation of TCR */
                                /*||||||||||||         [3]Independent operation */
                                /*||||||||||||         [4]Setting of starting counting and interrupt */
                                /*|||||||||||| : 0 0 0 0 : [1]Interval timer mode */
                                /*||||||||||||             [2]Counting down */
                                /*||||||||||||             [3]Possible */
                                /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
                                /*||||||||||||                (timer output does not change, either). */
                                /*|||||||||||| */
                                /*|||||||||||| : 0 0 0 1 : [1]Interval timer mode */
                                /*||||||||||||             [2]Counting down */
```

```
/*||||||||||||                      [3]Possible */
/*||||||||||||                      [4]Timer interrupt is generated when counting is started */
/*||||||||||||                         (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 0 0 : [1]Capture mode */
/*||||||||||||                      [2]Counting up */
/*||||||||||||                      [3]Possible */
/*||||||||||||                      [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                         (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 0 1 0 1 : [1]Capture mode */
/*||||||||||||                      [2]Counting up */
/*||||||||||||                      [3]Possible */
/*||||||||||||                      [4]Timer interrupt is generated when counting is started */
/*||||||||||||                         (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 1 0 : [1]Event counter mode */
/*||||||||||||                      [2]Counting down */
/*||||||||||||                      [3]Possible */
/*||||||||||||                      [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                         (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 0 : [1]One-count mode */
/*||||||||||||                      [2]Counting down */
/*||||||||||||                      [3]Impossible */
/*||||||||||||                      [4]Start trigger is invalid during counting operation. */
/*||||||||||||                         At that time, interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 1 : [1]One-count mode */
/*||||||||||||                      [2]Counting down */
/*||||||||||||                      [3]Impossible */
/*||||||||||||                      [4]Start trigger is valid during counting operation. */
/*||||||||||||                         At that time, interrupt is also generated. */
/*|||||||||||| */
/*|||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*||||||||||||                      [2]Counting up */
/*||||||||||||                      [3]Possible */
/*||||||||||||                      [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                         (timer output does not change, either). */
```

```
/*||||||||||||                    Start trigger is invalid during counting operation. */
/*||||||||||||                    At that time interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : Other than above : Setting prohibited */
/*|||||||||||| */
/*||||||||||++------- : Be sure to set 00 */
/*|||||||||| */
/*||||||||++--------- : Selection of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 valid edge */
/*||||||||                    (the timer input used with channel 4 is selected by using TIS0 register). */
/*|||||||| : 0 0 : Falling edge */
/*|||||||| : 0 1 : Rising edge */
/*|||||||| : 1 0 : Both edges (when low-level width is measured) */
/*||||||||            Start trigger: Falling edge, Capture trigger: Rising edge */
/*|||||||| : 1 1 : Both edges (when high-level width is measured) */
/*||||||||            Start trigger: Rising edge, Capture trigger: Falling edge */
/*|||||||| */
/*|||||+++----------- : Setting of start trigger or capture trigger of channel 4 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are unselected). */
/*||||| : 0 0 1 : Valid edge of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 is used as both the start trigger and capture trigger. */
/*||||| : 0 1 0 : Both the edges of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 are used as a start trigger and a capture trigger. */
/*||||| : 1 0 0 : Interrupt signal of the master channel is used (when the channel is used as a slave channel with the combination operation function). */
/*||||| : Other than above : Setting prohibited */
/*||||| */
/*||||+-------------- : Selection of slave/master of channel 4 */
/*|||| : 0 : Operates as slave channel with combination operation function. */
/*|||| : 1 : Operates as master channel with combination operation function. */
/*|||| */
/*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
/*||| : 0 : Operation clock MCK specified by CKS04 bit */
/*||| : 1 : Valid edge of input signal input from TI04 pin, fSUB/2, fSUB/4, or INTRTC1 */
/*|||          (the timer input used with channel 4 is selected by using TIS0 register). */
/*||| */
/*|++---------------- : Be sure to set 00 */
```

```
                        /*| */
                        /*+------------------ : Selection of operation clock (MCK) of channel 4 */
                        /* : 0 : Operation clock CK00 set by TPS0 register */
                        /* : 1 : Operation clock CK01 set by TPS0 register */
    TDR04 = 200;        /* set interval time to 10us(=0.05us * 200) */
    TMMK04 = 1;                 /* disable interrupt */

}

/*--------------------------------------------------------------*/
/* Module:         fn_InitLvi
                   */
/* Description:    Setting of Low-voltage detectior                    */
/*     parameter: --
                        */
/*     return     : --
                        */
/*--------------------------------------------------------------*/
void    fn_InitLvi(void)
{
    unsigned short      loop;   /* waiting counter */

    LVIMK = 1;                          /* disable LVI interrupt */

    LVIS =     0b00001001;              /* Low-Voltage Detection Level Select Register */
                        /*||||++++--- : Detection level */
                        /*|||| : 0 0 0 0 : VLVI0 (4.22V) */
                        /*|||| : 0 0 0 1 : VLVI1 (4.07V) */
                        /*|||| : 0 0 1 0 : VLVI2 (3.92V) */
                        /*|||| : 0 0 1 1 : VLVI3 (3.76V) */
                        /*|||| : 0 1 0 0 : VLVI4 (3.61V) */
                        /*|||| : 0 1 0 1 : VLVI5 (3.45V) */
                        /*|||| : 0 1 1 0 : VLVI6 (3.30V) */
                        /*|||| : 0 1 1 1 : VLVI7 (3.15V) */
                        /*|||| : 1 0 0 0 : VLVI8 (2.99V) */
                        /*|||| : 1 0 0 1 : VLVI9 (2.84V) */
                        /*|||| : 1 0 1 0 : VLVI10 (2.68V) */
                        /*|||| : 1 0 1 1 : VLVI11 (2.53V) */
                        /*|||| : 1 1 0 0 : VLVI12 (2.38V) */
```

```
                    /*||||  : 1 1 0 1 : VLVI13 (2.22V) */
                    /*||||  : 1 1 1 0 : VLVI14 (2.07V) */
                    /*||||  : 1 1 1 1 : VLVI15 (1.91V) */
                    /*||||  */
                    /*++++------- : Be sure to set 0000 */

    LVIM =    0b10000000;            /* Low-Voltage Detection Register */
                    /*|||||||+--- : LVIF Low-voltage detection flag */
                    /*||||||| : 0 : * LVISEL = 0: VDD >= VLVI, or when LVI operation is disabled */
                    /*|||||||         * LVISEL = 1: EXLVI >= VEXLVI, or when LVI operation is disabled */
                    /*||||||| : 1 : * LVISEL = 0: VDD < VLVI */
                    /*|||||||         * LVISEL = 1: EXLVI < VEXLVI */
                    /*||||||| */
                    /*||||||+---- : Low-voltage detection operation mode (interrupt/reset) selection(LVIMD) */
                    /*|||||| : 0 : * LVISEL = 0: Generates an internal interrupt signal */
                    /*||||||                        when VDD drops lower than VLVI (VDD < VLVI) */
                    /*||||||                        or when VDD becomes VLVI or higher (VDD >= VLVI).
*/
                    /*||||||          * LVISEL = 1: Generates an interrupt signal */
                    /*||||||                        when EXLVI drops lower than VEXLVI (EXLVI <
VEXLVI) */
                    /*||||||                        or when EXLVI becomes VEXLVI or higher (EXLVI >=
VEXLVI). */
                    /*|||||| : 1 : * LVISEL = 0: Generates an internal reset signal when VDD < VLVI */
                    /*||||||                        and releases the reset signal when VDD >= VLVI. */
                    /*||||||          * LVISEL = 1: Generates an internal reset signal when EXLVI < VEXLVI
*/
                    /*||||||                        and releases the reset signal when EXLVI >= VEXLVI.
*/
                    /*|||||| */
                    /*|||||+----- : Voltage detection selection(LVISEL) */
                    /*||||| : 0 : Detects level of supply voltage (VDD) */
                    /*||||| : 1 : Detects level of input voltage from external input pin (EXLVI) */
                    /*||||| */
                    /*|++++------ : Be sure to set 0000 */
                    /*| */
                    /*+---------- : Enables low-voltage detection operation */
                    /*  : 0 : Disables operation */
                    /*  : 1 : Enables operation */
```

```
        /* software to wait for the operation stabilization time (210us) */
        for(loop = 500; loop > 0; loop--){
                NOP();
        }
        /* wait until VLVI≤VDD */
        while( LVIF ){
                NOP();
        }

        LVIIF = 0;                      /* clear LVI interrupt request flag */
}



/*-------------------------------------------------------------*/
/* Module:          fn_DmaInit
                                */
/* Description:     Setting of DMA contoroler                                        */
/*      parameter: --
                                */
/*      return     : --
                                */
/*-------------------------------------------------------------*/
void    fn_DmaInit(void)
{

        DEN0 = 1;                                       /* Enables operation of DMA channel 0 */

        DSA0 = (unsigned char)&DACSW0;                          /* DMA SFR Address Register 0
*/
        DRA0 = (unsigned short)&ushDacData[0];          /* DMA RAM Address Register 0 */
        DBC0 = 100;                                     /* DMA Byte Count Register 0 */
        DMC0 = 0b01100100;                              /* DMA Mode Control Register 0 */
                    /*|||||||| */
                    /*||||++++------- IFC3-IFC0 */
                    /*||||              [Selection of DMA start source] */
                    /*||||               0000:Disables DMA transfer by interrupt */
                    /*||||                              (Only software trigger is enabled.) */
```

```
                /*||||                  0010:INTTM00 */
                /*||||                  0011:INTTM01 */
                /*||||                  0100:INTTM04 */
                /*||||                  0101:INTTM05 */
                /*||||                  0110:INTST0/INTCSI00 */
                /*||||                  0111:INTSR0/INTCSI01 */
                /*||||                  1000:INTST1/INTCSI10/INTIIC10 */
                /*||||                  1001:INTSR1 */
                /*||||                  1010:INTST3 */
                /*||||                  1011:INTSR3 */
                /*||||                  1100:INTAD */
                /*|||+----------- DWAIT0 */
                /*|||                  [Pending of DMA transfer] */
                /*|||                   0:Executes DMA transfer upon DMA start request (not held pending). */
                /*|||                   1:Holds DMA start request pending if any. */
                /*||+------------ DS0 */
                /*||                   [Specification of transfer data size for DMA transfer] */
                /*||                    0:8bits */
                /*||                    1:16bits */
                /*|+------------- DRS0 */
                /*|                    [Selection of DMA transfer direction] */
                /*|                     0:SFR to internal RAM */
                /*|                     1:Internal RAM to SFR */
                /*+------------- STG0 */
                /*                     [DMA transfer start software trigger] */
                /*                      0:No trigger operation */
                /*                      1:DMA transfer is started when DMA operation is enabled (DEN0 = 1). */

        DST0 = 1;                       /* DMA transfer is started when DMA operation is enabled (DEN0 = 1). */

}



/*---------------------------------------------------------------*/
/* Module:         fn_Wait500usBase
        */
/* Description:    Delays the program for (Time * 500us)                      */
/*      parameter: wait time(Time)
        */
```

```
/*      return    : --
                              */
/*---------------------------------------------------------------*/
void    fn_Wait500usBase(unsigned short Time)
{
        TS0L.1 = 1;                         /* start TAU0 CH1 */
        TMIF01 = 0;

        for(; Time > 0; Time--){        /* wait for (parameter * 500)us */
                while(!TMIF01){
                        NOP();
                }
                TMIF01 = 0;
        }

        TT0L.1 = 1;                         /* stop TAU0 CH1 */
}



/*---------------------------------------------------------------*/
/* Module:         fn_AdcDacInit
                */
/* Description:    Initialization of A/D D/A OPAmp module                    */
/*      parameter: --
                              */
/*      return    : --
                              */
/*---------------------------------------------------------------*/
void    fn_AdcDacInit(void)
{
        ADCEN = 1;                          /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

        /*-------------------------------------------*/
        /*   Initialization of Voltage reference              */
        /*-------------------------------------------*/
        ADVRC = 0b00001000;         /* Analog reference voltage control register */
                              /*||||+|++--- : [1]Positive reference voltage supplies selection of A/D and D/A
converters */
```

```
/*IIII I          [2]Operationcontrol of voltage reference */
/*IIII I          [3]Output voltage selection of voltage reference */
/*IIII I          [4]Operation control of input gate voltage boost circuit for A/D converter
*/
/*IIII I          [5]Relationship with the conversion mode used */
/*IIII I */
/*IIII I : 0 0 0 : [1]AVREFP (external voltage reference input) */
/*IIII I             [2]Stops operation (Hi-Z) */
/*IIII I             [3]2.5 V */
/*IIII I             [4]Stops operation */
/*IIII I             [5]Can be set in conversion mode 1 */
/*IIII I */
/*IIII I : 0 1 0 : [1]AVREFP (external voltage reference input) */
/*IIII I             [2]Stops operation (Hi-Z) */
/*IIII I             [3]2.0 V */
/*IIII I             [4]Enables operation */
/*IIII I             [5]Can be set in conversion mode 2 or 3 */
/*IIII I */
/*IIII I : 1 0 0 : [1]VREFOUT (voltage reference output) */
/*IIII I             [2]Stops operation (pull-down output) */
/*IIII I             [3]2.5 V */
/*IIII I             [4]Stops operation */
/*IIII I             [5] - */
/*IIII I */
/*IIII I : 1 0 1 : [1]VREFOUT (voltage reference output) */
/*IIII I             [2]Enables operation */
/*IIII I             [3]2.5 V */
/*IIII I             [4]Enables operation */
/*IIII I             [5]Can be set in conversion mode 2 or 3 */
/*IIII I */
/*IIII I : 1 1 0 : [1]VREFOUT (voltage reference output) */
/*IIII I             [2]Stops operation (pull-down output) */
/*IIII I             [3]2.0 V */
/*IIII I             [4]Enables operation */
/*IIII I             [5] - */
/*IIII I */
/*IIII I : 1 1 1 : [1]VREFOUT (voltage reference output) */
/*IIII I             [2]Enables operation */
/*IIII I             [3]2.0 V */
```

```
                    /*|||| |              [4]Enables operation */
                    /*|||| |              [5]Can be set in conversion mode 2 or 3 */
                    /*|||| | */
                    /*|||| | : Other than the above : Setting prohibited */
                    /*|||| | */
                    /*|+++-+----- : Be sure to set 0000 */
                    /*| */
                    /*+---------- : Reference voltage supply (negative side) of A/D converter selection */
                    /* : 0 : AVSS */
                    /* : 1 : AVREFM (external voltage reference input) */

ADVRC.0 = 1;            /* Enables operation */
ADVRC.1 = 1;            /* Output 2.0 V */

ADPC =  0b00000000;     /* A/D Port Configuration Register */
                    /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                    /*||| :              +------------------------ ANI15/AVREFM/P157 */
                    /*||| :              | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                    /*||| :              | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
                    /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                    /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                    /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                    /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                    /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                    /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                    /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                    /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                    /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                    /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                    /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                    /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                    /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                    /*||| */
                    /*+++-------- : Be sure to set 000 */

/* wait for settling time to 17msec */
fn_Wait500usBase(17000/500);

/*------------------------------------------*/
```

```
/*    Initialization of operational amplifier    */
/*-------------------------------------------*/
PM2.1 = 1;                          /* input port mode (AMP0O) */
PM2.0 = 1;                          /* input port mode (AMP0-) */
PM2.2 = 1;                          /* input port mode (AMP0+) */

PM2.4 = 1;                          /* input port mode (AMP1O) */
PM2.3 = 1;                          /* input port mode (AMP1-) */
PM2.5 = 1;                          /* input port mode (AMP1+) */

PM2.7 = 1;                          /* input port mode (AMP2O) */
PM2.6 = 1;                          /* input port mode (AMP2-) */
PM15.0 = 1;                         /* input port mode (AMP2+) */

PM11.0 = 1;                         /* input port mode (ANO0) */

ADCEN = 1;                          /* supplies input clock to OPAmp A/D converter */
OAC.0 = 1;                          /* operational amplifier (AMP0) enable */
OAC.1 = 0;                          /* operational amplifier (AMP1) disable */
OAC.2 = 0;                          /* operational amplifier (AMP2) disable */

/*-------------------------------------------*/
/*          Initialization of D/A converter          */
/*-------------------------------------------*/
DACEN = 1;                                  /* supplies input clock to D/A converter */

DAM.0 = 1;                                  /* [DAMD0] D/A converter operation is Real-time
output mode */
DAM.2 = 1;                                  /* [DARES0]D/A conversion resolution is 12-bit */
DAM.6 = 1;                                  /* [DAREF] D/A converter positive reference voltage
is VREFOUT */
DACSW0 = aushDacData[99];           /* D/A conversion value setting reigster 0 */

/* set D/A chanel1 */
DAM.1 = 0;                                  /* [DAMD1] D/A converter operation is Normal mode
*/
DAM.3 = 1;                                  /* [DARES1]D/A conversion resolution is 12-bit */
DACSW1 = aushDacData[80];           /* D/A conversion value setting reigster 1 */
```

```
}
/*-------------------------------------------------------------*/
/* Module:          main
                              */
/* Description:     Main process
          */
/*          parameter: --
                              */
/*          return      : --
                              */
/*-------------------------------------------------------------*/
void main(void)
{
          unsigned short        count;               /* counter for initializing D/A convert data */

          ushCycleCount = 0;                         /* Cycle counter */

          DACE1 = 1;                                                      /* D/A converter CH1 enable */
          DACE0 = 1;                                                      /* D/A converter CH0 enable */


          /*=====================================*/
          /*----------------------------------*/
          /*                  Main Loop                  */
          /*----------------------------------*/
          /*=====================================*/

          /*    initialize D/Aconverter data */
          for(count = 0; count<100; count++){
                    ushDacData[count] = aushDacData[count];
          }

          TS0L.4 = 1;                                /* start TAU0 CH4 */
          TMIF04 = 0;                                /* reset INTTM04 interrupt request */

          DMAMK0 = 0;                                /* enable DMA CH0 interrupt */

          EI();                                      /* enable all interrupts */
```

```
	/*==============================================================*/
	/* if system have nothing to do, go to standby for power-saving */
	/*==============================================================*/


	while (1){
		HALT();
		NOP();
	}
}



/*------------------------------------------------------------*/
/* Module:		fn_intdma0
				*/
/* Description:	DMA ch0 interruput process					*/
/*	parameter: --
				*/
/*	return	: --
				*/
/*------------------------------------------------------------*/
__interrupt void fn_intdma0(void)
{
	TT0L.4 = 1;				/* stop TAU0 CH4 */


	/* start A/D conversion */
/*	DI();			/* disable all interrupts */
	ADCS = 1;		/* A/D converter enable */
	ADIF = 0;	/* clear A/D interrupt request flag */
	ADMK = 0;		/* enable A/D interrupt */
	HALT();			/* set HALT mode and wait conversion finished */
	ADMK = 1;		/* disable A/D interrupt */
	ADIF = 0;	/* clear A/D interrupt request flag */
/*	EI();			/* enable all interrupts */


	shVoltage = (short)(ADCR & 0x0fff);		/* get A/D result */
	shVoltage -= (2048 - 1);
```

```
        if(shVoltage < 0){
                DACSW1 += 1;                /* D/A conversion value setting reigster 1 */
        }
        else if(shVoltage > 2){
                DACSW1 -= 1;                /* D/A conversion value setting reigster 1 */
        }

        ushCycleCount++;                    /* Cycle counter */
        if(ushCycleCount > 100){                    /* Cycle counter */
                DACSW1 = ushDacData[95];    /* D/A conversion value setting reigster 1 */
                ushCycleCount = 0;                  /* reset Cycle counter */
        }

        DRA0 = (unsigned short)&ushDacData[0];          /* DMA RAM Address Register 0 */
        DBC0 = 100;                                             /* DMA Byte Count Register 0 */

        DST0 = 1;                           /* DMA transfer is started when DMA operation is enabled
(DEN0 = 1). */

        TS0L.4 = 1;                         /* start TAU0 CH4 */
        TMIF04 = 0;                         /* reset INTTM04 interrupt request */

    }
```

(4) Offset adjustment of inverting amplifier circuit + low-pass filter

● main.c

```
/*
 * Copyright (C) NEC Electronics Corporation 2006
 * NEC ELECTRONICS CONFIDENTIAL AND PROPRIETARY
 * All rights reserved by NEC Electronics Corporation.
 * This program must be used solely for the purpose for which
 * it was furnished by NEC Electronics Corporation. No part of this
 * program may be reproduced or disclosed to others, in any
 * form, without the prior written permission of NEC Electronics
 * Corporation. Use of copyright notice dose not evidence
 * publication of the program.
 */

/*--------------------------------------------------------------------*/
/* #pragma directive for CC78K0
                                    */
/*--------------------------------------------------------------------*/
#pragma         SFR
#pragma         DI
#pragma         EI
#pragma         HALT
#pragma         NOP

#pragma interrupt INTDMA0 fn_intdma0 RB1

/*--------------------------------------------------------------------*/
/* Include files
                                                */
/*--------------------------------------------------------------------*/
/* TAU:TDR0n value operation by CK00 (fCLK/2^3 = 5MHz) */
#define CCK00_500USEC   (2500 - 1)          /* 500us (0.2[us/clk] * 2500[count]) */



/*--------------------------------------------------------------------*/
/* Function prototyps
                                                */
/*--------------------------------------------------------------------*/
```

```
                void    fn_Wait500usBase(unsigned short);    /* Delays the program for (Time * 500us) */
                void    fn_InitPort(void);                                /* Setting of I/O ports
*/
                void    fn_InitTau0(void);                                /* Setting of Timer
array unit 0 */
                void    fn_InitLvi(void);                                 /*    Setting    of
Low-voltage detectior */
                void    fn_InitVr(void);                                  /* Setting of Voltage
reference */
                void    fn_AdcDacInit(void);                         /* Setting of D/A converter */
                void    fn_DmaInit(void);                                 /* Setting of DMA
controler */


  /*--------------------------------------------------------------------*/
  /* Extern variables/constants
                */
  /*--------------------------------------------------------------------*/

  /*--------------------------------------------------------------------*/
  /* Local constants
                                */
  /*--------------------------------------------------------------------*/
  /*--------------------------------------------------------------------*/
  /* Global variables
                                */
  /*--------------------------------------------------------------------*/

  /*--------------------------------------------------------------------*/
  /* Local variables
                                */
  /*--------------------------------------------------------------------*/
  static  short         shVoltage;/* voltage (Hex) */
  static  unsigned short    ushCycleCount;    /* Cycle counter */
  static  unsigned char     ucCycleStatus;    /* Cycle status */

  /*--------------------------------------------------------------------*/
  /* Code
                                              */
```

```
/*----------------------------------------------------------------------*/
/*==============================================================================

        D/A output data

==============================================================================*/
static unsigned short ushDacData[100];          /* D/A data for DMA */
const unsigned short aushDacData[100] = {       /* defined D/A data */
                                                /* n        sin(0.2pi*n/100) */
/*              2048,               /* 0        0.100000 */
                2061,               /* 1        0.106279 */
                2074,               /* 2        0.112533 */
                2086,               /* 3        0.118738 */
                2099,               /* 4        0.124869 */
                2111,               /* 5        0.130902 */
                2123,               /* 6        0.136812 */
                2135,               /* 7        0.142578 */
                2147,               /* 8        0.148175 */
                2158,               /* 9        0.153583 */
                2168,               /* 10       0.158779 */
                2179,               /* 11       0.163742 */
                2188,               /* 12       0.168455 */
                2197,               /* 13       0.172897 */
                2206,               /* 14       0.177051 */
                2214,               /* 15       0.180902 */
                2221,               /* 16       0.184433 */
                2227,               /* 17       0.187631 */
                2233,               /* 18       0.190483 */
                2238,               /* 19       0.192978 */
                2243,               /* 20       0.195106 */
                2246,               /* 21       0.196858 */
                2249,               /* 22       0.198229 */
                2251,               /* 23       0.199211 */
                2252,               /* 24       0.199803 */
                2253,               /* 25       0.200000 */
                2252,               /* 26       0.199803 */
                2251,               /* 27       0.199211 */
                2249,               /* 28       0.198229 */
                2246,               /* 29       0.196858 */
```

```
2243,       /* 30   0.195106 */
2238,       /* 31   0.192978 */
2233,       /* 32   0.190483 */
2227,       /* 33   0.187631 */
2221,       /* 34   0.184433 */
2214,       /* 35   0.180902 */
2206,       /* 36   0.177051 */
2197,       /* 37   0.172897 */
2188,       /* 38   0.168455 */
2179,       /* 39   0.163742 */
2168,       /* 40   0.158779 */
2158,       /* 41   0.153583 */
2147,       /* 42   0.148175 */
2135,       /* 43   0.142578 */
2123,       /* 44   0.136812 */
2111,       /* 45   0.130902 */
2099,       /* 46   0.124869 */
2086,       /* 47   0.118738 */
2074,       /* 48   0.112533 */
2061,       /* 49   0.106279 */
2048,       /* 50   0.100000 */
2035,       /* 51   0.093721 */
2022,       /* 52   0.087467 */
2010,       /* 53   0.081262 */
1997,       /* 54   0.075131 */
1985,       /* 55   0.069098 */
1973,       /* 56   0.063188 */
1961,       /* 57   0.057422 */
1949,       /* 58   0.051825 */
1938,       /* 59   0.046417 */
1928,       /* 60   0.041221 */
1917,       /* 61   0.036258 */
1908,       /* 62   0.031545 */
1899,       /* 63   0.027103 */
1890,       /* 64   0.022949 */
1882,       /* 65   0.019098 */
1875,       /* 66   0.015567 */
1869,       /* 67   0.012369 */
1863,       /* 68   0.009517 */
```

```
        1858,           /* 69     0.007022 */
        1853,           /* 70     0.004894 */
        1850,           /* 71     0.003142 */
        1847,           /* 72     0.001771 */
        1845,           /* 73     0.000789 */
        1844,           /* 74     0.000197 */
        1843,           /* 75     0.000000 */
        1844,           /* 76     0.000197 */
        1845,           /* 77     0.000789 */
        1847,           /* 78     0.001771 */
        1850,           /* 79     0.003142 */
        1853,           /* 80     0.004894 */
        1858,           /* 81     0.007022 */
        1863,           /* 82     0.009517 */
        1869,           /* 83     0.012369 */
        1875,           /* 84     0.015567 */
        1882,           /* 85     0.019098 */
        1890,           /* 86     0.022949 */
        1899,           /* 87     0.027103 */
        1908,           /* 88     0.031545 */
        1917,           /* 89     0.036258 */
        1928,           /* 90     0.041221 */
        1938,           /* 91     0.046417 */
        1949,           /* 92     0.051825 */
        1961,           /* 93     0.057422 */
        1973,           /* 94     0.063188 */
        1985,           /* 95     0.069098 */
        1997,           /* 96     0.075131 */
        2010,           /* 97     0.081262 */
        2022,           /* 98     0.087467 */
        2035,           /* 99     0.093721 */
        2048            /* 100    0.100000 */

};

/*----------------------------------------------------------------------*/
/* Hardware initialization
                                        */
/*----------------------------------------------------------------------*/
```

```
void hdwinit(void)
{
        DI();                           /* disable all interrupts */

        /*----------------------------------------*/
        /*          Initialization of port        */
        /*----------------------------------------*/
        fn_InitPort();

        /*---------------------------------------*/
        /*          Initialization of clock      */
        /*---------------------------------------*/
        CMC =   0b01010011;             /* Clock Operation Mode Control Register */
                        /*|||||||+--- : Control of high-speed system clock oscillation frequency */
                        /*||||||| : 0 : 2 MHz <= fMX <= 10 MHz */
                        /*||||||| : 1 : 10 MHz < fMX <= 20 MHz */
                        /*||||||| */
                        /*|||||++---- : XT1 oscillator oscillation mode selection */
                        /*||||| : 0 0 : Low-consumption oscillation */
                        /*||||| : 0 1 : Normal oscillation */
                        /*||||| : 1 x : Super-low-consumption oscillation */
                        /*|||||    x = don't care */
                        /*||||| */
                        /*||||+------ : Be sure to set 0 */
                        /*|||| */
                        /*|||+------- : [1] Subsystem clock pin operation mode */
                        /*|||             [2] XT1/P123 pin and XT2/P124 pin */
                        /*||| : 0 : [1]Input port mode */
                        /*|||         [2]Input port */
                        /*||| */
                        /*||| : 1 : [1]XT1 oscillation mode */
                        /*|||         [2]Crystal resonator connection */
                        /*||| */
                        /*||+-------- : Be sure to set 0 */
                        /*|| */
                        /*++--------- : [1]EXCLK OSCSEL High-speed system clock pin operation mode */
                        /*                [2]X1/P121 pin */
                        /*                [3]X2/EXCLK/P122 pin */
                        /* : 0 0 : [1]Input port mode */
```

```
                        /*              [2][3]Input port */
                        /* */
                        /* : 0 1 : [1]X1 oscillation mode */
                        /*              [2][3]Crystal/ceramic resonator connection */
                        /* */
                        /* : 1 0 : [1]Input port mode */
                        /*              [2][3]Input port */
                        /* */
                        /* : 1 1 : [1]External clock input mode */
                        /*              [2]Input port */
                        /*              [3]External clock input */

        MSTOP = 0;                                  /* X1 oscillator operating */
        XTSTOP = 0;                                 /* XT1 oscillator operating */

        OSMC =  0b00000001;                         /* Operation Speed Mode Control Register */
                        /*||||||++--- : fCLK frequency selection */
                        /*|||||| : 0 0 : Operates at a frequency of 10 MHz or less. */
                        /*|||||| : 0 1 : Operates at a frequency higher than 10 MHz. */
                        /*|||||| : 1 0 : Operates at a frequency of 1 MHz. */
                        /*|||||| */
                        /*|+++++----- : Be sure to set 00000 */
                        /*| */
                        /*+---------- : Setting in subsystem clock HALT mode */
                        /* : 0 : Enables subsystem clock supply to peripheral functions. */
                        /*          (See Table 21-1 Operating Statuses in HALT Mode (2/3) */
                        /*           for the peripheral functions whose operations are enabled.) */
                        /* : 1 : Stops subsystem clock supply to peripheral functions except real-time counter,
*/
                        /*           clock output/buzzer output, and LCD controller/driver. */

        while(OSTC.0 != 1){                 /* wait X1 oscillation stabilization */
                NOP();
        }

        /*-- Caution ----------------------------------------------*/
        /* To increase fCLK to 10 MHz or higher, set FSEL to '1',    */
        /* then change fCLK after two or more clocks have elapsed. */
        /*--------------------------------------------------------*/
```

```
NOP();
NOP();

CKC =    0b00010000;                 /* System Clock Control Register */
                    /*|+|+++++--- : Selection of CPU/peripheral hardware clock (fCLK) */
                    /*| | : 0 0 x 0 0 0 : fIH */
                    /*| | : 0 0 x 0 0 1 : fIH/2 (default) */
                    /*| | : 0 0 x 0 1 0 : fIH/2^2 */
                    /*| | : 0 0 x 0 1 1 : fIH/2^3 */
                    /*| | : 0 0 x 1 0 0 : fIH/2^4 */
                    /*| | : 0 0 x 1 0 1 : fIH/2^5 */
                    /*| | : 0 1 x 0 0 0 : fMX */
                    /*| | : 0 1 x 0 0 1 : fMX/2 */
                    /*| | : 0 1 x 0 1 0 : fMX/2^2 */
                    /*| | : 0 1 x 0 1 1 : fMX/2^3 */
                    /*| | : 0 1 x 1 0 0 : fMX/2^4 */
                    /*| | : 0 1 x 1 0 1 : fMX/2^5 */
                    /*| | : 1 x 0 x x x : fSUB */
                    /*| | : 1 x 1 x x x : fSUB/2 */
                    /*| | : Other than above : Setting prohibited */
                    /*| |   x = don't care */
                    /*| | */
                    /*| +-------- : Status of Main system clock (fMAIN) */
                    /*| : 0 : Internal high-speed oscillation clock (fIH) */
                    /*| : 1 : High-speed system clock (fMX) */
                    /*| */
                    /*+---------- : Status of CPU/peripheral hardware clock (fCLK) */
                    /* : 0 : Main system clock (fMAIN) */
                    /* : 1 : Subsystem clock (fSUB) */

/* Confirming the CPU clock status */
while((CLS != 0)||(MCS != 1)){
        NOP();
}
/* CPU is operating on a High-speed system clock */
HIOSTOP = 1;                         /* internal high-speed oscillation stopped */

OSTS =   0b00000111;                 /* Oscillation Stabilization Time Select Register */
                    /*||||++++--- : Oscillation stabilization time selection */
```

```
                        /*llll : 0 0 0 : 2^8/fX */
                        /*llll : 0 0 1 : 2^9/fX */
                        /*llll : 0 1 0 : 2^10/fX */
                        /*llll : 0 1 1 : 2^11/fX */
                        /*llll : 1 0 0 : 2^13/fX */
                        /*llll : 1 0 1 : 2^15/fX */
                        /*llll : 1 1 0 : 2^17/fX */
                        /*llll : 1 1 1 : 2^18/fX */
                        /*llll */
                        /*++++------- : Be sure to set 000000 */

    /*----------------------------------------*/
    /*          Initialization of timer         */
    /*----------------------------------------*/
    fn_InitTau0();

    /* software to wait for the operation stabilization time */
    /* (over 200ms from when XT1 enable) */
    fn_Wait500usBase(200000/500);

    /*----------------------------------------*/
    /* Initialization of low-voltage detector */
    /*----------------------------------------*/
    fn_InitLvi();

    /*----------------------------------------*/
    /*  Initialization of A/D,D/A,OPAmp                 */
    /*----------------------------------------*/
    fn_AdcDacInit();

    /*----------------------------------------*/
    /*  Initialization of DMA                                  */
    /*----------------------------------------*/
    fn_DmaInit();

    EI();                               /* enable all interrupts */

}
```

```
/*---------------------------------------------------------------*/
/* Module:          fn_InitPort
                    */
/* Description:     Setting of I/O ports                                    */
/*      parameter: --
                              */
/*      return    : --
                              */
/*---------------------------------------------------------------*/
void    fn_InitPort(void)
{
/*-----------------------------------------------------------*/
/*      Ports configuration for digital input and output            */
/*-----------------------------------------------------------*/
        ADPC =   0b00010000;          /* A/D Port Configuration Register */
                              /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                              /*||| :              +------------------------ ANI15/AVREFM/P157 */
                              /*||| :              | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                              /*||| :              | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
                              /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                              /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                              /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                              /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                              /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                              /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                              /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                              /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                              /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                              /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                              /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                              /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                              /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                              /*||| */
                              /*+++-------- : Be sure to set 000 */

/*-----------------------------------------------------------*/
/*      Setting of Port 0
        */
/*-----------------------------------------------------------*/
```

```
        P0 =        0b00000000;         /* Set P00-P02 Output latch to Low */
        PM0 =       0b11111000;         /* Set P00-P02 to output port */
                                                        /* P00-P02:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 1
        */
/*-----------------------------------------------------------*/
        P1 =        0b00000000;         /* Set P10-P17 Output latch to Low */
        PM1 =       0b00000000;         /* Set P10-P17 to output port */
                                                        /* P10-P15:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 2
        */
/*-----------------------------------------------------------*/
        P2 =        0b00000000;         /* Set P20-P27 Output latch to Low */
        PM2 =       0b11111111;         /* Set P20-P27 to input port */
                                                        /* P20-P27:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 3
        */
/*-----------------------------------------------------------*/
        P3 =        0b00001100;         /* Set P30-P31,P34 Output latch to Low */
                                                        /* Set P33,P32 Output latch High */
        PM3 =       0b11100000;         /* Set P30-P34 to output port */
                                                        /* P30-P34:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 4
        */
/*-----------------------------------------------------------*/
        P4 =        0b00000000;         /* Set P40-P41 Output latch to Low */
        PM4 =       0b11111100;         /* Set P40-P41 to output port */
                                                        /* P40-P41:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 5
```

```
        */
/*-----------------------------------------------------------*/
        P5 =    0b00000000;     /* Set P50-P57 Output latch to Low */
        PM5 =   0b11110000;     /* Set P50-P57 to output port */
                                        /* P50-P57:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 6
        */
/*-----------------------------------------------------------*/
        P6 =    0b00000000;     /* Set P60-P61 Output latch to Low */
        PM6 =   0b11111100;     /* Set P60-P61 to output port */
                                        /* P60-P61:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 7
        */
/*-----------------------------------------------------------*/
        P7 =    0b00000000;     /* Set P70-P77 Output latch to Low */
        PM7 =   0b00000000;     /* Set P70-P77 to output port */
                                        /* P70-P77:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 8
        */
/*-----------------------------------------------------------*/
        P8 =    0b00000000;     /* Set P80-P88 Output latch to Low */
        PM8 =   0b00000000;     /* Set P80-P88 to output port */
                                        /* P80-P88:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 9
        */
/*-----------------------------------------------------------*/
        P9 =    0b00000000;     /* Set P90-P97 Output latch to Low */
        PM9 =   0b00000000;     /* Set P90-P97 to output port */
                                        /* P90-P97:Unused */

/*-----------------------------------------------------------*/
```

```
/*      Setting of Port 10
        */
/*----------------------------------------------------------*/
        P10 =   0b00000000;     /* Set P100-P102 Output latch to Low */
        PM10 =  0b11111000;     /* Set P100-P102 to output port */
                                        /* P100-P102:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 11
        */
/*----------------------------------------------------------*/
        P11 =   0b00000000;     /* Set P110-P111 Output latch to Low */
        PM11 =  0b11111100;     /* Set P110-P111 to output port */
                                        /* P110-P111:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 12
        */
/*----------------------------------------------------------*/
        P12 =   0b00000000;     /* Set P120 Output latch to Low */
        PM12 =  0b11111110;     /* Set P120 to output port */
                                        /* P120-P124:Unused */
                                        /* *P121-P124:Input port */

/*----------------------------------------------------------*/
/*      Setting of Port 13
        */
/*----------------------------------------------------------*/
        P13 =   0b00000000;     /* Set P130 Output latch to Low */
                                        /* P130:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 14
        */
/*----------------------------------------------------------*/
        P14 =   0b00000000;     /* Set P140-P147 Output latch to Low */
        PM14 =  0b00000000;     /* Set P140-P147 to output port */
                                        /* P140-P147:Unused */
```

```
/*----------------------------------------------------------*/
/*      Setting of Port 15
        */
/*----------------------------------------------------------*/
        P15 =   0b00000000;         /* Set P150-P152,P157 Output latch to Low */
        PM15 =  0b11111111;         /* Set P150-P152,P157 to input port */
                                                    /* P150-P152,P157:Unused */

}

/*--------------------------------------------------------------*/
/* Module:          fn_InitTau0
                            */
/* Description:     Setting of Timer array unit 0                                    */
/*      parameter: --
                            */
/*      return     : --
                            */
/*--------------------------------------------------------------*/
void    fn_InitTau0(void)
{
        TAU0EN = 1;                         /* supplies input clock to timer array unit 0 */
        TPS0L = 0b00000010;         /* Timer Clock Select Register 0 */
                            /*||||++++--- : Selection of operation clock (CK00) */
                            /*++++------- : Selection of operation clock (CK01) */
                            /* : 0 0 0 0 : CK0m = fCLK */
                            /* : 0 0 0 1 : CK0m = fCLK/2 */
                            /* : 0 0 1 0 : CK0m = fCLK/2^2 */
                            /* : 0 0 1 1 : CK0m = fCLK/2^3 */
                            /* : 0 1 0 0 : CK0m = fCLK/2^4 */
                            /* : 0 1 0 1 : CK0m = fCLK/2^5 */
                            /* : 0 1 1 0 : CK0m = fCLK/2^6 */
                            /* : 0 1 1 1 : CK0m = fCLK/2^7 */
                            /* : 1 0 0 0 : CK0m = fCLK/2^8 */
                            /* : 1 0 0 1 : CK0m = fCLK/2^9 */
                            /* : 1 0 1 0 : CK0m = fCLK/2^10 */
                            /* : 1 0 1 1 : CK0m = fCLK/2^11 */
                            /* : 1 1 0 0 : CK0m = fCLK/2^12 */
                            /* : 1 1 0 1 : CK0m = fCLK/2^13 */
```

```
                    /* : 1 1 1 0 : CK0m = fCLK/2^14 */
                    /* : 1 1 1 1 : CK0m = fCLK/2^15 */
                    /*    m = 0, 1 */

/* CH1:for wait */
TMR01 = 0b0000000000000000;         /* Timer Mode Register 01 */
                    /*||||||||||||++++--- : [1]Operation mode of channel 1 */
                    /*||||||||||||          [2]Count operation of TCR */
                    /*||||||||||||          [3]Independent operation */
                    /*||||||||||||          [4]Setting of starting counting and interrupt */
                    /*|||||||||||| : 0 0 0 0 : [1]Interval timer mode */
                    /*||||||||||||              [2]Counting down */
                    /*||||||||||||              [3]Possible */
                    /*||||||||||||              [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                 (timer output does not change, either). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 0 0 1 : [1]Interval timer mode */
                    /*||||||||||||              [2]Counting down */
                    /*||||||||||||              [3]Possible */
                    /*||||||||||||              [4]Timer interrupt is generated when counting is started */
                    /*||||||||||||                 (timer output also changes). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 0 0 : [1]Capture mode */
                    /*||||||||||||              [2]Counting up */
                    /*||||||||||||              [3]Possible */
                    /*||||||||||||              [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                 (timer output does not change, either). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 0 1 : [1]Capture mode */
                    /*||||||||||||              [2]Counting up */
                    /*||||||||||||              [3]Possible */
                    /*||||||||||||              [4]Timer interrupt is generated when counting is started */
                    /*||||||||||||                 (timer output also changes). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 1 0 : [1]Event counter mode */
                    /*||||||||||||              [2]Counting down */
                    /*||||||||||||              [3]Possible */
                    /*||||||||||||              [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                 (timer output does not change, either). */
```

```
/*|||||||||||| */
/*|||||||||||| : 1 0 0 0 : [1]One-count mode */
/*||||||||||||                    [2]Counting down */
/*||||||||||||                    [3]Impossible */
/*||||||||||||                    [4]Start trigger is invalid during counting operation. */
/*||||||||||||                        At that time, interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 1 : [1]One-count mode */
/*||||||||||||                    [2]Counting down */
/*||||||||||||                    [3]Impossible */
/*||||||||||||                    [4]Start trigger is valid during counting operation. */
/*||||||||||||                        At that time, interrupt is also generated. */
/*|||||||||||| */
/*|||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*||||||||||||                    [2]Counting up */
/*||||||||||||                    [3]Possible */
/*||||||||||||                    [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                        (timer output does not change, either). */
/*||||||||||||                        Start trigger is invalid during counting operation. */
/*||||||||||||                        At that time interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : Other than above : Setting prohibited */
/*|||||||||||| */
/*||||||||||++-------- : Be sure to set 00 */
/*|||||||||| */
/*||||||||++---------- : Selection of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 valid
edge */
/*||||||||                   (the timer input used with channel 1 is selected by using TIS0
register). */
/*|||||||| : 0 0 : Falling edge */
/*|||||||| : 0 1 : Rising edge */
/*|||||||| : 1 0 : Both edges (when low-level width is measured) */
/*||||||||              Start trigger: Falling edge, Capture trigger: Rising edge */
/*|||||||| : 1 1 : Both edges (when high-level width is measured) */
/*||||||||              Start trigger: Rising edge, Capture trigger: Falling edge */
/*|||||||| */
/*|||||+++------------ : Setting of start trigger or capture trigger of channel 1 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are unselected).
*/
```

```
                    /*||||| : 0 0 1 : Valid edge of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 is used
as both the start trigger and capture trigger. */
                    /*||||| : 0 1 0 : Both the edges of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 are
used as a start trigger and a capture trigger. */
                    /*||||| : 1 0 0 : Interrupt signal of the master channel is used (when the channel is used
as a slave channel with the combination operation function). */
                    /*||||| : Other than above : Setting prohibited */
                    /*||||| */
                    /*||||+-------------- : Selection of slave/master of channel 1 */
                    /*|||| : 0 : Operates as slave channel with combination operation function. */
                    /*|||| : 1 : Operates as master channel with combination operation function. */
                    /*|||| */
                    /*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
                    /*||| : 0 : Operation clock MCK specified by CKS01 bit */
                    /*||| : 1 : Valid edge of input signal input from TI01 pin, fSUB/2, fSUB/4, or INTRTC1 */
                    /*|||        (the timer input used with channel 1 is selected by using TIS0 register). */
                    /*||| */
                    /*|++---------------- : Be sure to set 00 */
                    /*| */
                    /*+------------------ : Selection of operation clock (MCK) of channel 1 */
                    /* : 0 : Operation clock CK00 set by TPS0 register */
                    /* : 1 : Operation clock CK01 set by TPS0 register */
    TDR01 = CCK00_500USEC;          /* set interval time to 500us */
    TMMK01 = 1;                                     /* disable interrupt */

    /* CH4:for DMA */
    TMR04 = 0b1000000000000000;     /* Timer Mode Register 04 */
                    /*||||||||||||++++--- : [1]Operation mode of channel 4 */
                    /*||||||||||||         [2]Count operation of TCR */
                    /*||||||||||||         [3]Independent operation */
                    /*||||||||||||         [4]Setting of starting counting and interrupt */
                    /*|||||||||||| : 0 0 0 0 : [1]Interval timer mode */
                    /*||||||||||||               [2]Counting down */
                    /*||||||||||||               [3]Possible */
                    /*||||||||||||               [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                  (timer output does not change, either). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 0 0 1 : [1]Interval timer mode */
                    /*||||||||||||               [2]Counting down */
```

```
/*||||||||||||                    [3]Possible */
/*||||||||||||                    [4]Timer interrupt is generated when counting is started */
/*||||||||||||                         (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 0 0 : [1]Capture mode */
/*||||||||||||                    [2]Counting up */
/*||||||||||||                    [3]Possible */
/*||||||||||||                    [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                         (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 0 1 0 1 : [1]Capture mode */
/*||||||||||||                    [2]Counting up */
/*||||||||||||                    [3]Possible */
/*||||||||||||                    [4]Timer interrupt is generated when counting is started */
/*||||||||||||                         (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 1 0 : [1]Event counter mode */
/*||||||||||||                    [2]Counting down */
/*||||||||||||                    [3]Possible */
/*||||||||||||                    [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                         (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 0 : [1]One-count mode */
/*||||||||||||                    [2]Counting down */
/*||||||||||||                    [3]Impossible */
/*||||||||||||                    [4]Start trigger is invalid during counting operation. */
/*||||||||||||                         At that time, interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 1 : [1]One-count mode */
/*||||||||||||                    [2]Counting down */
/*||||||||||||                    [3]Impossible */
/*||||||||||||                    [4]Start trigger is valid during counting operation. */
/*||||||||||||                         At that time, interrupt is also generated. */
/*|||||||||||| */
/*|||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*||||||||||||                    [2]Counting up */
/*||||||||||||                    [3]Possible */
/*||||||||||||                    [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                         (timer output does not change, either). */
```

```
/*||||||||||||                    Start trigger is invalid during counting operation. */
/*||||||||||||                    At that time interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : Other than above : Setting prohibited */
/*|||||||||||| */
/*||||||||||++------- : Be sure to set 00 */
/*|||||||||| */
/*||||||||++--------- : Selection of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 valid edge */
/*||||||||                (the timer input used with channel 4 is selected by using TIS0 register). */
/*|||||||| : 0 0 : Falling edge */
/*|||||||| : 0 1 : Rising edge */
/*|||||||| : 1 0 : Both edges (when low-level width is measured) */
/*||||||||           Start trigger: Falling edge, Capture trigger: Rising edge */
/*|||||||| : 1 1 : Both edges (when high-level width is measured) */
/*||||||||           Start trigger: Rising edge, Capture trigger: Falling edge */
/*|||||||| */
/*|||||+++----------- : Setting of start trigger or capture trigger of channel 4 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are unselected). */
/*||||| : 0 0 1 : Valid edge of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 is used as both the start trigger and capture trigger. */
/*||||| : 0 1 0 : Both the edges of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 are used as a start trigger and a capture trigger. */
/*||||| : 1 0 0 : Interrupt signal of the master channel is used (when the channel is used as a slave channel with the combination operation function). */
/*||||| : Other than above : Setting prohibited */
/*||||| */
/*||||+-------------- : Selection of slave/master of channel 4 */
/*|||| : 0 : Operates as slave channel with combination operation function. */
/*|||| : 1 : Operates as master channel with combination operation function. */
/*|||| */
/*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
/*||| : 0 : Operation clock MCK specified by CKS04 bit */
/*||| : 1 : Valid edge of input signal input from TI04 pin, fSUB/2, fSUB/4, or INTRTC1 */
/*|||        (the timer input used with channel 4 is selected by using TIS0 register). */
/*||| */
/*|++---------------- : Be sure to set 00 */
```

```
                        /*| */
                        /*+------------------ : Selection of operation clock (MCK) of channel 4 */
                        /* : 0 : Operation clock CK00 set by TPS0 register */
                        /* : 1 : Operation clock CK01 set by TPS0 register */
    TDR04 = 200;        /* set interval time to 10us(=0.05us * 200) */
    TMMK04 = 1;                 /* disable interrupt */

}

/*--------------------------------------------------------------*/
/* Module:          fn_InitLvi
                    */
/* Description:     Setting of Low-voltage detectior                         */
/*      parameter: --
                        */
/*      return     : --
                        */
/*--------------------------------------------------------------*/
void    fn_InitLvi(void)
{
    unsigned short      loop;       /* waiting counter */

    LVIMK = 1;                          /* disable LVI interrupt */

    LVIS =      0b00001001;             /* Low-Voltage Detection Level Select Register */
                        /*||||++++--- : Detection level */
                        /*|||| : 0 0 0 0 : VLVI0 (4.22V) */
                        /*|||| : 0 0 0 1 : VLVI1 (4.07V) */
                        /*|||| : 0 0 1 0 : VLVI2 (3.92V) */
                        /*|||| : 0 0 1 1 : VLVI3 (3.76V) */
                        /*|||| : 0 1 0 0 : VLVI4 (3.61V) */
                        /*|||| : 0 1 0 1 : VLVI5 (3.45V) */
                        /*|||| : 0 1 1 0 : VLVI6 (3.30V) */
                        /*|||| : 0 1 1 1 : VLVI7 (3.15V) */
                        /*|||| : 1 0 0 0 : VLVI8 (2.99V) */
                        /*|||| : 1 0 0 1 : VLVI9 (2.84V) */
                        /*|||| : 1 0 1 0 : VLVI10 (2.68V) */
                        /*|||| : 1 0 1 1 : VLVI11 (2.53V) */
                        /*|||| : 1 1 0 0 : VLVI12 (2.38V) */
```

```
                        /*||||    : 1 1 0 1 : VLVI13 (2.22V) */
                        /*||||    : 1 1 1 0 : VLVI14 (2.07V) */
                        /*||||    : 1 1 1 1 : VLVI15 (1.91V) */
                        /*||||  */
                        /*++++------- : Be sure to set 0000 */

        LVIM =    0b10000000;           /* Low-Voltage Detection Register */
                        /*|||||||+--- : LVIF Low-voltage detection flag */
                        /*||||||| : 0 : * LVISEL = 0: VDD >= VLVI, or when LVI operation is disabled */
                        /*|||||||       * LVISEL = 1: EXLVI >= VEXLVI, or when LVI operation is disabled */
                        /*||||||| : 1 : * LVISEL = 0: VDD < VLVI */
                        /*|||||||       * LVISEL = 1: EXLVI < VEXLVI */
                        /*||||||| */
                        /*||||||+---- : Low-voltage detection operation mode (interrupt/reset) selection(LVIMD) */
                        /*|||||| : 0 : * LVISEL = 0: Generates an internal interrupt signal */
                        /*||||||                 when VDD drops lower than VLVI (VDD < VLVI) */
                        /*||||||                 or when VDD becomes VLVI or higher (VDD >= VLVI).
*/
                        /*||||||        * LVISEL = 1: Generates an interrupt signal */
                        /*||||||                  when EXLVI drops lower than VEXLVI (EXLVI <
VEXLVI) */
                        /*||||||                 or when EXLVI becomes VEXLVI or higher (EXLVI >=
VEXLVI). */
                        /*|||||| : 1 : * LVISEL = 0: Generates an internal reset signal when VDD < VLVI */
                        /*||||||                 and releases the reset signal when VDD >= VLVI. */
                        /*||||||        * LVISEL = 1: Generates an internal reset signal when EXLVI < VEXLVI
*/
                        /*||||||                 and releases the reset signal when EXLVI >= VEXLVI.
*/
                        /*|||||| */
                        /*|||||+----- : Voltage detection selection(LVISEL) */
                        /*||||| : 0 : Detects level of supply voltage (VDD) */
                        /*||||| : 1 : Detects level of input voltage from external input pin (EXLVI) */
                        /*||||| */
                        /*|++++------ : Be sure to set 0000 */
                        /*| */
                        /*+---------- : Enables low-voltage detection operation */
                        /*   : 0 : Disables operation */
                        /*   : 1 : Enables operation */
```

```
        /* software to wait for the operation stabilization time (210us) */
        for(loop = 500; loop > 0; loop--){
                NOP();
        }
        /* wait until VLVI≤VDD */
        while( LVIF ){
                NOP();
        }

        LVIIF = 0;                      /* clear LVI interrupt request flag */
}




/*------------------------------------------------------------*/
/* Module:          fn_DmaInit
                                */
/* Description:     Setting of DMA contoroler                                        */
/*      parameter: --
                                */
/*      return     : --
                                */
/*------------------------------------------------------------*/
void    fn_DmaInit(void)
{

        DEN0 = 1;                                       /* Enables operation of DMA channel 0 */

        DSA0 = (unsigned char)&DACSW0;                              /* DMA SFR Address Register 0
*/
        DRA0 = (unsigned short)&ushDacData[0];          /* DMA RAM Address Register 0 */
        DBC0 = 100;                                             /* DMA Byte Count Register 0 */
        DMC0 = 0b01100100;                              /* DMA Mode Control Register 0 */
                  /*||||||||| */
                  /*||||++++------- IFC3-IFC0 */
                  /*||||              [Selection of DMA start source] */
                  /*||||               0000:Disables DMA transfer by interrupt */
```

```
                    /*||||                              (Only software trigger is enabled.) */
                    /*||||          0010:INTTM00 */
                    /*||||          0011:INTTM01 */
                    /*||||          0100:INTTM04 */
                    /*||||          0101:INTTM05 */
                    /*||||          0110:INTST0/INTCSI00 */
                    /*||||          0111:INTSR0/INTCSI01 */
                    /*||||          1000:INTST1/INTCSI10/INTIIC10 */
                    /*||||          1001:INTSR1 */
                    /*||||          1010:INTST3 */
                    /*||||          1011:INTSR3 */
                    /*||||          1100:INTAD */
                    /*|||+----------- DWAIT0 */
                    /*|||          [Pending of DMA transfer] */
                    /*|||           0:Executes DMA transfer upon DMA start request (not held pending). */
                    /*|||           1:Holds DMA start request pending if any. */
                    /*||+------------ DS0 */
                    /*||           [Specification of transfer data size for DMA transfer] */
                    /*||            0:8bits */
                    /*||            1:16bits */
                    /*|+------------- DRS0 */
                    /*|            [Selection of DMA transfer direction] */
                    /*|             0:SFR to internal RAM */
                    /*|             1:Internal RAM to SFR */
                    /*+------------- STG0 */
                    /*             [DMA transfer start software trigger] */
                    /*              0:No trigger operation */
                    /*              1:DMA transfer is started when DMA operation is enabled (DEN0 = 1). */

        DST0 = 1;               /* DMA transfer is started when DMA operation is enabled (DEN0 = 1). */

}


/*--------------------------------------------------------------*/
/* Module:          fn_Wait500usBase
        */
/* Description:     Delays the program for (Time * 500us)                   */
/*      parameter: wait time(Time)
```

```
          */
/*        return      : --
                              */
/*---------------------------------------------------------------*/
void      fn_Wait500usBase(unsigned short Time)
{
          TS0L.1 = 1;                         /* start TAU0 CH1 */
          TMIF01 = 0;

          for(; Time > 0; Time--){            /* wait for (parameter * 500)us */
                    while(!TMIF01){
                              NOP();
                    }
                    TMIF01 = 0;
          }

          TT0L.1 = 1;                         /* stop TAU0 CH1 */
}


/*---------------------------------------------------------------*/
/* Module:            fn_AdcDacInit
                    */
/* Description:       Initialization of A/D D/A OPAmp module                    */
/*        parameter: --
                              */
/*        return      : --
                              */
/*---------------------------------------------------------------*/
void      fn_AdcDacInit(void)
{
          ADCEN = 1;                          /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

          /*--------------------------------------------*/
          /*    Initialization of Voltage reference                */
          /*--------------------------------------------*/
          ADVRC = 0b00001000;           /* Analog reference voltage control register */
                              /*||||+|++--- : [1]Positive reference voltage supplies selection of A/D and D/A
```

```
converters */
                              /*IIII |        [2]Operationcontrol of voltage reference */
                              /*IIII |        [3]Output voltage selection of voltage reference */
                              /*IIII |        [4]Operation control of input gate voltage boost circuit for A/D converter
*/
                              /*IIII |        [5]Relationship with the conversion mode used */
                              /*IIII | */
                              /*IIII | : 0 0 0 : [1]AVREFP (external voltage reference input) */
                              /*IIII |          [2]Stops operation (Hi-Z) */
                              /*IIII |          [3]2.5 V */
                              /*IIII |          [4]Stops operation */
                              /*IIII |          [5]Can be set in conversion mode 1 */
                              /*IIII | */
                              /*IIII | : 0 1 0 : [1]AVREFP (external voltage reference input) */
                              /*IIII |          [2]Stops operation (Hi-Z) */
                              /*IIII |          [3]2.0 V */
                              /*IIII |          [4]Enables operation */
                              /*IIII |          [5]Can be set in conversion mode 2 or 3 */
                              /*IIII | */
                              /*IIII | : 1 0 0 : [1]VREFOUT (voltage reference output) */
                              /*IIII |          [2]Stops operation (pull-down output) */
                              /*IIII |          [3]2.5 V */
                              /*IIII |          [4]Stops operation */
                              /*IIII |          [5] - */
                              /*IIII | */
                              /*IIII | : 1 0 1 : [1]VREFOUT (voltage reference output) */
                              /*IIII |          [2]Enables operation */
                              /*IIII |          [3]2.5 V */
                              /*IIII |          [4]Enables operation */
                              /*IIII |          [5]Can be set in conversion mode 2 or 3 */
                              /*IIII | */
                              /*IIII | : 1 1 0 : [1]VREFOUT (voltage reference output) */
                              /*IIII |          [2]Stops operation (pull-down output) */
                              /*IIII |          [3]2.0 V */
                              /*IIII |          [4]Enables operation */
                              /*IIII |          [5] - */
                              /*IIII | */
                              /*IIII | : 1 1 1 : [1]VREFOUT (voltage reference output) */
                              /*IIII |          [2]Enables operation */
```

```
                    /*|||| |                [3]2.0 V */
                    /*|||| |                [4]Enables operation */
                    /*|||| |                [5]Can be set in conversion mode 2 or 3 */
                    /*|||| | */
                    /*|||| | : Other than the above : Setting prohibited */
                    /*|||| | */
                    /*|+++-+----- : Be sure to set 0000 */
                    /*| */
                    /*+---------- : Reference voltage supply (negative side) of A/D converter selection */
                    /* : 0 : AVSS */
                    /* : 1 : AVREFM (external voltage reference input) */

ADVRC.0 = 1;                    /* Enables operation */
ADVRC.1 = 1;                    /* Output 2.0 V */

ADPC =    0b00000000;           /* A/D Port Configuration Register */
                    /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                    /*||| :                  +------------------------ ANI15/AVREFM/P157 */
                    /*||| :                  | +-+-+------------------ ANI10/P152 - ANI8/AMP2+/P150 */
                    /*||| :                  | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
                    /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                    /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                    /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                    /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                    /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                    /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                    /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                    /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                    /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                    /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                    /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                    /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                    /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                    /*||| */
                    /*+++-------- : Be sure to set 000 */

/* wait for settling time to 17msec */
fn_Wait500usBase(17000/500);
```

```
        /*-------------------------------------------*/
        /*    Initialization of operational amplifier    */
        /*-------------------------------------------*/
        PM2.1 = 1;                    /* input port mode (AMP0O) */
        PM2.0 = 1;                    /* input port mode (AMP0-) */
        PM2.2 = 1;                    /* input port mode (AMP0+) */

        PM2.4 = 1;                    /* input port mode (AMP1O) */
        PM2.3 = 1;                    /* input port mode (AMP1-) */
        PM2.5 = 1;                    /* input port mode (AMP1+) */

        PM2.7 = 1;                    /* input port mode (AMP2O) */
        PM2.6 = 1;                    /* input port mode (AMP2-) */
        PM15.0 = 1;                   /* input port mode (AMP2+) */

        PM11.0 = 1;                   /* input port mode (ANO0) */

        ADCEN = 1;                    /* supplies input clock to OPAmp A/D converter */
        OAC.0 = 1;                    /* operational amplifier (AMP0) enable */
        OAC.1 = 0;                    /* operational amplifier (AMP1) disable */
        OAC.2 = 0;                    /* operational amplifier (AMP2) disable */

        /*-------------------------------------------*/
        /*          Initialization of D/A converter          */
        /*-------------------------------------------*/
        DACEN = 1;                            /* supplies input clock to D/A converter */

        DAM.0 = 1;                            /* [DAMD0] D/A converter operation is Real-time
output mode */
    /*  DAM.0 = 0;                            /* [DAMD0] D/A converter operation is Normal mode
*/
        DAM.2 = 1;                            /* [DARES0]D/A conversion resolution is 12-bit */
        DAM.6 = 1;                            /* [DAREF] D/A converter positive reference voltage
is VREFOUT */
        DACSW0 = aushDacData[99];     /* D/A conversion value setting reigster 0 */

        /* set D/A chanel1 */
        DAM.1 = 0;                            /* [DAMD1] D/A converter operation is Normal mode
*/
```

```
        DAM.3 = 1;                                      /* [DARES1]D/A conversion resolution is 12-bit */
        DACSW1 = aushDacData[90];               /* D/A conversion value setting reigster 1 */

}
/*--------------------------------------------------------------*/
/* Module:          main
                                */
/* Description:     Main process
        */
/*      parameter: --
                                */
/*      return      : --
                                */
/*--------------------------------------------------------------*/
void main(void)
{
        unsigned short          count;          /* counter for initializing D/A convert data */

        ushCycleCount = 0;                      /* Cycle counter */
        ucCycleStatus = 0;                      /* Cycle status */

        DACE1 = 1;                                              /* D/A converter CH1 enable */
        DACE0 = 1;                                              /* D/A converter CH0 enable */


        /*==================================*/
        /*----------------------------------*/
        /*              Main Loop               */
        /*----------------------------------*/
        /*==================================*/

        /*  initialize D/Aconverter data */
        for(count = 0; count<100; count++){
                ushDacData[count] = aushDacData[count];
        }

        TS0L.4 = 1;                             /* start TAU0 CH4 */
        TMIF04 = 0;                             /* reset INTTM04 interrupt request */
```

```
        DMAMK0 = 0;                             /* enable DMA CH0 interrupt */

        EI();                                   /* enable all interrupts */

        /*================================================================*/
        /* if system have nothing to do, go to standby for power-saving */
        /*================================================================*/


        while (1){
                HALT();
                NOP();
        }
}


/*--------------------------------------------------------------*/
/* Module:          fn_intdma0
                              */
/* Description:     DMA ch0 interruput process                                         */
/*      parameter: --
                              */
/*      return    : --
                              */
/*--------------------------------------------------------------*/
__interrupt void fn_intdma0(void)
{
        TT0L.4 = 1;                             /* stop TAU0 CH4 */


        /* start A/D conversion */
/*      DI();               /* disable all interrupts */
        ADCS = 1;           /* A/D converter enable */
        ADIF = 0;  /* clear A/D interrupt request flag */
        ADMK = 0;           /* enable A/D interrupt */
        HALT();             /* set HALT mode and wait conversion finished */
        ADMK = 1;           /* disable A/D interrupt */
        ADIF = 0;  /* clear A/D interrupt request flag */
/*      EI();               /* enable all interrupts */
```

```
        shVoltage = (short)(ADCR & 0x0fff);          /* get A/D result */
        shVoltage -= (2048 - 1);
        if(shVoltage < 0){
                DACSW1 += 1;                         /* D/A conversion value setting reigster 1 */
        }
        else if(shVoltage > 2){
                DACSW1 -= 1;                         /* D/A conversion value setting reigster 1 */
        }

        ushCycleCount++;                             /* Cycle counter */
        if(ushCycleCount > 50){                      /* Cycle counter */
                switch(ucCycleStatus){                       /* Cycle status */
                        case    0:
                                TDR04 = 2000;                        /*    set    interval    time    to
100us(=0.05us * 2000) */
                                ucCycleStatus = 1;           /* Cycle status (100Hz) */
                                ushCycleCount = 20;                  /* reset Cycle counter */
                                break;

                        case    1:
                                TDR04 = 200;                         /*    set    interval    time    to
10us(=0.05us * 200) */
                                ucCycleStatus = 0;           /* Cycle status (1kHz) */
                                ushCycleCount = 0;                   /* reset Cycle counter */
                                break;

                }
                DACSW1 = ushDacData[90];             /* D/A conversion value setting reigster 1 */
  /*            ushCycleCount = 0;                   /* reset Cycle counter */

        }

        DRA0 = (unsigned short)&ushDacData[0];       /* DMA RAM Address Register 0 */
        DBC0 = 100;                                          /* DMA Byte Count Register 0 */

        DST0 = 1;                                    /* DMA transfer is started when DMA operation is
enabled (DEN0 = 1). */
```

```
        TS0L.4 = 1;                              /* start TAU0 CH4 */
        TMIF04 = 0;                              /* reset INTTM04 interrupt request */

}
```

(5) Non-inverting amplifier circuit (AC)

● main.c

```
/*
 * Copyright (C) NEC Electronics Corporation 2006
 * NEC ELECTRONICS CONFIDENTIAL AND PROPRIETARY
 * All rights reserved by NEC Electronics Corporation.
 * This program must be used solely for the purpose for which
 * it was furnished by NEC Electronics Corporation. No part of this
 * program may be reproduced or disclosed to others, in any
 * form, without the prior written permission of NEC Electronics
 * Corporation. Use of copyright notice dose not evidence
 * publication of the program.
 */

/*--------------------------------------------------------------------*/
/* #pragma directive for CC78K0
                                        */
/*--------------------------------------------------------------------*/
#pragma         SFR
#pragma         DI
#pragma         EI
#pragma         HALT
#pragma         NOP

#pragma interrupt INTDMA0 fn_intdma0 RB1

/*--------------------------------------------------------------------*/
/* Include files
                                                      */
/*--------------------------------------------------------------------*/
/* TAU:TDR0n value operation by CK00 (fCLK/2^3 = 5MHz) */
#define CCK00_500USEC    (2500 - 1)          /* 500us (0.2[us/clk] * 2500[count]) */



/*--------------------------------------------------------------------*/
/* Function prototyps
                                                      */
/*--------------------------------------------------------------------*/
```

```
        void    fn_Wait500usBase(unsigned short);    /* Delays the program for (Time * 500us) */
        void    fn_InitPort(void);                                        /* Setting of I/O ports
*/
        void    fn_InitTau0(void);                                        /* Setting of Timer
array unit 0 */
        void    fn_InitLvi(void);                                         /* Setting of
Low-voltage detectior */
        void    fn_InitVr(void);                                          /* Setting of Voltage
reference */
        void    fn_AdcDacInit(void);                                  /* Setting of D/A converter */
        void    fn_DmaInit(void);                                         /* Setting of DMA
controler */


 /*-------------------------------------------------------------------*/
 /* Extern variables/constants
        */
 /*-------------------------------------------------------------------*/

 /*-------------------------------------------------------------------*/
 /* Local constants
                        */
 /*-------------------------------------------------------------------*/
 /*-------------------------------------------------------------------*/
 /* Global variables
                        */
 /*-------------------------------------------------------------------*/

 /*-------------------------------------------------------------------*/
 /* Local variables
                        */
 /*-------------------------------------------------------------------*/
 static  short          shVoltage;/* voltage (Hex) */
 static  unsigned short  ushCycleCount;   /* Cycle counter */
 static  unsigned char   ucCycleStatus;   /* Cycle status */

 /*-------------------------------------------------------------------*/
 /* Code
                                    */
```

```
/*-----------------------------------------------------------------------*/
/*===========================================================================

        D/A output data

===========================================================================*/
static unsigned short ushDacData[100];          /* D/A data for DMA */
const unsigned short aushDacData[100] = {       /* defined D/A data */
                                                /* n        sin(0.2pi*n/100) */
/*              2048,           /* 0            0.100000 */
                2061,           /* 1            0.106279 */
                2074,           /* 2            0.112533 */
                2086,           /* 3            0.118738 */
                2099,           /* 4            0.124869 */
                2111,           /* 5            0.130902 */
                2123,           /* 6            0.136812 */
                2135,           /* 7            0.142578 */
                2147,           /* 8            0.148175 */
                2158,           /* 9            0.153583 */
                2168,           /* 10           0.158779 */
                2179,           /* 11           0.163742 */
                2188,           /* 12           0.168455 */
                2197,           /* 13           0.172897 */
                2206,           /* 14           0.177051 */
                2214,           /* 15           0.180902 */
                2221,           /* 16           0.184433 */
                2227,           /* 17           0.187631 */
                2233,           /* 18           0.190483 */
                2238,           /* 19           0.192978 */
                2243,           /* 20           0.195106 */
                2246,           /* 21           0.196858 */
                2249,           /* 22           0.198229 */
                2251,           /* 23           0.199211 */
                2252,           /* 24           0.199803 */
                2253,           /* 25           0.200000 */
                2252,           /* 26           0.199803 */
                2251,           /* 27           0.199211 */
                2249,           /* 28           0.198229 */
                2246,           /* 29           0.196858 */
```

```
2243,           /* 30     0.195106 */
2238,           /* 31     0.192978 */
2233,           /* 32     0.190483 */
2227,           /* 33     0.187631 */
2221,           /* 34     0.184433 */
2214,           /* 35     0.180902 */
2206,           /* 36     0.177051 */
2197,           /* 37     0.172897 */
2188,           /* 38     0.168455 */
2179,           /* 39     0.163742 */
2168,           /* 40     0.158779 */
2158,           /* 41     0.153583 */
2147,           /* 42     0.148175 */
2135,           /* 43     0.142578 */
2123,           /* 44     0.136812 */
2111,           /* 45     0.130902 */
2099,           /* 46     0.124869 */
2086,           /* 47     0.118738 */
2074,           /* 48     0.112533 */
2061,           /* 49     0.106279 */
2048,           /* 50     0.100000 */
2035,           /* 51     0.093721 */
2022,           /* 52     0.087467 */
2010,           /* 53     0.081262 */
1997,           /* 54     0.075131 */
1985,           /* 55     0.069098 */
1973,           /* 56     0.063188 */
1961,           /* 57     0.057422 */
1949,           /* 58     0.051825 */
1938,           /* 59     0.046417 */
1928,           /* 60     0.041221 */
1917,           /* 61     0.036258 */
1908,           /* 62     0.031545 */
1899,           /* 63     0.027103 */
1890,           /* 64     0.022949 */
1882,           /* 65     0.019098 */
1875,           /* 66     0.015567 */
1869,           /* 67     0.012369 */
1863,           /* 68     0.009517 */
```

```
        1858,           /* 69    0.007022 */
        1853,           /* 70    0.004894 */
        1850,           /* 71    0.003142 */
        1847,           /* 72    0.001771 */
        1845,           /* 73    0.000789 */
        1844,           /* 74    0.000197 */
        1843,           /* 75    0.000000 */
        1844,           /* 76    0.000197 */
        1845,           /* 77    0.000789 */
        1847,           /* 78    0.001771 */
        1850,           /* 79    0.003142 */
        1853,           /* 80    0.004894 */
        1858,           /* 81    0.007022 */
        1863,           /* 82    0.009517 */
        1869,           /* 83    0.012369 */
        1875,           /* 84    0.015567 */
        1882,           /* 85    0.019098 */
        1890,           /* 86    0.022949 */
        1899,           /* 87    0.027103 */
        1908,           /* 88    0.031545 */
        1917,           /* 89    0.036258 */
        1928,           /* 90    0.041221 */
        1938,           /* 91    0.046417 */
        1949,           /* 92    0.051825 */
        1961,           /* 93    0.057422 */
        1973,           /* 94    0.063188 */
        1985,           /* 95    0.069098 */
        1997,           /* 96    0.075131 */
        2010,           /* 97    0.081262 */
        2022,           /* 98    0.087467 */
        2035,           /* 99    0.093721 */
        2048            /* 100   0.100000 */

};

/*---------------------------------------------------------------------*/
/* Hardware initialization
                                */
/*---------------------------------------------------------------------*/
```

```
void hdwinit(void)
{
        DI();                           /* disable all interrupts */

        /*----------------------------------------*/
        /*          Initialization of port          */
        /*----------------------------------------*/
        fn_InitPort();

        /*---------------------------------------*/
        /*         Initialization of clock         */
        /*---------------------------------------*/
        CMC =   0b01010011;              /* Clock Operation Mode Control Register */
                        /*|||||||+--- : Control of high-speed system clock oscillation frequency */
                        /*||||||| : 0 : 2 MHz <= fMX <= 10 MHz */
                        /*||||||| : 1 : 10 MHz < fMX <= 20 MHz */
                        /*||||||| */
                        /*|||||++---- : XT1 oscillator oscillation mode selection */
                        /*||||| : 0 0 : Low-consumption oscillation */
                        /*||||| : 0 1 : Normal oscillation */
                        /*||||| : 1 x : Super-low-consumption oscillation */
                        /*|||||    x = don't care */
                        /*||||| */
                        /*||||+------ : Be sure to set 0 */
                        /*|||| */
                        /*|||+------- : [1] Subsystem clock pin operation mode */
                        /*|||              [2] XT1/P123 pin and XT2/P124 pin */
                        /*||| : 0 : [1]Input port mode */
                        /*|||         [2]Input port */
                        /*||| */
                        /*||| : 1 : [1]XT1 oscillation mode */
                        /*|||         [2]Crystal resonator connection */
                        /*||| */
                        /*||+-------- : Be sure to set 0 */
                        /*|| */
                        /*++--------- : [1]EXCLK OSCSEL High-speed system clock pin operation mode */
                        /*                 [2]X1/P121 pin */
                        /*                 [3]X2/EXCLK/P122 pin */
                        /* : 0 0 : [1]Input port mode */
```

```
                    /*          [2][3]Input port */
                    /* */
                    /* : 0 1 : [1]X1 oscillation mode */
                    /*          [2][3]Crystal/ceramic resonator connection */
                    /* */
                    /* : 1 0 : [1]Input port mode */
                    /*          [2][3]Input port */
                    /* */
                    /* : 1 1 : [1]External clock input mode */
                    /*          [2]Input port */
                    /*          [3]External clock input */

    MSTOP = 0;                          /* X1 oscillator operating */
    XTSTOP = 0;                         /* XT1 oscillator operating */

    OSMC =  0b00000001;                 /* Operation Speed Mode Control Register */
                    /*||||||++--- : fCLK frequency selection */
                    /*|||||| : 0 0 : Operates at a frequency of 10 MHz or less. */
                    /*|||||| : 0 1 : Operates at a frequency higher than 10 MHz. */
                    /*|||||| : 1 0 : Operates at a frequency of 1 MHz. */
                    /*|||||| */
                    /*|+++++----- : Be sure to set 00000 */
                    /*| */
                    /*+---------- : Setting in subsystem clock HALT mode */
                    /* : 0 : Enables subsystem clock supply to peripheral functions. */
                    /*        (See Table 21-1 Operating Statuses in HALT Mode (2/3) */
                    /*         for the peripheral functions whose operations are enabled.) */
                    /* : 1 : Stops subsystem clock supply to peripheral functions except real-time counter,
*/
                    /*         clock output/buzzer output, and LCD controller/driver. */

    while(OSTC.0 != 1){             /* wait X1 oscillation stabilization */
            NOP();
    }

    /*-- Caution ---------------------------------------------*/
    /* To increase fCLK to 10 MHz or higher, set FSEL to '1',    */
    /* then change fCLK after two or more clocks have elapsed. */
    /*-------------------------------------------------------*/
```

```
NOP();
NOP();

CKC =   0b00010000;             /* System Clock Control Register */
                /*|+|+++++--- : Selection of CPU/peripheral hardware clock (fCLK) */
                /*| | : 0 0 x 0 0 0 : fIH */
                /*| | : 0 0 x 0 0 1 : fIH/2 (default) */
                /*| | : 0 0 x 0 1 0 : fIH/2^2 */
                /*| | : 0 0 x 0 1 1 : fIH/2^3 */
                /*| | : 0 0 x 1 0 0 : fIH/2^4 */
                /*| | : 0 0 x 1 0 1 : fIH/2^5 */
                /*| | : 0 1 x 0 0 0 : fMX */
                /*| | : 0 1 x 0 0 1 : fMX/2 */
                /*| | : 0 1 x 0 1 0 : fMX/2^2 */
                /*| | : 0 1 x 0 1 1 : fMX/2^3 */
                /*| | : 0 1 x 1 0 0 : fMX/2^4 */
                /*| | : 0 1 x 1 0 1 : fMX/2^5 */
                /*| | : 1 x 0 x x x : fSUB */
                /*| | : 1 x 1 x x x : fSUB/2 */
                /*| | : Other than above : Setting prohibited */
                /*| |    x = don't care */
                /*| | */
                /*| +-------- : Status of Main system clock (fMAIN) */
                /*| : 0 : Internal high-speed oscillation clock (fIH) */
                /*| : 1 : High-speed system clock (fMX) */
                /*| */
                /*+---------- : Status of CPU/peripheral hardware clock (fCLK) */
                /* : 0 : Main system clock (fMAIN) */
                /* : 1 : Subsystem clock (fSUB) */

/* Confirming the CPU clock status */
while((CLS != 0)||(MCS != 1)){
        NOP();
}
/* CPU is operating on a High-speed system clock */
HIOSTOP = 1;                    /* internal high-speed oscillation stopped */

OSTS =  0b00000111;             /* Oscillation Stabilization Time Select Register */
                /*||||++++--- : Oscillation stabilization time selection */
```

```
                        /*||||  : 0 0 0 : 2^8/fX */
                        /*||||  : 0 0 1 : 2^9/fX */
                        /*||||  : 0 1 0 : 2^10/fX */
                        /*||||  : 0 1 1 : 2^11/fX */
                        /*||||  : 1 0 0 : 2^13/fX */
                        /*||||  : 1 0 1 : 2^15/fX */
                        /*||||  : 1 1 0 : 2^17/fX */
                        /*||||  : 1 1 1 : 2^18/fX */
                        /*|||| */
                        /*++++------- : Be sure to set 000000 */

    /*----------------------------------------*/
    /*          Initialization of timer          */
    /*----------------------------------------*/
    fn_InitTau0();

    /* software to wait for the operation stabilization time */
    /* (over 200ms from when XT1 enable) */
    fn_Wait500usBase(200000/500);

    /*----------------------------------------*/
    /* Initialization of low-voltage detector */
    /*----------------------------------------*/
    fn_InitLvi();

    /*----------------------------------------*/
    /*  Initialization of A/D,D/A,OPAmp              */
    /*----------------------------------------*/
    fn_AdcDacInit();

    /*----------------------------------------*/
    /*  Initialization of DMA                           */
    /*----------------------------------------*/
    fn_DmaInit();

    EI();                             /* enable all interrupts */

}
```

```
/*--------------------------------------------------------------*/
/* Module:          fn_InitPort
                    */
/* Description:     Setting of I/O ports                                    */
/*      parameter: --
                              */
/*      return    : --
                              */
/*--------------------------------------------------------------*/
void    fn_InitPort(void)
{
/*----------------------------------------------------------*/
/*      Ports configuration for digital input and output                */
/*----------------------------------------------------------*/
        ADPC =  0b00010000;         /* A/D Port Configuration Register */
                              /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                              /*||| :                   +------------------------ ANI15/AVREFM/P157 */
                              /*||| :                   | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                              /*||| :                   | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
                              /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                              /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                              /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                              /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                              /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                              /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                              /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                              /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                              /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                              /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                              /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                              /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                              /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                              /*||| */
                              /*+++-------- : Be sure to set 000 */

/*----------------------------------------------------------*/
/*      Setting of Port 0
        */
/*----------------------------------------------------------*/
```

```
	P0 =	0b00000000;	/* Set P00-P02 Output latch to Low */
	PM0 =	0b11111000;	/* Set P00-P02 to output port */
					/* P00-P02:Unused */

/*----------------------------------------------------------*/
/*	Setting of Port 1
	*/
/*----------------------------------------------------------*/
	P1 =	0b00000000;	/* Set P10-P17 Output latch to Low */
	PM1 =	0b00000000;	/* Set P10-P17 to output port */
					/* P10-P15:Unused */

/*----------------------------------------------------------*/
/*	Setting of Port 2
	*/
/*----------------------------------------------------------*/
	P2 =	0b00000000;	/* Set P20-P27 Output latch to Low */
	PM2 =	0b11111111;	/* Set P20-P27 to input port */
					/* P20-P27:Unused */

/*----------------------------------------------------------*/
/*	Setting of Port 3
	*/
/*----------------------------------------------------------*/
	P3 =	0b00001100;	/* Set P30-P31,P34 Output latch to Low */
					/* Set P33,P32 Output latch High */
	PM3 =	0b11100000;	/* Set P30-P34 to output port */
					/* P30-P34:Unused */

/*----------------------------------------------------------*/
/*	Setting of Port 4
	*/
/*----------------------------------------------------------*/
	P4 =	0b00000000;	/* Set P40-P41 Output latch to Low */
	PM4 =	0b11111100;	/* Set P40-P41 to output port */
					/* P40-P41:Unused */

/*----------------------------------------------------------*/
/*	Setting of Port 5
```

```
        */
/*--------------------------------------------------------*/
        P5 =    0b00000000;     /* Set P50-P57 Output latch to Low */
        PM5 =   0b11110000;     /* Set P50-P57 to output port */
                                                /* P50-P57:Unused */

/*--------------------------------------------------------*/
/*      Setting of Port 6
        */
/*--------------------------------------------------------*/
        P6 =    0b00000000;     /* Set P60-P61 Output latch to Low */
        PM6 =   0b11111100;     /* Set P60-P61 to output port */
                                                /* P60-P61:Unused */

/*--------------------------------------------------------*/
/*      Setting of Port 7
        */
/*--------------------------------------------------------*/
        P7 =    0b00000000;     /* Set P70-P77 Output latch to Low */
        PM7 =   0b00000000;     /* Set P70-P77 to output port */
                                                /* P70-P77:Unused */

/*--------------------------------------------------------*/
/*      Setting of Port 8
        */
/*--------------------------------------------------------*/
        P8 =    0b00000000;     /* Set P80-P88 Output latch to Low */
        PM8 =   0b00000000;     /* Set P80-P88 to output port */
                                                /* P80-P88:Unused */

/*--------------------------------------------------------*/
/*      Setting of Port 9
        */
/*--------------------------------------------------------*/
        P9 =    0b00000000;     /* Set P90-P97 Output latch to Low */
        PM9 =   0b00000000;     /* Set P90-P97 to output port */
                                                /* P90-P97:Unused */

/*--------------------------------------------------------*/
```

```
/*      Setting of Port 10
        */
/*---------------------------------------------------------*/
        P10 =   0b00000000;     /* Set P100-P102 Output latch to Low */
        PM10 =  0b11111000;     /* Set P100-P102 to output port */
                                        /* P100-P102:Unused */

/*---------------------------------------------------------*/
/*      Setting of Port 11
        */
/*---------------------------------------------------------*/
        P11 =   0b00000000;     /* Set P110-P111 Output latch to Low */
        PM11 =  0b11111100;     /* Set P110-P111 to output port */
                                        /* P110-P111:Unused */

/*---------------------------------------------------------*/
/*      Setting of Port 12
        */
/*---------------------------------------------------------*/
        P12 =   0b00000000;     /* Set P120 Output latch to Low */
        PM12 =  0b11111110;     /* Set P120 to output port */
                                        /* P120-P124:Unused */
                                        /* *P121-P124:Input port */

/*---------------------------------------------------------*/
/*      Setting of Port 13
        */
/*---------------------------------------------------------*/
        P13 =   0b00000000;     /* Set P130 Output latch to Low */
                                        /* P130:Unused */

/*---------------------------------------------------------*/
/*      Setting of Port 14
        */
/*---------------------------------------------------------*/
        P14 =   0b00000000;     /* Set P140-P147 Output latch to Low */
        PM14 =  0b00000000;     /* Set P140-P147 to output port */
                                        /* P140-P147:Unused */
```

```
/*------------------------------------------------------------*/
/*      Setting of Port 15
        */
/*------------------------------------------------------------*/
        P15 =   0b00000000;         /* Set P150-P152,P157 Output latch to Low */
        PM15 =  0b11111111;         /* Set P150-P152,P157 to input port */
                                                /* P150-P152,P157:Unused */

}

/*----------------------------------------------------------------*/
/* Module:          fn_InitTau0
                            */
/* Description:     Setting of Timer array unit 0                                  */
/*      parameter: --
                            */
/*      return     : --
                            */
/*----------------------------------------------------------------*/
void    fn_InitTau0(void)
{
        TAU0EN = 1;                             /* supplies input clock to timer array unit 0 */
        TPS0L = 0b00000010;         /* Timer Clock Select Register 0 */
                            /*||||++++--- : Selection of operation clock (CK00) */
                            /*++++------- : Selection of operation clock (CK01) */
                            /* : 0 0 0 0 : CK0m = fCLK */
                            /* : 0 0 0 1 : CK0m = fCLK/2 */
                            /* : 0 0 1 0 : CK0m = fCLK/2^2 */
                            /* : 0 0 1 1 : CK0m = fCLK/2^3 */
                            /* : 0 1 0 0 : CK0m = fCLK/2^4 */
                            /* : 0 1 0 1 : CK0m = fCLK/2^5 */
                            /* : 0 1 1 0 : CK0m = fCLK/2^6 */
                            /* : 0 1 1 1 : CK0m = fCLK/2^7 */
                            /* : 1 0 0 0 : CK0m = fCLK/2^8 */
                            /* : 1 0 0 1 : CK0m = fCLK/2^9 */
                            /* : 1 0 1 0 : CK0m = fCLK/2^10 */
                            /* : 1 0 1 1 : CK0m = fCLK/2^11 */
                            /* : 1 1 0 0 : CK0m = fCLK/2^12 */
                            /* : 1 1 0 1 : CK0m = fCLK/2^13 */
```

```
                    /* : 1 1 1 0 : CK0m = fCLK/2^14 */
                    /* : 1 1 1 1 : CK0m = fCLK/2^15 */
                    /*    m = 0, 1 */

/* CH1:for wait */
TMR01 = 0b0000000000000000;         /* Timer Mode Register 01 */
                    /*||||||||||||++++--- : [1]Operation mode of channel 1 */
                    /*||||||||||||          [2]Count operation of TCR */
                    /*||||||||||||          [3]Independent operation */
                    /*||||||||||||          [4]Setting of starting counting and interrupt */
                    /*|||||||||||| : 0 0 0 0 : [1]Interval timer mode */
                    /*||||||||||||             [2]Counting down */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                (timer output does not change, either). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 0 0 1 : [1]Interval timer mode */
                    /*||||||||||||             [2]Counting down */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is generated when counting is started */
                    /*||||||||||||                (timer output also changes). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 0 0 : [1]Capture mode */
                    /*||||||||||||             [2]Counting up */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                (timer output does not change, either). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 0 1 : [1]Capture mode */
                    /*||||||||||||             [2]Counting up */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is generated when counting is started */
                    /*||||||||||||                (timer output also changes). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 1 0 : [1]Event counter mode */
                    /*||||||||||||             [2]Counting down */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                (timer output does not change, either). */
```

```
*/
/*|||||||||||| */
/*|||||||||||| : 1 0 0 0 : [1]One-count mode */
/*||||||||||||            [2]Counting down */
/*||||||||||||            [3]Impossible */
/*||||||||||||            [4]Start trigger is invalid during counting operation. */
/*||||||||||||               At that time, interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 1 : [1]One-count mode */
/*||||||||||||            [2]Counting down */
/*||||||||||||            [3]Impossible */
/*||||||||||||            [4]Start trigger is valid during counting operation. */
/*||||||||||||               At that time, interrupt is also generated. */
/*|||||||||||| */
/*|||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*||||||||||||            [2]Counting up */
/*||||||||||||            [3]Possible */
/*||||||||||||            [4]Timer interrupt is not generated when counting is started */
/*||||||||||||               (timer output does not change, either). */
/*||||||||||||               Start trigger is invalid during counting operation. */
/*||||||||||||               At that time interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : Other than above : Setting prohibited */
/*|||||||||||| */
/*||||||||||++------- : Be sure to set 00 */
/*|||||||||| */
/*||||||||++--------- : Selection of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 valid
edge */
/*||||||||            (the timer input used with channel 1 is selected by using TIS0
register). */
/*|||||||| : 0 0 : Falling edge */
/*|||||||| : 0 1 : Rising edge */
/*|||||||| : 1 0 : Both edges (when low-level width is measured) */
/*||||||||         Start trigger: Falling edge, Capture trigger: Rising edge */
/*|||||||| : 1 1 : Both edges (when high-level width is measured) */
/*||||||||         Start trigger: Rising edge, Capture trigger: Falling edge */
/*|||||||| */
/*|||||+++----------- : Setting of start trigger or capture trigger of channel 1 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are unselected).
*/
```

```
                    /*||||| : 0 0 1 : Valid edge of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 is used
as both the start trigger and capture trigger. */
                    /*||||| : 0 1 0 : Both the edges of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 are
used as a start trigger and a capture trigger. */
                    /*||||| : 1 0 0 : Interrupt signal of the master channel is used (when the channel is used
as a slave channel with the combination operation function). */
                    /*||||| : Other than above : Setting prohibited */
                    /*||||| */
                    /*||||+-------------- : Selection of slave/master of channel 1 */
                    /*|||| : 0 : Operates as slave channel with combination operation function. */
                    /*|||| : 1 : Operates as master channel with combination operation function. */
                    /*|||| */
                    /*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
                    /*||| : 0 : Operation clock MCK specified by CKS01 bit */
                    /*||| : 1 : Valid edge of input signal input from TI01 pin, fSUB/2, fSUB/4, or INTRTC1 */
                    /*|||       (the timer input used with channel 1 is selected by using TIS0 register). */
                    /*||| */
                    /*|++---------------- : Be sure to set 00 */
                    /*| */
                    /*+------------------ : Selection of operation clock (MCK) of channel 1 */
                    /* : 0 : Operation clock CK00 set by TPS0 register */
                    /* : 1 : Operation clock CK01 set by TPS0 register */
    TDR01 = CCK00_500USEC;          /* set interval time to 500us */
    TMMK01 = 1;                                 /* disable interrupt */

    /* CH4:for DMA */
    TMR04 = 0b1000000000000000;     /* Timer Mode Register 04 */
                    /*||||||||||||++++--- : [1]Operation mode of channel 4 */
                    /*||||||||||||          [2]Count operation of TCR */
                    /*||||||||||||          [3]Independent operation */
                    /*||||||||||||          [4]Setting of starting counting and interrupt */
                    /*|||||||||||| : 0 0 0 0 : [1]Interval timer mode */
                    /*||||||||||||              [2]Counting down */
                    /*||||||||||||              [3]Possible */
                    /*||||||||||||              [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                 (timer output does not change, either). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 0 0 1 : [1]Interval timer mode */
                    /*||||||||||||              [2]Counting down */
```

```
/*||||||||||||                    [3]Possible */
/*||||||||||||                    [4]Timer interrupt is generated when counting is started */
/*||||||||||||                         (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 0 0 : [1]Capture mode */
/*||||||||||||                    [2]Counting up */
/*||||||||||||                    [3]Possible */
/*||||||||||||                    [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                         (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 0 1 0 1 : [1]Capture mode */
/*||||||||||||                    [2]Counting up */
/*||||||||||||                    [3]Possible */
/*||||||||||||                    [4]Timer interrupt is generated when counting is started */
/*||||||||||||                         (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 1 0 : [1]Event counter mode */
/*||||||||||||                    [2]Counting down */
/*||||||||||||                    [3]Possible */
/*||||||||||||                    [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                         (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 0 : [1]One-count mode */
/*||||||||||||                    [2]Counting down */
/*||||||||||||                    [3]Impossible */
/*||||||||||||                    [4]Start trigger is invalid during counting operation. */
/*||||||||||||                         At that time, interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 1 : [1]One-count mode */
/*||||||||||||                    [2]Counting down */
/*||||||||||||                    [3]Impossible */
/*||||||||||||                    [4]Start trigger is valid during counting operation. */
/*||||||||||||                         At that time, interrupt is also generated. */
/*|||||||||||| */
/*|||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*||||||||||||                    [2]Counting up */
/*||||||||||||                    [3]Possible */
/*||||||||||||                    [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                         (timer output does not change, either). */
```

```
/*||||||||||||                    Start trigger is invalid during counting operation. */
/*||||||||||||                    At that time interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : Other than above : Setting prohibited */
/*|||||||||||| */
/*||||||||||++------- : Be sure to set 00 */
/*|||||||||| */
/*||||||||++--------- : Selection of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 valid edge */
/*||||||||                  (the timer input used with channel 4 is selected by using TIS0 register). */
/*|||||||| : 0 0 : Falling edge */
/*|||||||| : 0 1 : Rising edge */
/*|||||||| : 1 0 : Both edges (when low-level width is measured) */
/*||||||||           Start trigger: Falling edge, Capture trigger: Rising edge */
/*|||||||| : 1 1 : Both edges (when high-level width is measured) */
/*||||||||           Start trigger: Rising edge, Capture trigger: Falling edge */
/*|||||||| */
/*|||||+++----------- : Setting of start trigger or capture trigger of channel 4 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are unselected). */
/*||||| : 0 0 1 : Valid edge of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 is used as both the start trigger and capture trigger. */
/*||||| : 0 1 0 : Both the edges of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 are used as a start trigger and a capture trigger. */
/*||||| : 1 0 0 : Interrupt signal of the master channel is used (when the channel is used as a slave channel with the combination operation function). */
/*||||| : Other than above : Setting prohibited */
/*||||| */
/*||||+-------------- : Selection of slave/master of channel 4 */
/*|||| : 0 : Operates as slave channel with combination operation function. */
/*|||| : 1 : Operates as master channel with combination operation function. */
/*|||| */
/*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
/*||| : 0 : Operation clock MCK specified by CKS04 bit */
/*||| : 1 : Valid edge of input signal input from TI04 pin, fSUB/2, fSUB/4, or INTRTC1 */
/*|||        (the timer input used with channel 4 is selected by using TIS0 register). */
/*||| */
/*|++---------------- : Be sure to set 00 */
```

```
                            /*| */
                            /*+------------------ : Selection of operation clock (MCK) of channel 4 */
                            /* : 0 : Operation clock CK00 set by TPS0 register */
                            /* : 1 : Operation clock CK01 set by TPS0 register */
        TDR04 = 2000;       /* set interval time to 100us(=0.05us * 2000) */
        TMMK04 = 1;                 /* disable interrupt */

}

/*-------------------------------------------------------------*/
/* Module:          fn_InitLvi
                    */
/* Description:     Setting of Low-voltage detectior                          */
/*      parameter: --
                            */
/*      return     : --
                            */
/*-------------------------------------------------------------*/
void    fn_InitLvi(void)
{
        unsigned short      loop;       /* waiting counter */

        LVIMK = 1;                          /* disable LVI interrupt */

        LVIS =      0b00001001;             /* Low-Voltage Detection Level Select Register */
                            /*||||++++--- : Detection level */
                            /*|||| : 0 0 0 0 : VLVI0 (4.22V) */
                            /*|||| : 0 0 0 1 : VLVI1 (4.07V) */
                            /*|||| : 0 0 1 0 : VLVI2 (3.92V) */
                            /*|||| : 0 0 1 1 : VLVI3 (3.76V) */
                            /*|||| : 0 1 0 0 : VLVI4 (3.61V) */
                            /*|||| : 0 1 0 1 : VLVI5 (3.45V) */
                            /*|||| : 0 1 1 0 : VLVI6 (3.30V) */
                            /*|||| : 0 1 1 1 : VLVI7 (3.15V) */
                            /*|||| : 1 0 0 0 : VLVI8 (2.99V) */
                            /*|||| : 1 0 0 1 : VLVI9 (2.84V) */
                            /*|||| : 1 0 1 0 : VLVI10 (2.68V) */
                            /*|||| : 1 0 1 1 : VLVI11 (2.53V) */
                            /*|||| : 1 1 0 0 : VLVI12 (2.38V) */
```

```
                        /*|||| : 1 1 0 1 : VLVI13 (2.22V) */
                        /*|||| : 1 1 1 0 : VLVI14 (2.07V) */
                        /*|||| : 1 1 1 1 : VLVI15 (1.91V) */
                        /*|||| */
                        /*++++------- : Be sure to set 0000 */

    LVIM =   0b10000000;            /* Low-Voltage Detection Register */
                        /*|||||||+--- : LVIF Low-voltage detection flag */
                        /*||||||| : 0 : * LVISEL = 0: VDD >= VLVI, or when LVI operation is disabled */
                        /*|||||||        * LVISEL = 1: EXLVI >= VEXLVI, or when LVI operation is disabled */
                        /*||||||| : 1 : * LVISEL = 0: VDD < VLVI */
                        /*|||||||        * LVISEL = 1: EXLVI < VEXLVI */
                        /*||||||| */
                        /*||||||+---- : Low-voltage detection operation mode (interrupt/reset) selection(LVIMD) */
                        /*|||||| : 0 : * LVISEL = 0: Generates an internal interrupt signal */
                        /*||||||                  when VDD drops lower than VLVI (VDD < VLVI) */
                        /*||||||                  or when VDD becomes VLVI or higher (VDD >= VLVI).
*/
                        /*||||||        * LVISEL = 1: Generates an interrupt signal */
                        /*||||||                  when EXLVI drops lower than VEXLVI (EXLVI <
VEXLVI) */
                        /*||||||                  or when EXLVI becomes VEXLVI or higher (EXLVI >=
VEXLVI). */
                        /*|||||| : 1 : * LVISEL = 0: Generates an internal reset signal when VDD < VLVI */
                        /*||||||                  and releases the reset signal when VDD >= VLVI. */
                        /*||||||        * LVISEL = 1: Generates an internal reset signal when EXLVI < VEXLVI
*/
                        /*||||||                  and releases the reset signal when EXLVI >= VEXLVI.
*/
                        /*|||||| */
                        /*|||||+----- : Voltage detection selection(LVISEL) */
                        /*||||| : 0 : Detects level of supply voltage (VDD) */
                        /*||||| : 1 : Detects level of input voltage from external input pin (EXLVI) */
                        /*||||| */
                        /*|++++------ : Be sure to set 0000 */
                        /*| */
                        /*+---------- : Enables low-voltage detection operation */
                        /* : 0 : Disables operation */
                        /* : 1 : Enables operation */
```

```
        /* software to wait for the operation stabilization time (210us) */
        for(loop = 500; loop > 0; loop--){
                NOP();
        }
        /* wait until VLVI≤VDD */
        while( LVIF ){
                NOP();
        }

        LVIIF = 0;                      /* clear LVI interrupt request flag */
}



/*------------------------------------------------------------*/
/* Module:      fn_DmaInit
                        */
/* Description:     Setting of DMA contoroler                                    */
/*      parameter: --
                        */
/*      return     : --
                        */
/*------------------------------------------------------------*/
void    fn_DmaInit(void)
{

        DEN0 = 1;                                   /* Enables operation of DMA channel 0 */

        DSA0 = (unsigned char)&DACSW0;                            /* DMA SFR Address Register 0
*/
        DRA0 = (unsigned short)&ushDacData[0];      /* DMA RAM Address Register 0 */
/*      DBC0 = 100;                                         /* DMA Byte Count Register 0 */
        DBC0 = 25;                                          /* DMA Byte Count Register 0 */
        DMC0 = 0b01100100;                          /* DMA Mode Control Register 0 */
                    /*|||||||| */
                    /*||||++++------- IFC3-IFC0 */
                    /*||||            [Selection of DMA start source] */
                    /*||||             0000:Disables DMA transfer by interrupt */
                    /*||||                          (Only software trigger is enabled.) */
```

```
                    /*||||                  0010:INTTM00 */
                    /*||||                  0011:INTTM01 */
                    /*||||                  0100:INTTM04 */
                    /*||||                  0101:INTTM05 */
                    /*||||                  0110:INTST0/INTCSI00 */
                    /*||||                  0111:INTSR0/INTCSI01 */
                    /*||||                  1000:INTST1/INTCSI10/INTIIC10 */
                    /*||||                  1001:INTSR1 */
                    /*||||                  1010:INTST3 */
                    /*||||                  1011:INTSR3 */
                    /*||||                  1100:INTAD */
                    /*|||+----------- DWAIT0 */
                    /*|||                  [Pending of DMA transfer] */
                    /*|||                   0:Executes DMA transfer upon DMA start request (not held pending). */
                    /*|||                   1:Holds DMA start request pending if any. */
                    /*||+------------ DS0 */
                    /*||                   [Specification of transfer data size for DMA transfer] */
                    /*||                    0:8bits */
                    /*||                    1:16bits */
                    /*|+------------- DRS0 */
                    /*|                    [Selection of DMA transfer direction] */
                    /*|                     0:SFR to internal RAM */
                    /*|                     1:Internal RAM to SFR */
                    /*+------------- STG0 */
                    /*                     [DMA transfer start software trigger] */
                    /*                      0:No trigger operation */
                    /*                      1:DMA transfer is started when DMA operation is enabled (DEN0 = 1). */

        DST0 = 1;                   /* DMA transfer is started when DMA operation is enabled (DEN0 = 1). */

}


/*--------------------------------------------------------------*/
/* Module:          fn_Wait500usBase
        */
/* Description:     Delays the program for (Time * 500us)                        */
/*      parameter: wait time(Time)
        */
```

```
/*        return      : --
                               */
/*--------------------------------------------------------------*/
void    fn_Wait500usBase(unsigned short Time)
{
        TS0L.1 = 1;                             /* start TAU0 CH1 */
        TMIF01 = 0;

        for(; Time > 0; Time--){          /* wait for (parameter * 500)us */
                while(!TMIF01){
                        NOP();
                }
                TMIF01 = 0;
        }

        TT0L.1 = 1;                             /* stop TAU0 CH1 */
}



/*--------------------------------------------------------------*/
/* Module:          fn_AdcDacInit
                    */
/* Description:     Initialization of A/D D/A OPAmp module                      */
/*        parameter: --
                               */
/*        return      : --
                               */
/*--------------------------------------------------------------*/
void    fn_AdcDacInit(void)
{
        ADCEN = 1;                              /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

        /*-------------------------------------------*/
        /*   Initialization of Voltage reference                      */
        /*-------------------------------------------*/
        ADVRC = 0b00001000;           /* Analog reference voltage control register */
                               /*||||+|++--- : [1]Positive reference voltage supplies selection of A/D and D/A
converters */
```

```
/*|||| |             [2]Operationcontrol of voltage reference */
/*|||| |             [3]Output voltage selection of voltage reference */
/*|||| |             [4]Operation control of input gate voltage boost circuit for A/D converter
*/
/*|||| |             [5]Relationship with the conversion mode used */
/*|||| | */
/*|||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
/*|||| |                 [2]Stops operation (Hi-Z) */
/*|||| |                 [3]2.5 V */
/*|||| |                 [4]Stops operation */
/*|||| |                 [5]Can be set in conversion mode 1 */
/*|||| | */
/*|||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
/*|||| |                 [2]Stops operation (Hi-Z) */
/*|||| |                 [3]2.0 V */
/*|||| |                 [4]Enables operation */
/*|||| |                 [5]Can be set in conversion mode 2 or 3 */
/*|||| | */
/*|||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
/*|||| |                 [2]Stops operation (pull-down output) */
/*|||| |                 [3]2.5 V */
/*|||| |                 [4]Stops operation */
/*|||| |                 [5] - */
/*|||| | */
/*|||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
/*|||| |                 [2]Enables operation */
/*|||| |                 [3]2.5 V */
/*|||| |                 [4]Enables operation */
/*|||| |                 [5]Can be set in conversion mode 2 or 3 */
/*|||| | */
/*|||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
/*|||| |                 [2]Stops operation (pull-down output) */
/*|||| |                 [3]2.0 V */
/*|||| |                 [4]Enables operation */
/*|||| |                 [5] - */
/*|||| | */
/*|||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
/*|||| |                 [2]Enables operation */
/*|||| |                 [3]2.0 V */
```

```
                    /*|||| |            [4]Enables operation */
                    /*|||| |            [5]Can be set in conversion mode 2 or 3 */
                    /*|||| | */
                    /*|||| | : Other than the above : Setting prohibited */
                    /*|||| | */
                    /*|+++-+----- : Be sure to set 0000 */
                    /*| */
                    /*+---------- : Reference voltage supply (negative side) of A/D converter selection */
                    /* : 0 : AVSS */
                    /* : 1 : AVREFM (external voltage reference input) */

ADVRC.0 = 1;            /* Enables operation */
ADVRC.1 = 1;            /* Output 2.0 V */

ADPC =   0b00000000;    /* A/D Port Configuration Register */
                    /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                    /*||| :              +------------------------ ANI15/AVREFM/P157 */
                    /*||| :              | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                    /*||| :              | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
                    /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                    /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                    /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                    /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                    /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                    /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                    /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                    /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                    /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                    /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                    /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                    /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                    /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                    /*||| */
                    /*+++-------- : Be sure to set 000 */

/* wait for settling time to 17msec */
fn_Wait500usBase(17000/500);

/*------------------------------------------*/
```

```
        /*   Initialization of operational amplifier   */
        /*------------------------------------------*/
        PM2.1 = 1;                          /* input port mode (AMP0O) */
        PM2.0 = 1;                          /* input port mode (AMP0-) */
        PM2.2 = 1;                          /* input port mode (AMP0+) */

        PM2.4 = 1;                          /* input port mode (AMP1O) */
        PM2.3 = 1;                          /* input port mode (AMP1-) */
        PM2.5 = 1;                          /* input port mode (AMP1+) */

        PM2.7 = 1;                          /* input port mode (AMP2O) */
        PM2.6 = 1;                          /* input port mode (AMP2-) */
        PM15.0 = 1;                         /* input port mode (AMP2+) */

        PM11.0 = 1;                         /* input port mode (ANO0) */

        ADCEN = 1;                          /* supplies input clock to OPAmp A/D converter */
        OAC.0 = 1;                          /* operational amplifier (AMP0) enable */
        OAC.1 = 0;                          /* operational amplifier (AMP1) disable */
        OAC.2 = 0;                          /* operational amplifier (AMP2) disable */

        /*------------------------------------------*/
        /*        Initialization of D/A converter        */
        /*------------------------------------------*/
        DACEN = 1;                                   /* supplies input clock to D/A converter */

        DAM.0 = 1;                                   /* [DAMD0] D/A converter operation is Real-time
output mode */
        DAM.2 = 1;                                   /* [DARES0]D/A conversion resolution is 12-bit */
        DAM.6 = 1;                                   /* [DAREF] D/A converter positive reference voltage
is VREFOUT */
        DACSW0 = aushDacData[99];           /* D/A conversion value setting reigster 0 */

  }

  /*------------------------------------------------------------*/
  /* Module:          main
                                  */
  /* Description:     Main process
```

```
        */
/*      parameter: --
                          */
/*      return    : --
                          */
/*-------------------------------------------------------------*/
void main(void)
{
        unsigned short    count;          /* counter for initializing D/A convert data */

/*      ushCycleCount = 0;                /* Cycle counter */
        ucCycleStatus = 0;                /* Cycle status */

/*      DACE1 = 1;                                        /* D/A converter CH1 enable */
        DACE0 = 1;                                        /* D/A converter CH0 enable */



        /*===================================*/
        /*---------------------------------*/
        /*              Main Loop           */
        /*---------------------------------*/
        /*===================================*/

        /*  initialize D/Aconverter data */
        for(count = 0; count<100; count++){
                ushDacData[count] = aushDacData[count];
        }

        TS0L.4 = 1;                       /* start TAU0 CH4 */
        TMIF04 = 0;                       /* reset INTTM04 interrupt request */

        DMAMK0 = 0;                       /* enable DMA CH0 interrupt */

        EI();                             /* enable all interrupts */

        /*==============================================================*/
        /* if system have nothing to do, go to standby for power-saving */
        /*==============================================================*/
```

```
	while (1){
		HALT();
		NOP();
	}
}


/*------------------------------------------------------------*/
/* Module:		fn_intdma0
				*/
/* Description:	DMA ch0 interruput process						*/
/*	parameter: --
				*/
/*	return	: --
				*/
/*------------------------------------------------------------*/
__interrupt void fn_intdma0(void)
{
	TT0L.4 = 1;					/* stop TAU0 CH4 */


	switch(ucCycleStatus){				/* Cycle status */
		case	0:
				TDR04 = 20000;			/*	set	interval	time	to
1000us(=0.05us * 20000) */
				ucCycleStatus = 1;		/* Cycle status (Flat) */
/*				DRA0 = (unsigned short)&ushDacData[0];		/* DMA RAM Address
Register 0 */
				DBC0 = 1;				/*	DMA	Byte	Count
Register 0 */
				break;

		case	1:
				TDR04 = 200;			/*	set	interval	time	to
10us(=0.05us * 200) */
				ucCycleStatus = 2;		/* Cycle status (2/4,3/4 sign) */
				DBC0 = 50 - 1;			/* DMA Byte Count Register 0 */
				break;
```

```
                    case    2:
                                        TDR04 = 20000;                          /*  set  interval  time  to
1000us(=0.05us * 20000) */
                                        ucCycleStatus = 3;          /* Cycle status (Flat) */
                                        DBC0 = 1;                                          /* DMA Byte Count
Register 0 */
                                        break;

                    case    3:
                                        TDR04 = 200;                            /*  set  interval  time  to
10us(=0.05us * 200) */
                                        ucCycleStatus = 4;          /* Cycle status (1kHz) */
                                        DBC0 = 25 - 1;                          /* DMA Byte Count Register 0 */
                                        break;

                    case    4:
                                        DRA0 = (unsigned short)&ushDacData[0];          /* DMA RAM Address
Register 0 */
                                        DACSW0 = ushDacData[99];                /* D/A conversion value setting
reigster 0 */
                                        TMR04 = 0b0000000000000000;             /* Timer Mode Register 04 */
                                        TDR04 = 20000;                          /*  set  interval  time  to
1000us(=0.05us * 20000) */
                                        ucCycleStatus = 5;          /* Cycle status (10Hz) */
                                        DBC0 = 50;                                         /* DMA Byte Count
Register 0 */
                                        break;

                    case    5:
                                        DRA0 = (unsigned short)&ushDacData[0];          /* DMA RAM Address
Register 0 */
                                        DACSW0 = ushDacData[99];                /* D/A conversion value setting
reigster 0 */
                                        TMR04 = 0b1000000000000000;             /* Timer Mode Register 04 */
                                        TDR04 = 200;                            /*  set  interval  time  to
10us(=0.05us * 200) */
                                        ucCycleStatus = 0;          /* Cycle status (1kHz) */
                                        DBC0 = 25;                                                  /*  DMA
```

```
Byte Count Register 0 */
                                        break;

        }

  /*    DBC0 = 100;                                                 /* DMA Byte Count Register 0 */

        DST0 = 1;                                           /* DMA transfer is started when DMA operation is
enabled (DEN0 = 1). */

        TS0L.4 = 1;                         /* start TAU0 CH4 */
        TMIF04 = 0;                         /* reset INTTM04 interrupt request */

  }
```

(6) Inverting amplifier circuit (AC)

● main.c

```
/*
 * Copyright (C) NEC Electronics Corporation 2006
 * NEC ELECTRONICS CONFIDENTIAL AND PROPRIETARY
 * All rights reserved by NEC Electronics Corporation.
 * This program must be used solely for the purpose for which
 * it was furnished by NEC Electronics Corporation. No part of this
 * program may be reproduced or disclosed to others, in any
 * form, without the prior written permission of NEC Electronics
 * Corporation. Use of copyright notice dose not evidence
 * publication of the program.
 */

/*--------------------------------------------------------------------*/
/* #pragma directive for CC78K0
                                   */
/*--------------------------------------------------------------------*/
#pragma         SFR
#pragma         DI
#pragma         EI
#pragma         HALT
#pragma         NOP

#pragma interrupt INTDMA0 fn_intdma0 RB1

/*--------------------------------------------------------------------*/
/* Include files
                                                   */
/*--------------------------------------------------------------------*/
/* TAU:TDR0n value operation by CK00 (fCLK/2^3 = 5MHz) */
#define CCK00_500USEC   (2500 - 1)          /* 500us (0.2[us/clk] * 2500[count]) */



/*--------------------------------------------------------------------*/
/* Function prototyps
                                                   */
/*--------------------------------------------------------------------*/
```

```
                void    fn_Wait500usBase(unsigned short);       /* Delays the program for (Time * 500us) */
                void    fn_InitPort(void);                                          /* Setting of I/O ports
*/
                void    fn_InitTau0(void);                                          /* Setting of Timer
array unit 0 */
                void    fn_InitLvi(void);                                           /* Setting of
Low-voltage detectior */
                void    fn_InitVr(void);                                            /* Setting of Voltage
reference */
                void    fn_AdcDacInit(void);                              /* Setting of D/A converter */
                void    fn_DmaInit(void);                                           /* Setting of DMA
controler */


  /*-----------------------------------------------------------------------*/
  /* Extern variables/constants
                */
  /*-----------------------------------------------------------------------*/

  /*-----------------------------------------------------------------------*/
  /* Local constants
                                */
  /*-----------------------------------------------------------------------*/
  /*-----------------------------------------------------------------------*/
  /* Global variables
                                */
  /*-----------------------------------------------------------------------*/

  /*-----------------------------------------------------------------------*/
  /* Local variables
                                */
  /*-----------------------------------------------------------------------*/
  static   short         shVoltage;/* voltage (Hex) */
  static   unsigned short      ushCycleCount;      /* Cycle counter */
  static   unsigned char       ucCycleStatus;      /* Cycle status */

  /*-----------------------------------------------------------------------*/
  /* Code
                                                */
```

```
/*-----------------------------------------------------------------------*/
/*=============================================================================

        D/A output data

=============================================================================*/
static unsigned short ushDacData[100];          /* D/A data for DMA */
const unsigned short aushDacData[100] = {       /* defined D/A data */
                                                /* n      sin(0.2pi*n/100) */
/*                 2048,              /* 0      0.100000 */
                   2061,              /* 1      0.106279 */
                   2074,              /* 2      0.112533 */
                   2086,              /* 3      0.118738 */
                   2099,              /* 4      0.124869 */
                   2111,              /* 5      0.130902 */
                   2123,              /* 6      0.136812 */
                   2135,              /* 7      0.142578 */
                   2147,              /* 8      0.148175 */
                   2158,              /* 9      0.153583 */
                   2168,              /* 10     0.158779 */
                   2179,              /* 11     0.163742 */
                   2188,              /* 12     0.168455 */
                   2197,              /* 13     0.172897 */
                   2206,              /* 14     0.177051 */
                   2214,              /* 15     0.180902 */
                   2221,              /* 16     0.184433 */
                   2227,              /* 17     0.187631 */
                   2233,              /* 18     0.190483 */
                   2238,              /* 19     0.192978 */
                   2243,              /* 20     0.195106 */
                   2246,              /* 21     0.196858 */
                   2249,              /* 22     0.198229 */
                   2251,              /* 23     0.199211 */
                   2252,              /* 24     0.199803 */
                   2253,              /* 25     0.200000 */
                   2252,              /* 26     0.199803 */
                   2251,              /* 27     0.199211 */
                   2249,              /* 28     0.198229 */
                   2246,              /* 29     0.196858 */
```

```
2243,    /* 30    0.195106 */
2238,    /* 31    0.192978 */
2233,    /* 32    0.190483 */
2227,    /* 33    0.187631 */
2221,    /* 34    0.184433 */
2214,    /* 35    0.180902 */
2206,    /* 36    0.177051 */
2197,    /* 37    0.172897 */
2188,    /* 38    0.168455 */
2179,    /* 39    0.163742 */
2168,    /* 40    0.158779 */
2158,    /* 41    0.153583 */
2147,    /* 42    0.148175 */
2135,    /* 43    0.142578 */
2123,    /* 44    0.136812 */
2111,    /* 45    0.130902 */
2099,    /* 46    0.124869 */
2086,    /* 47    0.118738 */
2074,    /* 48    0.112533 */
2061,    /* 49    0.106279 */
2048,    /* 50    0.100000 */
2035,    /* 51    0.093721 */
2022,    /* 52    0.087467 */
2010,    /* 53    0.081262 */
1997,    /* 54    0.075131 */
1985,    /* 55    0.069098 */
1973,    /* 56    0.063188 */
1961,    /* 57    0.057422 */
1949,    /* 58    0.051825 */
1938,    /* 59    0.046417 */
1928,    /* 60    0.041221 */
1917,    /* 61    0.036258 */
1908,    /* 62    0.031545 */
1899,    /* 63    0.027103 */
1890,    /* 64    0.022949 */
1882,    /* 65    0.019098 */
1875,    /* 66    0.015567 */
1869,    /* 67    0.012369 */
1863,    /* 68    0.009517 */
```

```
        1858,           /* 69   0.007022 */
        1853,           /* 70   0.004894 */
        1850,           /* 71   0.003142 */
        1847,           /* 72   0.001771 */
        1845,           /* 73   0.000789 */
        1844,           /* 74   0.000197 */
        1843,           /* 75   0.000000 */
        1844,           /* 76   0.000197 */
        1845,           /* 77   0.000789 */
        1847,           /* 78   0.001771 */
        1850,           /* 79   0.003142 */
        1853,           /* 80   0.004894 */
        1858,           /* 81   0.007022 */
        1863,           /* 82   0.009517 */
        1869,           /* 83   0.012369 */
        1875,           /* 84   0.015567 */
        1882,           /* 85   0.019098 */
        1890,           /* 86   0.022949 */
        1899,           /* 87   0.027103 */
        1908,           /* 88   0.031545 */
        1917,           /* 89   0.036258 */
        1928,           /* 90   0.041221 */
        1938,           /* 91   0.046417 */
        1949,           /* 92   0.051825 */
        1961,           /* 93   0.057422 */
        1973,           /* 94   0.063188 */
        1985,           /* 95   0.069098 */
        1997,           /* 96   0.075131 */
        2010,           /* 97   0.081262 */
        2022,           /* 98   0.087467 */
        2035,           /* 99   0.093721 */
        2048            /* 100  0.100000 */

};

/*--------------------------------------------------------------------*/
/* Hardware initialization
                                    */
/*--------------------------------------------------------------------*/
```

```
void hdwinit(void)
{
        DI();                                   /* disable all interrupts */

        /*-----------------------------------------*/
        /*              Initialization of port              */
        /*-----------------------------------------*/
        fn_InitPort();

        /*----------------------------------------*/
        /*            Initialization of clock              */
        /*----------------------------------------*/
        CMC =     0b01010011;                   /* Clock Operation Mode Control Register */
                            /*|||||||+--- : Control of high-speed system clock oscillation frequency */
                            /*||||||| : 0 : 2 MHz <= fMX <= 10 MHz */
                            /*||||||| : 1 : 10 MHz < fMX <= 20 MHz */
                            /*||||||| */
                            /*|||||++---- : XT1 oscillator oscillation mode selection */
                            /*||||| : 0 0 : Low-consumption oscillation */
                            /*||||| : 0 1 : Normal oscillation */
                            /*||||| : 1 x : Super-low-consumption oscillation */
                            /*|||||    x = don't care */
                            /*||||| */
                            /*||||+------ : Be sure to set 0 */
                            /*|||| */
                            /*|||+------- : [1] Subsystem clock pin operation mode */
                            /*|||                [2] XT1/P123 pin and XT2/P124 pin */
                            /*||| : 0 : [1]Input port mode */
                            /*|||          [2]Input port */
                            /*||| */
                            /*||| : 1 : [1]XT1 oscillation mode */
                            /*|||          [2]Crystal resonator connection */
                            /*||| */
                            /*||+-------- : Be sure to set 0 */
                            /*|| */
                            /*++--------- : [1]EXCLK OSCSEL High-speed system clock pin operation mode */
                            /*                     [2]X1/P121 pin */
                            /*                     [3]X2/EXCLK/P122 pin */
                            /* : 0 0 : [1]Input port mode */
```

```
                    /*              [2][3]Input port */
                    /*                                */
                    /* : 0 1 : [1]X1 oscillation mode */
                    /*              [2][3]Crystal/ceramic resonator connection */
                    /*                                */
                    /* : 1 0 : [1]Input port mode */
                    /*              [2][3]Input port */
                    /*                                */
                    /* : 1 1 : [1]External clock input mode */
                    /*              [2]Input port */
                    /*              [3]External clock input */

    MSTOP = 0;                          /* X1 oscillator operating */
    XTSTOP = 0;                         /* XT1 oscillator operating */

    OSMC =  0b00000001;                 /* Operation Speed Mode Control Register */
                    /*||||||++--- : fCLK frequency selection */
                    /*|||||| : 0 0 : Operates at a frequency of 10 MHz or less. */
                    /*|||||| : 0 1 : Operates at a frequency higher than 10 MHz. */
                    /*|||||| : 1 0 : Operates at a frequency of 1 MHz. */
                    /*||||||                                  */
                    /*|+++++----- : Be sure to set 00000 */
                    /*|                                  */
                    /*+---------- : Setting in subsystem clock HALT mode */
                    /* : 0 : Enables subsystem clock supply to peripheral functions. */
                    /*          (See Table 21-1 Operating Statuses in HALT Mode (2/3) */
                    /*           for the peripheral functions whose operations are enabled.) */
                    /* : 1 : Stops subsystem clock supply to peripheral functions except real-time counter,
*/
                    /*           clock output/buzzer output, and LCD controller/driver. */

    while(OSTC.0 != 1){            /* wait X1 oscillation stabilization */
        NOP();
    }

    /*-- Caution ---------------------------------------------*/
    /* To increase fCLK to 10 MHz or higher, set FSEL to '1',    */
    /* then change fCLK after two or more clocks have elapsed. */
    /*-------------------------------------------------------*/
```

```
NOP();
NOP();

CKC =      0b00010000;                 /* System Clock Control Register */
                     /*|+|+++++--- : Selection of CPU/peripheral hardware clock (fCLK) */
                     /*| | : 0 0 x 0 0 0 : fIH */
                     /*| | : 0 0 x 0 0 1 : fIH/2 (default) */
                     /*| | : 0 0 x 0 1 0 : fIH/2^2 */
                     /*| | : 0 0 x 0 1 1 : fIH/2^3 */
                     /*| | : 0 0 x 1 0 0 : fIH/2^4 */
                     /*| | : 0 0 x 1 0 1 : fIH/2^5 */
                     /*| | : 0 1 x 0 0 0 : fMX */
                     /*| | : 0 1 x 0 0 1 : fMX/2 */
                     /*| | : 0 1 x 0 1 0 : fMX/2^2 */
                     /*| | : 0 1 x 0 1 1 : fMX/2^3 */
                     /*| | : 0 1 x 1 0 0 : fMX/2^4 */
                     /*| | : 0 1 x 1 0 1 : fMX/2^5 */
                     /*| | : 1 x 0 x x x : fSUB */
                     /*| | : 1 x 1 x x x : fSUB/2 */
                     /*| | : Other than above : Setting prohibited */
                     /*| |    x = don't care */
                     /*| | */
                     /*| +-------- : Status of Main system clock (fMAIN) */
                     /*| : 0 : Internal high-speed oscillation clock (fIH) */
                     /*| : 1 : High-speed system clock (fMX) */
                     /*| */
                     /*+---------- : Status of CPU/peripheral hardware clock (fCLK) */
                     /* : 0 : Main system clock (fMAIN) */
                     /* : 1 : Subsystem clock (fSUB) */

/* Confirming the CPU clock status */
while((CLS != 0)||(MCS != 1)){
        NOP();
}
/* CPU is operating on a High-speed system clock */
HIOSTOP = 1;                           /* internal high-speed oscillation stopped */

OSTS =     0b00000111;                 /* Oscillation Stabilization Time Select Register */
                     /*||||++++--- : Oscillation stabilization time selection */
```

```
                        /*||||  : 0 0 0 : 2^8/fX */
                        /*||||  : 0 0 1 : 2^9/fX */
                        /*||||  : 0 1 0 : 2^10/fX */
                        /*||||  : 0 1 1 : 2^11/fX */
                        /*||||  : 1 0 0 : 2^13/fX */
                        /*||||  : 1 0 1 : 2^15/fX */
                        /*||||  : 1 1 0 : 2^17/fX */
                        /*||||  : 1 1 1 : 2^18/fX */
                        /*|||| */
                        /*++++------- : Be sure to set 000000 */

        /*----------------------------------------*/
        /*          Initialization of timer          */
        /*----------------------------------------*/
        fn_InitTau0();

        /* software to wait for the operation stabilization time */
        /* (over 200ms from when XT1 enable) */
        fn_Wait500usBase(200000/500);

        /*----------------------------------------*/
        /* Initialization of low-voltage detector */
        /*----------------------------------------*/
        fn_InitLvi();

        /*----------------------------------------*/
        /*   Initialization of A/D,D/A,OPAmp                */
        /*----------------------------------------*/
        fn_AdcDacInit();

        /*----------------------------------------*/
        /*   Initialization of DMA                                  */
        /*----------------------------------------*/
        fn_DmaInit();

        EI();                               /* enable all interrupts */

}
```

```
/*---------------------------------------------------------------*/
/* Module:          fn_InitPort
                    */
/* Description:     Setting of I/O ports                                    */
/*      parameter: --
                              */
/*      return    : --
                              */
/*---------------------------------------------------------------*/
void    fn_InitPort(void)
{
/*-----------------------------------------------------------*/
/*      Ports configuration for digital input and output              */
/*-----------------------------------------------------------*/
        ADPC =  0b00010000;        /* A/D Port Configuration Register */
                              /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                              /*||| :              +------------------------- ANI15/AVREFM/P157 */
                              /*||| :              | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                              /*||| :              | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
                              /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                              /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                              /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                              /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                              /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                              /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                              /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                              /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                              /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                              /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                              /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                              /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                              /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                              /*||| */
                              /*+++-------- : Be sure to set 000 */

/*-----------------------------------------------------------*/
/*      Setting of Port 0
        */
/*-----------------------------------------------------------*/
```

```
        P0 =      0b00000000;       /* Set P00-P02 Output latch to Low */
        PM0 =     0b11111000;       /* Set P00-P02 to output port */
                                                   /* P00-P02:Unused */


/*----------------------------------------------------------*/
/*      Setting of Port 1
        */
/*----------------------------------------------------------*/
        P1 =      0b00000000;       /* Set P10-P17 Output latch to Low */
        PM1 =     0b00000000;       /* Set P10-P17 to output port */
                                                   /* P10-P15:Unused */


/*----------------------------------------------------------*/
/*      Setting of Port 2
        */
/*----------------------------------------------------------*/
        P2 =      0b00000000;       /* Set P20-P27 Output latch to Low */
        PM2 =     0b11111111;       /* Set P20-P27 to input port */
                                                   /* P20-P27:Unused */


/*----------------------------------------------------------*/
/*      Setting of Port 3
        */
/*----------------------------------------------------------*/
        P3 =      0b00001100;       /* Set P30-P31,P34 Output latch to Low */
                                                   /* Set P33,P32 Output latch High */
        PM3 =     0b11100000;       /* Set P30-P34 to output port */
                                                   /* P30-P34:Unused */


/*----------------------------------------------------------*/
/*      Setting of Port 4
        */
/*----------------------------------------------------------*/
        P4 =      0b00000000;       /* Set P40-P41 Output latch to Low */
        PM4 =     0b11111100;       /* Set P40-P41 to output port */
                                                   /* P40-P41:Unused */


/*----------------------------------------------------------*/
/*      Setting of Port 5
```

```
        */
/*----------------------------------------------------------*/
        P5 =    0b00000000;     /* Set P50-P57 Output latch to Low */
        PM5 =   0b11110000;     /* Set P50-P57 to output port */
                                        /* P50-P57:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 6
        */
/*----------------------------------------------------------*/
        P6 =    0b00000000;     /* Set P60-P61 Output latch to Low */
        PM6 =   0b11111100;     /* Set P60-P61 to output port */
                                        /* P60-P61:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 7
        */
/*----------------------------------------------------------*/
        P7 =    0b00000000;     /* Set P70-P77 Output latch to Low */
        PM7 =   0b00000000;     /* Set P70-P77 to output port */
                                        /* P70-P77:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 8
        */
/*----------------------------------------------------------*/
        P8 =    0b00000000;     /* Set P80-P88 Output latch to Low */
        PM8 =   0b00000000;     /* Set P80-P88 to output port */
                                        /* P80-P88:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 9
        */
/*----------------------------------------------------------*/
        P9 =    0b00000000;     /* Set P90-P97 Output latch to Low */
        PM9 =   0b00000000;     /* Set P90-P97 to output port */
                                        /* P90-P97:Unused */

/*----------------------------------------------------------*/
```

```
/*      Setting of Port 10
        */
/*--------------------------------------------------------*/
        P10 =   0b00000000;     /* Set P100-P102 Output latch to Low */
        PM10 =  0b11111000;     /* Set P100-P102 to output port */
                                        /* P100-P102:Unused */

/*--------------------------------------------------------*/
/*      Setting of Port 11
        */
/*--------------------------------------------------------*/
        P11 =   0b00000000;     /* Set P110-P111 Output latch to Low */
        PM11 =  0b11111100;     /* Set P110-P111 to output port */
                                        /* P110-P111:Unused */

/*--------------------------------------------------------*/
/*      Setting of Port 12
        */
/*--------------------------------------------------------*/
        P12 =   0b00000000;     /* Set P120 Output latch to Low */
        PM12 =  0b11111110;     /* Set P120 to output port */
                                        /* P120-P124:Unused */
                                        /* *P121-P124:Input port */

/*--------------------------------------------------------*/
/*      Setting of Port 13
        */
/*--------------------------------------------------------*/
        P13 =   0b00000000;     /* Set P130 Output latch to Low */
                                        /* P130:Unused */

/*--------------------------------------------------------*/
/*      Setting of Port 14
        */
/*--------------------------------------------------------*/
        P14 =   0b00000000;     /* Set P140-P147 Output latch to Low */
        PM14 =  0b00000000;     /* Set P140-P147 to output port */
                                        /* P140-P147:Unused */
```

```
/*----------------------------------------------------------*/
/*      Setting of Port 15
        */
/*----------------------------------------------------------*/
        P15 =   0b00000000;         /* Set P150-P152,P157 Output latch to Low */
        PM15 =  0b11111111;         /* Set P150-P152,P157 to input port */
                                                /* P150-P152,P157:Unused */


}

/*--------------------------------------------------------------*/
/* Module:          fn_InitTau0
                            */
/* Description:     Setting of Timer array unit 0                                */
/*      parameter: --
                            */
/*      return      : --
                            */
/*--------------------------------------------------------------*/
void    fn_InitTau0(void)
{
        TAU0EN = 1;                             /* supplies input clock to timer array unit 0 */
        TPS0L = 0b00000010;         /* Timer Clock Select Register 0 */
                            /*||||++++--- : Selection of operation clock (CK00) */
                            /*++++------- : Selection of operation clock (CK01) */
                            /* : 0 0 0 0 : CK0m = fCLK */
                            /* : 0 0 0 1 : CK0m = fCLK/2 */
                            /* : 0 0 1 0 : CK0m = fCLK/2^2 */
                            /* : 0 0 1 1 : CK0m = fCLK/2^3 */
                            /* : 0 1 0 0 : CK0m = fCLK/2^4 */
                            /* : 0 1 0 1 : CK0m = fCLK/2^5 */
                            /* : 0 1 1 0 : CK0m = fCLK/2^6 */
                            /* : 0 1 1 1 : CK0m = fCLK/2^7 */
                            /* : 1 0 0 0 : CK0m = fCLK/2^8 */
                            /* : 1 0 0 1 : CK0m = fCLK/2^9 */
                            /* : 1 0 1 0 : CK0m = fCLK/2^10 */
                            /* : 1 0 1 1 : CK0m = fCLK/2^11 */
                            /* : 1 1 0 0 : CK0m = fCLK/2^12 */
                            /* : 1 1 0 1 : CK0m = fCLK/2^13 */
```

```
                    /* : 1 1 1 0 : CK0m = fCLK/2^14 */
                    /* : 1 1 1 1 : CK0m = fCLK/2^15 */
                    /*    m = 0, 1 */

/* CH1:for wait */
TMR01 = 0b0000000000000000;        /* Timer Mode Register 01 */
                    /*||||||||||||++++--- : [1]Operation mode of channel 1 */
                    /*||||||||||||          [2]Count operation of TCR */
                    /*||||||||||||          [3]Independent operation */
                    /*||||||||||||          [4]Setting of starting counting and interrupt */
                    /*|||||||||||| : 0 0 0 0 : [1]Interval timer mode */
                    /*||||||||||||              [2]Counting down */
                    /*||||||||||||              [3]Possible */
                    /*||||||||||||              [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                 (timer output does not change, either). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 0 0 1 : [1]Interval timer mode */
                    /*||||||||||||              [2]Counting down */
                    /*||||||||||||              [3]Possible */
                    /*||||||||||||              [4]Timer interrupt is generated when counting is started */
                    /*||||||||||||                 (timer output also changes). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 0 0 : [1]Capture mode */
                    /*||||||||||||              [2]Counting up */
                    /*||||||||||||              [3]Possible */
                    /*||||||||||||              [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                 (timer output does not change, either). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 0 1 : [1]Capture mode */
                    /*||||||||||||              [2]Counting up */
                    /*||||||||||||              [3]Possible */
                    /*||||||||||||              [4]Timer interrupt is generated when counting is started */
                    /*||||||||||||                 (timer output also changes). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 1 0 : [1]Event counter mode */
                    /*||||||||||||              [2]Counting down */
                    /*||||||||||||              [3]Possible */
                    /*||||||||||||              [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                 (timer output does not change, either). */
```

```
/*||||||||||| */
/*||||||||||| : 1 0 0 0 : [1]One-count mode */
/*|||||||||||             [2]Counting down */
/*|||||||||||             [3]Impossible */
/*|||||||||||             [4]Start trigger is invalid during counting operation. */
/*|||||||||||                At that time, interrupt is not generated, either. */
/*||||||||||| */
/*||||||||||| : 1 0 0 1 : [1]One-count mode */
/*|||||||||||             [2]Counting down */
/*|||||||||||             [3]Impossible */
/*|||||||||||             [4]Start trigger is valid during counting operation. */
/*|||||||||||                At that time, interrupt is also generated. */
/*||||||||||| */
/*||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*|||||||||||             [2]Counting up */
/*|||||||||||             [3]Possible */
/*|||||||||||             [4]Timer interrupt is not generated when counting is started */
/*|||||||||||                (timer output does not change, either). */
/*|||||||||||                Start trigger is invalid during counting operation. */
/*|||||||||||                At that time interrupt is not generated, either. */
/*||||||||||| */
/*||||||||||| : Other than above : Setting prohibited */
/*||||||||||| */
/*||||||||||++------- : Be sure to set 00 */
/*|||||||||| */
/*||||||||++--------- : Selection of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 valid
edge */
/*||||||||            (the timer input used with channel 1 is selected by using TIS0
register). */
/*|||||||| : 0 0 : Falling edge */
/*|||||||| : 0 1 : Rising edge */
/*|||||||| : 1 0 : Both edges (when low-level width is measured) */
/*||||||||         Start trigger: Falling edge, Capture trigger: Rising edge */
/*|||||||| : 1 1 : Both edges (when high-level width is measured) */
/*||||||||         Start trigger: Rising edge, Capture trigger: Falling edge */
/*|||||||| */
/*|||||+++----------- : Setting of start trigger or capture trigger of channel 1 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are unselected).
*/
```

```
                    /*||||| : 0 0 1 : Valid edge of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 is used
as both the start trigger and capture trigger. */
                    /*||||| : 0 1 0 : Both the edges of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 are
used as a start trigger and a capture trigger. */
                    /*||||| : 1 0 0 : Interrupt signal of the master channel is used (when the channel is used
as a slave channel with the combination operation function). */
                    /*||||| : Other than above : Setting prohibited */
                    /*||||| */
                    /*||||+-------------- : Selection of slave/master of channel 1 */
                    /*|||| : 0 : Operates as slave channel with combination operation function. */
                    /*|||| : 1 : Operates as master channel with combination operation function. */
                    /*|||| */
                    /*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
                    /*||| : 0 : Operation clock MCK specified by CKS01 bit */
                    /*||| : 1 : Valid edge of input signal input from TI01 pin, fSUB/2, fSUB/4, or INTRTC1 */
                    /*|||        (the timer input used with channel 1 is selected by using TIS0 register). */
                    /*||| */
                    /*|++---------------- : Be sure to set 00 */
                    /*| */
                    /*+------------------ : Selection of operation clock (MCK) of channel 1 */
                    /* : 0 : Operation clock CK00 set by TPS0 register */
                    /* : 1 : Operation clock CK01 set by TPS0 register */
    TDR01 = CCK00_500USEC;          /* set interval time to 500us */
    TMMK01 = 1;                                     /* disable interrupt */

    /* CH4:for DMA */
    TMR04 = 0b1000000000000000;     /* Timer Mode Register 04 */
                    /*||||||||||||++++--- : [1]Operation mode of channel 4 */
                    /*||||||||||||          [2]Count operation of TCR */
                    /*||||||||||||          [3]Independent operation */
                    /*||||||||||||          [4]Setting of starting counting and interrupt */
                    /*|||||||||||| : 0 0 0 0 : [1]Interval timer mode */
                    /*||||||||||||              [2]Counting down */
                    /*||||||||||||              [3]Possible */
                    /*||||||||||||              [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                 (timer output does not change, either). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 0 0 1 : [1]Interval timer mode */
                    /*||||||||||||              [2]Counting down */
```

```
/*|||||||||||| [3]Possible */
/*|||||||||||| [4]Timer interrupt is generated when counting is started */
/*|||||||||||| (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 0 0 : [1]Capture mode */
/*|||||||||||| [2]Counting up */
/*|||||||||||| [3]Possible */
/*|||||||||||| [4]Timer interrupt is not generated when counting is started */
/*|||||||||||| (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 0 1 0 1 : [1]Capture mode */
/*|||||||||||| [2]Counting up */
/*|||||||||||| [3]Possible */
/*|||||||||||| [4]Timer interrupt is generated when counting is started */
/*|||||||||||| (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 1 0 : [1]Event counter mode */
/*|||||||||||| [2]Counting down */
/*|||||||||||| [3]Possible */
/*|||||||||||| [4]Timer interrupt is not generated when counting is started */
/*|||||||||||| (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 0 : [1]One-count mode */
/*|||||||||||| [2]Counting down */
/*|||||||||||| [3]Impossible */
/*|||||||||||| [4]Start trigger is invalid during counting operation. */
/*|||||||||||| At that time, interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 1 : [1]One-count mode */
/*|||||||||||| [2]Counting down */
/*|||||||||||| [3]Impossible */
/*|||||||||||| [4]Start trigger is valid during counting operation. */
/*|||||||||||| At that time, interrupt is also generated. */
/*|||||||||||| */
/*|||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*|||||||||||| [2]Counting up */
/*|||||||||||| [3]Possible */
/*|||||||||||| [4]Timer interrupt is not generated when counting is started */
/*|||||||||||| (timer output does not change, either). */
```

```
/*||||||||||||                Start trigger is invalid during counting operation. */
/*||||||||||||                At that time interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : Other than above : Setting prohibited */
/*|||||||||||| */
/*||||||||||++------- : Be sure to set 00 */
/*|||||||||| */
/*||||||||++--------- : Selection of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 valid edge */
/*||||||||                (the timer input used with channel 4 is selected by using TIS0 register). */
/*|||||||| : 0 0 : Falling edge */
/*|||||||| : 0 1 : Rising edge */
/*|||||||| : 1 0 : Both edges (when low-level width is measured) */
/*||||||||          Start trigger: Falling edge, Capture trigger: Rising edge */
/*|||||||| : 1 1 : Both edges (when high-level width is measured) */
/*||||||||          Start trigger: Rising edge, Capture trigger: Falling edge */
/*|||||||| */
/*|||||+++----------- : Setting of start trigger or capture trigger of channel 4 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are unselected). */
/*||||| : 0 0 1 : Valid edge of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 is used as both the start trigger and capture trigger. */
/*||||| : 0 1 0 : Both the edges of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 are used as a start trigger and a capture trigger. */
/*||||| : 1 0 0 : Interrupt signal of the master channel is used (when the channel is used as a slave channel with the combination operation function). */
/*||||| : Other than above : Setting prohibited */
/*||||| */
/*||||+-------------- : Selection of slave/master of channel 4 */
/*|||| : 0 : Operates as slave channel with combination operation function. */
/*|||| : 1 : Operates as master channel with combination operation function. */
/*|||| */
/*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
/*||| : 0 : Operation clock MCK specified by CKS04 bit */
/*||| : 1 : Valid edge of input signal input from TI04 pin, fSUB/2, fSUB/4, or INTRTC1 */
/*|||        (the timer input used with channel 4 is selected by using TIS0 register). */
/*||| */
/*|++---------------- : Be sure to set 00 */
```

```
                                /*| */
                                /*+------------------ : Selection of operation clock (MCK) of channel 4 */
                                /* : 0 : Operation clock CK00 set by TPS0 register */
                                /* : 1 : Operation clock CK01 set by TPS0 register */
        TDR04 = 2000;           /* set interval time to 100us(=0.05us * 2000) */
        TMMK04 = 1;                     /* disable interrupt */

}

/*--------------------------------------------------------------*/
/* Module:          fn_InitLvi
                    */
/* Description:     Setting of Low-voltage detectior                                    */
/*      parameter: --
                                */
/*      return     : --
                                */
/*--------------------------------------------------------------*/
void    fn_InitLvi(void)
{
        unsigned short          loop;       /* waiting counter */

        LVIMK = 1;                                  /* disable LVI interrupt */

        LVIS =  0b00001001;                         /* Low-Voltage Detection Level Select Register */
                                /*||||++++--- : Detection level */
                                /*|||| : 0 0 0 0 : VLVI0 (4.22V) */
                                /*|||| : 0 0 0 1 : VLVI1 (4.07V) */
                                /*|||| : 0 0 1 0 : VLVI2 (3.92V) */
                                /*|||| : 0 0 1 1 : VLVI3 (3.76V) */
                                /*|||| : 0 1 0 0 : VLVI4 (3.61V) */
                                /*|||| : 0 1 0 1 : VLVI5 (3.45V) */
                                /*|||| : 0 1 1 0 : VLVI6 (3.30V) */
                                /*|||| : 0 1 1 1 : VLVI7 (3.15V) */
                                /*|||| : 1 0 0 0 : VLVI8 (2.99V) */
                                /*|||| : 1 0 0 1 : VLVI9 (2.84V) */
                                /*|||| : 1 0 1 0 : VLVI10 (2.68V) */
                                /*|||| : 1 0 1 1 : VLVI11 (2.53V) */
                                /*|||| : 1 1 0 0 : VLVI12 (2.38V) */
```

```
                              /*||||  : 1 1 0 1 : VLVI13 (2.22V) */
                              /*||||  : 1 1 1 0 : VLVI14 (2.07V) */
                              /*||||  : 1 1 1 1 : VLVI15 (1.91V) */
                              /*|||| */
                              /*++++------- : Be sure to set 0000 */

      LVIM =    0b10000000;             /* Low-Voltage Detection Register */
                              /*|||||||+--- : LVIF Low-voltage detection flag */
                              /*||||||| : 0 : * LVISEL = 0: VDD >= VLVI, or when LVI operation is disabled */
                              /*|||||||        * LVISEL = 1: EXLVI >= VEXLVI, or when LVI operation is disabled */
                              /*||||||| : 1 : * LVISEL = 0: VDD < VLVI */
                              /*|||||||        * LVISEL = 1: EXLVI < VEXLVI */
                              /*||||||| */
                              /*||||||+---- : Low-voltage detection operation mode (interrupt/reset) selection(LVIMD) */
                              /*|||||| : 0 : * LVISEL = 0: Generates an internal interrupt signal */
                              /*||||||                      when VDD drops lower than VLVI (VDD < VLVI) */
                              /*||||||                      or when VDD becomes VLVI or higher (VDD >= VLVI).
*/
                              /*||||||       * LVISEL = 1: Generates an interrupt signal */
                              /*||||||                      when EXLVI drops lower than VEXLVI (EXLVI <
VEXLVI) */
                              /*||||||                      or when EXLVI becomes VEXLVI or higher (EXLVI >=
VEXLVI). */
                              /*|||||| : 1 : * LVISEL = 0: Generates an internal reset signal when VDD < VLVI */
                              /*||||||                      and releases the reset signal when VDD >= VLVI. */
                              /*||||||       * LVISEL = 1: Generates an internal reset signal when EXLVI < VEXLVI
*/
                              /*||||||                      and releases the reset signal when EXLVI >= VEXLVI.
*/
                              /*|||||| */
                              /*|||||+----- : Voltage detection selection(LVISEL) */
                              /*||||| : 0 : Detects level of supply voltage (VDD) */
                              /*||||| : 1 : Detects level of input voltage from external input pin (EXLVI) */
                              /*||||| */
                              /*|++++------ : Be sure to set 0000 */
                              /*| */
                              /*+---------- : Enables low-voltage detection operation */
                              /* : 0 : Disables operation */
                              /* : 1 : Enables operation */
```

```
	/* software to wait for the operation stabilization time (210us) */
	for(loop = 500; loop > 0; loop--){
		NOP();
	}
	/* wait until VLVI≤VDD */
	while( LVIF ){
		NOP();
	}

	LVIIF = 0;				/* clear LVI interrupt request flag */
}



/*--------------------------------------------------------------*/
/* Module:		fn_DmaInit
					*/
/* Description:	Setting of DMA contoroler							*/
/*	parameter: --
					*/
/*	return	: --
					*/
/*--------------------------------------------------------------*/
void	fn_DmaInit(void)
{

	DEN0 = 1;						/* Enables operation of DMA channel 0 */

	DSA0 = (unsigned char)&DACSW0;				/* DMA SFR Address Register 0
*/
	DRA0 = (unsigned short)&ushDacData[0];		/* DMA RAM Address Register 0 */
	DBC0 = 25;							/* DMA Byte Count Register 0 */
	DMC0 = 0b01100100;				/* DMA Mode Control Register 0 */
			/*||||||||| */
			/*||||++++------- IFC3-IFC0 */
			/*||||		[Selection of DMA start source] */
			/*||||		 0000:Disables DMA transfer by interrupt */
			/*||||				(Only software trigger is enabled.) */
			/*||||		 0010:INTTM00 */
```

```
                /*||||            0011:INTTM01 */
                /*||||            0100:INTTM04 */
                /*||||            0101:INTTM05 */
                /*||||            0110:INTST0/INTCSI00 */
                /*||||            0111:INTSR0/INTCSI01 */
                /*||||            1000:INTST1/INTCSI10/INTIIC10 */
                /*||||            1001:INTSR1 */
                /*||||            1010:INTST3 */
                /*||||            1011:INTSR3 */
                /*||||            1100:INTAD */
                /*|||+----------- DWAIT0 */
                /*|||            [Pending of DMA transfer] */
                /*|||             0:Executes DMA transfer upon DMA start request (not held pending). */
                /*|||             1:Holds DMA start request pending if any. */
                /*||+------------ DS0 */
                /*||             [Specification of transfer data size for DMA transfer] */
                /*||              0:8bits */
                /*||              1:16bits */
                /*|+------------- DRS0 */
                /*|              [Selection of DMA transfer direction] */
                /*|               0:SFR to internal RAM */
                /*|               1:Internal RAM to SFR */
                /*+------------- STG0 */
                /*               [DMA transfer start software trigger] */
                /*                0:No trigger operation */
                /*                1:DMA transfer is started when DMA operation is enabled (DEN0 = 1). */

        DST0 = 1;               /* DMA transfer is started when DMA operation is enabled (DEN0 = 1). */

}



/*------------------------------------------------------------*/
/* Module:         fn_Wait500usBase
        */
/* Description:    Delays the program for (Time * 500us)                    */
/*      parameter: wait time(Time)
        */
/*      return     : --
```

```
                                        */
/*------------------------------------------------------------*/
void    fn_Wait500usBase(unsigned short Time)
{
        TS0L.1 = 1;                             /* start TAU0 CH1 */
        TMIF01 = 0;

        for(; Time > 0; Time--){        /* wait for (parameter * 500)us */
                while(!TMIF01){
                        NOP();
                }
                TMIF01 = 0;
        }

        TT0L.1 = 1;                             /* stop TAU0 CH1 */
}



/*------------------------------------------------------------*/
/* Module:          fn_AdcDacInit
                    */
/* Description:     Initialization of A/D D/A OPAmp module                  */
/*      parameter: --
                                        */
/*      return     : --
                                        */
/*------------------------------------------------------------*/
void    fn_AdcDacInit(void)
{
        ADCEN = 1;                              /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

        /*------------------------------------------*/
        /*   Initialization of Voltage reference                  */
        /*------------------------------------------*/
        ADVRC = 0b00001000;             /* Analog reference voltage control register */
                                /*||||+|++--- : [1]Positive reference voltage supplies selection of A/D and D/A
converters */
                                /*|||| |            [2]Operationcontrol of voltage reference */
```

```
/*|||| |              [3]Output voltage selection of voltage reference */
/*|||| |              [4]Operation control of input gate voltage boost circuit for A/D converter
*/
/*|||| |              [5]Relationship with the conversion mode used */
/*|||| | */
/*|||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
/*|||| |                  [2]Stops operation (Hi-Z) */
/*|||| |                  [3]2.5 V */
/*|||| |                  [4]Stops operation */
/*|||| |                  [5]Can be set in conversion mode 1 */
/*|||| | */
/*|||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
/*|||| |                  [2]Stops operation (Hi-Z) */
/*|||| |                  [3]2.0 V */
/*|||| |                  [4]Enables operation */
/*|||| |                  [5]Can be set in conversion mode 2 or 3 */
/*|||| | */
/*|||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
/*|||| |                  [2]Stops operation (pull-down output) */
/*|||| |                  [3]2.5 V */
/*|||| |                  [4]Stops operation */
/*|||| |                  [5] - */
/*|||| | */
/*|||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
/*|||| |                  [2]Enables operation */
/*|||| |                  [3]2.5 V */
/*|||| |                  [4]Enables operation */
/*|||| |                  [5]Can be set in conversion mode 2 or 3 */
/*|||| | */
/*|||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
/*|||| |                  [2]Stops operation (pull-down output) */
/*|||| |                  [3]2.0 V */
/*|||| |                  [4]Enables operation */
/*|||| |                  [5] - */
/*|||| | */
/*|||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
/*|||| |                  [2]Enables operation */
/*|||| |                  [3]2.0 V */
/*|||| |                  [4]Enables operation */
```

```
                          /*|||| |          [5]Can be set in conversion mode 2 or 3 */
                          /*|||| | */
                          /*|||| | : Other than the above : Setting prohibited */
                          /*|||| | */
                          /*|+++-+----- : Be sure to set 0000 */
                          /*| */
                          /*+---------- : Reference voltage supply (negative side) of A/D converter selection */
                          /* : 0 : AVSS */
                          /* : 1 : AVREFM (external voltage reference input) */

    ADVRC.0 = 1;                  /* Enables operation */
    ADVRC.1 = 1;                  /* Output 2.0 V */

    ADPC =    0b00000000;         /* A/D Port Configuration Register */
                          /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                          /*||| :           +------------------------ ANI15/AVREFM/P157 */
                          /*||| :           | +-+-+------------------ ANI10/P152 - ANI8/AMP2+/P150 */
                          /*||| :           | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
                          /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                          /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                          /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                          /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                          /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                          /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                          /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                          /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                          /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                          /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                          /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                          /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                          /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                          /*||| */
                          /*+++-------- : Be sure to set 000 */

    /* wait for settling time to 17msec */
    fn_Wait500usBase(17000/500);

    /*------------------------------------------*/
    /*    Initialization of operational amplifier    */
```

```
    /*-------------------------------------------*/
    PM2.1 = 1;                          /* input port mode (AMP0O) */
    PM2.0 = 1;                          /* input port mode (AMP0-) */
    PM2.2 = 1;                          /* input port mode (AMP0+) */

    PM2.4 = 1;                          /* input port mode (AMP1O) */
    PM2.3 = 1;                          /* input port mode (AMP1-) */
    PM2.5 = 1;                          /* input port mode (AMP1+) */

    PM2.7 = 1;                          /* input port mode (AMP2O) */
    PM2.6 = 1;                          /* input port mode (AMP2-) */
    PM15.0 = 1;                         /* input port mode (AMP2+) */

    PM11.0 = 1;                         /* input port mode (ANO0) */

    ADCEN = 1;                          /* supplies input clock to OPAmp A/D converter */
    OAC.0 = 1;                          /* operational amplifier (AMP0) enable */
    OAC.1 = 0;                          /* operational amplifier (AMP1) disable */
    OAC.2 = 0;                          /* operational amplifier (AMP2) disable */

    /*-------------------------------------------*/
    /*      Initialization of D/A converter      */
    /*-------------------------------------------*/
    DACEN = 1;                                  /* supplies input clock to D/A converter */

    DAM.0 = 1;                                  /* [DAMD0] D/A converter operation is Real-time
output mode */
    DAM.2 = 1;                                  /* [DARES0]D/A conversion resolution is 12-bit */
    DAM.6 = 1;                                  /* [DAREF] D/A converter positive reference voltage
is VREFOUT */
    DACSW0 = aushDacData[99];           /* D/A conversion value setting reigster 0 */

    /* set D/A chanel1 */
    DAM.1 = 0;                                  /* [DAMD1] D/A converter operation is Normal mode
*/
    DAM.3 = 1;                                  /* [DARES1]D/A conversion resolution is 12-bit */
    DACSW1 = aushDacData[99];           /* D/A conversion value setting reigster 1 */

}
```

```
/*--------------------------------------------------------------*/
/* Module:          main
                                */
/* Description:     Main process
          */
/*        parameter: --
                                */
/*        return    : --
                                */
/*--------------------------------------------------------------*/
void main(void)
{
          unsigned short          count;                    /* counter for initializing D/A convert data */

/*        ushCycleCount = 0;                                /* Cycle counter */
          ucCycleStatus = 0;                                /* Cycle status */

          DACE1 = 1;                                                                        /* D/A converter CH1 enable */
          DACE0 = 1;                                                                        /* D/A converter CH0 enable */


          /*==================================*/
          /*----------------------------------*/
          /*                    Main Loop                    */
          /*----------------------------------*/
          /*==================================*/

          /*   initialize D/Aconverter data */
          for(count = 0; count<100; count++){
                    ushDacData[count] = aushDacData[count];
          }

          TS0L.4 = 1;                                       /* start TAU0 CH4 */
          TMIF04 = 0;                                       /* reset INTTM04 interrupt request */

          DMAMK0 = 0;                                       /* enable DMA CH0 interrupt */

          EI();                                             /* enable all interrupts */
```

```
        /*==============================================================*/
        /* if system have nothing to do, go to standby for power-saving */
        /*==============================================================*/


        while (1){
                HALT();
                NOP();
        }
}


/*------------------------------------------------------------*/
/* Module:          fn_intdma0
                              */
/* Description:     DMA ch0 interruput process                                      */
/*      parameter: --
                              */
/*      return    : --
                              */
/*------------------------------------------------------------*/
__interrupt void fn_intdma0(void)
{
        TT0L.4 = 1;                     /* stop TAU0 CH4 */


        switch(ucCycleStatus){                          /* Cycle status */
                case    0:
                        TDR04 = 20000;                  /*  set  interval  time  to
1000us(=0.05us * 20000) */
                        ucCycleStatus = 1;          /* Cycle status (Flat) */
/*                      DRA0 = (unsigned short)&ushDacData[0];          /* DMA RAM Address
Register 0 */
                        DBC0 = 1;                                       /*  DMA  Byte  Count
Register 0 */
                        break;

                case    1:
```

```
                        TDR04 = 200;                        /*  set  interval  time  to
10us(=0.05us * 200) */
                        ucCycleStatus = 2;          /* Cycle status (2/4 sign) */
                        DBC0 = 25;                                          /*   DMA
Byte Count Register 0 */
                        break;

            case    2:
                        DRA0 = (unsigned short)&ushDacData[0];        /* DMA RAM Address
Register 0 */
                        DACSW0 = ushDacData[99];            /* D/A conversion value setting
reigster 0 */
                        TMR04 = 0b0000000000000000;         /* Timer Mode Register 04 */
                        TDR04 = 20000;                      /*  set  interval  time  to
1000us(=0.05us * 20000) */
                        ucCycleStatus = 3;          /* Cycle status (slow) */
                        DBC0 = 50;                                          /*   DMA
Byte Count Register 0 */
                        break;

            case    3:
                        DRA0 = (unsigned short)&ushDacData[0];        /* DMA RAM Address
Register 0 */
                        DACSW0 = ushDacData[99];            /* D/A conversion value setting
reigster 0 */
                        TMR04 = 0b1000000000000000;         /* Timer Mode Register 04 */
                        TDR04 = 200;                        /*  set  interval  time  to
10us(=0.05us * 200) */
                        ucCycleStatus = 0;          /* Cycle status (1kHz) */
                        DBC0 = 25;                                          /*   DMA
Byte Count Register 0 */
                        break;

        }
 /*     DBC0 = 100;                                 /* DMA Byte Count Register 0 */
        DST0 = 1;                           /* DMA transfer is started when DMA operation is
enabled (DEN0 = 1). */

        TS0L.4 = 1;                 /* start TAU0 CH4 */
```

```
        TMIF04 = 0;                         /* reset INTTM04 interrupt request */

}
```

(7) Instrumentation amplifier

● main.c

```
/*
 * Copyright (C) NEC Electronics Corporation 2006
 * NEC ELECTRONICS CONFIDENTIAL AND PROPRIETARY
 * All rights reserved by NEC Electronics Corporation.
 * This program must be used solely for the purpose for which
 * it was furnished by NEC Electronics Corporation. No part of this
 * program may be reproduced or disclosed to others, in any
 * form, without the prior written permission of NEC Electronics
 * Corporation. Use of copyright notice dose not evidence
 * publication of the program.
 */

/*--------------------------------------------------------------------*/
/* #pragma directive for CC78K0
                                */
/*--------------------------------------------------------------------*/
#pragma         SFR
#pragma         DI
#pragma         EI
#pragma         HALT
#pragma         NOP

#pragma interrupt INTRTC fn_intrtc RB1

/*--------------------------------------------------------------------*/
/* Include files
                                          */
/*--------------------------------------------------------------------*/
/* TAU:TDR0n value operation by CK00 (fCLK/2^3 = 5MHz) */
#define CCK00_500USEC   (2500 - 1)          /* 500us (0.2[us/clk] * 2500[count]) */

/* TAU:TDR0n value operation by CK01 (fCLK = 20MHz) */
#define CCK01_PLAY                  (2500 - 1) /* 0.125ms = about 8kHz for play */
                                                                        /*
(0.05[us/clk] * 2500[count]) */
```

```
#define CDAC0_50V                 ( 1024 )
#define CDAC0_52V                 ( 1065 )
#define CDAC0_54V                 ( 1105 )

/*----------------------------------------------------------------------*/
/* Function prototyps
                                  */
/*----------------------------------------------------------------------*/
            void    fn_Wait500usBase(unsigned short);    /* Delays the program for (Time * 500us) */
            void    fn_InitPort(void);                              /* Setting of I/O ports
*/
            void    fn_InitTau0(void);                              /*  Setting  of  Timer
array unit 0 */
            void    fn_InitLvi(void);                               /*    Setting    of
Low-voltage detectior */
            void    fn_InitVr(void);                                /*  Setting  of  Voltage
reference */
            void    fn_InitRtc(void);                               /*    Setting    of
Real-time counter */
            void    fn_AdcDacInit(void);                    /* Setting of D/A converter */



/*----------------------------------------------------------------------*/
/* Extern variables/constants
            */
/*----------------------------------------------------------------------*/

/*----------------------------------------------------------------------*/
/* Local constants
                                  */
/*----------------------------------------------------------------------*/
/*----------------------------------------------------------------------*/
/* Global variables
                                  */
/*----------------------------------------------------------------------*/

/*----------------------------------------------------------------------*/
/* Local variables
                                  */
```

```
/*----------------------------------------------------------------------*/

/*----------------------------------------------------------------------*/
/* Code
                                                                  */
/*----------------------------------------------------------------------*/
/*----------------------------------------------------------------------*/
/* Hardware initialization
                                  */
/*----------------------------------------------------------------------*/
void hdwinit(void)
{
        DI();                           /* disable all interrupts */

        /*------------------------------------------*/
        /*          Initialization of port          */
        /*------------------------------------------*/
        fn_InitPort();

        /*------------------------------------------*/
        /*          Initialization of clock         */
        /*------------------------------------------*/
        CMC =   0b01010011;             /* Clock Operation Mode Control Register */
                        /*|||||||+--- : Control of high-speed system clock oscillation frequency */
                        /*||||||| : 0 : 2 MHz <= fMX <= 10 MHz */
                        /*||||||| : 1 : 10 MHz < fMX <= 20 MHz */
                        /*||||||| */
                        /*|||||++---- : XT1 oscillator oscillation mode selection */
                        /*||||| : 0 0 : Low-consumption oscillation */
                        /*||||| : 0 1 : Normal oscillation */
                        /*||||| : 1 x : Super-low-consumption oscillation */
                        /*|||||   x = don't care */
                        /*||||| */
                        /*||||+------ : Be sure to set 0 */
                        /*|||| */
                        /*|||+------- : [1] Subsystem clock pin operation mode */
                        /*|||               [2] XT1/P123 pin and XT2/P124 pin */
                        /*||| : 0 : [1]Input port mode */
                        /*|||         [2]Input port */
```

*/

```
                    /*||| */
                    /*||| : 1 : [1]XT1 oscillation mode */
                    /*|||       [2]Crystal resonator connection */
                    /*||| */
                    /*||+-------- : Be sure to set 0 */
                    /*|| */
                    /*++--------- : [1]EXCLK OSCSEL High-speed system clock pin operation mode */
                    /*                [2]X1/P121 pin */
                    /*                [3]X2/EXCLK/P122 pin */
                    /* : 0 0 : [1]Input port mode */
                    /*         [2][3]Input port */
                    /* */
                    /* : 0 1 : [1]X1 oscillation mode */
                    /*         [2][3]Crystal/ceramic resonator connection */
                    /* */
                    /* : 1 0 : [1]Input port mode */
                    /*         [2][3]Input port */
                    /* */
                    /* : 1 1 : [1]External clock input mode */
                    /*         [2]Input port */
                    /*         [3]External clock input */

MSTOP = 0;                              /* X1 oscillator operating */
XTSTOP = 0;                             /* XT1 oscillator operating */

OSMC =   0b00000001;                    /* Operation Speed Mode Control Register */
                    /*||||||++--- : fCLK frequency selection */
                    /*|||||| : 0 0 : Operates at a frequency of 10 MHz or less. */
                    /*|||||| : 0 1 : Operates at a frequency higher than 10 MHz. */
                    /*|||||| : 1 0 : Operates at a frequency of 1 MHz. */
                    /*|||||| */
                    /*|+++++----- : Be sure to set 00000 */
                    /*| */
                    /*+---------- : Setting in subsystem clock HALT mode */
                    /* : 0 : Enables subsystem clock supply to peripheral functions. */
                    /*      (See Table 21-1 Operating Statuses in HALT Mode (2/3) */
                    /*       for the peripheral functions whose operations are enabled.) */
                    /* : 1 : Stops subsystem clock supply to peripheral functions except real-time counter,
*/
```

```
                        /*          clock output/buzzer output, and LCD controller/driver. */

while(OSTC.0 != 1){                 /* wait X1 oscillation stabilization */
        NOP();
}

/*-- Caution ---------------------------------------------*/
/* To increase fCLK to 10 MHz or higher, set FSEL to '1',    */
/* then change fCLK after two or more clocks have elapsed. */
/*------------------------------------------------------*/
NOP();
NOP();

CKC =   0b00010000;                     /* System Clock Control Register */
                        /*|+|+++++--- : Selection of CPU/peripheral hardware clock (fCLK) */
                        /*| | : 0 0 x 0 0 0 : fIH */
                        /*| | : 0 0 x 0 0 1 : fIH/2 (default) */
                        /*| | : 0 0 x 0 1 0 : fIH/2^2 */
                        /*| | : 0 0 x 0 1 1 : fIH/2^3 */
                        /*| | : 0 0 x 1 0 0 : fIH/2^4 */
                        /*| | : 0 0 x 1 0 1 : fIH/2^5 */
                        /*| | : 0 1 x 0 0 0 : fMX */
                        /*| | : 0 1 x 0 0 1 : fMX/2 */
                        /*| | : 0 1 x 0 1 0 : fMX/2^2 */
                        /*| | : 0 1 x 0 1 1 : fMX/2^3 */
                        /*| | : 0 1 x 1 0 0 : fMX/2^4 */
                        /*| | : 0 1 x 1 0 1 : fMX/2^5 */
                        /*| | : 1 x 0 x x x : fSUB */
                        /*| | : 1 x 1 x x x : fSUB/2 */
                        /*| | : Other than above : Setting prohibited */
                        /*| |    x = don't care */
                        /*| | */
                        /*| +-------- : Status of Main system clock (fMAIN) */
                        /*| : 0 : Internal high-speed oscillation clock (fIH) */
                        /*| : 1 : High-speed system clock (fMX) */
                        /*| */
                        /*+---------- : Status of CPU/peripheral hardware clock (fCLK) */
                        /* : 0 : Main system clock (fMAIN) */
                        /* : 1 : Subsystem clock (fSUB) */
```

```
/* Confirming the CPU clock status */

while((CLS != 0)||(MCS != 1)){
        NOP();
}

/* CPU is operating on a High-speed system clock */
HIOSTOP = 1;                    /* internal high-speed oscillation stopped */

OSTS =  0b00000111;             /* Oscillation Stabilization Time Select Register */
                /*||||++++--- : Oscillation stabilization time selection */
                /*|||| : 0 0 0 : 2^8/fX */
                /*|||| : 0 0 1 : 2^9/fX */
                /*|||| : 0 1 0 : 2^10/fX */
                /*|||| : 0 1 1 : 2^11/fX */
                /*|||| : 1 0 0 : 2^13/fX */
                /*|||| : 1 0 1 : 2^15/fX */
                /*|||| : 1 1 0 : 2^17/fX */
                /*|||| : 1 1 1 : 2^18/fX */
                /*|||| */
                /*++++------- : Be sure to set 000000 */

/*----------------------------------------*/
/*         Initialization of timer        */
/*----------------------------------------*/
fn_InitTau0();

/* software to wait for the operation stabilization time */
/* (over 500ms from when XT1 enable) */
fn_Wait500usBase(500000/500);


/*----------------------------------------*/
/* Initialization of low-voltage detector */
/*----------------------------------------*/
fn_InitLvi();

/*----------------------------------------*/
```

```
        /*   Initialization of real-time counter      */
        /*----------------------------------------*/
        fn_InitRtc();

        /*----------------------------------------*/
        /*   Initialization of A/D,D/A,OPAmp                          */
        /*----------------------------------------*/
        fn_AdcDacInit();



        /* CPU is operating on a High-speed system clock */
        CKC =      0b01000000;                        /* CPU/peripheral hardware clock by fSUB*/

        while(CLS != 1){
                   NOP();
        }

        HIOSTOP = 1;                                   /* internal high-speed oscillation stopped */
        MSTOP = 1;                                              /* X1 oscillator stopped */

        EI();                                          /* enable all interrupts */
}

/*------------------------------------------------------------*/
/* Module:          fn_InitPort
                    */
/* Description:     Setting of I/O ports                                                              */
/*       parameter: --
                              */
/*       return     : --
                              */
/*------------------------------------------------------------*/
void    fn_InitPort(void)
{
/*--------------------------------------------------------*/
/*       Ports configuration for digital input and output                 */
/*--------------------------------------------------------*/
        ADPC =     0b00010000;          /* A/D Port Configuration Register */
                              /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
```

```
                    /*||| :            +------------------------ ANI15/AVREFM/P157 */
                    /*||| :            | +-+-+------------------ ANI10/P152 - ANI8/AMP2+/P150 */
                    /*||| :            | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
                    /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                    /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                    /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                    /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                    /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                    /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                    /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                    /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                    /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                    /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                    /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                    /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                    /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                    /*||| */
                    /*+++-------- : Be sure to set 000 */

/*-----------------------------------------------------------*/
/*      Setting of Port 0
        */
/*-----------------------------------------------------------*/
        P0 =    0b00000000;     /* Set P00-P02 Output latch to Low */
        PM0 =   0b11111000;     /* Set P00-P02 to output port */
                                                /* P00-P02:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 1
        */
/*-----------------------------------------------------------*/
        P1 =    0b00000000;     /* Set P10-P17 Output latch to Low */
        PM1 =   0b00000000;     /* Set P10-P17 to output port */
                                                /* P10-P15:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 2
        */
/*-----------------------------------------------------------*/
```

```
        P2 =        0b00000000;         /* Set P20-P27 Output latch to Low */
        PM2 =       0b11111111;         /* Set P20-P27 to input port */
                                                            /* P20-P27:AMP0-2 */

/*-----------------------------------------------------------*/
/*      Setting of Port 3
        */
/*-----------------------------------------------------------*/
        P3 =        0b00001100;         /* Set P30-P31,P34 Output latch to Low */
                                                            /* Set P33,P32 Output latch High */
        PM3 =       0b11100000;         /* Set P30-P34 to output port */
                                                            /* P30-P34:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 4
        */
/*-----------------------------------------------------------*/
        P4 =        0b00000000;         /* Set P40-P41 Output latch to Low */
        PM4 =       0b11111100;         /* Set P40-P41 to output port */
                                                            /* P40-P41:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 5
        */
/*-----------------------------------------------------------*/
        P5 =        0b00000000;         /* Set P50-P57 Output latch to Low */
        PM5 =       0b11110000;         /* Set P50-P57 to output port */
                                                            /* P50-P57:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 6
        */
/*-----------------------------------------------------------*/
        P6 =        0b00000000;         /* Set P60-P61 Output latch to Low */
        PM6 =       0b11111100;         /* Set P60-P61 to output port */
                                                            /* P60-P61:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 7
```

```
        */
/*----------------------------------------------------------*/
        P7 =    0b00000000;     /* Set P70-P77 Output latch to Low */
        PM7 =   0b00000000;     /* Set P70-P77 to output port */
                                        /* P70-P77:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 8
        */
/*----------------------------------------------------------*/
        P8 =    0b00000000;     /* Set P80-P88 Output latch to Low */
        PM8 =   0b00000000;     /* Set P80-P88 to output port */
                                        /* P80-P88:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 9
        */
/*----------------------------------------------------------*/
        P9 =    0b00000000;     /* Set P90-P97 Output latch to Low */
        PM9 =   0b00000000;     /* Set P90-P97 to output port */
                                        /* P90-P97:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 10
        */
/*----------------------------------------------------------*/
        P10 =   0b00000000;     /* Set P100-P102 Output latch to Low */
        PM10 =  0b11111000;     /* Set P100-P102 to output port */
                                        /* P100-P102:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 11
        */
/*----------------------------------------------------------*/
        P11 =   0b00000000;     /* Set P110-P111 Output latch to Low */
        PM11 =  0b11111100;     /* Set P110-P111 to output port */
                                        /* P110-P111:Unused */

/*----------------------------------------------------------*/
```

```
/*      Setting of Port 12
        */
/*------------------------------------------------------------*/
        P12 =   0b00000000;     /* Set P120 Output latch to Low */
        PM12 =  0b11111110;     /* Set P120 to output port */
                                                /* P120-P124:Unused */
                                                /* *P121-P124:Input port */

/*------------------------------------------------------------*/
/*      Setting of Port 13
        */
/*------------------------------------------------------------*/
        P13 =   0b00000000;     /* Set P130 Output latch to Low */
                                                /* P130:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 14
        */
/*------------------------------------------------------------*/
        P14 =   0b00000000;     /* Set P140-P147 Output latch to Low */
        PM14 =  0b00000000;     /* Set P140-P147 to output port */
                                                /* P140-P147:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 15
        */
/*------------------------------------------------------------*/
        P15 =   0b00000000;     /* Set P150-P152,P157 Output latch to Low */
        PM15 =  0b11111111;     /* Set P150-P152,P157 to input port */
                                                /* P150:AMP2 */
                                                /* P151,P152,P157:Unused */

}

/*----------------------------------------------------------------*/
/* Module:          fn_InitTau0
                            */
/* Description:     Setting of Timer array unit 0                          */
/*      parameter: --
```

```
                                */
/*      return      : --
                                */
/*--------------------------------------------------------------*/
void    fn_InitTau0(void)
{
        TAU0EN = 1;                             /* supplies input clock to timer array unit 0 */
        TPS0L =   0b00000010;           /* Timer Clock Select Register 0 */
                                /*||||++++--- : Selection of operation clock (CK00) */
                                /*++++------- : Selection of operation clock (CK01) */
                                /* : 0 0 0 0 : CK0m = fCLK */
                                /* : 0 0 0 1 : CK0m = fCLK/2 */
                                /* : 0 0 1 0 : CK0m = fCLK/2^2 */
                                /* : 0 0 1 1 : CK0m = fCLK/2^3 */
                                /* : 0 1 0 0 : CK0m = fCLK/2^4 */
                                /* : 0 1 0 1 : CK0m = fCLK/2^5 */
                                /* : 0 1 1 0 : CK0m = fCLK/2^6 */
                                /* : 0 1 1 1 : CK0m = fCLK/2^7 */
                                /* : 1 0 0 0 : CK0m = fCLK/2^8 */
                                /* : 1 0 0 1 : CK0m = fCLK/2^9 */
                                /* : 1 0 1 0 : CK0m = fCLK/2^10 */
                                /* : 1 0 1 1 : CK0m = fCLK/2^11 */
                                /* : 1 1 0 0 : CK0m = fCLK/2^12 */
                                /* : 1 1 0 1 : CK0m = fCLK/2^13 */
                                /* : 1 1 1 0 : CK0m = fCLK/2^14 */
                                /* : 1 1 1 1 : CK0m = fCLK/2^15 */
                                /*    m = 0, 1 */

        /* CH1:for wait */
        TMR01 = 0b0000000000000000;             /* Timer Mode Register 01 */
                                /*||||||||||||++++--- : [1]Operation mode of channel 1 */
                                /*||||||||||||          [2]Count operation of TCR */
                                /*||||||||||||          [3]Independent operation */
                                /*||||||||||||          [4]Setting of starting counting and interrupt */
                                /*|||||||||||| : 0 0 0 0 : [1]Interval timer mode */
                                /*||||||||||||              [2]Counting down */
                                /*||||||||||||              [3]Possible */
                                /*||||||||||||              [4]Timer interrupt is not generated when counting is started */
                                /*||||||||||||                 (timer output does not change, either). */
```

```
/*|||||||||||| */
/*|||||||||||| : 0 0 0 1 : [1]Interval timer mode */
/*||||||||||||            [2]Counting down */
/*||||||||||||            [3]Possible */
/*||||||||||||            [4]Timer interrupt is generated when counting is started */
/*||||||||||||               (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 0 0 : [1]Capture mode */
/*||||||||||||            [2]Counting up */
/*||||||||||||            [3]Possible */
/*||||||||||||            [4]Timer interrupt is not generated when counting is started */
/*||||||||||||               (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 0 1 0 1 : [1]Capture mode */
/*||||||||||||            [2]Counting up */
/*||||||||||||            [3]Possible */
/*||||||||||||            [4]Timer interrupt is generated when counting is started */
/*||||||||||||               (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 1 0 : [1]Event counter mode */
/*||||||||||||            [2]Counting down */
/*||||||||||||            [3]Possible */
/*||||||||||||            [4]Timer interrupt is not generated when counting is started */
/*||||||||||||               (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 0 : [1]One-count mode */
/*||||||||||||            [2]Counting down */
/*||||||||||||            [3]Impossible */
/*||||||||||||            [4]Start trigger is invalid during counting operation. */
/*||||||||||||               At that time, interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 1 : [1]One-count mode */
/*||||||||||||            [2]Counting down */
/*||||||||||||            [3]Impossible */
/*||||||||||||            [4]Start trigger is valid during counting operation. */
/*||||||||||||               At that time, interrupt is also generated. */
/*|||||||||||| */
/*|||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*||||||||||||            [2]Counting up */
```

```
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                (timer output does not change, either). */
/*||||||||||||                Start trigger is invalid during counting operation. */
/*||||||||||||                At that time interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : Other than above : Setting prohibited */
/*|||||||||||| */
/*||||||||||++------- : Be sure to set 00 */
/*|||||||||| */
/*||||||||++--------- : Selection of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 valid
edge */
/*||||||||            (the timer input used with channel 1 is selected by using TIS0
register). */
/*|||||||| : 0 0 : Falling edge */
/*|||||||| : 0 1 : Rising edge */
/*|||||||| : 1 0 : Both edges (when low-level width is measured) */
/*||||||||         Start trigger: Falling edge, Capture trigger: Rising edge */
/*|||||||| : 1 1 : Both edges (when high-level width is measured) */
/*||||||||         Start trigger: Rising edge, Capture trigger: Falling edge */
/*|||||||| */
/*|||||+++----------- : Setting of start trigger or capture trigger of channel 1 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are unselected).
*/
/*||||| : 0 0 1 : Valid edge of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 is used
as both the start trigger and capture trigger. */
/*||||| : 0 1 0 : Both the edges of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 are
used as a start trigger and a capture trigger. */
/*||||| : 1 0 0 : Interrupt signal of the master channel is used (when the channel is used
as a slave channel with the combination operation function). */
/*||||| : Other than above : Setting prohibited */
/*||||| */
/*||||+-------------- : Selection of slave/master of channel 1 */
/*|||| : 0 : Operates as slave channel with combination operation function. */
/*|||| : 1 : Operates as master channel with combination operation function. */
/*|||| */
/*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
/*||| : 0 : Operation clock MCK specified by CKS01 bit */
/*||| : 1 : Valid edge of input signal input from TI01 pin, fSUB/2, fSUB/4, or INTRTC1 */
```

```
                        /*|||          (the timer input used with channel 1 is selected by using TIS0 register). */
                        /*||| */
                        /*|++---------------- : Be sure to set 00 */
                        /*| */
                        /*+------------------ : Selection of operation clock (MCK) of channel 1 */
                        /* : 0 : Operation clock CK00 set by TPS0 register */
                        /* : 1 : Operation clock CK01 set by TPS0 register */
    TDR01 = CCK00_500USEC;              /* set interval time to 500us */
    TMMK01 = 1;                                             /* disable interrupt */

}

/*--------------------------------------------------------------*/
/* Module:          fn_InitLvi
                    */
/* Description:     Setting of Low-voltage detectior                        */
/*      parameter: --
                        */
/*      return     : --
                        */
/*--------------------------------------------------------------*/
void    fn_InitLvi(void)
{
    unsigned short      loop;       /* waiting counter */

    LVIMK = 1;                          /* disable LVI interrupt */

    LVIS =      0b00001001;             /* Low-Voltage Detection Level Select Register */
                        /*||||++++--- : Detection level */
                        /*|||| : 0 0 0 0 : VLVI0 (4.22V) */
                        /*|||| : 0 0 0 1 : VLVI1 (4.07V) */
                        /*|||| : 0 0 1 0 : VLVI2 (3.92V) */
                        /*|||| : 0 0 1 1 : VLVI3 (3.76V) */
                        /*|||| : 0 1 0 0 : VLVI4 (3.61V) */
                        /*|||| : 0 1 0 1 : VLVI5 (3.45V) */
                        /*|||| : 0 1 1 0 : VLVI6 (3.30V) */
                        /*|||| : 0 1 1 1 : VLVI7 (3.15V) */
                        /*|||| : 1 0 0 0 : VLVI8 (2.99V) */
                        /*|||| : 1 0 0 1 : VLVI9 (2.84V) */
```

```
                    /*||||  :  1 0 1 0 : VLVI10 (2.68V) */
                    /*||||  :  1 0 1 1 : VLVI11 (2.53V) */
                    /*||||  :  1 1 0 0 : VLVI12 (2.38V) */
                    /*||||  :  1 1 0 1 : VLVI13 (2.22V) */
                    /*||||  :  1 1 1 0 : VLVI14 (2.07V) */
                    /*||||  :  1 1 1 1 : VLVI15 (1.91V) */
                    /*|||| */
                    /*++++------- : Be sure to set 0000 */

    LVIM =  0b10000000;             /* Low-Voltage Detection Register */
                    /*|||||||+--- : LVIF Low-voltage detection flag */
                    /*||||||| : 0 : * LVISEL = 0: VDD >= VLVI, or when LVI operation is disabled */
                    /*|||||||       * LVISEL = 1: EXLVI >= VEXLVI, or when LVI operation is disabled */
                    /*||||||| : 1 : * LVISEL = 0: VDD < VLVI */
                    /*|||||||       * LVISEL = 1: EXLVI < VEXLVI */
                    /*||||||| */
                    /*||||||+---- : Low-voltage detection operation mode (interrupt/reset) selection(LVIMD) */
                    /*|||||| : 0 : * LVISEL = 0: Generates an internal interrupt signal */
                    /*||||||                    when VDD drops lower than VLVI (VDD < VLVI) */
                    /*||||||                    or when VDD becomes VLVI or higher (VDD >= VLVI).
*/
                    /*||||||       * LVISEL = 1: Generates an interrupt signal */
                    /*||||||                    when EXLVI drops lower than VEXLVI (EXLVI <
VEXLVI) */
                    /*||||||                    or when EXLVI becomes VEXLVI or higher (EXLVI >=
VEXLVI). */
                    /*|||||| : 1 : * LVISEL = 0: Generates an internal reset signal when VDD < VLVI */
                    /*||||||                    and releases the reset signal when VDD >= VLVI. */
                    /*||||||       * LVISEL = 1: Generates an internal reset signal when EXLVI < VEXLVI
*/
                    /*||||||                    and releases the reset signal when EXLVI >= VEXLVI.
*/
                    /*|||||| */
                    /*|||||+----- : Voltage detection selection(LVISEL) */
                    /*||||| : 0 : Detects level of supply voltage (VDD) */
                    /*||||| : 1 : Detects level of input voltage from external input pin (EXLVI) */
                    /*||||| */
                    /*|++++------ : Be sure to set 0000 */
                    /*| */
```

```
                                /*+---------- : Enables low-voltage detection operation */
                                /* : 0 : Disables operation */
                                /* : 1 : Enables operation */

        /* software to wait for the operation stabilization time (210us) */
        for(loop = 500; loop > 0; loop--){
                NOP();
        }
        /* wait until VLVI≤VDD */
        while( LVIF ){
                NOP();
        }

        LVIIF = 0;                      /* clear LVI interrupt request flag */
}



/*-------------------------------------------------------------*/
/* Module:      fn_InitRtc
                */
/* Description: Setting of Real-time counter                              */
/*      parameter: --
                                */
/*      return    : --
                                */
/*-------------------------------------------------------------*/
void    fn_InitRtc(void)
{
        RTCEN = 1;                      /* supplies operational real-time counter (RTC) input clock. */

        RTCC0 = 0b00001010;             /* Real-Time Counter Control Register 0 */
                                /*|||||+++--- : Constant-period interrupt (INTRTC) selection */
                                /*||||| : 0 0 0 : Does not use constant-period interrupt function. */
                                /*||||| : 0 0 1 : Once per 0.5 s */
                                /*||||| : 0 1 0 : Once per 1 s */
                                /*||||| : 0 1 1 : Once per 1 m */
                                /*||||| : 1 0 0 : Once per 1 hour */
                                /*||||| : 1 0 1 : Once per 1 day */
                                /*||||| : 1 1 x : Once per 1 month */
```

```
                        /*|||||    x = don't care */
                        /*||||| */
                        /*||||+------ : Selection of 12-/24-hour system */
                        /*|||| : 0 : 12-hour system */
                        /*|||| : 1 : 24-hour system */
                        /*|||| */
                        /*|||+------- : RTCCL pin output control */
                        /*||| : 0 : Disables output of RTCCL pin (32 kHz). */
                        /*||| : 1 : Enables output of RTCCL pin (32 kHz). */
                        /*||| */
                        /*||+-------- : RTC1HZ pin output control */
                        /*|| : 0 : Disables output of RTC1HZ pin (1 Hz). */
                        /*|| : 1 : Enables output of RTC1HZ pin (1 Hz). */
                        /*|| */
                        /*|+--------- : Be sure to set 0 */
                        /*| */
                        /*+---------- : Real-time counter operation control */
                        /* : 0 : Stops counter operation. */
                        /* : 1 : Starts counter operation. */
        RTCMK = 1;                          /* disable RTC interrupt */
        RTCIF = 0;                          /* clear RTC interrupt request flag */
/*      RTCE = 1;                   /* starts counter operation */

}



/*--------------------------------------------------------------*/
/* Module:          fn_Wait500usBase
        */
/* Description:     Delays the program for (Time * 500us)                   */
/*      parameter: wait time(Time)
        */
/*      return     : --
                        */
/*--------------------------------------------------------------*/
void    fn_Wait500usBase(unsigned short Time)
{
        TS0L.1 = 1;                         /* start TAU0 CH1 */
        TMIF01 = 0;
```

```
        for(; Time > 0; Time--){          /* wait for (parameter * 500)us */
                while(!TMIF01){
                        NOP();
                }
                TMIF01 = 0;
        }

        TT0L.1 = 1;                              /* stop TAU0 CH1 */
}



/*--------------------------------------------------------------*/
/* Module:          fn_AdcDacInit
                    */
/* Description:     Initialization of A/D D/A OPAmp module                    */
/*      parameter: --
                            */
/*      return      : --
                            */
/*--------------------------------------------------------------*/
void    fn_AdcDacInit(void)
{
        ADCEN = 1;                               /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

        /*--------------------------------------------*/
        /*   Initialization of Voltage reference                 */
        /*--------------------------------------------*/
        ADVRC = 0b00001000;           /* Analog reference voltage control register */
                            /*||||+|++---  :  [1]Positive reference voltage supplies selection of A/D and D/A
converters */
                            /*|||| |             [2]Operationcontrol of voltage reference */
                            /*|||| |             [3]Output voltage selection of voltage reference */
                            /*|||| |             [4]Operation control of input gate voltage boost circuit for A/D converter
*/
                            /*|||| |             [5]Relationship with the conversion mode used */
                            /*|||| | */
                            /*|||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
```

```
/*|||| |                    [2]Stops operation (Hi-Z) */
/*|||| |                    [3]2.5 V */
/*|||| |                    [4]Stops operation */
/*|||| |                    [5]Can be set in conversion mode 1 */
/*|||| | */
/*|||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
/*|||| |                    [2]Stops operation (Hi-Z) */
/*|||| |                    [3]2.0 V */
/*|||| |                    [4]Enables operation */
/*|||| |                    [5]Can be set in conversion mode 2 or 3 */
/*|||| | */
/*|||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
/*|||| |                    [2]Stops operation (pull-down output) */
/*|||| |                    [3]2.5 V */
/*|||| |                    [4]Stops operation */
/*|||| |                    [5] - */
/*|||| | */
/*|||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
/*|||| |                    [2]Enables operation */
/*|||| |                    [3]2.5 V */
/*|||| |                    [4]Enables operation */
/*|||| |                    [5]Can be set in conversion mode 2 or 3 */
/*|||| | */
/*|||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
/*|||| |                    [2]Stops operation (pull-down output) */
/*|||| |                    [3]2.0 V */
/*|||| |                    [4]Enables operation */
/*|||| |                    [5] - */
/*|||| | */
/*|||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
/*|||| |                    [2]Enables operation */
/*|||| |                    [3]2.0 V */
/*|||| |                    [4]Enables operation */
/*|||| |                    [5]Can be set in conversion mode 2 or 3 */
/*|||| | */
/*|||| | : Other than the above : Setting prohibited */
/*|||| | */
/*|+++-+----- : Be sure to set 0000 */
/*| */
```

```
                    /*+---------- : Reference voltage supply (negative side) of A/D converter selection */
                    /* : 0 : AVSS */
                    /* : 1 : AVREFM (external voltage reference input) */

ADVRC.0 = 1;                /* Enables operation */
ADVRC.1 = 1;                /* Output 2.0 V */

ADPC =    0b00000000;       /* A/D Port Configuration Register */
                    /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                    /*||| :             +------------------------ ANI15/AVREFM/P157 */
                    /*||| :             | +-+-+------------------ ANI10/P152 - ANI8/AMP2+/P150 */
                    /*||| :             | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
                    /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                    /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                    /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                    /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                    /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                    /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                    /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                    /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                    /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                    /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                    /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                    /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                    /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                    /*||| */
                    /*+++-------- : Be sure to set 000 */

/* wait for settling time to 17msec */
fn_Wait500usBase(17000/500);

/*-------------------------------------------*/
/*    Initialization of operational amplifier    */
/*-------------------------------------------*/
PM2.1 = 1;                          /* input port mode (AMP0O) */
PM2.0 = 1;                          /* input port mode (AMP0-) */
PM2.2 = 1;                          /* input port mode (AMP0+) */

PM2.4 = 1;                          /* input port mode (AMP1O) */
```

```
        PM2.3 = 1;                      /* input port mode (AMP1-) */
        PM2.5 = 1;                      /* input port mode (AMP1+) */

        PM2.7 = 1;                      /* input port mode (AMP2O) */
        PM2.6 = 1;                      /* input port mode (AMP2-) */
        PM15.0 = 1;                     /* input port mode (AMP2+) */

        PM11.0 = 1;                     /* input port mode (ANO0) */

        OAC.0 = 1;                      /* operational amplifier (AMP0) enable */
        OAC.1 = 1;                      /* operational amplifier (AMP1) enable */
        OAC.2 = 1;                      /* operational amplifier (AMP2) enable */

        /*-------------------------------------------*/
        /*      Initialization of D/A converter      */
        /*-------------------------------------------*/
        DACEN = 1;                              /* supplies input clock to D/A converter */

/*      DAMD0 = 1;                              /* [DAMD0] D/A converter operation is Real-time
output mode */
        DAM.0 = 0;                              /* [DAMD0] D/A converter operation is Normal mode
*/
        DAM.2 = 1;                              /* [DARES0]D/A conversion resolution is 12-bit */
        DAM.6 = 1;                              /* [DAREF] D/A converter positive reference voltage
is VREFOUT */
        DACSW0 = CDAC0_52V;             /* D/A conversion value setting reigster 0 */

        /* set D/A chanel1 */
        DAM.1 = 0;                              /* [DAMD1] D/A converter operation is Normal mode
*/
        DAM.3 = 1;                              /* [DARES1]D/A conversion resolution is 12-bit */
        DACSW1 = CDAC0_50V;             /* D/A conversion value setting reigster 1 */

}

/*-------------------------------------------------------------*/
/* Module:         main
                                */
/* Description:    Main process
```

```
        */
/*      parameter: --
                                */
/*      return    : --
                                */
/*--------------------------------------------------------------*/
void main(void)
{
        DACE1 = 1;                      /* D/A converter CH1 enable */
        DACE0 = 1;                      /* D/A converter CH0 enable */



        /*==================================*/
        /*----------------------------------*/
        /*              Main Loop           */
        /*----------------------------------*/
        /*==================================*/

        RTCIF = 0;                      /* reset RCT interrupt request */
        RTCMK = 0;                      /* enable RTC interrupt */
        RTCE  = 1;                      /* starts counter operation */

        EI();                           /* enable all interrupts */

        /*==============================================================*/
        /* if system have nothing to do, go to standby for power-saving */
        /*==============================================================*/

        while (1)
        {
                HALT();
                NOP();
        }
}


/*--------------------------------------------------------------*/
/* Module:      fn_intrtc
                */
```

```
/* Description:      RTC interruput process
        */
/*        parameter: --
                         */
/*        return     : --
                         */
/*-------------------------------------------------------------*/
__interrupt void fn_intrtc(void)
{
        switch(DACSW0)
        {
                case    CDAC0_52V:                      /* DAC0 output voltage : 0.52V */
                        DACSW0 = CDAC0_54V;             /* DAC0 output voltage : 0.52V -> 0.54V */
                        break;
                case    CDAC0_54V:                      /* DAC0 output voltage : 0.54V */
                default:
                        DACSW0 = CDAC0_52V;             /* DAC0 output voltage : 0.54V -> 0.52V */
                        break;
        }

}
```

(8) Comparator circuit

● main.c

```
/*
 * Copyright (C) NEC Electronics Corporation 2006
 * NEC ELECTRONICS CONFIDENTIAL AND PROPRIETARY
 * All rights reserved by NEC Electronics Corporation.
 * This program must be used solely for the purpose for which
 * it was furnished by NEC Electronics Corporation. No part of this
 * program may be reproduced or disclosed to others, in any
 * form, without the prior written permission of NEC Electronics
 * Corporation. Use of copyright notice dose not evidence
 * publication of the program.
 */

/*--------------------------------------------------------------------*/
/* #pragma directive for CC78K0
                                    */
/*--------------------------------------------------------------------*/
#pragma         SFR
#pragma         DI
#pragma         EI
#pragma         HALT
#pragma         NOP

#pragma interrupt INTDMA0 fn_intdma0 RB1
#pragma interrupt INTP0                 fn_intp0 RB2

/*--------------------------------------------------------------------*/
/* Include files
                                                */
/*--------------------------------------------------------------------*/
/* TAU:TDR0n value operation by CK00 (fCLK/2^3 = 5MHz) */
#define CCK00_500USEC    (2500 - 1)          /* 500us (0.2[us/clk] * 2500[count]) */



/*--------------------------------------------------------------------*/
/* Function prototyps
                                                */
```

```
/*--------------------------------------------------------------------*/
            void    fn_Wait500usBase(unsigned short);     /* Delays the program for (Time * 500us) */
            void    fn_InitPort(void);                          /* Setting of I/O ports
*/
            void    fn_InitTau0(void);                          /* Setting of Timer
array unit 0 */
            void    fn_InitLvi(void);                           /* Setting of
Low-voltage detectior */
            void    fn_InitVr(void);                            /* Setting of Voltage
reference */
            void    fn_AdcDacInit(void);                   /* Setting of D/A converter */
            void    fn_DmaInit(void);                           /* Setting of DMA
controler */


/*--------------------------------------------------------------------*/
/* Extern variables/constants
            */
/*--------------------------------------------------------------------*/

/*--------------------------------------------------------------------*/
/* Local constants
                          */
/*--------------------------------------------------------------------*/
/*--------------------------------------------------------------------*/
/* Global variables
                          */
/*--------------------------------------------------------------------*/

/*--------------------------------------------------------------------*/
/* Local variables
                          */
/*--------------------------------------------------------------------*/

/*--------------------------------------------------------------------*/
/* Code
                                      */
/*--------------------------------------------------------------------*/
/*==========================================================================
```

D/A output data

```
==================================================================================*/
static unsigned short ushDacData[100];          /* D/A data for DMA */
const unsigned short aushDacData[100] = {       /* defined D/A data */
                                                /* n       sin(0.2pi*n/100) */
/*              2048,               /* 0        0.100000 */
                2061,               /* 1        0.106279 */
                2074,               /* 2        0.112533 */
                2086,               /* 3        0.118738 */
                2099,               /* 4        0.124869 */
                2111,               /* 5        0.130902 */
                2123,               /* 6        0.136812 */
                2135,               /* 7        0.142578 */
                2147,               /* 8        0.148175 */
                2158,               /* 9        0.153583 */
                2168,               /* 10       0.158779 */
                2179,               /* 11       0.163742 */
                2188,               /* 12       0.168455 */
                2197,               /* 13       0.172897 */
                2206,               /* 14       0.177051 */
                2214,               /* 15       0.180902 */
                2221,               /* 16       0.184433 */
                2227,               /* 17       0.187631 */
                2233,               /* 18       0.190483 */
                2238,               /* 19       0.192978 */
                2243,               /* 20       0.195106 */
                2246,               /* 21       0.196858 */
                2249,               /* 22       0.198229 */
                2251,               /* 23       0.199211 */
                2252,               /* 24       0.199803 */
                2253,               /* 25       0.200000 */
                2252,               /* 26       0.199803 */
                2251,               /* 27       0.199211 */
                2249,               /* 28       0.198229 */
                2246,               /* 29       0.196858 */
                2243,               /* 30       0.195106 */
                2238,               /* 31       0.192978 */
```

```
2233,          /* 32     0.190483 */
2227,          /* 33     0.187631 */
2221,          /* 34     0.184433 */
2214,          /* 35     0.180902 */
2206,          /* 36     0.177051 */
2197,          /* 37     0.172897 */
2188,          /* 38     0.168455 */
2179,          /* 39     0.163742 */
2168,          /* 40     0.158779 */
2158,          /* 41     0.153583 */
2147,          /* 42     0.148175 */
2135,          /* 43     0.142578 */
2123,          /* 44     0.136812 */
2111,          /* 45     0.130902 */
2099,          /* 46     0.124869 */
2086,          /* 47     0.118738 */
2074,          /* 48     0.112533 */
2061,          /* 49     0.106279 */
2048,          /* 50     0.100000 */
2035,          /* 51     0.093721 */
2022,          /* 52     0.087467 */
2010,          /* 53     0.081262 */
1997,          /* 54     0.075131 */
1985,          /* 55     0.069098 */
1973,          /* 56     0.063188 */
1961,          /* 57     0.057422 */
1949,          /* 58     0.051825 */
1938,          /* 59     0.046417 */
1928,          /* 60     0.041221 */
1917,          /* 61     0.036258 */
1908,          /* 62     0.031545 */
1899,          /* 63     0.027103 */
1890,          /* 64     0.022949 */
1882,          /* 65     0.019098 */
1875,          /* 66     0.015567 */
1869,          /* 67     0.012369 */
1863,          /* 68     0.009517 */
1858,          /* 69     0.007022 */
1853,          /* 70     0.004894 */
```

```
        1850,           /* 71     0.003142 */
        1847,           /* 72     0.001771 */
        1845,           /* 73     0.000789 */
        1844,           /* 74     0.000197 */
        1843,           /* 75     0.000000 */
        1844,           /* 76     0.000197 */
        1845,           /* 77     0.000789 */
        1847,           /* 78     0.001771 */
        1850,           /* 79     0.003142 */
        1853,           /* 80     0.004894 */
        1858,           /* 81     0.007022 */
        1863,           /* 82     0.009517 */
        1869,           /* 83     0.012369 */
        1875,           /* 84     0.015567 */
        1882,           /* 85     0.019098 */
        1890,           /* 86     0.022949 */
        1899,           /* 87     0.027103 */
        1908,           /* 88     0.031545 */
        1917,           /* 89     0.036258 */
        1928,           /* 90     0.041221 */
        1938,           /* 91     0.046417 */
        1949,           /* 92     0.051825 */
        1961,           /* 93     0.057422 */
        1973,           /* 94     0.063188 */
        1985,           /* 95     0.069098 */
        1997,           /* 96     0.075131 */
        2010,           /* 97     0.081262 */
        2022,           /* 98     0.087467 */
        2035,           /* 99     0.093721 */
        2048                      /* 100    0.100000 */

};

/*--------------------------------------------------------------------*/
/* Hardware initialization
                                    */
/*--------------------------------------------------------------------*/
void hdwinit(void)
{
```

```
DI();                           /* disable all interrupts */

/*-----------------------------------------*/
/*          Initialization of port         */
/*-----------------------------------------*/
fn_InitPort();

/*----------------------------------------*/
/*         Initialization of clock        */
/*----------------------------------------*/
CMC =   0b01010011;             /* Clock Operation Mode Control Register */
                    /*|||||||+--- : Control of high-speed system clock oscillation frequency */
                    /*||||||| : 0 : 2 MHz <= fMX <= 10 MHz */
                    /*||||||| : 1 : 10 MHz < fMX <= 20 MHz */
                    /*||||||| */
                    /*|||||++---- : XT1 oscillator oscillation mode selection */
                    /*||||| : 0 0 : Low-consumption oscillation */
                    /*||||| : 0 1 : Normal oscillation */
                    /*||||| : 1 x : Super-low-consumption oscillation */
                    /*|||||   x = don't care */
                    /*||||| */
                    /*||||+------ : Be sure to set 0 */
                    /*|||| */
                    /*|||+------- : [1] Subsystem clock pin operation mode */
                    /*|||           [2] XT1/P123 pin and XT2/P124 pin */
                    /*||| : 0 : [1]Input port mode */
                    /*|||         [2]Input port */
                    /*||| */
                    /*||| : 1 : [1]XT1 oscillation mode */
                    /*|||         [2]Crystal resonator connection */
                    /*||| */
                    /*||+-------- : Be sure to set 0 */
                    /*|| */
                    /*++--------- : [1]EXCLK OSCSEL High-speed system clock pin operation mode */
                    /*              [2]X1/P121 pin */
                    /*              [3]X2/EXCLK/P122 pin */
                    /* : 0 0 : [1]Input port mode */
                    /*         [2][3]Input port */
                    /* */
```

```
                    /* : 0 1 : [1]X1 oscillation mode */
                    /*         [2][3]Crystal/ceramic resonator connection */
                    /* */
                    /* : 1 0 : [1]Input port mode */
                    /*         [2][3]Input port */
                    /* */
                    /* : 1 1 : [1]External clock input mode */
                    /*         [2]Input port */
                    /*         [3]External clock input */

    MSTOP = 0;                      /* X1 oscillator operating */

    OSMC =  0b00000001;             /* Operation Speed Mode Control Register */
                    /*||||||++--- : fCLK frequency selection */
                    /*|||||| : 0 0 : Operates at a frequency of 10 MHz or less. */
                    /*|||||| : 0 1 : Operates at a frequency higher than 10 MHz. */
                    /*|||||| : 1 0 : Operates at a frequency of 1 MHz. */
                    /*|||||| */
                    /*|+++++----- : Be sure to set 00000 */
                    /*| */
                    /*+---------- : Setting in subsystem clock HALT mode */
                    /* : 0 : Enables subsystem clock supply to peripheral functions. */
                    /*      (See Table 21-1 Operating Statuses in HALT Mode (2/3) */
                    /*       for the peripheral functions whose operations are enabled.) */
                    /* : 1 : Stops subsystem clock supply to peripheral functions except real-time counter,
*/
                    /*       clock output/buzzer output, and LCD controller/driver. */

    while(OSTC.0 != 1){         /* wait X1 oscillation stabilization */
            NOP();
    }

    /*-- Caution ---------------------------------------------*/
    /* To increase fCLK to 10 MHz or higher, set FSEL to '1',   */
    /* then change fCLK after two or more clocks have elapsed. */
    /*--------------------------------------------------------*/
    NOP();
    NOP();
```

```
CKC =   0b00010000;                 /* System Clock Control Register */
                /*|+|+++++--- : Selection of CPU/peripheral hardware clock (fCLK) */
                /*| | : 0 0 x 0 0 0 : fIH */
                /*| | : 0 0 x 0 0 1 : fIH/2 (default) */
                /*| | : 0 0 x 0 1 0 : fIH/2^2 */
                /*| | : 0 0 x 0 1 1 : fIH/2^3 */
                /*| | : 0 0 x 1 0 0 : fIH/2^4 */
                /*| | : 0 0 x 1 0 1 : fIH/2^5 */
                /*| | : 0 1 x 0 0 0 : fMX */
                /*| | : 0 1 x 0 0 1 : fMX/2 */
                /*| | : 0 1 x 0 1 0 : fMX/2^2 */
                /*| | : 0 1 x 0 1 1 : fMX/2^3 */
                /*| | : 0 1 x 1 0 0 : fMX/2^4 */
                /*| | : 0 1 x 1 0 1 : fMX/2^5 */
                /*| | : 1 x 0 x x x : fSUB */
                /*| | : 1 x 1 x x x : fSUB/2 */
                /*| | : Other than above : Setting prohibited */
                /*| |   x = don't care */
                /*| | */
                /*| +-------- : Status of Main system clock (fMAIN) */
                /*| : 0 : Internal high-speed oscillation clock (fIH) */
                /*| : 1 : High-speed system clock (fMX) */
                /*| */
                /*+---------- : Status of CPU/peripheral hardware clock (fCLK) */
                /* : 0 : Main system clock (fMAIN) */
                /* : 1 : Subsystem clock (fSUB) */

/* Confirming the CPU clock status */
while((CLS != 0)||(MCS != 1)){
        NOP();
}
/* CPU is operating on a High-speed system clock */
HIOSTOP = 1;                        /* internal high-speed oscillation stopped */

OSTS =  0b00000111;                 /* Oscillation Stabilization Time Select Register */
                /*||||++++--- : Oscillation stabilization time selection */
                /*|||| : 0 0 0 : 2^8/fX */
                /*|||| : 0 0 1 : 2^9/fX */
                /*|||| : 0 1 0 : 2^10/fX */
```

```
                        /*|||| : 0 1 1 : 2^11/fX */
                        /*|||| : 1 0 0 : 2^13/fX */
                        /*|||| : 1 0 1 : 2^15/fX */
                        /*|||| : 1 1 0 : 2^17/fX */
                        /*|||| : 1 1 1 : 2^18/fX */
                        /*|||| */
                        /*++++------- : Be sure to set 000000 */

        /*----------------------------------------*/
        /*          Initialization of timer          */
        /*----------------------------------------*/
        fn_InitTau0();

        /* software to wait for the operation stabilization time */
        /* (over 200ms from when XT1 enable) */
        fn_Wait500usBase(200000/500);

        /*----------------------------------------*/
        /* Initialization of low-voltage detector */
        /*----------------------------------------*/
        fn_InitLvi();

        /*----------------------------------------*/
        /*   Initialization of A/D,D/A,OPAmp                */
        /*----------------------------------------*/
        fn_AdcDacInit();

        /*----------------------------------------*/
        /*   Initialization of DMA                              */
        /*----------------------------------------*/
        fn_DmaInit();

        EI();                                   /* enable all interrupts */

}

/*------------------------------------------------------------*/
/* Module:          fn_InitPort
                    */
```

```
/* Description:    Setting of I/O ports                                              */
/*        parameter: --
                        */
/*        return    : --
                        */
/*-------------------------------------------------------------*/
void    fn_InitPort(void)
{
/*----------------------------------------------------------*/
/*        Ports configuration for digital input and output             */
/*----------------------------------------------------------*/
        ADPC =    0b00010000;        /* A/D Port Configuration Register */
                        /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                        /*||| :              +------------------------ ANI15/AVREFM/P157 */
                        /*||| :              | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                        /*||| :              | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
                        /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                        /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                        /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                        /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                        /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                        /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                        /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                        /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                        /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                        /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                        /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                        /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                        /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                        /*||| */
                        /*+++-------- : Be sure to set 000 */

/*----------------------------------------------------------*/
/*        Setting of Port 0
        */
/*----------------------------------------------------------*/
        P0 =      0b00000000;        /* Set P00-P02 Output latch to Low */
        PM0 =     0b11111000;        /* Set P00-P02 to output port */
                                                /* P00-P02:Unused */
```

```
/*------------------------------------------------------------*/
/*      Setting of Port 1
        */
/*------------------------------------------------------------*/
        P1 =    0b00000000;     /* Set P10-P17 Output latch to Low */
        PM1 =   0b00000000;     /* Set P10-P17 to output port */
                                        /* P10-P15:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 2
        */
/*------------------------------------------------------------*/
        P2 =    0b00000000;     /* Set P20-P27 Output latch to Low */
        PM2 =   0b11111111;     /* Set P20-P27 to input port */
                                        /* P20-P27:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 3
        */
/*------------------------------------------------------------*/
        P3 =    0b00001100;     /* Set P30-P31,P34 Output latch to Low */
                                        /* Set P33,P32 Output latch High */
        PM3 =   0b11100000;     /* Set P30-P34 to output port */
                                        /* P30-P34:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 4
        */
/*------------------------------------------------------------*/
        P4 =    0b00000000;     /* Set P40-P41 Output latch to Low */
        PM4 =   0b11111100;     /* Set P40-P41 to output port */
                                        /* P40-P41:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 5
        */
/*------------------------------------------------------------*/
        P5 =    0b00000000;     /* Set P50-P57 Output latch to Low */
```

```
        PM5 =      0b11110000;       /* Set P50-P57 to output port */
                                                         /* P50-P57:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 6
        */
/*------------------------------------------------------------*/
        P6 =       0b00000000;       /* Set P60-P61 Output latch to Low */
        PM6 =      0b11111100;       /* Set P60-P61 to output port */
                                                         /* P60-P61:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 7
        */
/*------------------------------------------------------------*/
        P7 =       0b00000000;       /* Set P70-P77 Output latch to Low */
        PM7 =      0b00000000;       /* Set P70-P77 to output port */
                                                         /* P70-P77:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 8
        */
/*------------------------------------------------------------*/
        P8 =       0b00000000;       /* Set P80-P88 Output latch to Low */
        PM8 =      0b00000000;       /* Set P80-P88 to output port */
                                                         /* P80-P88:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 9
        */
/*------------------------------------------------------------*/
        P9 =       0b00000000;       /* Set P90-P97 Output latch to Low */
        PM9 =      0b00000000;       /* Set P90-P97 to output port */
                                                         /* P90-P97:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 10
        */
/*------------------------------------------------------------*/
```

```
        P10 =   0b00000000;     /* Set P100-P102 Output latch to Low */
        PM10 =  0b11111000;     /* Set P100-P102 to output port */
                                                /* P100-P102:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 11
        */
/*-----------------------------------------------------------*/
        P11 =   0b00000000;     /* Set P110-P111 Output latch to Low */
        PM11 =  0b11111100;     /* Set P110-P111 to output port */
                                                /* P110-P111:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 12
        */
/*-----------------------------------------------------------*/
        P12 =   0b00000000;     /* Set P120 Output latch to Low */
        PM12 =  0b11111110;     /* Set P120 to output port */
                                                /* P120-P124:Unused */
                                                /* *P121-P124:Input port */

/*-----------------------------------------------------------*/
/*      Setting of Port 13
        */
/*-----------------------------------------------------------*/
        P13 =   0b00000000;     /* Set P130 Output latch to Low */
                                                /* P130:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 14
        */
/*-----------------------------------------------------------*/
        P14 =   0b00000000;     /* Set P140-P147 Output latch to Low */
        PM14 =  0b00000000;     /* Set P140-P147 to output port */
                                                /* P140-P147:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 15
        */
```

```
/*----------------------------------------------------------*/
        P15 =    0b00000000;         /* Set P150-P152,P157 Output latch to Low */
        PM15 =   0b11111111;         /* Set P150-P152,P157 to input port */
                                                     /* P150-P152,P157:Unused */

}

/*--------------------------------------------------------------*/
/* Module:          fn_InitTau0
                                */
/* Description:     Setting of Timer array unit 0                                    */
/*      parameter: --
                                */
/*      return    : --
                                */
/*--------------------------------------------------------------*/
void    fn_InitTau0(void)
{
        TAU0EN = 1;                          /* supplies input clock to timer array unit 0 */
        TPS0L =  0b00000010;         /* Timer Clock Select Register 0 */
                                /*||||++++--- : Selection of operation clock (CK00) */
                                /*++++------- : Selection of operation clock (CK01) */
                                /* : 0 0 0 0 : CK0m = fCLK */
                                /* : 0 0 0 1 : CK0m = fCLK/2 */
                                /* : 0 0 1 0 : CK0m = fCLK/2^2 */
                                /* : 0 0 1 1 : CK0m = fCLK/2^3 */
                                /* : 0 1 0 0 : CK0m = fCLK/2^4 */
                                /* : 0 1 0 1 : CK0m = fCLK/2^5 */
                                /* : 0 1 1 0 : CK0m = fCLK/2^6 */
                                /* : 0 1 1 1 : CK0m = fCLK/2^7 */
                                /* : 1 0 0 0 : CK0m = fCLK/2^8 */
                                /* : 1 0 0 1 : CK0m = fCLK/2^9 */
                                /* : 1 0 1 0 : CK0m = fCLK/2^10 */
                                /* : 1 0 1 1 : CK0m = fCLK/2^11 */
                                /* : 1 1 0 0 : CK0m = fCLK/2^12 */
                                /* : 1 1 0 1 : CK0m = fCLK/2^13 */
                                /* : 1 1 1 0 : CK0m = fCLK/2^14 */
                                /* : 1 1 1 1 : CK0m = fCLK/2^15 */
                                /*    m = 0, 1 */
```

```
/* CH1:for wait */
TMR01 = 0b0000000000000000;         /* Timer Mode Register 01 */
                        /*||||||||||||++++--- : [1]Operation mode of channel 1 */
                        /*||||||||||||          [2]Count operation of TCR */
                        /*||||||||||||          [3]Independent operation */
                        /*||||||||||||          [4]Setting of starting counting and interrupt */
                        /*|||||||||||| : 0 0 0 0 : [1]Interval timer mode */
                        /*||||||||||||             [2]Counting down */
                        /*||||||||||||             [3]Possible */
                        /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
                        /*||||||||||||                (timer output does not change, either). */
                        /*|||||||||||| */
                        /*|||||||||||| : 0 0 0 1 : [1]Interval timer mode */
                        /*||||||||||||             [2]Counting down */
                        /*||||||||||||             [3]Possible */
                        /*||||||||||||             [4]Timer interrupt is generated when counting is started */
                        /*||||||||||||                (timer output also changes). */
                        /*|||||||||||| */
                        /*|||||||||||| : 0 1 0 0 : [1]Capture mode */
                        /*||||||||||||             [2]Counting up */
                        /*||||||||||||             [3]Possible */
                        /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
                        /*||||||||||||                (timer output does not change, either). */
                        /*|||||||||||| */
                        /*|||||||||||| : 0 1 0 1 : [1]Capture mode */
                        /*||||||||||||             [2]Counting up */
                        /*||||||||||||             [3]Possible */
                        /*||||||||||||             [4]Timer interrupt is generated when counting is started */
                        /*||||||||||||                (timer output also changes). */
                        /*|||||||||||| */
                        /*|||||||||||| : 0 1 1 0 : [1]Event counter mode */
                        /*||||||||||||             [2]Counting down */
                        /*||||||||||||             [3]Possible */
                        /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
                        /*||||||||||||                (timer output does not change, either). */
                        /*|||||||||||| */
                        /*|||||||||||| : 1 0 0 0 : [1]One-count mode */
                        /*||||||||||||             [2]Counting down */
```

```
/*||||||||||||                    [3]Impossible */
/*||||||||||||                    [4]Start trigger is invalid during counting operation. */
/*||||||||||||                       At that time, interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 1 : [1]One-count mode */
/*||||||||||||                    [2]Counting down */
/*||||||||||||                    [3]Impossible */
/*||||||||||||                    [4]Start trigger is valid during counting operation. */
/*||||||||||||                       At that time, interrupt is also generated. */
/*|||||||||||| */
/*|||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*||||||||||||                    [2]Counting up */
/*||||||||||||                    [3]Possible */
/*||||||||||||                    [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                       (timer output does not change, either). */
/*||||||||||||                       Start trigger is invalid during counting operation. */
/*||||||||||||                       At that time interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : Other than above : Setting prohibited */
/*|||||||||||| */
/*||||||||||++------- : Be sure to set 00 */
/*|||||||||| */
/*||||||||++--------- : Selection of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 valid edge */
/*||||||||                  (the timer input used with channel 1 is selected by using TIS0 register). */
/*|||||||| : 0 0 : Falling edge */
/*|||||||| : 0 1 : Rising edge */
/*|||||||| : 1 0 : Both edges (when low-level width is measured) */
/*||||||||           Start trigger: Falling edge, Capture trigger: Rising edge */
/*|||||||| : 1 1 : Both edges (when high-level width is measured) */
/*||||||||           Start trigger: Rising edge, Capture trigger: Falling edge */
/*|||||||| */
/*|||||+++----------- : Setting of start trigger or capture trigger of channel 1 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are unselected). */
/*||||| : 0 0 1 : Valid edge of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 is used as both the start trigger and capture trigger. */
/*||||| : 0 1 0 : Both the edges of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 are
```

```
used as a start trigger and a capture trigger. */
                              /*||||| : 1 0 0 : Interrupt signal of the master channel is used (when the channel is used
as a slave channel with the combination operation function). */
                              /*||||| : Other than above : Setting prohibited */
                              /*||||| */
                              /*||||+-------------- : Selection of slave/master of channel 1 */
                              /*|||| : 0 : Operates as slave channel with combination operation function. */
                              /*|||| : 1 : Operates as master channel with combination operation function. */
                              /*|||| */
                              /*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
                              /*||| : 0 : Operation clock MCK specified by CKS01 bit */
                              /*||| : 1 : Valid edge of input signal input from TI01 pin, fSUB/2, fSUB/4, or INTRTC1 */
                              /*|||       (the timer input used with channel 1 is selected by using TIS0 register). */
                              /*||| */
                              /*|++---------------- : Be sure to set 00 */
                              /*| */
                              /*+------------------ : Selection of operation clock (MCK) of channel 1 */
                              /* : 0 : Operation clock CK00 set by TPS0 register */
                              /* : 1 : Operation clock CK01 set by TPS0 register */
        TDR01 = CCK00_500USEC;          /* set interval time to 500us */
        TMMK01 = 1;                                     /* disable interrupt */

        /* CH4:for DMA */
        TMR04 = 0b1000000000000000;     /* Timer Mode Register 04 */
                              /*||||||||||||++++--- : [1]Operation mode of channel 4 */
                              /*||||||||||||        [2]Count operation of TCR */
                              /*||||||||||||        [3]Independent operation */
                              /*||||||||||||        [4]Setting of starting counting and interrupt */
                              /*|||||||||||| : 0 0 0 0 : [1]Interval timer mode */
                              /*||||||||||||             [2]Counting down */
                              /*||||||||||||             [3]Possible */
                              /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
                              /*||||||||||||                (timer output does not change, either). */
                              /*|||||||||||| */
                              /*|||||||||||| : 0 0 0 1 : [1]Interval timer mode */
                              /*||||||||||||             [2]Counting down */
                              /*||||||||||||             [3]Possible */
                              /*||||||||||||             [4]Timer interrupt is generated when counting is started */
                              /*||||||||||||                (timer output also changes). */
```

```
/*|||||||||||| */
/*|||||||||||| : 0 1 0 0 : [1]Capture mode */
/*||||||||||||             [2]Counting up */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 0 1 0 1 : [1]Capture mode */
/*||||||||||||             [2]Counting up */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is generated when counting is started */
/*||||||||||||                (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 1 0 : [1]Event counter mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 0 : [1]One-count mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Impossible */
/*||||||||||||             [4]Start trigger is invalid during counting operation. */
/*||||||||||||                At that time, interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 1 : [1]One-count mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Impossible */
/*||||||||||||             [4]Start trigger is valid during counting operation. */
/*||||||||||||                At that time, interrupt is also generated. */
/*|||||||||||| */
/*|||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*||||||||||||             [2]Counting up */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                (timer output does not change, either). */
/*||||||||||||                Start trigger is invalid during counting operation. */
/*||||||||||||                At that time interrupt is not generated, either. */
/*|||||||||||| */
```

/*|||||||||||| : Other than above : Setting prohibited */
/*|||||||||||| */
/*||||||||||++------- : Be sure to set 00 */
/*|||||||||| */
/*||||||||++--------- : Selection of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 valid edge */
/*|||||||| (the timer input used with channel 4 is selected by using TIS0 register). */
/*|||||||| : 0 0 : Falling edge */
/*|||||||| : 0 1 : Rising edge */
/*|||||||| : 1 0 : Both edges (when low-level width is measured) */
/*|||||||| Start trigger: Falling edge, Capture trigger: Rising edge */
/*|||||||| : 1 1 : Both edges (when high-level width is measured) */
/*|||||||| Start trigger: Rising edge, Capture trigger: Falling edge */
/*|||||||| */
/*|||||+++----------- : Setting of start trigger or capture trigger of channel 4 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are unselected). */
/*||||| : 0 0 1 : Valid edge of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 is used as both the start trigger and capture trigger. */
/*||||| : 0 1 0 : Both the edges of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 are used as a start trigger and a capture trigger. */
/*||||| : 1 0 0 : Interrupt signal of the master channel is used (when the channel is used as a slave channel with the combination operation function). */
/*||||| : Other than above : Setting prohibited */
/*||||| */
/*||||+-------------- : Selection of slave/master of channel 4 */
/*|||| : 0 : Operates as slave channel with combination operation function. */
/*|||| : 1 : Operates as master channel with combination operation function. */
/*|||| */
/*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
/*||| : 0 : Operation clock MCK specified by CKS04 bit */
/*||| : 1 : Valid edge of input signal input from TI04 pin, fSUB/2, fSUB/4, or INTRTC1 */
/*||| (the timer input used with channel 4 is selected by using TIS0 register). */
/*||| */
/*|++---------------- : Be sure to set 00 */
/*| */
/*+------------------ : Selection of operation clock (MCK) of channel 4 */
/* : 0 : Operation clock CK00 set by TPS0 register */

```
                        /* : 1 : Operation clock CK01 set by TPS0 register */
    TDR04 = 200;        /* set interval time to 10us(=0.05us * 200) */
    TMMK04 = 1;                 /* disable interrupt */

}

/*------------------------------------------------------------*/
/* Module:          fn_InitLvi
                    */
/* Description:     Setting of Low-voltage detectior                    */
/*      parameter: --
                        */
/*      return     : --
                        */
/*------------------------------------------------------------*/
void    fn_InitLvi(void)
{
    unsigned short      loop;       /* waiting counter */

    LVIMK = 1;                              /* disable LVI interrupt */

    LVIS =      0b00001001;                 /* Low-Voltage Detection Level Select Register */
                        /*||||++++--- : Detection level */
                        /*|||| : 0 0 0 0 : VLVI0 (4.22V) */
                        /*|||| : 0 0 0 1 : VLVI1 (4.07V) */
                        /*|||| : 0 0 1 0 : VLVI2 (3.92V) */
                        /*|||| : 0 0 1 1 : VLVI3 (3.76V) */
                        /*|||| : 0 1 0 0 : VLVI4 (3.61V) */
                        /*|||| : 0 1 0 1 : VLVI5 (3.45V) */
                        /*|||| : 0 1 1 0 : VLVI6 (3.30V) */
                        /*|||| : 0 1 1 1 : VLVI7 (3.15V) */
                        /*|||| : 1 0 0 0 : VLVI8 (2.99V) */
                        /*|||| : 1 0 0 1 : VLVI9 (2.84V) */
                        /*|||| : 1 0 1 0 : VLVI10 (2.68V) */
                        /*|||| : 1 0 1 1 : VLVI11 (2.53V) */
                        /*|||| : 1 1 0 0 : VLVI12 (2.38V) */
                        /*|||| : 1 1 0 1 : VLVI13 (2.22V) */
                        /*|||| : 1 1 1 0 : VLVI14 (2.07V) */
                        /*|||| : 1 1 1 1 : VLVI15 (1.91V) */
```

```
                    /*|||| */
                    /*++++------- : Be sure to set 0000 */

    LVIM =    0b10000000;          /* Low-Voltage Detection Register */
                    /*|||||||+--- : LVIF Low-voltage detection flag */
                    /*||||||| : 0 : * LVISEL = 0: VDD >= VLVI, or when LVI operation is disabled */
                    /*|||||||       * LVISEL = 1: EXLVI >= VEXLVI, or when LVI operation is disabled */
                    /*||||||| : 1 : * LVISEL = 0: VDD < VLVI */
                    /*|||||||       * LVISEL = 1: EXLVI < VEXLVI */
                    /*||||||| */
                    /*||||||+---- : Low-voltage detection operation mode (interrupt/reset) selection(LVIMD) */
                    /*|||||| : 0 : * LVISEL = 0: Generates an internal interrupt signal */
                    /*||||||                 when VDD drops lower than VLVI (VDD < VLVI) */
                    /*||||||                 or when VDD becomes VLVI or higher (VDD >= VLVI). */
                    /*||||||       * LVISEL = 1: Generates an interrupt signal */
                    /*||||||                   when EXLVI drops lower than VEXLVI (EXLVI <
VEXLVI) */
                    /*||||||                 or when EXLVI becomes VEXLVI or higher (EXLVI >=
VEXLVI). */
                    /*|||||| : 1 : * LVISEL = 0: Generates an internal reset signal when VDD < VLVI */
                    /*||||||                 and releases the reset signal when VDD >= VLVI. */
                    /*||||||       * LVISEL = 1: Generates an internal reset signal when EXLVI < VEXLVI
*/
                    /*||||||                 and releases the reset signal when EXLVI >= VEXLVI. */
                    /*|||||| */
                    /*|||||+----- : Voltage detection selection(LVISEL) */
                    /*||||| : 0 : Detects level of supply voltage (VDD) */
                    /*||||| : 1 : Detects level of input voltage from external input pin (EXLVI) */
                    /*||||| */
                    /*|++++------ : Be sure to set 0000 */
                    /*| */
                    /*+---------- : Enables low-voltage detection operation */
                    /* : 0 : Disables operation */
                    /* : 1 : Enables operation */

    /* software to wait for the operation stabilization time (210us) */
    for(loop = 500; loop > 0; loop--){
        NOP();
    }
```

```
        /* wait until VLVI≤VDD */
        while( LVIF ){
                NOP();
        }

        LVIIF = 0;                      /* clear LVI interrupt request flag */
}



/*------------------------------------------------------------*/
/* Module:         fn_DmaInit
                                */
/* Description:    Setting of DMA contoroler                                    */
/*      parameter: --
                                */
/*      return    : --
                                */
/*------------------------------------------------------------*/
void    fn_DmaInit(void)
{

        DEN0 = 1;                               /* Enables operation of DMA channel 0 */

        DSA0 = (unsigned char)&DACSW0;                          /* DMA SFR Address Register 0
*/
        DRA0 = (unsigned short)&ushDacData[0];          /* DMA RAM Address Register 0 */
        DBC0 = 100;                                     /* DMA Byte Count Register 0 */
        DMC0 = 0b01100100;                      /* DMA Mode Control Register 0 */
                /*|||||||| */
                /*||||++++------- IFC3-IFC0 */
                /*||||            [Selection of DMA start source] */
                /*||||             0000:Disables DMA transfer by interrupt */
                /*||||                          (Only software trigger is enabled.) */
                /*||||             0010:INTTM00 */
                /*||||             0011:INTTM01 */
                /*||||             0100:INTTM04 */
                /*||||             0101:INTTM05 */
                /*||||             0110:INTST0/INTCSI00 */
                /*||||             0111:INTSR0/INTCSI01 */
```

```
                    /*||||              1000:INTST1/INTCSI10/INTIIC10 */
                    /*||||              1001:INTSR1 */
                    /*||||              1010:INTST3 */
                    /*||||              1011:INTSR3 */
                    /*||||              1100:INTAD */
                    /*|||+----------- DWAIT0 */
                    /*|||              [Pending of DMA transfer] */
                    /*|||               0:Executes DMA transfer upon DMA start request (not held pending). */
                    /*|||               1:Holds DMA start request pending if any. */
                    /*||+------------ DS0 */
                    /*||               [Specification of transfer data size for DMA transfer] */
                    /*||                0:8bits */
                    /*||                1:16bits */
                    /*|+------------- DRS0 */
                    /*|                [Selection of DMA transfer direction] */
                    /*|                 0:SFR to internal RAM */
                    /*|                 1:Internal RAM to SFR */
                    /*+------------- STG0 */
                    /*                 [DMA transfer start software trigger] */
                    /*                  0:No trigger operation */
                    /*                  1:DMA transfer is started when DMA operation is enabled (DEN0 = 1). */

        DST0 = 1;                       /* DMA transfer is started when DMA operation is enabled (DEN0 = 1). */

}


/*--------------------------------------------------------------*/
/* Module:         fn_Wait500usBase
        */
/* Description:    Delays the program for (Time * 500us)                    */
/*      parameter: wait time(Time)
        */
/*      return    : --
                              */
/*--------------------------------------------------------------*/
void    fn_Wait500usBase(unsigned short Time)
{
        TS0L.1 = 1;                             /* start TAU0 CH1 */
```

```
        TMIF01 = 0;

        for(; Time > 0; Time--){        /* wait for (parameter * 500)us */
                while(!TMIF01){
                        NOP();
                }
                TMIF01 = 0;
        }

        TT0L.1 = 1;                             /* stop TAU0 CH1 */
}



/*--------------------------------------------------------------*/
/* Module:          fn_AdcDacInit
                    */
/* Description:     Initialization of A/D D/A OPAmp module                    */
/*      parameter: --
                            */
/*      return    : --
                            */
/*--------------------------------------------------------------*/
void    fn_AdcDacInit(void)
{
        ADCEN = 1;                              /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

        /*------------------------------------------*/
        /*   Initialization of Voltage reference                     */
        /*------------------------------------------*/
        ADVRC = 0b00001000;         /* Analog reference voltage control register */
                        /*||||+|++---  :  [1]Positive reference voltage supplies selection of A/D and D/A
converters */
                        /*|||| |            [2]Operationcontrol of voltage reference */
                        /*|||| |            [3]Output voltage selection of voltage reference */
                        /*|||| |            [4]Operation control of input gate voltage boost circuit for A/D converter
*/
                        /*|||| |            [5]Relationship with the conversion mode used */
                        /*|||| | */
```

```
/*|||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
/*|||| |             [2]Stops operation (Hi-Z) */
/*|||| |             [3]2.5 V */
/*|||| |             [4]Stops operation */
/*|||| |             [5]Can be set in conversion mode 1 */
/*|||| | */
/*|||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
/*|||| |             [2]Stops operation (Hi-Z) */
/*|||| |             [3]2.0 V */
/*|||| |             [4]Enables operation */
/*|||| |             [5]Can be set in conversion mode 2 or 3 */
/*|||| | */
/*|||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
/*|||| |             [2]Stops operation (pull-down output) */
/*|||| |             [3]2.5 V */
/*|||| |             [4]Stops operation */
/*|||| |             [5] - */
/*|||| | */
/*|||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
/*|||| |             [2]Enables operation */
/*|||| |             [3]2.5 V */
/*|||| |             [4]Enables operation */
/*|||| |             [5]Can be set in conversion mode 2 or 3 */
/*|||| | */
/*|||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
/*|||| |             [2]Stops operation (pull-down output) */
/*|||| |             [3]2.0 V */
/*|||| |             [4]Enables operation */
/*|||| |             [5] - */
/*|||| | */
/*|||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
/*|||| |             [2]Enables operation */
/*|||| |             [3]2.0 V */
/*|||| |             [4]Enables operation */
/*|||| |             [5]Can be set in conversion mode 2 or 3 */
/*|||| | */
/*|||| | : Other than the above : Setting prohibited */
/*|||| | */
/*|+++-+----- : Be sure to set 0000 */
```

```
                    /*| */
                    /*+---------- : Reference voltage supply (negative side) of A/D converter selection */
                    /* : 0 : AVSS */
                    /* : 1 : AVREFM (external voltage reference input) */

    ADVRC.0 = 1;              /* Enables operation */
    ADVRC.1 = 1;              /* Output 2.0 V */

    ADPC =    0b00000000;     /* A/D Port Configuration Register */
                    /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                    /*||| :               +------------------------ ANI15/AVREFM/P157 */
                    /*||| :               | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                    /*||| :               | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
                    /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                    /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                    /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                    /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                    /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                    /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                    /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                    /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                    /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                    /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                    /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                    /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                    /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                    /*||| */
                    /*+++-------- : Be sure to set 000 */

    /* wait for settling time to 17msec */
    fn_Wait500usBase(17000/500);

    /*-------------------------------------------*/
    /*    Initialization of operational amplifier    */
    /*-------------------------------------------*/
    PM2.1 = 1;                          /* input port mode (AMP0O) */
    PM2.0 = 1;                          /* input port mode (AMP0-) */
    PM2.2 = 1;                          /* input port mode (AMP0+) */
```

```
        PM2.4 = 1;                  /* input port mode (AMP1O) */
        PM2.3 = 1;                  /* input port mode (AMP1-) */
        PM2.5 = 1;                  /* input port mode (AMP1+) */

        PM2.7 = 1;                  /* input port mode (AMP2O) */
        PM2.6 = 1;                  /* input port mode (AMP2-) */
        PM15.0 = 1;                 /* input port mode (AMP2+) */

        PM11.0 = 1;                 /* input port mode (ANO0) */

        ADCEN = 1;                  /* supplies input clock to OPAmp A/D converter */
        OAC.0 = 1;                  /* operational amplifier (AMP0) enable */
        OAC.1 = 0;                  /* operational amplifier (AMP1) disable */
        OAC.2 = 0;                  /* operational amplifier (AMP2) disable */

        /*------------------------------------------*/
        /*      Initialization of D/A converter     */
        /*------------------------------------------*/
        DACEN = 1;                          /* supplies input clock to D/A converter */

    /*  DAMD0 = 1;                          /* [DAMD0] D/A converter operation is Real-time
output mode */
        DAM.0 = 0;                          /* [DAMD0] D/A converter operation is Normal mode
*/
        DAM.2 = 1;                          /* [DARES0]D/A conversion resolution is 12-bit */
        DAM.6 = 1;                          /* [DAREF] D/A converter positive reference voltage
is VREFOUT */
        DACSW0 = aushDacData[99];   /* D/A conversion value setting reigster 0 */

        /* set D/A chanel1 */
        DAM.1 = 0;                          /* [DAMD1] D/A converter operation is Normal mode
*/
        DAM.3 = 1;                          /* [DARES1]D/A conversion resolution is 12-bit */
        DACSW1 = aushDacData[99];   /* D/A conversion value setting reigster 1 */

    }

    /*------------------------------------------------------------*/
    /* Module:         main
```

```
                                        */
/* Description:    Main process
          */
/*        parameter: --
                                        */
/*        return    : --
                                        */
/*-------------------------------------------------------------*/
void main(void)
{
        unsigned short       count;              /* counter for initializing D/A convert data */

        /* Port setting */
        P12.0 = 0;
        PM12.0 = 1;
        P7.3 = 0;
        PM7.3 = 0;

        /* INTP0 pin valid edge selection */
        EGP0.0 = 1;                                         /* Both rising and falling edges */
        EGN0.0 = 1;

        DACE1 = 1;                                          /* D/A converter CH1 enable */
        DACE0 = 1;                                          /* D/A converter CH0 enable */


        /*==================================*/
        /*----------------------------------*/
        /*              Main Loop           */
        /*----------------------------------*/
        /*==================================*/

        /*   initialize D/Aconverter data */
        for(count = 0; count<100; count++){
                ushDacData[count] = aushDacData[count];
        }

        TS0L.4 = 1;                              /* start TAU0 CH4 */
        TMIF04 = 0;                              /* reset INTTM04 interrupt request */
```

```
        DMAMK0 = 0;                         /* enable DMA CH0 interrupt */

        PIF0 = 0;                   /* reset INTP0 interrupt request */
        PMK0 = 0;                   /* enable INTP0 interrupt */

        EI();                               /* enable all interrupts */

        /*=============================================================*/
        /* if system have nothing to do, go to standby for power-saving */
        /*=============================================================*/


        while (1){
                HALT();
                NOP();
        }
  }



  /*------------------------------------------------------------*/
  /* Module:        fn_intdma0
                              */
  /* Description:   DMA ch0 interruput process                                          */
  /*      parameter: --
                              */
  /*      return    : --
                              */
  /*------------------------------------------------------------*/
  __interrupt void fn_intdma0(void)
  {
        DRA0 = (unsigned short)&ushDacData[0];          /* DMA RAM Address Register 0 */
        DBC0 = 100;                                             /* DMA Byte Count Register 0 */

        DST0 = 1;                           /* DMA transfer is started when DMA operation is enabled
(DEN0 = 1). */
  }

  /*------------------------------------------------------------*/
```

```
/* Module:          fn_intp0
                    */
/* Description:     INTP0 interruput process                                  */
/*       parameter: --
                              */
/*       return    : --
                              */
/*-------------------------------------------------------------*/
__interrupt void fn_intp0(void)
{
        P7.3 = P12.0;
}
```

(9) CR oscillator 1

● main.c

```
/*
 * Copyright (C) NEC Electronics Corporation 2006
 * NEC ELECTRONICS CONFIDENTIAL AND PROPRIETARY
 * All rights reserved by NEC Electronics Corporation.
 * This program must be used solely for the purpose for which
 * it was furnished by NEC Electronics Corporation. No part of this
 * program may be reproduced or disclosed to others, in any
 * form, without the prior written permission of NEC Electronics
 * Corporation. Use of copyright notice dose not evidence
 * publication of the program.
 */

/*--------------------------------------------------------------------*/
/* #pragma directive for CC78K0
                                        */
/*--------------------------------------------------------------------*/
#pragma         SFR
#pragma         DI
#pragma         EI
#pragma         HALT
#pragma         NOP

/*--------------------------------------------------------------------*/
/* Include files
                                                      */
/*--------------------------------------------------------------------*/
/* TAU:TDR0n value operation by CK00 (fCLK/2^3 = 5MHz) */
#define CCK00_500USEC   (2500 - 1)          /* 500us (0.2[us/clk] * 2500[count]) */



/*--------------------------------------------------------------------*/
/* Function prototyps
                                                      */
/*--------------------------------------------------------------------*/
                void    fn_Wait500usBase(unsigned short);       /* Delays the program for (Time * 500us) */
                void    fn_InitPort(void);                                  /* Setting of I/O ports
```

```
*/
                void    fn_InitTau0(void);                          /* Setting of Timer
array unit 0 */
                void    fn_InitLvi(void);                           /*    Setting    of
Low-voltage detectior */
                void    fn_InitVr(void);                            /* Setting of Voltage
reference */
                void    fn_AdcDacInit(void);                /* Setting of D/A converter */

  extern  void  fn_DisplayInit(void);                 /* Setting of LCD driver */
  extern  void  fn_Display(unsigned short ushCount);/* LCD display */

  /*----------------------------------------------------------------------*/
  /* Extern variables/constants
                */
  /*----------------------------------------------------------------------*/

  /*----------------------------------------------------------------------*/
  /* Local constants
                              */
  /*----------------------------------------------------------------------*/
  /*----------------------------------------------------------------------*/
  /* Global variables
                              */
  /*----------------------------------------------------------------------*/

  /*----------------------------------------------------------------------*/
  /* Local variables
                              */
  /*----------------------------------------------------------------------*/

  /*----------------------------------------------------------------------*/
  /* Code
                                          */
  /*----------------------------------------------------------------------*/

  /*----------------------------------------------------------------------*/
  /* Hardware initialization
                    */
```

```
/*----------------------------------------------------------------------*/
void hdwinit(void)
{
        DI();                                   /* disable all interrupts */

        /*-----------------------------------------*/
        /*              Initialization of port                  */
        /*-----------------------------------------*/
        fn_InitPort();

        /*-----------------------------------------*/
        /*              Initialization of clock                 */
        /*-----------------------------------------*/
        CMC =   0b01010011;                     /* Clock Operation Mode Control Register */
                        /*|||||||+--- : Control of high-speed system clock oscillation frequency */
                        /*||||||| : 0 : 2 MHz <= fMX <= 10 MHz */
                        /*||||||| : 1 : 10 MHz < fMX <= 20 MHz */
                        /*||||||| */
                        /*|||||++---- : XT1 oscillator oscillation mode selection */
                        /*||||| : 0 0 : Low-consumption oscillation */
                        /*||||| : 0 1 : Normal oscillation */
                        /*||||| : 1 x : Super-low-consumption oscillation */
                        /*|||||   x = don't care */
                        /*||||| */
                        /*||||+------ : Be sure to set 0 */
                        /*|||| */
                        /*|||+------- : [1] Subsystem clock pin operation mode */
                        /*|||               [2] XT1/P123 pin and XT2/P124 pin */
                        /*||| : 0 : [1]Input port mode */
                        /*|||         [2]Input port */
                        /*||| */
                        /*||| : 1 : [1]XT1 oscillation mode */
                        /*|||         [2]Crystal resonator connection */
                        /*||| */
                        /*||+-------- : Be sure to set 0 */
                        /*|| */
                        /*++--------- : [1]EXCLK OSCSEL High-speed system clock pin operation mode */
                        /*                  [2]X1/P121 pin */
                        /*                  [3]X2/EXCLK/P122 pin */
```

```
                    /* : 0 0 : [1]Input port mode */
                    /*         [2][3]Input port */
                    /* */
                    /* : 0 1 : [1]X1 oscillation mode */
                    /*         [2][3]Crystal/ceramic resonator connection */
                    /* */
                    /* : 1 0 : [1]Input port mode */
                    /*         [2][3]Input port */
                    /* */
                    /* : 1 1 : [1]External clock input mode */
                    /*         [2]Input port */
                    /*         [3]External clock input */

    MSTOP = 0;                          /* X1 oscillator operating */
    XTSTOP = 0;                         /* XT1 oscillator operating */

    OSMC =  0b00000001;                 /* Operation Speed Mode Control Register */
                    /*||||||++--- : fCLK frequency selection */
                    /*|||||| : 0 0 : Operates at a frequency of 10 MHz or less. */
                    /*|||||| : 0 1 : Operates at a frequency higher than 10 MHz. */
                    /*|||||| : 1 0 : Operates at a frequency of 1 MHz. */
                    /*|||||| */
                    /*|+++++----- : Be sure to set 00000 */
                    /*| */
                    /*+---------- : Setting in subsystem clock HALT mode */
                    /* : 0 : Enables subsystem clock supply to peripheral functions. */
                    /*     (See Table 21-1 Operating Statuses in HALT Mode (2/3) */
                    /*      for the peripheral functions whose operations are enabled.) */
                    /* : 1 : Stops subsystem clock supply to peripheral functions except real-time counter,
*/
                    /*      clock output/buzzer output, and LCD controller/driver. */

    while(OSTC.0 != 1){             /* wait X1 oscillation stabilization */
            NOP();
    }

    /*-- Caution ---------------------------------------------*/
    /* To increase fCLK to 10 MHz or higher, set FSEL to '1',    */
    /* then change fCLK after two or more clocks have elapsed. */
```

```
/*---------------------------------------------------------*/
NOP();
NOP();

CKC =     0b00010000;                  /* System Clock Control Register */
                        /*|+|+++++--- : Selection of CPU/peripheral hardware clock (fCLK) */
                        /*| | : 0 0 x 0 0 0 : fIH */
                        /*| | : 0 0 x 0 0 1 : fIH/2 (default) */
                        /*| | : 0 0 x 0 1 0 : fIH/2^2 */
                        /*| | : 0 0 x 0 1 1 : fIH/2^3 */
                        /*| | : 0 0 x 1 0 0 : fIH/2^4 */
                        /*| | : 0 0 x 1 0 1 : fIH/2^5 */
                        /*| | : 0 1 x 0 0 0 : fMX */
                        /*| | : 0 1 x 0 0 1 : fMX/2 */
                        /*| | : 0 1 x 0 1 0 : fMX/2^2 */
                        /*| | : 0 1 x 0 1 1 : fMX/2^3 */
                        /*| | : 0 1 x 1 0 0 : fMX/2^4 */
                        /*| | : 0 1 x 1 0 1 : fMX/2^5 */
                        /*| | : 1 x 0 x x x : fSUB */
                        /*| | : 1 x 1 x x x : fSUB/2 */
                        /*| | : Other than above : Setting prohibited */
                        /*| |    x = don't care */
                        /*| | */
                        /*| +-------- : Status of Main system clock (fMAIN) */
                        /*| : 0 : Internal high-speed oscillation clock (fIH) */
                        /*| : 1 : High-speed system clock (fMX) */
                        /*| */
                        /*+---------- : Status of CPU/peripheral hardware clock (fCLK) */
                        /* : 0 : Main system clock (fMAIN) */
                        /* : 1 : Subsystem clock (fSUB) */

/* Confirming the CPU clock status */
while((CLS != 0)||(MCS != 1)){
          NOP();
}
/* CPU is operating on a High-speed system clock */
HIOSTOP = 1;                           /* internal high-speed oscillation stopped */

OSTS =    0b00000111;                  /* Oscillation Stabilization Time Select Register */
```

```
                    /*||||++++--- : Oscillation stabilization time selection */
                    /*|||| : 0 0 0 : 2^8/fX */
                    /*|||| : 0 0 1 : 2^9/fX */
                    /*|||| : 0 1 0 : 2^10/fX */
                    /*|||| : 0 1 1 : 2^11/fX */
                    /*|||| : 1 0 0 : 2^13/fX */
                    /*|||| : 1 0 1 : 2^15/fX */
                    /*|||| : 1 1 0 : 2^17/fX */
                    /*|||| : 1 1 1 : 2^18/fX */
                    /*|||| */
                    /*++++------- : Be sure to set 000000 */

/*----------------------------------------*/
/*        Initialization of timer         */
/*----------------------------------------*/
fn_InitTau0();

/* software to wait for the operation stabilization time */
/* (over 200ms from when XT1 enable) */
fn_Wait500usBase(200000/500);

/*----------------------------------------*/
/* Initialization of low-voltage detector */
/*----------------------------------------*/
fn_InitLvi();

/*----------------------------------------*/
/*   Initialization of A/D,D/A,OPAmp                */
/*----------------------------------------*/
fn_AdcDacInit();

/*----------------------------------------*/
/*   Initialization of LCD                                  */
/*----------------------------------------*/
fn_DisplayInit();


EI();                                  /* enable all interrupts */
```

```
}

/*----------------------------------------------------------------*/
/* Module:         fn_InitPort
                   */
/* Description:    Setting of I/O ports                                          */
/*      parameter: --
                          */
/*      return    : --
                          */
/*----------------------------------------------------------------*/
void    fn_InitPort(void)
{
/*------------------------------------------------------------*/
/*      Ports configuration for digital input and output          */
/*------------------------------------------------------------*/
        ADPC =  0b00010000;         /* A/D Port Configuration Register */
                          /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                          /*||| :                +------------------------ ANI15/AVREFM/P157 */
                          /*||| :                | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                          /*||| :                | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
                          /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                          /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                          /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                          /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                          /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                          /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                          /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                          /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                          /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                          /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                          /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                          /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                          /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                          /*||| */
                          /*+++-------- : Be sure to set 000 */

/*------------------------------------------------------------*/
/*      Setting of Port 0
```

```
        */
/*----------------------------------------------------------*/
        P0 =    0b00000000;     /* Set P00-P02 Output latch to Low */
        PM0 =   0b11111000;     /* Set P00-P02 to output port */
                                        /* P00-P02:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 1
        */
/*----------------------------------------------------------*/
        P1 =    0b00000000;     /* Set P10-P17 Output latch to Low */
        PM1 =   0b00000000;     /* Set P10-P17 to output port */
                                        /* P10-P15:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 2
        */
/*----------------------------------------------------------*/
        P2 =    0b00000000;     /* Set P20-P27 Output latch to Low */
        PM2 =   0b11111111;     /* Set P20-P27 to input port */
                                        /* P20-P27:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 3
        */
/*----------------------------------------------------------*/
        P3 =    0b00001100;     /* Set P30-P31,P34 Output latch to Low */
                                        /* Set P33,P32 Output latch High */
        PM3 =   0b11110000;     /* Set P30-P33 to output port */
                                        /* Set P34(TI06) to input port */
                                        /* P30-P33:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 4
        */
/*----------------------------------------------------------*/
        P4 =    0b00000000;     /* Set P40-P41 Output latch to Low */
        PM4 =   0b11111100;     /* Set P40-P41 to output port */
                                        /* P40-P41:Unused */
```

```
/*----------------------------------------------------------*/
/*      Setting of Port 5
        */
/*----------------------------------------------------------*/
        P5 =    0b00000000;     /* Set P50-P57 Output latch to Low */
        PM5 =   0b11110000;     /* Set P50-P57 to output port */
                                        /* P50-P57:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 6
        */
/*----------------------------------------------------------*/
        P6 =    0b00000000;     /* Set P60-P61 Output latch to Low */
        PM6 =   0b11111100;     /* Set P60-P61 to output port */
                                        /* P60-P61:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 7
        */
/*----------------------------------------------------------*/
        P7 =    0b00000000;     /* Set P70-P77 Output latch to Low */
        PM7 =   0b00000000;     /* Set P70-P77 to output port */
                                        /* P70-P77:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 8
        */
/*----------------------------------------------------------*/
        P8 =    0b00000000;     /* Set P80-P88 Output latch to Low */
        PM8 =   0b00000000;     /* Set P80-P88 to output port */
                                        /* P80-P88:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 9
        */
/*----------------------------------------------------------*/
        P9 =    0b00000000;     /* Set P90-P97 Output latch to Low */
        PM9 =   0b00000000;     /* Set P90-P97 to output port */
```

```
                                                        /* P90-P97:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 10
        */
/*-----------------------------------------------------------*/
        P10 =   0b00000000;     /* Set P100-P102 Output latch to Low */
        PM10 =  0b11111000;     /* Set P100-P102 to output port */
                                                        /* P100-P102:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 11
        */
/*-----------------------------------------------------------*/
        P11 =   0b00000000;     /* Set P110-P111 Output latch to Low */
        PM11 =  0b11111100;     /* Set P110-P111 to output port */
                                                        /* P110-P111:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 12
        */
/*-----------------------------------------------------------*/
        P12 =   0b00000000;     /* Set P120 Output latch to Low */
        PM12 =  0b11111110;     /* Set P120 to output port */
                                                        /* P120-P124:Unused */
                                                        /* *P121-P124:Input port */

/*-----------------------------------------------------------*/
/*      Setting of Port 13
        */
/*-----------------------------------------------------------*/
        P13 =   0b00000000;     /* Set P130 Output latch to Low */
                                                        /* P130:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 14
        */
/*-----------------------------------------------------------*/
        P14 =   0b00000000;     /* Set P140-P147 Output latch to Low */
```

```
        PM14 =   0b00000000;         /* Set P140-P147 to output port */
                                                      /* P140-P147:Unused */

/*-----------------------------------------------------------*/
/*      Setting of Port 15
        */
/*-----------------------------------------------------------*/
        P15 =    0b00000000;         /* Set P150-P152,P157 Output latch to Low */
        PM15 =   0b11111111;         /* Set P150-P152,P157 to input port */
                                                      /* P150-P152,P157:Unused */
}

/*---------------------------------------------------------------*/
/* Module:          fn_InitTau0
                              */
/* Description:     Setting of Timer array unit 0                              */
/*      parameter: --
                              */
/*      return     : --
                              */
/*---------------------------------------------------------------*/
void    fn_InitTau0(void)
{
        TAU0EN = 1;                          /* supplies input clock to timer array unit 0 */
        TPS0L =  0b00000010;         /* Timer Clock Select Register 0 */
                              /*||||++++--- : Selection of operation clock (CK00) */
                              /*++++------- : Selection of operation clock (CK01) */
                              /* : 0 0 0 0 : CK0m = fCLK */
                              /* : 0 0 0 1 : CK0m = fCLK/2 */
                              /* : 0 0 1 0 : CK0m = fCLK/2^2 */
                              /* : 0 0 1 1 : CK0m = fCLK/2^3 */
                              /* : 0 1 0 0 : CK0m = fCLK/2^4 */
                              /* : 0 1 0 1 : CK0m = fCLK/2^5 */
                              /* : 0 1 1 0 : CK0m = fCLK/2^6 */
                              /* : 0 1 1 1 : CK0m = fCLK/2^7 */
                              /* : 1 0 0 0 : CK0m = fCLK/2^8 */
                              /* : 1 0 0 1 : CK0m = fCLK/2^9 */
                              /* : 1 0 1 0 : CK0m = fCLK/2^10 */
                              /* : 1 0 1 1 : CK0m = fCLK/2^11 */
```

```
                    /* : 1 1 0 0 : CK0m = fCLK/2^12 */
                    /* : 1 1 0 1 : CK0m = fCLK/2^13 */
                    /* : 1 1 1 0 : CK0m = fCLK/2^14 */
                    /* : 1 1 1 1 : CK0m = fCLK/2^15 */
                    /*    m = 0, 1 */

/* CH1:for wait */
TMR01 = 0b0000000000000000;         /* Timer Mode Register 01 */
                    /*||||||||||||++++--- : [1]Operation mode of channel 1 */
                    /*||||||||||||          [2]Count operation of TCR */
                    /*||||||||||||          [3]Independent operation */
                    /*||||||||||||          [4]Setting of starting counting and interrupt */
                    /*|||||||||||| : 0 0 0 0 : [1]Interval timer mode */
                    /*||||||||||||             [2]Counting down */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                (timer output does not change, either). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 0 0 1 : [1]Interval timer mode */
                    /*||||||||||||             [2]Counting down */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is generated when counting is started */
                    /*||||||||||||                (timer output also changes). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 0 0 : [1]Capture mode */
                    /*||||||||||||             [2]Counting up */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                (timer output does not change, either). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 0 1 : [1]Capture mode */
                    /*||||||||||||             [2]Counting up */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is generated when counting is started */
                    /*||||||||||||                (timer output also changes). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 1 0 : [1]Event counter mode */
                    /*||||||||||||             [2]Counting down */
                    /*||||||||||||             [3]Possible */
```

```
/*||||||||||||                    [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                         (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 0 : [1]One-count mode */
/*||||||||||||                    [2]Counting down */
/*||||||||||||                    [3]Impossible */
/*||||||||||||                    [4]Start trigger is invalid during counting operation. */
/*||||||||||||                         At that time, interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 1 : [1]One-count mode */
/*||||||||||||                    [2]Counting down */
/*||||||||||||                    [3]Impossible */
/*||||||||||||                    [4]Start trigger is valid during counting operation. */
/*||||||||||||                         At that time, interrupt is also generated. */
/*|||||||||||| */
/*|||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*||||||||||||                    [2]Counting up */
/*||||||||||||                    [3]Possible */
/*||||||||||||                    [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                         (timer output does not change, either). */
/*||||||||||||                         Start trigger is invalid during counting operation. */
/*||||||||||||                         At that time interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : Other than above : Setting prohibited */
/*|||||||||||| */
/*||||||||||++------- : Be sure to set 00 */
/*|||||||||| */
/*||||||||++--------- : Selection of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 valid
edge */
/*||||||||                   (the timer input used with channel 1 is selected by using TIS0
register). */
/*|||||||| : 0 0 : Falling edge */
/*|||||||| : 0 1 : Rising edge */
/*|||||||| : 1 0 : Both edges (when low-level width is measured) */
/*||||||||              Start trigger: Falling edge, Capture trigger: Rising edge */
/*|||||||| : 1 1 : Both edges (when high-level width is measured) */
/*||||||||              Start trigger: Rising edge, Capture trigger: Falling edge */
/*|||||||| */
/*|||||+++----------- : Setting of start trigger or capture trigger of channel 1 */
```

```
                              /*|||||  : 0 0 0 : Only software trigger start is valid (other trigger sources are unselected).
*/
                              /*|||||  : 0 0 1 : Valid edge of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 is used
as both the start trigger and capture trigger. */
                              /*|||||  : 0 1 0 : Both the edges of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 are
used as a start trigger and a capture trigger. */
                              /*|||||  : 1 0 0 : Interrupt signal of the master channel is used (when the channel is used
as a slave channel with the combination operation function). */
                              /*|||||  : Other than above : Setting prohibited */
                              /*||||| */
                              /*||||+-------------- : Selection of slave/master of channel 1 */
                              /*||||  : 0 : Operates as slave channel with combination operation function. */
                              /*||||  : 1 : Operates as master channel with combination operation function. */
                              /*|||| */
                              /*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
                              /*|||  : 0 : Operation clock MCK specified by CKS01 bit */
                              /*|||  : 1 : Valid edge of input signal input from TI01 pin, fSUB/2, fSUB/4, or INTRTC1 */
                              /*|||        (the timer input used with channel 1 is selected by using TIS0 register). */
                              /*||| */
                              /*|++---------------- : Be sure to set 00 */
                              /*| */
                              /*+------------------ : Selection of operation clock (MCK) of channel 1 */
                              /*  : 0 : Operation clock CK00 set by TPS0 register */
                              /*  : 1 : Operation clock CK01 set by TPS0 register */
      TDR01 = CCK00_500USEC;          /* set interval time to 500us */
      TMMK01 = 1;                                        /* disable interrupt */


      /* CH4:for capturing interval */
      TMR04 = 0b1000000000000000;     /* Timer Mode Register 04 */
                              /*||||||||||||++++--- : [1]Operation mode of channel 4 */
                              /*||||||||||||          [2]Count operation of TCR */
                              /*||||||||||||          [3]Independent operation */
                              /*||||||||||||          [4]Setting of starting counting and interrupt */
                              /*|||||||||||| : 0 0 0 0 : [1]Interval timer mode */
                              /*||||||||||||             [2]Counting down */
                              /*||||||||||||             [3]Possible */
                              /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
                              /*||||||||||||                (timer output does not change, either). */
```

```
/*|||||||||||| */
/*|||||||||||| : 0 0 0 1 : [1]Interval timer mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is generated when counting is started */
/*||||||||||||                (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 0 0 : [1]Capture mode */
/*||||||||||||             [2]Counting up */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 0 1 0 1 : [1]Capture mode */
/*||||||||||||             [2]Counting up */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is generated when counting is started */
/*||||||||||||                (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 1 0 : [1]Event counter mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 0 : [1]One-count mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Impossible */
/*||||||||||||             [4]Start trigger is invalid during counting operation. */
/*||||||||||||                At that time, interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 1 : [1]One-count mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Impossible */
/*||||||||||||             [4]Start trigger is valid during counting operation. */
/*||||||||||||                At that time, interrupt is also generated. */
/*|||||||||||| */
/*|||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*||||||||||||             [2]Counting up */
```

```
/*||||||||||||                    [3]Possible */
/*||||||||||||                    [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                       (timer output does not change, either). */
/*||||||||||||                       Start trigger is invalid during counting operation. */
/*||||||||||||                       At that time interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : Other than above : Setting prohibited */
/*|||||||||||| */
/*||||||||||++------- : Be sure to set 00 */
/*|||||||||| */
/*||||||||++--------- : Selection of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 valid edge */
/*||||||||                    (the timer input used with channel 4 is selected by using TIS0 register). */
/*|||||||| : 0 0 : Falling edge */
/*|||||||| : 0 1 : Rising edge */
/*|||||||| : 1 0 : Both edges (when low-level width is measured) */
/*||||||||              Start trigger: Falling edge, Capture trigger: Rising edge */
/*|||||||| : 1 1 : Both edges (when high-level width is measured) */
/*||||||||              Start trigger: Rising edge, Capture trigger: Falling edge */
/*|||||||| */
/*|||||+++----------- : Setting of start trigger or capture trigger of channel 4 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are unselected). */
/*||||| : 0 0 1 : Valid edge of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 is used as both the start trigger and capture trigger. */
/*||||| : 0 1 0 : Both the edges of TI04 pin input signal, fSUB/2, fSUB/4, or INTRTC1 are used as a start trigger and a capture trigger. */
/*||||| : 1 0 0 : Interrupt signal of the master channel is used (when the channel is used as a slave channel with the combination operation function). */
/*||||| : Other than above : Setting prohibited */
/*||||| */
/*||||+-------------- : Selection of slave/master of channel 4 */
/*|||| : 0 : Operates as slave channel with combination operation function. */
/*|||| : 1 : Operates as master channel with combination operation function. */
/*|||| */
/*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
/*||| : 0 : Operation clock MCK specified by CKS04 bit */
/*||| : 1 : Valid edge of input signal input from TI04 pin, fSUB/2, fSUB/4, or INTRTC1 */
```

```
                    /*|||          (the timer input used with channel 4 is selected by using TIS0 register). */
                    /*||| */
                    /*|++---------------- : Be sure to set 00 */
                    /*| */
                    /*+------------------ : Selection of operation clock (MCK) of channel 4 */
                    /* : 0 : Operation clock CK00 set by TPS0 register */
                    /* : 1 : Operation clock CK01 set by TPS0 register */
TDR04 = 20000;      /* set interval time to 1ms(=0.05us * 20000) */
TMMK04 = 1;                  /* disable interrupt */



/* CH6:for TI06 */
TMR06 = 0b1001000101000110;          /* Timer Mode Register 06 */
                    /*||||||||||||++++--- : [1]Operation mode of channel 6 */
                    /*||||||||||||          [2]Count operation of TCR */
                    /*||||||||||||          [3]Independent operation */
                    /*||||||||||||          [4]Setting of starting counting and interrupt */
                    /*|||||||||||| : 0 0 0 0 : [1]Interval timer mode */
                    /*||||||||||||             [2]Counting down */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                (timer output does not change, either). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 0 0 1 : [1]Interval timer mode */
                    /*||||||||||||             [2]Counting down */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is generated when counting is started */
                    /*||||||||||||                (timer output also changes). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 0 0 : [1]Capture mode */
                    /*||||||||||||             [2]Counting up */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                (timer output does not change, either). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 0 1 : [1]Capture mode */
                    /*||||||||||||             [2]Counting up */
                    /*||||||||||||             [3]Possible */
                    /*||||||||||||             [4]Timer interrupt is generated when counting is started */
```

```
/*||||||||||||                    (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 1 0 : [1]Event counter mode */
/*||||||||||||                [2]Counting down */
/*||||||||||||                [3]Possible */
/*||||||||||||                [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                    (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 0 : [1]One-count mode */
/*||||||||||||                [2]Counting down */
/*||||||||||||                [3]Impossible */
/*||||||||||||                [4]Start trigger is invalid during counting operation. */
/*||||||||||||                    At that time, interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 1 : [1]One-count mode */
/*||||||||||||                [2]Counting down */
/*||||||||||||                [3]Impossible */
/*||||||||||||                [4]Start trigger is valid during counting operation. */
/*||||||||||||                    At that time, interrupt is also generated. */
/*|||||||||||| */
/*|||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*||||||||||||                [2]Counting up */
/*||||||||||||                [3]Possible */
/*||||||||||||                [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                    (timer output does not change, either). */
/*||||||||||||                    Start trigger is invalid during counting operation. */
/*||||||||||||                    At that time interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : Other than above : Setting prohibited */
/*|||||||||||| */
/*||||||||||++------- : Be sure to set 00 */
/*|||||||||| */
/*||||||||++--------- : Selection of TI06 pin input signal, fSUB/2, fSUB/4, or INTRTC1 valid
edge */
/*||||||||                  (the timer input used with channel 6 is selected by using TIS0
register). */
/*|||||||| : 0 0 : Falling edge */
/*|||||||| : 0 1 : Rising edge */
/*|||||||| : 1 0 : Both edges (when low-level width is measured) */
```

```
                                        /*||||||||              Start trigger: Falling edge, Capture trigger: Rising edge */
                                        /*|||||||| : 1 1 : Both edges (when high-level width is measured) */
                                        /*||||||||              Start trigger: Rising edge, Capture trigger: Falling edge */
                                        /*|||||||| */
                                        /*|||||+++----------- : Setting of start trigger or capture trigger of channel 6 */
                                        /*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are unselected).
*/
                                        /*||||| : 0 0 1 : Valid edge of TI06 pin input signal, fSUB/2, fSUB/4, or INTRTC1 is used
as both the start trigger and capture trigger. */
                                        /*||||| : 0 1 0 : Both the edges of TI06 pin input signal, fSUB/2, fSUB/4, or INTRTC1 are
used as a start trigger and a capture trigger. */
                                        /*||||| : 1 0 0 : Interrupt signal of the master channel is used (when the channel is used
as a slave channel with the combination operation function). */
                                        /*||||| : Other than above : Setting prohibited */
                                        /*||||| */
                                        /*||||+-------------- : Selection of slave/master of channel 6 */
                                        /*|||| : 0 : Operates as slave channel with combination operation function. */
                                        /*|||| : 1 : Operates as master channel with combination operation function. */
                                        /*|||| */
                                        /*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
                                        /*||| : 0 : Operation clock MCK specified by CKS04 bit */
                                        /*||| : 1 : Valid edge of input signal input from TI04 pin, fSUB/2, fSUB/4, or INTRTC1 */
                                        /*|||          (the timer input used with channel 4 is selected by using TIS0 register). */
                                        /*||| */
                                        /*|++---------------- : Be sure to set 00 */
                                        /*| */
                                        /*+------------------ : Selection of operation clock (MCK) of channel 4 */
                                        /* : 0 : Operation clock CK00 set by TPS0 register */
                                        /* : 1 : Operation clock CK01 set by TPS0 register */
        TDR06 = 0xffff;                 /* set initial count */
        TMMK06 = 1;                              /* disable interrupt */
        TIS0.6 = 0;                              /* select TI06 */

    }

    /*-------------------------------------------------------------*/
    /* Module:          fn_InitLvi
                        */
    /* Description:     Setting of Low-voltage detectior                                   */
```

```
/*        parameter: --
                            */
/*        return    : --
                            */
/*------------------------------------------------------------*/
void      fn_InitLvi(void)
{
          unsigned short        loop;      /* waiting counter */

          LVIMK = 1;                              /* disable LVI interrupt */

          LVIS =    0b00001001;                   /* Low-Voltage Detection Level Select Register */
                            /*||||++++--- : Detection level */
                            /*|||| : 0 0 0 0 : VLVI0 (4.22V) */
                            /*|||| : 0 0 0 1 : VLVI1 (4.07V) */
                            /*|||| : 0 0 1 0 : VLVI2 (3.92V) */
                            /*|||| : 0 0 1 1 : VLVI3 (3.76V) */
                            /*|||| : 0 1 0 0 : VLVI4 (3.61V) */
                            /*|||| : 0 1 0 1 : VLVI5 (3.45V) */
                            /*|||| : 0 1 1 0 : VLVI6 (3.30V) */
                            /*|||| : 0 1 1 1 : VLVI7 (3.15V) */
                            /*|||| : 1 0 0 0 : VLVI8 (2.99V) */
                            /*|||| : 1 0 0 1 : VLVI9 (2.84V) */
                            /*|||| : 1 0 1 0 : VLVI10 (2.68V) */
                            /*|||| : 1 0 1 1 : VLVI11 (2.53V) */
                            /*|||| : 1 1 0 0 : VLVI12 (2.38V) */
                            /*|||| : 1 1 0 1 : VLVI13 (2.22V) */
                            /*|||| : 1 1 1 0 : VLVI14 (2.07V) */
                            /*|||| : 1 1 1 1 : VLVI15 (1.91V) */
                            /*|||| */
                            /*++++------- : Be sure to set 0000 */


          LVIM =    0b10000000;                   /* Low-Voltage Detection Register */
                            /*|||||||+--- : LVIF Low-voltage detection flag */
                            /*||||||| : 0 : * LVISEL = 0: VDD >= VLVI, or when LVI operation is disabled */
                            /*|||||||       * LVISEL = 1: EXLVI >= VEXLVI, or when LVI operation is disabled */
                            /*||||||| : 1 : * LVISEL = 0: VDD < VLVI */
                            /*|||||||       * LVISEL = 1: EXLVI < VEXLVI */
```

```
                                    /*||||||| */
                                    /*||||||+---- : Low-voltage detection operation mode (interrupt/reset) selection(LVIMD) */
                                    /*|||||| : 0 : * LVISEL = 0: Generates an internal interrupt signal */
                                    /*||||||                  when VDD drops lower than VLVI (VDD < VLVI) */
                                    /*||||||                  or when VDD becomes VLVI or higher (VDD >= VLVI). */
                                    /*||||||       * LVISEL = 1: Generates an interrupt signal */
                                    /*||||||                     when EXLVI drops lower than VEXLVI (EXLVI <
VEXLVI) */
                                    /*||||||                  or when EXLVI becomes VEXLVI or higher (EXLVI >=
VEXLVI). */
                                    /*|||||| : 1 : * LVISEL = 0: Generates an internal reset signal when VDD < VLVI */
                                    /*||||||                  and releases the reset signal when VDD >= VLVI. */
                                    /*||||||       * LVISEL = 1: Generates an internal reset signal when EXLVI < VEXLVI
*/
                                    /*||||||                  and releases the reset signal when EXLVI >= VEXLVI. */
                                    /*|||||| */
                                    /*|||||+----- : Voltage detection selection(LVISEL) */
                                    /*||||| : 0 : Detects level of supply voltage (VDD) */
                                    /*||||| : 1 : Detects level of input voltage from external input pin (EXLVI) */
                                    /*||||| */
                                    /*|++++------ : Be sure to set 0000 */
                                    /*| */
                                    /*+---------- : Enables low-voltage detection operation */
                                    /* : 0 : Disables operation */
                                    /* : 1 : Enables operation */

        /* software to wait for the operation stabilization time (210us) */
        for(loop = 500; loop > 0; loop--){
                NOP();
        }
        /* wait until VLVI≤VDD */
        while( LVIF ){
                NOP();
        }

        LVIIF = 0;                      /* clear LVI interrupt request flag */
}
```

```
/*------------------------------------------------------------*/
/* Module:          fn_Wait500usBase
        */
/* Description:     Delays the program for (Time * 500us)              */
/*      parameter: wait time(Time)
        */
/*      return    : --
                        */
/*------------------------------------------------------------*/
void    fn_Wait500usBase(unsigned short Time)
{
        TS0L.1 = 1;                             /* start TAU0 CH1 */
        TMIF01 = 0;

        for(; Time > 0; Time--){        /* wait for (parameter * 500)us */
                while(!TMIF01){
                        NOP();
                }
                TMIF01 = 0;
        }

        TT0L.1 = 1;                             /* stop TAU0 CH1 */
}


/*------------------------------------------------------------*/
/* Module:          fn_AdcDacInit
                */
/* Description:     Initialization of A/D D/A OPAmp module             */
/*      parameter: --
                        */
/*      return    : --
                        */
/*------------------------------------------------------------*/
void    fn_AdcDacInit(void)
{
        ADCEN = 1;                              /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */
```

```
        /*------------------------------------------*/
        /*  Initialization of Voltage reference                */
        /*------------------------------------------*/
/*      ADVRC = 0b00001000;         /* Analog reference voltage control register */
                          /*||||+|++--- : [1]Positive reference voltage supplies selection of A/D and D/A
converters */
                          /*|||| |        [2]Operationcontrol of voltage reference */
                          /*|||| |        [3]Output voltage selection of voltage reference */
                          /*|||| |        [4]Operation control of input gate voltage boost circuit for A/D converter
*/
                          /*|||| |        [5]Relationship with the conversion mode used */
                          /*|||| | */
                          /*|||| | : 0 0 0 : [1]AVREFP (external voltage reference input) */
                          /*|||| |           [2]Stops operation (Hi-Z) */
                          /*|||| |           [3]2.5 V */
                          /*|||| |           [4]Stops operation */
                          /*|||| |           [5]Can be set in conversion mode 1 */
                          /*|||| | */
                          /*|||| | : 0 1 0 : [1]AVREFP (external voltage reference input) */
                          /*|||| |           [2]Stops operation (Hi-Z) */
                          /*|||| |           [3]2.0 V */
                          /*|||| |           [4]Enables operation */
                          /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                          /*|||| | */
                          /*|||| | : 1 0 0 : [1]VREFOUT (voltage reference output) */
                          /*|||| |           [2]Stops operation (pull-down output) */
                          /*|||| |           [3]2.5 V */
                          /*|||| |           [4]Stops operation */
                          /*|||| |           [5] - */
                          /*|||| | */
                          /*|||| | : 1 0 1 : [1]VREFOUT (voltage reference output) */
                          /*|||| |           [2]Enables operation */
                          /*|||| |           [3]2.5 V */
                          /*|||| |           [4]Enables operation */
                          /*|||| |           [5]Can be set in conversion mode 2 or 3 */
                          /*|||| | */
                          /*|||| | : 1 1 0 : [1]VREFOUT (voltage reference output) */
                          /*|||| |           [2]Stops operation (pull-down output) */
```

```
                        /*|||| |            [3]2.0 V */
                        /*|||| |            [4]Enables operation */
                        /*|||| |            [5] - */
                        /*|||| | */
                        /*|||| | : 1 1 1 : [1]VREFOUT (voltage reference output) */
                        /*|||| |            [2]Enables operation */
                        /*|||| |            [3]2.0 V */
                        /*|||| |            [4]Enables operation */
                        /*|||| |            [5]Can be set in conversion mode 2 or 3 */
                        /*|||| | */
                        /*|||| | : Other than the above : Setting prohibited */
                        /*|||| | */
                        /*|+++-+----- : Be sure to set 0000 */
                        /*| */
                        /*+---------- : Reference voltage supply (negative side) of A/D converter selection */
                        /* : 0 : AVSS */
                        /* : 1 : AVREFM (external voltage reference input) */

/*      ADVRC.0 = 1;            /* Enables operation */
/*      ADVRC.1 = 1;            /* Output 2.0 V */

        ADPC =  0b00000000;     /* A/D Port Configuration Register */
                        /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                        /*||| :             +------------------------- ANI15/AVREFM/P157 */
                        /*||| :             | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                        /*||| :             | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
                        /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                        /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                        /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                        /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                        /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                        /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                        /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                        /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                        /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                        /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                        /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                        /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                        /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
```

```
                                /*||| */
                                /*+++-------- : Be sure to set 000 */

        /* wait for settling time to 17msec */
/*      fn_Wait500usBase(17000/500); */

        /*-------------------------------------------*/
        /*   Initialization of operational amplifier    */
        /*-------------------------------------------*/
        PM2.1 = 1;                      /* input port mode (AMP0O) */
        PM2.0 = 1;                      /* input port mode (AMP0-) */
        PM2.2 = 1;                      /* input port mode (AMP0+) */

        PM2.4 = 1;                      /* input port mode (AMP1O) */
        PM2.3 = 1;                      /* input port mode (AMP1-) */
        PM2.5 = 1;                      /* input port mode (AMP1+) */

        PM2.7 = 1;                      /* input port mode (AMP2O) */
        PM2.6 = 1;                      /* input port mode (AMP2-) */
        PM15.0 = 1;                     /* input port mode (AMP2+) */

        PM11.0 = 1;                     /* input port mode (ANO0) */

        ADCEN = 1;                      /* supplies input clock to OPAmp A/D converter */
        OAC.0 = 1;                      /* operational amplifier (AMP0) enable */
        OAC.1 = 0;                      /* operational amplifier (AMP1) disable */
        OAC.2 = 0;                      /* operational amplifier (AMP2) disable */

        /*-------------------------------------------*/
        /*       Initialization of D/A converter        */
        /*-------------------------------------------*/
        DACEN = 1;                              /* supplies input clock to D/A converter */

        /* set D/A chanel1 */
        DAM.1 = 0;                              /* [DAMD1] D/A converter operation is Normal mode
*/
        DAM.3 = 1;                              /* [DARES1]D/A conversion resolution is 12-bit */
        DACSW1 = 2048;                  /* D/A conversion value setting reigster 1 */
                                                        /* ANO1 = 1.5V (1/2 * 2^12) */
```

```
}

/*---------------------------------------------------------------*/
/* Module:          main
                              */
/* Description:     Main process
          */
/*          parameter: --
                              */
/*          return     : --
                              */
/*---------------------------------------------------------------*/
void main(void)
{
          unsigned short          ushCount; /* Pulse counter */

          DACE1 = 1;                                        /* D/A converter CH1 enable */
          ushCount = 0;

          /*==================================*/
          /*---------------------------------*/
          /*                    Main Loop                    */
          /*---------------------------------*/
          /*==================================*/

          /*================================================================*/
          /* if system have nothing to do, go to standby for power-saving */
          /*================================================================*/

          while (1){
                    TS0L.6 = 1;                             /* start TAU0 CH6 (pulse counter) */
                    TS0L.4 = 1;                             /* start TAU0 CH4 (interval timer) */
                    TMIF04 = 0;                             /* clear interval timer request */

                    while(TMIF04 == 0){                /* wait interval */
                              NOP();
                    }
```

```
			ushCount = TCR06;		/* get pulse count */

			TT0L.6 = 1;					/* stop TAU0 CH6   (pulse counter)*/
			TT0L.4 = 1;					/* stop TAU0 CH4 (interval timer) */

			ushCount = 0xffff - ushCount;		/* get pulse count */
			fn_Display(ushCount);			/* display pulse count */
		}
}
```

● display.c

```
/*
 * Copyright (C) NEC Electronics Corporation 2006
 * NEC ELECTRONICS CONFIDENTIAL AND PROPRIETARY
 * All rights reserved by NEC Electronics Corporation.
 * This program must be used solely for the purpose for which
 * it was furnished by NEC Electronics Corporation. No part of this
 * program may be reproduced or disclosed to others, in any
 * form, without the prior written permission of NEC Electronics
 * Corporation. Use of copyright notice dose not evidence
 * publication of the program.
 */

/*--------------------------------------------------------------------*/
/* #pragma directive for CC78K0
				*/
/*--------------------------------------------------------------------*/
#pragma		SFR

/*--------------------------------------------------------------------*/
/* Include files
						*/
/*--------------------------------------------------------------------*/
/*#include		"defines.h"*/
#include		<string.h>

/*--------------------------------------------------------------------*/
```

```
/* Function prototyps
                                */
/*-------------------------------------------------------------------*/
               void      fn_DisplayAllClear(void);                  /* Display all clear */
extern  void   fn_Wait500usBase(unsigned short);       /* Delays the program (500us base) */

/*-------------------------------------------------------------------*/
/* Extern variables/constants
               */
/*-------------------------------------------------------------------*/
/*-------------------------------------------------------------------*/
/* Local constants
                                */
/*-------------------------------------------------------------------*/
/* size of area for displaying */
#define CLCDSIZE_ALL      (unsigned char)(&SEG53 - &SEG4 + 1)    /* all area */
/*#define          CLCDSIZE_NUMBER          4                /* number */
#define CLCDSIZE_NUMBER            5                 /* number */

/* Display starting position */
#define CLCDPOS_START   &SEG4              /* start of all area */
/*#define          CLCDPOS_<1>     &SEG5               /* tenths place of <1> */
/*#define          CLCDPOS_<1>     &SEG10              /* ones place of <1> */
/*#define          CLCDPOS_<2>     &SEG18              /* tenths place of <2> */
/*#define          CLCDPOS_<2>     &SEG23              /* ones place of <2> */
/*#define          CLCDPOS_<3>     &SEG31              /* tenths place of <3> */
/*#define          CLCDPOS_<3>     &SEG36              /* ones place of <3> */
/*#define          CLCDPOS_<4>     &SEG44              /* tenths place of <4> */
/*#define          CLCDPOS_<4>     &SEG49              /* ones place of <4> */


/*#define          CLCDPOS_REF              &SEG5    /* sign of REF */
/*#define          CLCDPOS_REF100  &SEG12  /* hundreds place of REF */
/*#define          CLCDPOS_REF10   &SEG17  /* tenths place of REF */
/*#define          CLCDPOS_REF1    &SEG22  /* ones place of REF */
/*#define          CLCDPOS_TH               &SEG32   /* sign of TH */
/*#define          CLCDPOS_TH100   &SEG39  /* hundreds place of TH */
/*#define          CLCDPOS_TH10    &SEG44  /* tenths place of TH */
/*#define          CLCDPOS_TH1              &SEG49   /* ones place of TH */
```

```
#define CLCDPOS_REF1000     &SEG5   /* thousands place of REF */
#define CLCDPOS_REF100  &SEG10  /* hundreds place of REF */
#define CLCDPOS_REF10   &SEG15  /* tenths place of REF */
#define CLCDPOS_REF1    &SEG20  /* ones place of REF */
#define CLCDPOS_1000    &SEG34  /* thousands place of TH */
#define CLCDPOS_100         &SEG39  /* hundreds place of TH */
#define CLCDPOS_10          &SEG44  /* tenths place of TH */
#define CLCDPOS_1           &SEG49  /* ones place of TH */


/* Display data */
/*=========================================================================
  < LCD PANEL >

          SEG4   SEG5   SEG6   SEG7   SEG8    = = =    SEG49  SEG50  SEG51  SEG52
SEG53
       +------+------+------+------+------+-  = = =  -+------+------+------+------+------+
  COM0 |      |      |      |      |      |           |      |      |      |      |      |
       +------+------+------+------+------+-  = = =  -+------+------+------+------+------+
  COM1 |      |      |      |      |      |           |      |      |      |      |      |
       +------+------+------+------+------+-  = = =  -+------+------+------+------+------+
  COM2 |      |      |      |      |      |           |      |      |      |      |      |
       +------+------+------+------+------+-  = = =  -+------+------+------+------+------+
  COM3 |      |      |      |      |      |           |      |      |      |      |      |
       +------+------+------+------+------+-  = = =  -+------+------+------+------+------+
  COM4 |      |      |      |      |      |           |      |      |      |      |      |
       +------+------+------+------+------+-  = = =  -+------+------+------+------+------+
  COM5 |      |      |      |      |      |           |      |      |      |      |      |
       +------+------+------+------+------+-  = = =  -+------+------+------+------+------+
  COM6 |      |      |      |      |      |           |      |      |      |      |      |
       +------+------+------+------+------+-  = = =  -+------+------+------+------+------+
  COM7 |      |      |      |      |      |           |      |      |      |      |      |
       +------+------+------+------+------+-  = = =  -+------+------+------+------+------+


  < example of the data setting >
     when you display "A" at area from SEG4 to SEG8
```

```
          +-----+-----+-----+-----+-----+
     bit0 |     |     |/////|     |     |    <- Blank area is 0.
          +-----+-----+-----+-----+-----+       Not blank area is 1.
     bit1 |     |/////|     |/////|     |
          +-----+-----+-----+-----+-----+
     bit2 |/////|     |     |     |/////|
          +-----+-----+-----+-----+-----+
     bit3 |/////|     |     |     |/////|
          +-----+-----+-----+-----+-----+
     bit4 |/////|/////|/////|/////|/////|
          +-----+-----+-----+-----+-----+
     bit5 |/////|     |     |     |/////|
          +-----+-----+-----+-----+-----+
     bit6 |/////|     |     |     |/////|
          +-----+-----+-----+-----+-----+
     bit7 |     |     |     |     |     |
          +-----+-----+-----+-----+-----+
             |     |     |     |     |
           0x7C  0x12  0x11  0x12  0x7C
             |     |     |     |     |
    SEG4 <---+     |     |     |     |
    SEG5 <---------+     |     |     |
    SEG6 <---------------+     |     |
    SEG7 <---------------------+     |
    SEG8 <---------------------------+
                   Write data to SFR

==========================================================================*/
/*-----------------------*/
/*    Number indication  */
/*-----------------------*/
static const unsigned char aDispNumber[][CLCDSIZE_NUMBER] = {
/* COM 76543210 */
        {0b01111100                 /* '0' */
        ,0b10000010
        ,0b10000010
        ,0b01111100
        ,0b00000000}
```

```
/* COM 76543210 */
,       {0b00000100             /* '1' */
        ,0b00000100
        ,0b11111110
        ,0b00000000
        ,0b00000000}

/* COM 76543210 */
,       {0b11000100             /* '2' */
        ,0b10100010
        ,0b10010010
        ,0b10001100
        ,0b00000000}

/* COM 76543210 */
,       {0b01000100             /* '3' */
        ,0b10010010
        ,0b10010010
        ,0b01101100
        ,0b00000000}

/* COM 76543210 */
,       {0b01111000             /* '4' */
        ,0b01000100
        ,0b11111110
        ,0b01000000
        ,0b00000000}

/* COM 76543210 */
,       {0b01001110             /* '5' */
        ,0b10001010
        ,0b10001010
        ,0b01110010
        ,0b00000000}

/* COM 76543210 */
,       {0b01111100             /* '6' */
        ,0b10010010
        ,0b10010010
```

```
            ,0b01100000
            ,0b00000000}

/* COM 76543210 */
,           {0b00000010                     /* '7' */
            ,0b11100010
            ,0b00011010
            ,0b00000110
            ,0b00000000}

/* COM 76543210 */
,           {0b01101100                     /* '8' */
            ,0b10010010
            ,0b10010010
            ,0b01101100
            ,0b00000000}

/* COM 76543210 */
,           {0b00001100                     /* '9' */
            ,0b10010010
            ,0b01010010
            ,0b00111100
            ,0b00000000}
};

/*------------------------*/
/*          sign of Ref          */
/*------------------------*/
static const unsigned char aDispRef[] = {
/* COM 76543210 */
             0b11111110                     /* R */
            ,0b00010010
            ,0b00110010
            ,0b01010010
            ,0b10001100
};


/*------------------------*/
```

```
/*          sign of Th          */
/*------------------------*/
static const unsigned char aDispTh[] = {
            0b00000010                  /* T */
           ,0b00000010
           ,0b11111110
           ,0b00000010
           ,0b00000010
};

/*------------------------*/
/*       Button frame           */
/*------------------------*/
static const unsigned char aDispButtonFrame[] = {
/* COM 76543210 */
            0b00000000
           ,0b01111111                  /*   */
           ,0b01001001
           ,0b01001001
           ,0b01001001
           ,0b01001001
           ,0b01001001
           ,0b01001001
           ,0b01001001
           ,0b01001001
           ,0b01001001
           ,0b01111111
           ,0b00000000
           ,0b01111111                  /*   */
           ,0b01001001
           ,0b01001001
           ,0b01001001
           ,0b01001001
           ,0b01001001
           ,0b01001001
           ,0b01001001
           ,0b01001001
           ,0b01001001
           ,0b01111111
```

```
            ,0b00000000
            ,0b01111111                 /*   */
            ,0b01001001
            ,0b01001001
            ,0b01001001
            ,0b01001001
            ,0b01001001
            ,0b01001001
            ,0b01001001
            ,0b01001001
            ,0b01001001
            ,0b01111111
            ,0b00000000
            ,0b01111111                 /*   */
            ,0b01001001
            ,0b01001001
            ,0b01001001
            ,0b01001001
            ,0b01001001
            ,0b01001001
            ,0b01001001
            ,0b01001001
            ,0b01001001
            ,0b01111111
            ,0b00000000
            ,0b00000000
};

static const unsigned char aDispClear[] = {
/* COM 76543210 */
             0b00000000                 /* ' ' */
            ,0b00000000
            ,0b00000000
            ,0b00000000
            ,0b00000000
            ,0b00000000                 /* ' ' */
            ,0b00000000
            ,0b00000000
            ,0b00000000
```

```
,0b00000000
,0b00000000                    /* ' ' */
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000                    /* ' ' */
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000                    /* ' ' */
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000                    /* ' ' */
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000                    /* ' ' */
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000                    /* ' ' */
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000                    /* ' ' */
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000                    /* ' ' */
,0b00000000
,0b00000000
```

```
        ,0b00000000
        ,0b00000000
};

/*-------------------------------------------------------------------*/
/* Global variables
                                        */
/*-------------------------------------------------------------------*/
/*boolean                bDispUpdate;        /* Display update request */
/*boolean                bBlinkOn;           /* Blink status (0:LCD OFF, 1:LCD ON) */

/*-------------------------------------------------------------------*/
/* Local variables
                                        */
/*-------------------------------------------------------------------*/

/*-------------------------------------------------------------------*/
/* Code
                                                              */
/*-------------------------------------------------------------------*/
/*===========================================================================*/
/*                                                                          */
/*                              Common function                             */
/*                                                                          */
/*===========================================================================*/
/*------------------------------------------------------------*/
/* Module:          fn_DisplayInit
                    */
/* Description:     Initialization of Display module                     */
/*      parameter: --
                              */
/*      return     : --
                              */
/*------------------------------------------------------------*/
void    fn_DisplayInit(void)
{
        /*---------------------------------------*/
        /* Initialization of LCD controler/driver */
        /*---------------------------------------*/
```

```
LCDMD = 0b00010000;                 /* LCD Mode Register */
                    /*++||++++--- : Be sure to set 000000 */
                    /*   || */
                    /*   ++------- : LCD drive voltage generator selection */
                    /*     : 0 0 : External resistance division method */
                    /*     : 0 1 : Internal voltage boosting method */
                    /*     : 1 0 : Capacitance split method */
                    /*     : 1 1 : Setting prohibited */
SEGEN = 0b00011111;                 /* Segment Enable Register */
                    /*|||||||+--- : Control segment signal output from pins SEG8-SEG11 */
                    /*||||||+---- : Control segment signal output from pins SEG12-SEG15 */
                    /*|||||+----- : Control segment signal output from pins SEG16-SEG19 */
                    /*||||+------ : Control segment signal output from pins SEG20-SEG23 */
                    /*|||+------- : Control segment signal output from pins SEG24-SEG26 */
                    /*||| : 0 : segment signal output disable */
                    /*||| : 1 : segment signal output enable */
                    /*||| */
                    /*+++-------- : Be sure to set 000 */
PFALL =  0b01111111;                /* Port Function Register ALL */
                    /*|||||||+--- : Pins P50-P53 port/segment output specification */
                    /*||||||+---- : Pins P54-P57 port/segment output specification */
                    /*|||||+----- : Pins P90-P93 port/segment output specification */
                    /*||||+------ : Pins P94-P97 port/segment output specification */
                    /*|||+------- : Pins P100-P102 port/segment output specification */
                    /*||+-------- : Pins P140-P143 port/segment output specification */
                    /*|+--------- : Pins P144-P147 port/segment output specification */
                    /*| : 0 : Used the pins as port (other than segment output) */
                    /*| : 1 : Used the pins as segment output */
                    /*| */
                    /*+---------- : Be sure to set 0 */
ISC =    0b00000000;                /* Input Switch Control Register */
                    /*|||||||+--- : Switching external interrupt (INTP0) input */
                    /*||||||| : 0 : Uses the input signal of the INTP0 pin as an external interrupt (normal
operation). */
                    /*||||||| : 1 : Uses the input signal of the RXD3 pin as an external interrupt */
                    /*|||||||       (to measure the pulse widths of the sync break field and sync field). */
                    /*||||||| */
                    /*||||||+---- : ISC1 Switching channel 7 input of timer array unit TAUS */
                    /*|||||| : 0 : Uses the input signal of the TI07 pin as a timer input (normal operation). */
```

```
                    /*||||||  : 1 : Input signal of RXD3 pin is used as timer input (wakeup signal detection). */
                    /*|||||| */
                    /*|||||+----- : RxD3/SEG53/P50 pin schmitt-triggered buffer control */
                    /*||||| : 0 : Disables input */
                    /*||||| : 1 : Enables input */
                    /*||||| */
                    /*||||+------ : TI02/SEG51/P52 pin schmitt-triggered buffer control */
                    /*|||| : 0 : Disables input */
                    /*|||| : 1 : Enables input */
                    /*|||| */
                    /*|||+------- : TI04/SEG50/P53 pin schmitt-triggered buffer control */
                    /*||| : 0 : Disables input */
                    /*||| : 1 : Enables input */
                    /*||| */
                    /*+++-------- : Be sure to set 000 */

fn_DisplayAllClear();          /* clear all LCD String area */

LCDC0 =  0b00110011;                  /* LCD Clock Control Register */
                    /*|||||+++--- : LCD clock (LCDCL) selection */
                    /*||||| : 0 0 0 : fLCD/2^4 */
                    /*||||| : 0 0 1 : fLCD/2^5 */
                    /*||||| : 0 1 0 : fLCD/2^6 */
                    /*||||| : 0 1 1 : fLCD/2^7 */
                    /*||||| : 1 0 0 : fLCD/2^8 */
                    /*||||| : 1 0 1 : fLCD/2^9 */
                    /*||||| : Other than above : Setting prohibited */
                    /*||||| */
                    /*++||+------ : Be sure to set 000 */
                    /*   || */
                    /*   ++------- : LCD source clock (fLCD) selection */
                    /* : 0 0 : fSUB */
                    /* : 0 1 : fCLK/2^6 */
                    /* : 1 0 : fCLK/2^7 */
                    /* : 1 1 : fCLK/2^8 */

VLCD = 0x0A;                          /* set LCD boost level to 5V */
/* wait for the reference voltage setup time (2ms(min.)) */
fn_Wait500usBase(2000/500);
```

```
LCDM =    0b00100111;                 /* LCD Display Mode Register */
                    /*|||||+++--- : LCD controller/driver display mode selection */
                    /*|||||    *When the external resistance division method is used */
                    /*|||||    : 0 0 0 : Four-time-slice mode & 1/3 bias method */
                    /*|||||    : 0 0 1 : Three-time-slice mode & 1/3 bias method */
                    /*|||||    : 0 1 0 : Two-time-slice mode & 1/2 bias method */
                    /*|||||    : 0 1 1 : Three-time-slice mode & 1/2 bias method */
                    /*|||||    : 1 0 0 : Static */
                    /*|||||    : 1 1 1 : Eight-time-slice mode & 1/4 bias method */
                    /*||||| */
                    /*|||||    *When the internal voltage boosting method is used */
                    /*|||||    : 0 0 0 : Four-time-slice mode & 1/3 bias method */
                    /*|||||    : 0 0 1 : Three-time-slice mode & 1/3 bias method */
                    /*|||||    : 0 1 0 : Four-time-slice mode & 1/3 bias method */
                    /*|||||    : 0 1 1 : Four-time-slice mode & 1/3 bias method */
                    /*|||||    : 1 0 0 : Setting prohibited */
                    /*|||||    : 1 1 1 : Eight-time-slice mode & 1/4 bias method */
                    /*||||| */
                    /*|||||    *When the capacitor split method is used */
                    /*|||||    : 0 0 0 : Four-time-slice mode & 1/3 bias method */
                    /*|||||    : 0 0 1 : Three-time-slice mode & 1/3 bias method */
                    /*|||||    : 0 1 0 : Four-time-slice mode & 1/3 bias method */
                    /*|||||    : 0 1 1 : Four-time-slice mode & 1/3 bias method */
                    /*|||||    : 1 0 0 : Setting prohibited */
                    /*|||||    : 1 1 1 : Four-time-slice mode & 1/3 bias method */
                    /*||||| */
                    /*|||||    : Other than above : Setting prohibited
                    /*||||| */
                    /*|||++------ : LCD display data area control */
                    /*||| : 0 0 : Display the data of an A pattern area */
                    /*|||           (lower 4 bits of LCD display data memory) */
                    /*||| : 0 1 : Display the data of an A pattern area */
                    /*|||           (higher 4 bits of LCD display data memory) */
                    /*||| : 1 0 : Display the data of an A pattern area and the B pattern area in turn. */
                    /*|||           (The on and off light indication which synchronized */
                    /*|||            in a constant-period interrupt timing of RTC) */
                    /*||| : 1 1 : Display the data of an A pattern area and the B pattern area in turn. */
                    /*|||           (The on and off light indication which synchronized */
```

```
                                /*|||           in a constant-period interrupt timing of RTC) */
                                /*||| */
                                /*||+-------- : Voltage boost circuit and capacitor split circuit operation enable/disable */
                                /*|| : 0 : Stops voltage boost circuit and capacitor split circuit operation */
                                /*|| : 1 : Enables voltage boost circuit and capacitor split circuit operation */
                                /*|| */
                                /*++--------- : LCD display enable/disable */
                                /* : 0 0 : Output ground level to segment/common pin */
                                /* : 0 1 : Display off (all segment outputs are deselected.) */
                                /* : 1 0 : Output ground level to segment/common pin */
                                /* : 1 1 : Display on */

        /* software to wait for the operation stabilization time (over 500ms) */
        fn_Wait500usBase(500000/500);

        SCOC = 1;                               /* output deselect level to SEG and LCD waveform to COM */
        LCDON = 1;                              /* display on */

        /*-----------------------------------------*/
        /*          Initialization of variables          */
        /*-----------------------------------------*/
/*      bDispUpdate = 1;                /* initialize display update request flag */
/*      bBlinkOn = 0;                           /* initialize blink on flag */
/*      ushOpeningMessage = 0;          /* initialize opening message counter */
}

/*-------------------------------------------------------------*/
/* Module:          fn_LcdWrite
                                */
/* Description:     Write data to LCD RAM
        */
/*      parameter: position in whitch display begins                                    */
/*                                  address of display data
        */
/*                                  size of display data
                */
/*      return     : --
                                */
/*-------------------------------------------------------------*/
```

```
static void fn_LcdWrite(unsigned char *Position, unsigned char *DataAddr, unsigned char DataSize)
{
        memcpy(Position, DataAddr, DataSize);
}

/*--------------------------------------------------------------*/
/* Module:          fn_DisplayAllClear
        */
/* Description:     LCD RAM all clear
        */
/*      parameter: --
                                */
/*      return    : --
                                */
/*--------------------------------------------------------------*/
void fn_DisplayAllClear(void)
{
        fn_LcdWrite(CLCDPOS_START, aDispClear, CLCDSIZE_ALL);
}


/*--------------------------------------------------------------*/
/* Module:          fn_Display
                    */
/* Description:     Display count to LCD                                                      */
/*      parameter: Count data(Hex)
        */
/*      return    : --
                                */
/*--------------------------------------------------------------*/
void fn_Display(unsigned short ushCount)
{
        unsigned short          ushNumber;                      /* number work for display */
        /* space */
/*      fn_LcdWrite(CLCDPOS_START, aDispClear, (unsigned char)(CLCDPOS_REF - CLCDPOS_START));
*/
        fn_LcdWrite(CLCDPOS_START, aDispClear, (unsigned char)(CLCDPOS_1000 - CLCDPOS_START));
```

```
        /************************/
        /*   display Count data    */
        /************************/
        /* display sign of TH */
/*      fn_LcdWrite(CLCDPOS_TH, aDispTh, sizeof aDispTh);
*/

        /* space */
/*      fn_LcdWrite((CLCDPOS_TH100 - 2), aDispClear, 2);
*/

        /* display thousands place of Count */
        fn_LcdWrite         (CLCDPOS_1000,
                                    aDispNumber[(unsigned char)(ushCount / 1000)],
                                    CLCDSIZE_NUMBER);

        /* clear digit gap */
/*      fn_LcdWrite((CLCDPOS_TH100 - 1), aDispClear, 1);
*/

        /* display hundreds place of TH */
        ushCount %= 1000;
        fn_LcdWrite         (CLCDPOS_100,
                                    aDispNumber[(unsigned char)(ushCount / 100)],
                                    CLCDSIZE_NUMBER);

        /* clear digit gap */
/*      fn_LcdWrite((CLCDPOS_TH10 - 1), aDispClear, 1);
*/
        /* display tenths place of TH */
        ushCount %= 100;
        fn_LcdWrite         (CLCDPOS_10,
                                    aDispNumber[(unsigned char)(ushCount / 10)],
                                    CLCDSIZE_NUMBER);

        /* clear digit gap */
/*      fn_LcdWrite((CLCDPOS_TH1 - 1), aDispClear, 1);
*/
```

```
        /* display ones place of TH */
        fn_LcdWrite         (CLCDPOS_1,
                            aDispNumber[(unsigned char)(ushCount % 10)],
                            CLCDSIZE_NUMBER);

        /* space */
/*      fn_LcdWrite((CLCDPOS_TH1 + CLCDSIZE_NUMBER), aDispClear, 1);
*/
}
```

(10) CR oscillator 2

● main.c

```
/*
 * Copyright (C) NEC Electronics Corporation 2006
 * NEC ELECTRONICS CONFIDENTIAL AND PROPRIETARY
 * All rights reserved by NEC Electronics Corporation.
 * This program must be used solely for the purpose for which
 * it was furnished by NEC Electronics Corporation. No part of this
 * program may be reproduced or disclosed to others, in any
 * form, without the prior written permission of NEC Electronics
 * Corporation. Use of copyright notice dose not evidence
 * publication of the program.
 */

/*-----------------------------------------------------------------------*/
/* #pragma directive for CC78K0
                                      */
/*-----------------------------------------------------------------------*/
#pragma         SFR
#pragma         DI
#pragma         EI
#pragma         HALT
#pragma         NOP

/* #pragma interrupt INTDMA0 fn_intdma0 RB1 */

/*-----------------------------------------------------------------------*/
/* Include files
                                                  */
/*-----------------------------------------------------------------------*/
/* TAU:TDR0n value operation by CK00 (fCLK/2^3 = 5MHz) */
#define CCK00_500USEC   (2500 - 1)          /* 500us (0.2[us/clk] * 2500[count]) */



/*-----------------------------------------------------------------------*/
/* Function prototyps
                                                  */
/*-----------------------------------------------------------------------*/
```

```
                    void      fn_Wait500usBase(unsigned short);      /* Delays the program for (Time * 500us) */
                    void      fn_InitPort(void);                                        /* Setting of I/O ports
*/
                    void      fn_InitTau0(void);                                        /* Setting of Timer
array unit 0 */
                    void      fn_InitLvi(void);                                         /* Setting of
Low-voltage detectior */
                    void      fn_InitVr(void);                                          /* Setting of Voltage
reference */
                    void      fn_AdcDacInit(void);                            /* Setting of D/A converter */

    extern  void      fn_DisplayInit(void);                          /* Setting of LCD driver */
    extern  void      fn_Display(unsigned short ushRef, unsigned short ushTh);      /* LCD display */

    /*------------------------------------------------------------------------*/
    /* Extern variables/constants
                    */
    /*------------------------------------------------------------------------*/

    /*------------------------------------------------------------------------*/
    /* Local constants
                                        */
    /*------------------------------------------------------------------------*/
    /*------------------------------------------------------------------------*/
    /* Global variables
                                        */
    /*------------------------------------------------------------------------*/

    /*------------------------------------------------------------------------*/
    /* Local variables
                                        */
    /*------------------------------------------------------------------------*/
    static  short           shVoltage;/* voltage (Hex) */
    static  unsigned char   ucMeasureStatus;   /* measuring status */

    static  unsigned short  ushR_REF1;         /* R_REF at 1V */
    static  unsigned short  ushR_REF2;         /* R_REF at 2V */
    static  unsigned short  ushR_TH1;          /* R_TH at 1V */
    static  unsigned short  ushR_TH2;          /* R_TH at 2V */
```

```
/*--------------------------------------------------------------------*/
/* Code
                                                                  */
/*--------------------------------------------------------------------*/

/*--------------------------------------------------------------------*/
/* Hardware initialization
                              */
/*--------------------------------------------------------------------*/
void hdwinit(void)
{
        DI();                           /* disable all interrupts */

        /*------------------------------------------*/
        /*          Initialization of port          */
        /*------------------------------------------*/
        fn_InitPort();

        /*-----------------------------------------*/
        /*         Initialization of clock         */
        /*-----------------------------------------*/
        CMC =   0b01010011;             /* Clock Operation Mode Control Register */
                        /*|||||||+--- : Control of high-speed system clock oscillation frequency */
                        /*||||||| : 0 : 2 MHz <= fMX <= 10 MHz */
                        /*||||||| : 1 : 10 MHz < fMX <= 20 MHz */
                        /*||||||| */
                        /*|||||++---- : XT1 oscillator oscillation mode selection */
                        /*||||| : 0 0 : Low-consumption oscillation */
                        /*||||| : 0 1 : Normal oscillation */
                        /*||||| : 1 x : Super-low-consumption oscillation */
                        /*|||||    x = don't care */
                        /*||||| */
                        /*||||+------ : Be sure to set 0 */
                        /*|||| */
                        /*|||+------- : [1] Subsystem clock pin operation mode */
                        /*|||             [2] XT1/P123 pin and XT2/P124 pin */
                        /*||| : 0 : [1]Input port mode */
                        /*|||         [2]Input port */
```

```
                    /*||| */
                    /*||| : 1 : [1]XT1 oscillation mode */
                    /*|||         [2]Crystal resonator connection */
                    /*||| */
                    /*||+-------- : Be sure to set 0 */
                    /*|| */
                    /*++--------- : [1]EXCLK OSCSEL High-speed system clock pin operation mode */
                    /*                   [2]X1/P121 pin */
                    /*                   [3]X2/EXCLK/P122 pin */
                    /* : 0 0 : [1]Input port mode */
                    /*          [2][3]Input port */
                    /* */
                    /* : 0 1 : [1]X1 oscillation mode */
                    /*          [2][3]Crystal/ceramic resonator connection */
                    /* */
                    /* : 1 0 : [1]Input port mode */
                    /*          [2][3]Input port */
                    /* */
                    /* : 1 1 : [1]External clock input mode */
                    /*          [2]Input port */
                    /*          [3]External clock input */

    MSTOP = 0;                          /* X1 oscillator operating */
    XTSTOP = 0;                         /* XT1 oscillator operating */

    OSMC =   0b00000001;                /* Operation Speed Mode Control Register */
                    /*||||||++--- : fCLK frequency selection */
                    /*|||||| : 0 0 : Operates at a frequency of 10 MHz or less. */
                    /*|||||| : 0 1 : Operates at a frequency higher than 10 MHz. */
                    /*|||||| : 1 0 : Operates at a frequency of 1 MHz. */
                    /*|||||| */
                    /*|+++++----- : Be sure to set 00000 */
                    /*| */
                    /*+---------- : Setting in subsystem clock HALT mode */
                    /* : 0 : Enables subsystem clock supply to peripheral functions. */
                    /*       (See Table 21-1 Operating Statuses in HALT Mode (2/3) */
                    /*        for the peripheral functions whose operations are enabled.) */
                    /* : 1 : Stops subsystem clock supply to peripheral functions except real-time counter,
*/
```

```
                    /*          clock output/buzzer output, and LCD controller/driver. */

while(OSTC.0 != 1){                     /* wait X1 oscillation stabilization */
        NOP();
}

/*-- Caution ---------------------------------------------*/
/* To increase fCLK to 10 MHz or higher, set FSEL to '1',    */
/* then change fCLK after two or more clocks have elapsed. */
/*---------------------------------------------------------*/
NOP();
NOP();

CKC =   0b00010000;                     /* System Clock Control Register */
                    /*|+|+++++--- : Selection of CPU/peripheral hardware clock (fCLK) */
                    /*| | : 0 0 x 0 0 0 : fIH */
                    /*| | : 0 0 x 0 0 1 : fIH/2 (default) */
                    /*| | : 0 0 x 0 1 0 : fIH/2^2 */
                    /*| | : 0 0 x 0 1 1 : fIH/2^3 */
                    /*| | : 0 0 x 1 0 0 : fIH/2^4 */
                    /*| | : 0 0 x 1 0 1 : fIH/2^5 */
                    /*| | : 0 1 x 0 0 0 : fMX */
                    /*| | : 0 1 x 0 0 1 : fMX/2 */
                    /*| | : 0 1 x 0 1 0 : fMX/2^2 */
                    /*| | : 0 1 x 0 1 1 : fMX/2^3 */
                    /*| | : 0 1 x 1 0 0 : fMX/2^4 */
                    /*| | : 0 1 x 1 0 1 : fMX/2^5 */
                    /*| | : 1 x 0 x x x : fSUB */
                    /*| | : 1 x 1 x x x : fSUB/2 */
                    /*| | : Other than above : Setting prohibited */
                    /*| |    x = don't care */
                    /*| | */
                    /*| +-------- : Status of Main system clock (fMAIN) */
                    /*| : 0 : Internal high-speed oscillation clock (fIH) */
                    /*| : 1 : High-speed system clock (fMX) */
                    /*| */
                    /*+---------- : Status of CPU/peripheral hardware clock (fCLK) */
                    /* : 0 : Main system clock (fMAIN) */
                    /* : 1 : Subsystem clock (fSUB) */
```

```
/* Confirming the CPU clock status */
while((CLS != 0)||(MCS != 1)){
		NOP();
}
/* CPU is operating on a High-speed system clock */
HIOSTOP = 1;				/* internal high-speed oscillation stopped */

OSTS =	0b00000111;			/* Oscillation Stabilization Time Select Register */
			/*||||++++--- : Oscillation stabilization time selection */
			/*|||| : 0 0 0 : 2^8/fX */
			/*|||| : 0 0 1 : 2^9/fX */
			/*|||| : 0 1 0 : 2^10/fX */
			/*|||| : 0 1 1 : 2^11/fX */
			/*|||| : 1 0 0 : 2^13/fX */
			/*|||| : 1 0 1 : 2^15/fX */
			/*|||| : 1 1 0 : 2^17/fX */
			/*|||| : 1 1 1 : 2^18/fX */
			/*|||| */
			/*++++------- : Be sure to set 000000 */

/*----------------------------------------*/
/*		Initialization of timer		*/
/*----------------------------------------*/
fn_InitTau0();

/* software to wait for the operation stabilization time */
/* (over 200ms from when XT1 enable) */
fn_Wait500usBase(200000/500);

/*----------------------------------------*/
/* Initialization of low-voltage detector */
/*----------------------------------------*/
fn_InitLvi();

/*----------------------------------------*/
/*	Initialization of A/D,D/A,OPAmp			*/
/*----------------------------------------*/
fn_AdcDacInit();
```

```
        /*----------------------------------------*/
        /*   Initialization of LCD                                      */
        /*----------------------------------------*/
        fn_DisplayInit();


        EI();                                   /* enable all interrupts */

}

/*--------------------------------------------------------------*/
/* Module:          fn_InitPort
                    */
/* Description:     Setting of I/O ports                                                        */
/*          parameter: --
                                */
/*          return     : --
                                */
/*--------------------------------------------------------------*/
void    fn_InitPort(void)
{
/*----------------------------------------------------------*/
/*          Ports configuration for digital input and output                    */
/*----------------------------------------------------------*/
        ADPC =   0b00010000;            /* A/D Port Configuration Register */
                                /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                                /*||| :                +------------------------- ANI15/AVREFM/P157 */
                                /*||| :                | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                                /*||| :                | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
                                /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                                /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                                /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                                /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                                /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                                /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                                /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                                /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                                /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
```

```
                    /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                    /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
                    /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                    /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                    /*||| */
                    /*+++-------- : Be sure to set 000 */

/*------------------------------------------------------------*/
/*      Setting of Port 0
        */
/*------------------------------------------------------------*/
        P0 =    0b00000000;     /* Set P00-P02 Output latch to Low */
        PM0 =   0b11111000;     /* Set P00-P02 to output port */
                                        /* P00-P02:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 1
        */
/*------------------------------------------------------------*/
        P1 =    0b00000000;     /* Set P10-P17 Output latch to Low */
        PM1 =   0b00000000;     /* Set P10-P17 to output port */
                                        /* P10-P15:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 2
        */
/*------------------------------------------------------------*/
        P2 =    0b00000000;     /* Set P20-P27 Output latch to Low */
        PM2 =   0b11111111;     /* Set P20-P27 to input port */
                                        /* P20-P27:Unused */

/*------------------------------------------------------------*/
/*      Setting of Port 3
        */
/*------------------------------------------------------------*/
        P3 =    0b00001100;     /* Set P30-P31,P34 Output latch to Low */
                                        /* Set P33,P32 Output latch High */
        PM3 =   0b11110000;     /* Set P30-P33 to output port */
                                        /* Set P34(TI06) to input port */
```

```
                                                        /* P30-P33:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 4
        */
/*----------------------------------------------------------*/
        P4 =    0b00000000;     /* Set P40-P41 Output latch to Low */
        PM4 =   0b11111100;     /* Set P40-P41 to output port */
                                                        /* P40-P41:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 5
        */
/*----------------------------------------------------------*/
        P5 =    0b00000000;     /* Set P50-P57 Output latch to Low */
        PM5 =   0b11110000;     /* Set P50-P57 to output port */
                                                        /* P50-P57:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 6
        */
/*----------------------------------------------------------*/
        P6 =    0b00000000;     /* Set P60-P61 Output latch to Low */
        PM6 =   0b11111100;     /* Set P60-P61 to output port */
                                                        /* P60-P61:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 7
        */
/*----------------------------------------------------------*/
        P7 =    0b00000000;     /* Set P70-P77 Output latch to Low */
        PM7 =   0b00000000;     /* Set P70-P77 to output port */
                                                        /* P70-P77:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 8
        */
/*----------------------------------------------------------*/
        P8 =    0b00000000;     /* Set P80-P88 Output latch to Low */
```

```
        PM8 =     0b00000000;       /* Set P80-P88 to output port */
                                                  /* P80-P88:Unused */

/*-------------------------------------------------------*/
/*      Setting of Port 9
        */
/*-------------------------------------------------------*/
        P9 =      0b00000000;       /* Set P90-P97 Output latch to Low */
        PM9 =     0b00000000;       /* Set P90-P97 to output port */
                                                  /* P90-P97:Unused */

/*-------------------------------------------------------*/
/*      Setting of Port 10
        */
/*-------------------------------------------------------*/
        P10 =     0b00000000;       /* Set P100-P102 Output latch to Low */
        PM10 =    0b11111000;       /* Set P100-P102 to output port */
                                                  /* P100-P102:Unused */

/*-------------------------------------------------------*/
/*      Setting of Port 11
        */
/*-------------------------------------------------------*/
        P11 =     0b00000000;       /* Set P110-P111 Output latch to Low */
        PM11 =    0b11111100;       /* Set P110-P111 to output port */
                                                  /* P110-P111:Unused */

/*-------------------------------------------------------*/
/*      Setting of Port 12
        */
/*-------------------------------------------------------*/
        P12 =     0b00000000;       /* Set P120 Output latch to Low */
        PM12 =    0b11111111;       /* Set P120 to output port */
                                                  /* P121-P124:Unused */
                                                  /* *P120-P124:Input port */

/*-------------------------------------------------------*/
/*      Setting of Port 13
        */
```

```
/*----------------------------------------------------------*/
        P13 =       0b00000000;         /* Set P130 Output latch to Low */
                                                              /* P130:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 14
        */
/*----------------------------------------------------------*/
        P14 =       0b00000000;         /* Set P140-P147 Output latch to Low */
        PM14 =      0b00000000;         /* Set P140-P147 to output port */
                                                              /* P140-P147:Unused */

/*----------------------------------------------------------*/
/*      Setting of Port 15
        */
/*----------------------------------------------------------*/
        P15 =       0b00000000;         /* Set P150-P152,P157 Output latch to Low */
        PM15 =      0b11111111;         /* Set P150-P152,P157 to input port */
                                                              /* P150-P152,P157:Unused */

}

/*--------------------------------------------------------------*/
/* Module:          fn_InitTau0
                            */
/* Description:     Setting of Timer array unit 0                                          */
/*      parameter: --
                            */
/*      return     : --
                            */
/*--------------------------------------------------------------*/
void    fn_InitTau0(void)
{
        TAU0EN = 1;                             /* supplies input clock to timer array unit 0 */
        TPS0L =     0b00000010;         /* Timer Clock Select Register 0 */
                            /*||||++++--- : Selection of operation clock (CK00) */
                            /*++++------- : Selection of operation clock (CK01) */
                            /* : 0 0 0 0 : CK0m = fCLK */
                            /* : 0 0 0 1 : CK0m = fCLK/2 */
```

```
                    /* : 0 0 1 0 : CK0m = fCLK/2^2 */
                    /* : 0 0 1 1 : CK0m = fCLK/2^3 */
                    /* : 0 1 0 0 : CK0m = fCLK/2^4 */
                    /* : 0 1 0 1 : CK0m = fCLK/2^5 */
                    /* : 0 1 1 0 : CK0m = fCLK/2^6 */
                    /* : 0 1 1 1 : CK0m = fCLK/2^7 */
                    /* : 1 0 0 0 : CK0m = fCLK/2^8 */
                    /* : 1 0 0 1 : CK0m = fCLK/2^9 */
                    /* : 1 0 1 0 : CK0m = fCLK/2^10 */
                    /* : 1 0 1 1 : CK0m = fCLK/2^11 */
                    /* : 1 1 0 0 : CK0m = fCLK/2^12 */
                    /* : 1 1 0 1 : CK0m = fCLK/2^13 */
                    /* : 1 1 1 0 : CK0m = fCLK/2^14 */
                    /* : 1 1 1 1 : CK0m = fCLK/2^15 */
                    /*    m = 0, 1 */

/* CH1:for wait */
TMR01 = 0b0000000000000000;         /* Timer Mode Register 01 */
                    /*||||||||||||++++--- : [1]Operation mode of channel 1 */
                    /*||||||||||||          [2]Count operation of TCR */
                    /*||||||||||||          [3]Independent operation */
                    /*||||||||||||          [4]Setting of starting counting and interrupt */
                    /*|||||||||||| : 0 0 0 0 : [1]Interval timer mode */
                    /*||||||||||||               [2]Counting down */
                    /*||||||||||||               [3]Possible */
                    /*||||||||||||               [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                  (timer output does not change, either). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 0 0 1 : [1]Interval timer mode */
                    /*||||||||||||               [2]Counting down */
                    /*||||||||||||               [3]Possible */
                    /*||||||||||||               [4]Timer interrupt is generated when counting is started */
                    /*||||||||||||                  (timer output also changes). */
                    /*|||||||||||| */
                    /*|||||||||||| : 0 1 0 0 : [1]Capture mode */
                    /*||||||||||||               [2]Counting up */
                    /*||||||||||||               [3]Possible */
                    /*||||||||||||               [4]Timer interrupt is not generated when counting is started */
                    /*||||||||||||                  (timer output does not change, either). */
```

```
/*|||||||||||| */
/*|||||||||||| : 0 1 0 1 : [1]Capture mode */
/*||||||||||||              [2]Counting up */
/*||||||||||||              [3]Possible */
/*||||||||||||              [4]Timer interrupt is generated when counting is started */
/*||||||||||||                 (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 1 0 : [1]Event counter mode */
/*||||||||||||              [2]Counting down */
/*||||||||||||              [3]Possible */
/*||||||||||||              [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                 (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 0 : [1]One-count mode */
/*||||||||||||              [2]Counting down */
/*||||||||||||              [3]Impossible */
/*||||||||||||              [4]Start trigger is invalid during counting operation. */
/*||||||||||||                 At that time, interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 1 : [1]One-count mode */
/*||||||||||||              [2]Counting down */
/*||||||||||||              [3]Impossible */
/*||||||||||||              [4]Start trigger is valid during counting operation. */
/*||||||||||||                 At that time, interrupt is also generated. */
/*|||||||||||| */
/*|||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*||||||||||||              [2]Counting up */
/*||||||||||||              [3]Possible */
/*||||||||||||              [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                 (timer output does not change, either). */
/*||||||||||||                 Start trigger is invalid during counting operation. */
/*||||||||||||                 At that time interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : Other than above : Setting prohibited */
/*|||||||||||| */
/*||||||||||++------- : Be sure to set 00 */
/*|||||||||| */
/*||||||||++--------- : Selection of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 valid
edge */
```

```
                                        /*||||||||                    (the timer input used with channel 1 is selected by using TIS0
register). */
                                        /*|||||||| : 0 0 : Falling edge */
                                        /*|||||||| : 0 1 : Rising edge */
                                        /*|||||||| : 1 0 : Both edges (when low-level width is measured) */
                                        /*||||||||             Start trigger: Falling edge, Capture trigger: Rising edge */
                                        /*|||||||| : 1 1 : Both edges (when high-level width is measured) */
                                        /*||||||||             Start trigger: Rising edge, Capture trigger: Falling edge */
                                        /*|||||||| */
                                        /*|||||+++----------- : Setting of start trigger or capture trigger of channel 1 */
                                        /*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are unselected).
*/
                                        /*||||| : 0 0 1 : Valid edge of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 is used
as both the start trigger and capture trigger. */
                                        /*||||| : 0 1 0 : Both the edges of TI01 pin input signal, fSUB/2, fSUB/4, or INTRTC1 are
used as a start trigger and a capture trigger. */
                                        /*||||| : 1 0 0 : Interrupt signal of the master channel is used (when the channel is used
as a slave channel with the combination operation function). */
                                        /*||||| : Other than above : Setting prohibited */
                                        /*||||| */
                                        /*||||+-------------- : Selection of slave/master of channel 1 */
                                        /*|||| : 0 : Operates as slave channel with combination operation function. */
                                        /*|||| : 1 : Operates as master channel with combination operation function. */
                                        /*|||| */
                                        /*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
                                        /*||| : 0 : Operation clock MCK specified by CKS01 bit */
                                        /*||| : 1 : Valid edge of input signal input from TI01 pin, fSUB/2, fSUB/4, or INTRTC1 */
                                        /*|||        (the timer input used with channel 1 is selected by using TIS0 register). */
                                        /*||| */
                                        /*|++---------------- : Be sure to set 00 */
                                        /*| */
                                        /*+------------------ : Selection of operation clock (MCK) of channel 1 */
                                        /* : 0 : Operation clock CK00 set by TPS0 register */
                                        /* : 1 : Operation clock CK01 set by TPS0 register */
        TDR01 = CCK00_500USEC;                  /* set interval time to 500us */
        TMMK01 = 1;                                                     /* disable interrupt */

        /* CH6:for TI06 */
        TMR06 = 0b1000000101000100;             /* Timer Mode Register 06 */
```

```
/*||||||||||||++++--- : [1]Operation mode of channel 6 */
/*||||||||||||          [2]Count operation of TCR */
/*||||||||||||          [3]Independent operation */
/*||||||||||||          [4]Setting of starting counting and interrupt */
/*|||||||||||| : 0 0 0 0 : [1]Interval timer mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 0 0 0 1 : [1]Interval timer mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is generated when counting is started */
/*||||||||||||                (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 0 0 : [1]Capture mode */
/*||||||||||||             [2]Counting up */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 0 1 0 1 : [1]Capture mode */
/*||||||||||||             [2]Counting up */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is generated when counting is started */
/*||||||||||||                (timer output also changes). */
/*|||||||||||| */
/*|||||||||||| : 0 1 1 0 : [1]Event counter mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                (timer output does not change, either). */
/*|||||||||||| */
/*|||||||||||| : 1 0 0 0 : [1]One-count mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Impossible */
/*||||||||||||             [4]Start trigger is invalid during counting operation. */
/*||||||||||||                At that time, interrupt is not generated, either. */
```

```
/*|||||||||||| */
/*|||||||||||| : 1 0 0 1 : [1]One-count mode */
/*||||||||||||             [2]Counting down */
/*||||||||||||             [3]Impossible */
/*||||||||||||             [4]Start trigger is valid during counting operation. */
/*||||||||||||                At that time, interrupt is also generated. */
/*|||||||||||| */
/*|||||||||||| : 1 1 0 0 : [1]Capture & one-count mode */
/*||||||||||||             [2]Counting up */
/*||||||||||||             [3]Possible */
/*||||||||||||             [4]Timer interrupt is not generated when counting is started */
/*||||||||||||                (timer output does not change, either). */
/*||||||||||||                Start trigger is invalid during counting operation. */
/*||||||||||||                At that time interrupt is not generated, either. */
/*|||||||||||| */
/*|||||||||||| : Other than above : Setting prohibited */
/*|||||||||||| */
/*||||||||||++------- : Be sure to set 00 */
/*|||||||||| */
/*||||||||++--------- : Selection of TI06 pin input signal, fSUB/2, fSUB/4, or INTRTC1 valid edge */
/*||||||||                  (the timer input used with channel 4 is selected by using TIS0 register). */
/*|||||||| : 0 0 : Falling edge */
/*|||||||| : 0 1 : Rising edge */
/*|||||||| : 1 0 : Both edges (when low-level width is measured) */
/*||||||||          Start trigger: Falling edge, Capture trigger: Rising edge */
/*|||||||| : 1 1 : Both edges (when high-level width is measured) */
/*||||||||          Start trigger: Rising edge, Capture trigger: Falling edge */
/*|||||||| */
/*|||||+++----------- : Setting of start trigger or capture trigger of channel 6 */
/*||||| : 0 0 0 : Only software trigger start is valid (other trigger sources are unselected). */
/*||||| : 0 0 1 : Valid edge of TI06 pin input signal, fSUB/2, fSUB/4, or INTRTC1 is used as both the start trigger and capture trigger. */
/*||||| : 0 1 0 : Both the edges of TI06 pin input signal, fSUB/2, fSUB/4, or INTRTC1 are used as a start trigger and a capture trigger. */
/*||||| : 1 0 0 : Interrupt signal of the master channel is used (when the channel is used as a slave channel with the combination operation function). */
```

```
                            /*||||| : Other than above : Setting prohibited */
                            /*||||| */
                            /*||||+-------------- : Selection of slave/master of channel 6 */
                            /*|||| : 0 : Operates as slave channel with combination operation function. */
                            /*|||| : 1 : Operates as master channel with combination operation function. */
                            /*|||| */
                            /*|||+--------------- : Selection of count clock (TCLK) of channel 0 */
                            /*||| : 0 : Operation clock MCK specified by CKS04 bit */
                            /*||| : 1 : Valid edge of input signal input from TI04 pin, fSUB/2, fSUB/4, or INTRTC1 */
                            /*|||       (the timer input used with channel 4 is selected by using TIS0 register). */
                            /*||| */
                            /*|++---------------- : Be sure to set 00 */
                            /*| */
                            /*+------------------ : Selection of operation clock (MCK) of channel 4 */
                            /* : 0 : Operation clock CK00 set by TPS0 register */
                            /* : 1 : Operation clock CK01 set by TPS0 register */
        TMMK06 = 1;                 /* disable interrupt */

}

/*--------------------------------------------------------------*/
/* Module:          fn_InitLvi
                    */
/* Description:     Setting of Low-voltage detectior                        */
/*      parameter: --
                            */
/*      return    : --
                            */
/*--------------------------------------------------------------*/
void    fn_InitLvi(void)
{
        unsigned short      loop;       /* waiting counter */

        LVIMK = 1;                              /* disable LVI interrupt */

        LVIS =     0b00001001;                  /* Low-Voltage Detection Level Select Register */
                            /*||||++++--- : Detection level */
                            /*|||| : 0 0 0 0 : VLVI0 (4.22V) */
                            /*|||| : 0 0 0 1 : VLVI1 (4.07V) */
```

```
                        /*||||  : 0 0 1 0 : VLVI2 (3.92V) */
                        /*||||  : 0 0 1 1 : VLVI3 (3.76V) */
                        /*||||  : 0 1 0 0 : VLVI4 (3.61V) */
                        /*||||  : 0 1 0 1 : VLVI5 (3.45V) */
                        /*||||  : 0 1 1 0 : VLVI6 (3.30V) */
                        /*||||  : 0 1 1 1 : VLVI7 (3.15V) */
                        /*||||  : 1 0 0 0 : VLVI8 (2.99V) */
                        /*||||  : 1 0 0 1 : VLVI9 (2.84V) */
                        /*||||  : 1 0 1 0 : VLVI10 (2.68V) */
                        /*||||  : 1 0 1 1 : VLVI11 (2.53V) */
                        /*||||  : 1 1 0 0 : VLVI12 (2.38V) */
                        /*||||  : 1 1 0 1 : VLVI13 (2.22V) */
                        /*||||  : 1 1 1 0 : VLVI14 (2.07V) */
                        /*||||  : 1 1 1 1 : VLVI15 (1.91V) */
                        /*||||  */
                        /*++++------- : Be sure to set 0000 */

        LVIM =    0b10000000;              /* Low-Voltage Detection Register */
                        /*|||||||+--- : LVIF Low-voltage detection flag */
                        /*||||||| : 0 : * LVISEL = 0: VDD >= VLVI, or when LVI operation is disabled */
                        /*|||||||       * LVISEL = 1: EXLVI >= VEXLVI, or when LVI operation is disabled */
                        /*||||||| : 1 : * LVISEL = 0: VDD < VLVI */
                        /*|||||||       * LVISEL = 1: EXLVI < VEXLVI */
                        /*||||||| */
                        /*||||||+---- : Low-voltage detection operation mode (interrupt/reset) selection(LVIMD) */
                        /*|||||| : 0 : * LVISEL = 0: Generates an internal interrupt signal */
                        /*||||||                when VDD drops lower than VLVI (VDD < VLVI) */
                        /*||||||                or when VDD becomes VLVI or higher (VDD >= VLVI). */
                        /*||||||        * LVISEL = 1: Generates an interrupt signal */
                        /*||||||                when EXLVI drops lower than VEXLVI (EXLVI < 
VEXLVI) */
                        /*||||||                or when EXLVI becomes VEXLVI or higher (EXLVI >= 
VEXLVI). */
                        /*|||||| : 1 : * LVISEL = 0: Generates an internal reset signal when VDD < VLVI */
                        /*||||||                and releases the reset signal when VDD >= VLVI. */
                        /*||||||        * LVISEL = 1: Generates an internal reset signal when EXLVI < VEXLVI 
*/
                        /*||||||                and releases the reset signal when EXLVI >= VEXLVI. */
                        /*|||||| */
```

```
                                /*|||||+----- : Voltage detection selection(LVISEL) */
                                /*||||| : 0 : Detects level of supply voltage (VDD) */
                                /*||||| : 1 : Detects level of input voltage from external input pin (EXLVI) */
                                /*||||| */
                                /*|++++------ : Be sure to set 0000 */
                                /*| */
                                /*+---------- : Enables low-voltage detection operation */
                                /* : 0 : Disables operation */
                                /* : 1 : Enables operation */

        /* software to wait for the operation stabilization time (210us) */
        for(loop = 500; loop > 0; loop--){
                NOP();
        }
        /* wait until VLVI≤VDD */
        while( LVIF ){
                NOP();
        }

        LVIIF = 0;                      /* clear LVI interrupt request flag */
}



/*--------------------------------------------------------------*/
/* Module:          fn_Wait500usBase
        */
/* Description:     Delays the program for (Time * 500us)                     */
/*      parameter: wait time(Time)
        */
/*      return    : --
                                */
/*--------------------------------------------------------------*/
void    fn_Wait500usBase(unsigned short Time)
{
        TS0L.1 = 1;                             /* start TAU0 CH1 */
        TMIF01 = 0;

        for(; Time > 0; Time--){        /* wait for (parameter * 500)us */
                while(!TMIF01){
```

```
                        NOP();
                }
                TMIF01 = 0;
        }

        TT0L.1 = 1;                             /* stop TAU0 CH1 */
}


/*------------------------------------------------------------*/
/* Module:          fn_AdcDacInit
                */
/* Description:     Initialization of A/D D/A OPAmp module                  */
/*      parameter: --
                        */
/*      return     : --
                        */
/*------------------------------------------------------------*/
void    fn_AdcDacInit(void)
{
        ADCEN = 1;                              /* supplies input clock to A/D converter, operational amplifier,
and voltage reference */

        ADPC =  0b00000000;             /* A/D Port Configuration Register */
                        /*|||+++++--- : Analog input (A)/digital I/O (D) switching */
                        /*||| :                 +------------------------ ANI15/AVREFM/P157 */
                        /*||| :                 | +-+-+------------------- ANI10/P152 - ANI8/AMP2+/P150 */
                        /*||| :                 | | | | +-+-+-+-+-+-+-+--- ANI7/AMP2O/P27 - ANI0/AMP0-/P20 */
                        /*||| : 0 0 0 0 0 : A A A A A A A A A A A A */
                        /*||| : 0 0 0 0 1 : A A A A A A A A A A A D */
                        /*||| : 0 0 0 1 0 : A A A A A A A A A A D D */
                        /*||| : 0 0 0 1 1 : A A A A A A A A A D D D */
                        /*||| : 0 0 1 0 0 : A A A A A A A A D D D D */
                        /*||| : 0 0 1 0 1 : A A A A A A A D D D D D */
                        /*||| : 0 0 1 1 0 : A A A A A A D D D D D D */
                        /*||| : 0 0 1 1 1 : A A A A A D D D D D D D */
                        /*||| : 0 1 0 0 0 : A A A A D D D D D D D D */
                        /*||| : 0 1 0 0 1 : A A A D D D D D D D D D */
                        /*||| : 0 1 0 1 0 : A A D D D D D D D D D D */
```

```
                                        /*||| : 0 1 1 1 1 : A D D D D D D D D D D D */
                                        /*||| : 1 0 0 0 0 : D D D D D D D D D D D D */
                                        /*||| */
                                        /*+++-------- : Be sure to set 000 */

        /* wait for settling time to 17msec */
/*      fn_Wait500usBase(17000/500); */

        /*-------------------------------------------*/
        /*   Initialization of operational amplifier   */
        /*-------------------------------------------*/
        PM2.1 = 1;                          /* input port mode (AMP0O) */
        PM2.0 = 1;                          /* input port mode (AMP0-) */
        PM2.2 = 1;                          /* input port mode (AMP0+) */

        PM2.4 = 1;                          /* input port mode (AMP1O) */
        PM2.3 = 1;                          /* input port mode (AMP1-) */
        PM2.5 = 1;                          /* input port mode (AMP1+) */

        PM2.7 = 1;                          /* input port mode (AMP2O) */
        PM2.6 = 1;                          /* input port mode (AMP2-) */
        PM15.0 = 1;                         /* input port mode (AMP2+) */

        PM11.0 = 1;                         /* input port mode (ANO0) */

        ADCEN = 1;                          /* supplies input clock to OPAmp A/D converter */
        OAC.0 = 1;                          /* operational amplifier (AMP0) enable */
        OAC.1 = 0;                          /* operational amplifier (AMP1) disable */
        OAC.2 = 0;                          /* operational amplifier (AMP2) disable */

        /*-------------------------------------------*/
        /*        Initialization of D/A converter        */
        /*-------------------------------------------*/
        DACEN = 1;                                   /* supplies input clock to D/A converter */

        /* set D/A chanel1 */
        DAM.1 = 0;                                   /* [DAMD1] D/A converter operation is Normal mode
*/
        DAM.3 = 1;                                   /* [DARES1]D/A conversion resolution is 12-bit */
```

```
        DACSW1 = 1365;                      /* D/A conversion value setting reigster 1 */
                                                        /* ANO1 = 1V (1/3 * 2^12) */

}

/*------------------------------------------------------------*/
/* Module:          main
                                */
/* Description:     Main process
        */
/*      parameter: --
                                */
/*      return    : --
                                */
/*------------------------------------------------------------*/
void main(void)
{
        unsigned short      count;          /* counter for initializing D/A convert data */
        unsigned short      *pMeasure;      /* pointer for measuring data */


        DACE1 = 1;                                      /* D/A converter CH1 enable */

        ucMeasureStatus = 0;                /* measuring status */

        /*===================================*/
        /*---------------------------------*/
        /*              Main Loop               */
        /*---------------------------------*/
        /*===================================*/

        PM7.1 = 1;                                  /* set T_REF HiZ(disconnect) */
        PM7.2 = 1;                                  /* set T_TH HiZ(disconnect) */


        fn_Display(0, 0);               /* initial display */

        /*============================================================*/
        /* if system have nothing to do, go to standby for power-saving */
```

```
    /*==============================================================*/

    while (1){

        switch(ucMeasureStatus){            /* measuring status */
            case    0:                                          /* T_REF1 */
                        DACSW1 = 1365;                  /*    D/A    conversion
value setting reigster 1 */

    /* ANO1 = 1V (1/3 * 2^12) */
                        P7.0 = 1;
                        PM7.0 = 0;                              /*    charge
CAPACITOR */
                        NOP();
                        NOP();
                        NOP();
                        NOP();
                        NOP();
                        NOP();                                  /*        wait
charge complete */
                        PM7.0 = 1;                              /*        end
charge */

                        P7.1 = 0;
                        PM7.1 = 0;                              /*         set
T_REF low(start discharge) */
                        TS0L.6 = 1;                             /*       start
TIMER06 */
                        TMIF06 = 0;                             /*       clear
measure end */
                        while(TMIF06 == 0);             /*  wait  measure  end
*/
  /*                    while(P3.4 == 0);               /*  wait  measure  end
*/
                        ushR_REF1 = TDR06;              /* get measuring data
*/
  /*                    ushR_REF1 = TCR06;              /* get measuring data
*/
```

```
                                                TT0L.6 = 1;                             /*      stop
TAU0 CH6 */
                                                PM7.1 = 1;                              /*       set
T_REF HiZ(stop discharge) */

                                                ucMeasureStatus = 1;        /* set T_REF2 measure */
                                                break;


                                case    1:                                  /* T_REF2 */
                                                DACSW1 = 2730;                      /*   D/A   conversion
value setting reigster 1 */

        /* ANO1 = 2V (2/3 * 2^12) */
                                                P7.0 = 1;
                                                PM7.0 = 0;                              /*   charge
CAPACITOR */
                                                NOP();
                                                NOP();
                                                NOP();
                                                NOP();
                                                NOP();
                                                NOP();                                  /*      wait
charge complete */
                                                PM7.0 = 1;                              /*       end
charge */

                                                P7.1 = 0;
                                                PM7.1 = 0;                              /*       set
T_REF low(start discharge) */
                                                TS0L.6 = 1;                             /*     start
TIMER06 */
                                                TMIF06 = 0;                             /*     clear
measure end */
                                                while(TMIF06 == 0);         /* wait measure end */
                                                ushR_REF2 = TDR06;                  /* get measuring data
*/
                                                TT0L.6 = 1;                             /*      stop
TAU0 CH6 */
```

```
                                        PM7.1 = 1;                          /*      set
T_REF HiZ(stop discharge) */

                                        ucMeasureStatus = 2;        /* set T_TH1 measure */
                                        break;


                        case      2:                                 /* T_TH1 */
                                        DACSW1 = 1365;               /*   D/A   conversion
value setting reigster 1 */

        /* ANO1 = 1V (1/3 * 2^12) */
                                        P7.0 = 1;
                                        PM7.0 = 0;                          /*   charge
CAPACITOR */
                                        NOP();
                                        NOP();
                                        NOP();
                                        NOP();
                                        NOP();
                                        NOP();                              /*      wait
charge complete */
                                        PM7.0 = 1;                          /*      end
charge */

                                        P7.2 = 0;
                                        PM7.2 = 0;                          /*      set
T_TH low(start discharge) */
                                        TS0L.6 = 1;                         /*    start
TIMER06 */
                                        TMIF06 = 0;                         /*    clear
measure end */
                                        while(TMIF06 == 0);         /* wait measure end */
                                        ushR_TH1 = TDR06;           /* get measuring data */
                                        TT0L.6 = 1;                         /*     stop
TAU0 CH6 */
                                        PM7.2 = 1;                          /*      set
T_TH HiZ(stop discharge) */
```

```
ucMeasureStatus = 3;            /* set T_TH2 measure */
break;


case 3:                                          /* T_TH2 */
DACSW1 = 2730;                                   /* D/A conversion
value setting reigster 1 */

/* ANO1 = 2V (2/3 * 2^12) */
P7.0 = 1;
PM7.0 = 0;                                       /* charge
CAPACITOR */
NOP();
NOP();
NOP();
NOP();
NOP();
NOP();                                           /* wait
charge complete */
PM7.0 = 1;                                       /* end
charge */

P7.2 = 0;
PM7.2 = 0;                                       /* set
T_TH low(start discharge) */
TS0L.6 = 1;                                      /* start
TIMER06 */
TMIF06 = 0;                                      /* clear
measure end */
while(TMIF06 == 0);             /* wait measure end */
ushR_TH2 = TDR06;               /* get measuring data */
TT0L.6 = 1;                                      /* stop
TAU0 CH6 */
PM7.2 = 1;                                       /* set
T_TH HiZ(stop discharge) */

fn_Display((ushR_REF1 - ushR_REF2), (ushR_TH1 -
ushR_TH2));
```

```
        /* display T_TH,R_REF data to LCD */

                                        ucMeasureStatus = 0;        /* set T_REF1 measure */

                }
        }
}
```

● display.c

```
/*
 * Copyright (C) NEC Electronics Corporation 2006
 * NEC ELECTRONICS CONFIDENTIAL AND PROPRIETARY
 * All rights reserved by NEC Electronics Corporation.
 * This program must be used solely for the purpose for which
 * it was furnished by NEC Electronics Corporation. No part of this
 * program may be reproduced or disclosed to others, in any
 * form, without the prior written permission of NEC Electronics
 * Corporation. Use of copyright notice dose not evidence
 * publication of the program.
 */

/*--------------------------------------------------------------------*/
/* #pragma directive for CC78K0
                                    */
/*--------------------------------------------------------------------*/
#pragma         SFR

/*--------------------------------------------------------------------*/
/* Include files
                                              */
/*--------------------------------------------------------------------*/
/*#include      "defines.h"*/
#include        <string.h>

/*--------------------------------------------------------------------*/
/* Function prototyps
```

```
                                        */
/*--------------------------------------------------------------------*/
            void        fn_DisplayAllClear(void);                           /* Display all clear */
extern  void        fn_Wait500usBase(unsigned short);       /* Delays the program (500us base) */

/*--------------------------------------------------------------------*/
/* Extern variables/constants
            */
/*--------------------------------------------------------------------*/
/*--------------------------------------------------------------------*/
/* Local constants
                                        */
/*--------------------------------------------------------------------*/
/* size of area for displaying */
#define CLCDSIZE_ALL        (unsigned char)(&SEG53 - &SEG4 + 1)     /* all area */
/*#define           CLCDSIZE_NUMBER             4                   /* number */
#define CLCDSIZE_NUMBER             5                   /* number */

/* Display starting position */
#define CLCDPOS_START   &SEG4                   /* start of all area */
/*#define           CLCDPOS_<1>     &SEG5                   /* tenths place of <1> */
/*#define           CLCDPOS_<1>     &SEG10                  /* ones place of <1> */
/*#define           CLCDPOS_<2>     &SEG18                  /* tenths place of <2> */
/*#define           CLCDPOS_<2>     &SEG23                  /* ones place of <2> */
/*#define           CLCDPOS_<3>     &SEG31                  /* tenths place of <3> */
/*#define           CLCDPOS_<3>     &SEG36                  /* ones place of <3> */
/*#define           CLCDPOS_<4>     &SEG44                  /* tenths place of <4> */
/*#define           CLCDPOS_<4>     &SEG49                  /* ones place of <4> */


/*#define           CLCDPOS_REF                 &SEG5   /* sign of REF */
/*#define           CLCDPOS_REF100  &SEG12  /* hundreds place of REF */
/*#define           CLCDPOS_REF10   &SEG17  /* tenths place of REF */
/*#define           CLCDPOS_REF1    &SEG22  /* ones place of REF */
/*#define           CLCDPOS_TH                  &SEG32  /* sign of TH */
/*#define           CLCDPOS_TH100   &SEG39  /* hundreds place of TH */
/*#define           CLCDPOS_TH10    &SEG44  /* tenths place of TH */
/*#define           CLCDPOS_TH1                 &SEG49  /* ones place of TH */
```

```
#define CLCDPOS_REF1000        &SEG5   /* thousands place of REF */
#define CLCDPOS_REF100  &SEG10  /* hundreds place of REF */
#define CLCDPOS_REF10   &SEG15  /* tenths place of REF */
#define CLCDPOS_REF1    &SEG20  /* ones place of REF */
#define CLCDPOS_TH1000  &SEG34  /* thousands place of TH */
#define CLCDPOS_TH100   &SEG39  /* hundreds place of TH */
#define CLCDPOS_TH10    &SEG44  /* tenths place of TH */
#define CLCDPOS_TH1            &SEG49  /* ones place of TH */


/* Display data */
/*=====================================================================
  < LCD PANEL >

           SEG4    SEG5    SEG6    SEG7    SEG8      = = =     SEG49   SEG50   SEG51   SEG52
SEG53
          +------+------+------+------+------+-  = = =  -+------+------+------+------+------+
    COM0 |      |      |      |      |      |          |      |      |      |      |      |
          +------+------+------+------+------+-  = = =  -+------+------+------+------+------+
    COM1 |      |      |      |      |      |          |      |      |      |      |      |
          +------+------+------+------+------+-  = = =  -+------+------+------+------+------+
    COM2 |      |      |      |      |      |          |      |      |      |      |      |
          +------+------+------+------+------+-  = = =  -+------+------+------+------+------+
    COM3 |      |      |      |      |      |          |      |      |      |      |      |
          +------+------+------+------+------+-  = = =  -+------+------+------+------+------+
    COM4 |      |      |      |      |      |          |      |      |      |      |      |
          +------+------+------+------+------+-  = = =  -+------+------+------+------+------+
    COM5 |      |      |      |      |      |          |      |      |      |      |      |
          +------+------+------+------+------+-  = = =  -+------+------+------+------+------+
    COM6 |      |      |      |      |      |          |      |      |      |      |      |
          +------+------+------+------+------+-  = = =  -+------+------+------+------+------+
    COM7 |      |      |      |      |      |          |      |      |      |      |      |
          +------+------+------+------+------+-  = = =  -+------+------+------+------+------+


  < example of the data setting >
     when you display "A" at area from SEG4 to SEG8


             +-----+-----+-----+-----+-----+
```

```
         bit0 |     |     |/////|     |     |   <- Blank area is 0.
              +-----+-----+-----+-----+-----+      Not blank area is 1.
         bit1 |     |/////|     |/////|     |
              +-----+-----+-----+-----+-----+
         bit2 |/////|     |     |     |/////|
              +-----+-----+-----+-----+-----+
         bit3 |/////|     |     |     |/////|
              +-----+-----+-----+-----+-----+
         bit4 |/////|/////|/////|/////|/////|
              +-----+-----+-----+-----+-----+
         bit5 |/////|     |     |     |/////|
              +-----+-----+-----+-----+-----+
         bit6 |/////|     |     |     |/////|
              +-----+-----+-----+-----+-----+
         bit7 |     |     |     |     |     |
              +-----+-----+-----+-----+-----+
                 |     |     |     |     |
                0x7C  0x12  0x11  0x12  0x7C
                 |     |     |     |     |
        SEG4 <---+     |     |     |     |
        SEG5 <---------+     |     |     |
        SEG6 <---------------+     |     |
        SEG7 <---------------------+     |
        SEG8 <---------------------------+
                    Write data to SFR

=====================================================================*/
/*-----------------------*/
/*    Number indication     */
/*-----------------------*/
static const unsigned char aDispNumber[][CLCDSIZE_NUMBER] = {
/* COM 76543210 */
        {0b01111100                /* '0' */
        ,0b10000010
        ,0b10000010
        ,0b01111100
        ,0b00000000}

/* COM 76543210 */
```

```
,       {0b00000100             /* '1' */
        ,0b00000100
        ,0b11111110
        ,0b00000000
        ,0b00000000}

/* COM 76543210 */
,       {0b11000100             /* '2' */
        ,0b10100010
        ,0b10010010
        ,0b10001100
        ,0b00000000}

/* COM 76543210 */
,       {0b01000100             /* '3' */
        ,0b10010010
        ,0b10010010
        ,0b01101100
        ,0b00000000}

/* COM 76543210 */
,       {0b01111000             /* '4' */
        ,0b01000100
        ,0b11111110
        ,0b01000000
        ,0b00000000}

/* COM 76543210 */
,       {0b01001110             /* '5' */
        ,0b10001010
        ,0b10001010
        ,0b01110010
        ,0b00000000}

/* COM 76543210 */
,       {0b01111100             /* '6' */
        ,0b10010010
        ,0b10010010
        ,0b01100000
```

```
            ,0b00000000}

/* COM 76543210 */
,           {0b00000010                   /* '7' */
            ,0b11100010
            ,0b00011010
            ,0b00000110
            ,0b00000000}

/* COM 76543210 */
,           {0b01101100                   /* '8' */
            ,0b10010010
            ,0b10010010
            ,0b01101100
            ,0b00000000}

/* COM 76543210 */
,           {0b00001100                   /* '9' */
            ,0b10010010
            ,0b01010010
            ,0b00111100
            ,0b00000000}
};

/*------------------------*/
/*          sign of Ref           */
/*------------------------*/
static const unsigned char aDispRef[] = {
/* COM 76543210 */
             0b11111110                   /* R */
            ,0b00010010
            ,0b00110010
            ,0b01010010
            ,0b10001100
};


/*------------------------*/
/*          sign of Th            */
```

```
/*-----------------------*/
static const unsigned char aDispTh[] = {
        0b00000010              /* T */
       ,0b00000010
       ,0b11111110
       ,0b00000010
       ,0b00000010
};

/*-----------------------*/
/*      Button frame        */
/*-----------------------*/
static const unsigned char aDispButtonFrame[] = {
/* COM 76543210 */
        0b00000000
       ,0b01111111              /*    */
       ,0b01001001
       ,0b01001001
       ,0b01001001
       ,0b01001001
       ,0b01001001
       ,0b01001001
       ,0b01001001
       ,0b01001001
       ,0b01001001
       ,0b01111111
       ,0b00000000
       ,0b01111111              /*    */
       ,0b01001001
       ,0b01001001
       ,0b01001001
       ,0b01001001
       ,0b01001001
       ,0b01001001
       ,0b01001001
       ,0b01001001
       ,0b01001001
       ,0b01111111
       ,0b00000000
```

```
        ,0b01111111             /*   */
        ,0b01001001
        ,0b01001001
        ,0b01001001
        ,0b01001001
        ,0b01001001
        ,0b01001001
        ,0b01001001
        ,0b01001001
        ,0b01001001
        ,0b01111111
        ,0b00000000
        ,0b01111111             /*   */
        ,0b01001001
        ,0b01001001
        ,0b01001001
        ,0b01001001
        ,0b01001001
        ,0b01001001
        ,0b01001001
        ,0b01001001
        ,0b01001001
        ,0b01111111
        ,0b00000000
        ,0b00000000
};

static const unsigned char aDispClear[] = {
/* COM 76543210 */
         0b00000000             /* ' ' */
        ,0b00000000
        ,0b00000000
        ,0b00000000
        ,0b00000000
        ,0b00000000             /* ' ' */
        ,0b00000000
        ,0b00000000
        ,0b00000000
        ,0b00000000
```

```
,0b00000000          /* ' ' */
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000          /* ' ' */
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000          /* ' ' */
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000          /* ' ' */
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000          /* ' ' */
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000          /* ' ' */
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000          /* ' ' */
,0b00000000
,0b00000000
,0b00000000
,0b00000000
,0b00000000          /* ' ' */
,0b00000000
,0b00000000
,0b00000000
```

```
        ,0b00000000
};

/*----------------------------------------------------------------------*/
/* Global variables
                                                */
/*----------------------------------------------------------------------*/
/*boolean                bDispUpdate;        /* Display update request */
/*boolean                bBlinkOn;           /* Blink status (0:LCD OFF, 1:LCD ON) */

/*----------------------------------------------------------------------*/
/* Local variables
                                                */
/*----------------------------------------------------------------------*/

/*----------------------------------------------------------------------*/
/* Code
                                                              */
/*----------------------------------------------------------------------*/
/*==============================================================================*/
/*                                                                  */
/*                          Common function                          */
/*                                                                  */
/*==============================================================================*/
/*--------------------------------------------------------------*/
/* Module:          fn_DisplayInit
                    */
/* Description:     Initialization of Display module                    */
/*      parameter: --
                         */
/*      return     : --
                         */
/*--------------------------------------------------------------*/
void    fn_DisplayInit(void)
{
        /*----------------------------------------*/
        /* Initialization of LCD controler/driver */
        /*----------------------------------------*/
        LCDMD = 0b00010000;                /* LCD Mode Register */
```

```
                    /*++||++++--- : Be sure to set 000000 */
                    /*   || */
                    /*   ++------- : LCD drive voltage generator selection */
                    /*     : 0 0 : External resistance division method */
                    /*     : 0 1 : Internal voltage boosting method */
                    /*     : 1 0 : Capacitance split method */
                    /*     : 1 1 : Setting prohibited */
    SEGEN = 0b00011111;                 /* Segment Enable Register */
                    /*|||||||+--- : Control segment signal output from pins SEG8-SEG11 */
                    /*||||||+---- : Control segment signal output from pins SEG12-SEG15 */
                    /*|||||+----- : Control segment signal output from pins SEG16-SEG19 */
                    /*||||+------ : Control segment signal output from pins SEG20-SEG23 */
                    /*|||+------- : Control segment signal output from pins SEG24-SEG26 */
                    /*||| : 0 : segment signal output disable */
                    /*||| : 1 : segment signal output enable */
                    /*||| */
                    /*+++-------- : Be sure to set 000 */
    PFALL =  0b01111111;                /* Port Function Register ALL */
                    /*|||||||+--- : Pins P50-P53 port/segment output specification */
                    /*||||||+---- : Pins P54-P57 port/segment output specification */
                    /*|||||+----- : Pins P90-P93 port/segment output specification */
                    /*||||+------ : Pins P94-P97 port/segment output specification */
                    /*|||+------- : Pins P100-P102 port/segment output specification */
                    /*||+-------- : Pins P140-P143 port/segment output specification */
                    /*|+--------- : Pins P144-P147 port/segment output specification */
                    /*| : 0 : Used the pins as port (other than segment output) */
                    /*| : 1 : Used the pins as segment output */
                    /*| */
                    /*+---------- : Be sure to set 0 */
    ISC =      0b00000000;              /* Input Switch Control Register */
                    /*|||||||+--- : Switching external interrupt (INTP0) input */
                    /*||||||| : 0 : Uses the input signal of the INTP0 pin as an external interrupt (normal
operation). */
                    /*||||||| : 1 : Uses the input signal of the RXD3 pin as an external interrupt */
                    /*|||||||        (to measure the pulse widths of the sync break field and sync field). */
                    /*||||||| */
                    /*||||||+---- : ISC1 Switching channel 7 input of timer array unit TAUS */
                    /*|||||| : 0 : Uses the input signal of the TI07 pin as a timer input (normal operation). */
                    /*|||||| : 1 : Input signal of RXD3 pin is used as timer input (wakeup signal detection). */
```

```
                    /*|||||| */
                    /*|||||+----- : RxD3/SEG53/P50 pin schmitt-triggered buffer control */
                    /*||||| : 0 : Disables input */
                    /*||||| : 1 : Enables input */
                    /*||||| */
                    /*||||+------ : TI02/SEG51/P52 pin schmitt-triggered buffer control */
                    /*|||| : 0 : Disables input */
                    /*|||| : 1 : Enables input */
                    /*|||| */
                    /*|||+------- : TI04/SEG50/P53 pin schmitt-triggered buffer control */
                    /*||| : 0 : Disables input */
                    /*||| : 1 : Enables input */
                    /*||| */
                    /*+++-------- : Be sure to set 000 */

fn_DisplayAllClear();          /* clear all LCD String area */

LCDC0 =  0b00110011;                 /* LCD Clock Control Register */
                    /*|||||+++--- : LCD clock (LCDCL) selection */
                    /*||||| : 0 0 0 : fLCD/2^4 */
                    /*||||| : 0 0 1 : fLCD/2^5 */
                    /*||||| : 0 1 0 : fLCD/2^6 */
                    /*||||| : 0 1 1 : fLCD/2^7 */
                    /*||||| : 1 0 0 : fLCD/2^8 */
                    /*||||| : 1 0 1 : fLCD/2^9 */
                    /*||||| : Other than above : Setting prohibited */
                    /*||||| */
                    /*++||+------ : Be sure to set 000 */
                    /*   || */
                    /*   ++------- : LCD source clock (fLCD) selection */
                    /* : 0 0 : fSUB */
                    /* : 0 1 : fCLK/2^6 */
                    /* : 1 0 : fCLK/2^7 */
                    /* : 1 1 : fCLK/2^8 */

VLCD = 0x0A;                         /* set LCD boost level to 5V */
/* wait for the reference voltage setup time (2ms(min.)) */
fn_Wait500usBase(2000/500);
```

```
LCDM =    0b00100111;                    /* LCD Display Mode Register */
                     /*|||||+++--- : LCD controller/driver display mode selection */
                     /*|||||    *When the external resistance division method is used */
                     /*|||||    : 0 0 0 : Four-time-slice mode & 1/3 bias method */
                     /*|||||    : 0 0 1 : Three-time-slice mode & 1/3 bias method */
                     /*|||||    : 0 1 0 : Two-time-slice mode & 1/2 bias method */
                     /*|||||    : 0 1 1 : Three-time-slice mode & 1/2 bias method */
                     /*|||||    : 1 0 0 : Static */
                     /*|||||    : 1 1 1 : Eight-time-slice mode & 1/4 bias method */
                     /*||||| */
                     /*|||||    *When the internal voltage boosting method is used */
                     /*|||||    : 0 0 0 : Four-time-slice mode & 1/3 bias method */
                     /*|||||    : 0 0 1 : Three-time-slice mode & 1/3 bias method */
                     /*|||||    : 0 1 0 : Four-time-slice mode & 1/3 bias method */
                     /*|||||    : 0 1 1 : Four-time-slice mode & 1/3 bias method */
                     /*|||||    : 1 0 0 : Setting prohibited */
                     /*|||||    : 1 1 1 : Eight-time-slice mode & 1/4 bias method */
                     /*||||| */
                     /*|||||    *When the capacitor split method is used */
                     /*|||||    : 0 0 0 : Four-time-slice mode & 1/3 bias method */
                     /*|||||    : 0 0 1 : Three-time-slice mode & 1/3 bias method */
                     /*|||||    : 0 1 0 : Four-time-slice mode & 1/3 bias method */
                     /*|||||    : 0 1 1 : Four-time-slice mode & 1/3 bias method */
                     /*|||||    : 1 0 0 : Setting prohibited */
                     /*|||||    : 1 1 1 : Four-time-slice mode & 1/3 bias method */
                     /*||||| */
                     /*|||||    : Other than above : Setting prohibited
                     /*||||| */
                     /*|||++------ : LCD display data area control */
                     /*||| : 0 0 : Display the data of an A pattern area */
                     /*|||           (lower 4 bits of LCD display data memory) */
                     /*||| : 0 1 : Display the data of an A pattern area */
                     /*|||           (higher 4 bits of LCD display data memory) */
                     /*||| : 1 0 : Display the data of an A pattern area and the B pattern area in turn. */
                     /*|||           (The on and off light indication which synchronized */
                     /*|||            in a constant-period interrupt timing of RTC) */
                     /*||| : 1 1 : Display the data of an A pattern area and the B pattern area in turn. */
                     /*|||           (The on and off light indication which synchronized */
                     /*|||            in a constant-period interrupt timing of RTC) */
```

```
                              /*||| */
                              /*||+-------- : Voltage boost circuit and capacitor split circuit operation enable/disable */
                              /*|| : 0 : Stops voltage boost circuit and capacitor split circuit operation */
                              /*|| : 1 : Enables voltage boost circuit and capacitor split circuit operation */
                              /*|| */
                              /*++--------- : LCD display enable/disable */
                              /* : 0 0 : Output ground level to segment/common pin */
                              /* : 0 1 : Display off (all segment outputs are deselected.) */
                              /* : 1 0 : Output ground level to segment/common pin */
                              /* : 1 1 : Display on */

        /* software to wait for the operation stabilization time (over 500ms) */
        fn_Wait500usBase(500000/500);

        SCOC = 1;                       /* output deselect level to SEG and LCD waveform to COM */
        LCDON = 1;                      /* display on */

        /*----------------------------------------*/
        /*      Initialization of variables       */
        /*----------------------------------------*/
/*      bDispUpdate = 1;        /* initialize display update request flag */
/*      bBlinkOn = 0;                   /* initialize blink on flag */
/*      ushOpeningMessage = 0;  /* initialize opening message counter */
}

/*------------------------------------------------------------*/
/* Module:         fn_LcdWrite
                              */
/* Description:    Write data to LCD RAM
        */
/*      parameter: position in whitch display begins                      */
/*                         address of display data
        */
/*                         size of display data
                */
/*      return     : --
                              */
/*------------------------------------------------------------*/
static void fn_LcdWrite(unsigned char *Position, unsigned char *DataAddr, unsigned char DataSize)
```

```
{
        memcpy(Position, DataAddr, DataSize);
}

/*------------------------------------------------------------*/
/* Module:        fn_DisplayAllClear
        */
/* Description:   LCD RAM all clear
        */
/*      parameter: --
                          */
/*      return     : --
                          */
/*------------------------------------------------------------*/
void fn_DisplayAllClear(void)
{
        fn_LcdWrite(CLCDPOS_START, aDispClear, CLCDSIZE_ALL);
}



/*------------------------------------------------------------*/
/* Module:        fn_Display
               */
/* Description:   Display count to LCD                                    */
/*      parameter: REF data(Hex), TH data(Hex)                            */
/*      return     : --
                          */
/*------------------------------------------------------------*/
void fn_Display(unsigned short ushRef, unsigned short ushTh)
{
        unsigned short      ushNumber;              /* number work for display */
        /* space */
/*      fn_LcdWrite(CLCDPOS_START, aDispClear, (unsigned char)(CLCDPOS_REF - CLCDPOS_START));
*/
        fn_LcdWrite(CLCDPOS_START,     aDispClear,     (unsigned     char)(CLCDPOS_REF1000     -
CLCDPOS_START));

        /************************/
        /*    display REF data    */
```

```
        /************************/
        /* display sign of REF */
/*      fn_LcdWrite(CLCDPOS_REF, aDispRef, sizeof aDispRef);
*/

        /* space */
/*      fn_LcdWrite((CLCDPOS_REF100 - 2), aDispClear, 2);
*/

        /* display thousands place of REF */
        fn_LcdWrite(CLCDPOS_REF1000,
                    aDispNumber[(unsigned char)(ushRef / 1000)],
                    CLCDSIZE_NUMBER);

        /* clear digit gap */
/*      fn_LcdWrite((CLCDPOS_REF100 - 1), aDispClear, 1);
*/
        /* display hundreds place of REF */
        ushRef %= 1000;
        fn_LcdWrite(CLCDPOS_REF100,
                    aDispNumber[(unsigned char)(ushRef / 100)],
                    CLCDSIZE_NUMBER);

        /* clear digit gap */
/*      fn_LcdWrite((CLCDPOS_REF10 - 1), aDispClear, 1);
*/

        /* display tenths place of REF */
        ushRef %= 100;
        fn_LcdWrite(CLCDPOS_REF10,
                    aDispNumber[(unsigned char)(ushRef / 10)],
                    CLCDSIZE_NUMBER);

        /* clear digit gap */
/*      fn_LcdWrite((CLCDPOS_REF1 - 1), aDispClear, 1);
*/

        /* display ones place of REF */
        fn_LcdWrite(CLCDPOS_REF1,
```

```
                    aDispNumber[(unsigned char)(ushRef % 10)],
                    CLCDSIZE_NUMBER);

        /* space */
/*      fn_LcdWrite((CLCDPOS_REF1 + CLCDSIZE_NUMBER), aDispClear, 6);*/
        fn_LcdWrite((CLCDPOS_REF1 + CLCDSIZE_NUMBER), aDispClear, 5);

        /************************/
        /*   display TH data    */
        /************************/
        /* display sign of TH */
/*      fn_LcdWrite(CLCDPOS_TH, aDispTh, sizeof aDispTh);
*/

        /* space */
/*      fn_LcdWrite((CLCDPOS_TH100 - 2), aDispClear, 2);
*/

        /* display thousands place of TH */
        fn_LcdWrite(CLCDPOS_TH1000,
                    aDispNumber[(unsigned char)(ushTh / 1000)],
                    CLCDSIZE_NUMBER);

        /* clear digit gap */
/*      fn_LcdWrite((CLCDPOS_TH100 - 1), aDispClear, 1);
*/

        /* display hundreds place of TH */
        ushTh %= 1000;
        fn_LcdWrite(CLCDPOS_TH100,
                    aDispNumber[(unsigned char)(ushTh / 100)],
                    CLCDSIZE_NUMBER);

        /* clear digit gap */
/*      fn_LcdWrite((CLCDPOS_TH10 - 1), aDispClear, 1);
*/
        /* display tenths place of TH */
        ushTh %= 100;
        fn_LcdWrite(CLCDPOS_TH10,
```

```
                        aDispNumber[(unsigned char)(ushTh / 10)],
                        CLCDSIZE_NUMBER);

        /* clear digit gap */
/*      fn_LcdWrite((CLCDPOS_TH1 - 1), aDispClear, 1);
*/

        /* display ones place of TH */
        fn_LcdWrite(CLCDPOS_TH1,
                        aDispNumber[(unsigned char)(ushTh % 10)],
                        CLCDSIZE_NUMBER);

        /* space */
/*      fn_LcdWrite((CLCDPOS_TH1 + CLCDSIZE_NUMBER), aDispClear, 1);
*/
}
```

APPENDIX B REVISION HISTORY

Edition	Date Published	Page	Revision
1st edition	March 2010	–	–

Published by: NEC Electronics Corporation (http://www.necel.com/)
Contact: http://www.necel.com/support/