Ballistic Chronograph
SLG46620V, SLG47910V

Abstract

This application note describes the design procedure to create a ballistic chronograph that can measure speeds ranging from 1 m/s to 1500 m/s with an accuracy of + - 1 m/s using the GreenPAK and ForgeFPGA. SLG46620V is used to track the moment of the flight of a bullet or other object through the optical sensors. SLG47910V measures the time between the moment of the intersection of the first sensor and the moment of the intersection of the second sensor, calculates the speed of the object, and displays the result on a seven-segment indicator. SLG46582V is used as LDO for power SLG47910V and SLG46620V.

This application note comes complete with design files which can be found in the References section.

Contents

1. Terms and Definitions

FPGA	Field Programmable Gate Array
FPGA Editor	Main FPGA design and simulation window
Go Configure Software Hub	Main window for device selection
GPIO	General Purpose Input/Output
IC	Integrated Circuit
OSC	Oscillator
ACMP	Analog comparator
LDO	Low-DropOut regulator
POR	Power On Reset
BCD	Binary Code Decimal
FF	Flip Flop
CLB	Configurable Logic Blocks
LUT	Look Up Table
FSM	Finite States Machine

2. References

For related documents and software, please visit:

https://www.dialog-semiconductor.com/products/greenpak/low-power-low-cost-forgefpga

Download our free Go Configure Software Hub software [1] to open the .ffpga and .gp files. The application note [2] contains the proposed circuit design. Use the GreenPAK development tools [3] to freeze the design into your own customized IC in a matter of minutes. Find out more in the complete library of application notes [4] featuring design examples as well as explanations of features and blocks within the GreenPAK IC.

[1] Go Configure Software Hub, Software Download and User Guide

[2] Ballistic Chronograph.gp, ForgeFPGA and GreenPAK Designs File

[3] GreenPAK Development Tools, GreenPAK Development Tools Webpage

[4] GreenPAK Application Notes, GreenPAK Application Notes Webpage

[5] SLG47910V, Datasheet

[6] SLG46620V, Datasheet

[7] SLG46582V, Datasheet

3. Introduction

A ballistic chronograph is a device for measuring the initial velocity of a small-arms bullet. Knowing the initial velocity of the bullet allows you to calculate its energy and the vertical correction when shooting at a distance.

This application note describes how to use GreenPAK (SLG46620V, SLG46582V) and ForgeFPGA (SLG47910V) to create a ballistic chronograph that can measure speeds ranging from 1 m/s to 1500 m/s with an accuracy of +/- 1 m/s.

The principle of operation of the chronograph is as follows, there are two optical sensors located at a certain distance, when the opaque object crosses the first sensor, the counter which counts clock pulses is started.

When the object crosses the second sensor the time measurement ends. Knowing the distance between the sensors (constant) and the measured time, you can calculate the speed.

The optical sensor consists of infrared LEDs, infrared phototransistors, and analog comparators inside SLG46620V, which are triggered when the light intensity changes. The outputs of the two optical sensors are connected to the SLG47910V (ForgeFPGA). The FPGA measures the time between the rising edges of the signals from the optical sensors, calculates the speed, and displays the result on a seven-segment indicator.

The SLG46582V is used as LDO, one to power the SLG47910V and one for each sensor based on the SLG46620V.

The Verilog code and design for GreenPAK (project link) for the ballistic chronograph are available on the ForgeFPGA webpage.

4. Ingredients

- ForgeFPGA SLG47910V IC
- GreenPAK SLG46620V IC
- GreenPAK SLG46582V IC
- Seven-segment display module with common cathode
- IR 5mm 940nm emitting diodes
- IR 5mm 940nm phototransistor
- ForgeFPGA Development- Board with a USB cable and power supply
- GreenPAK Development Board
- Latest Revision of Go Configure Software Hub software

5. Design Overview

The ballistic chronograph consists of two optical sensors and the main board as shown in Figure 1. The optical sensor consists of two SLG46620V chips, one SLG46582V chip, and infrared diodes and infrared phototransistors, which are opposite each other. Optical sensors are located at a distance of 30 cm from each other. The main board consists of one SLG47910V chip, one SLG46582V, and a seven-segment indicator. SLG46582V chips are used as LDO to power SLG46620V and SLG47910V.

The wiring diagram of the phototransistors is shown in Figure 2. -In1 is connected to the negative input of the ACMP SLG46620V, and + In1 to the positive input of the ACMP. When phototransistors are illuminated by infrared radiation, their resistance decreases. When the object crosses the sensor, the intensity of light incident on the phototransistor decreases, respectively, its resistance increases, and a pulse is formed on + In1, from which the comparator is triggered. To increase the area operation of the sensor 12 phototransistors were used which are combined in pairs of 2 pcs. After the operation of one of the 6 comparators, the sensor sets a high level at the output Trigger. The main board measures the time between the operation of the first and second sensors, calculates the speed, and displays the result on a seven-segment indicator. When measuring the speed in the range from 1 m/s to 1000 m/s, the result is displayed with a step of 0.1 m/s, in the range from 1000 m/s to 1500 m/s, the result is displayed with a step of 1 m/s. The measurement accuracy is + -1 m/s. After completing the previous measurement, the chronograph waits for 2s and is ready for the next one. At rest or 60 seconds after the last measurement, 4 dashes will light up on the indicator.

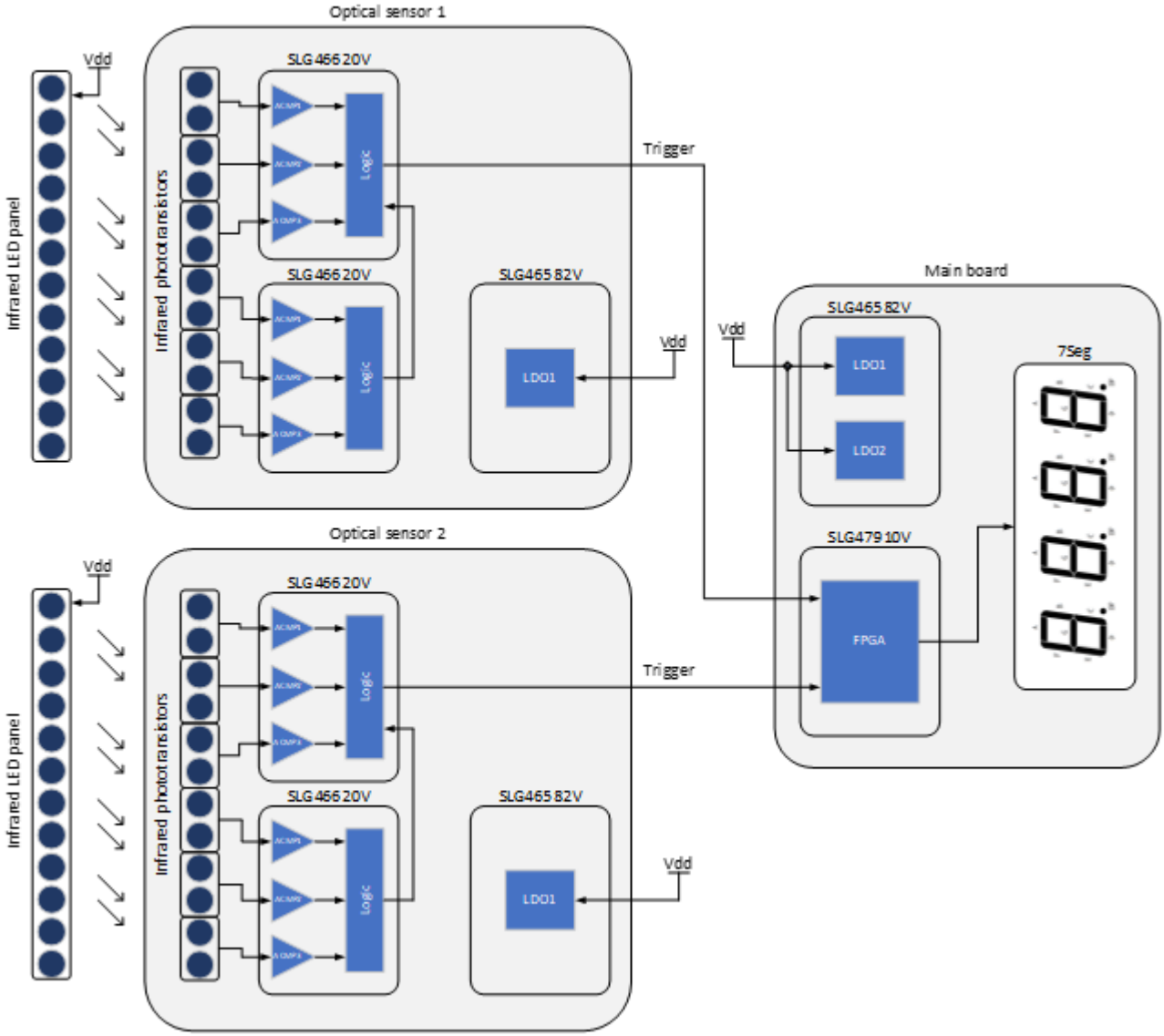

Figure 1: Functional Diagram

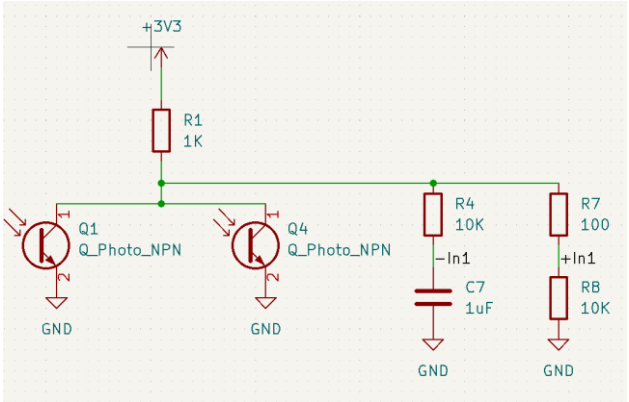

Figure 2: The Phototransistors Connection Diagram

6. Optical Sensor Design

The two optical sensors have the same design. The full schematic optical sensor was designed using the KiCad and is shown in Figure 3.

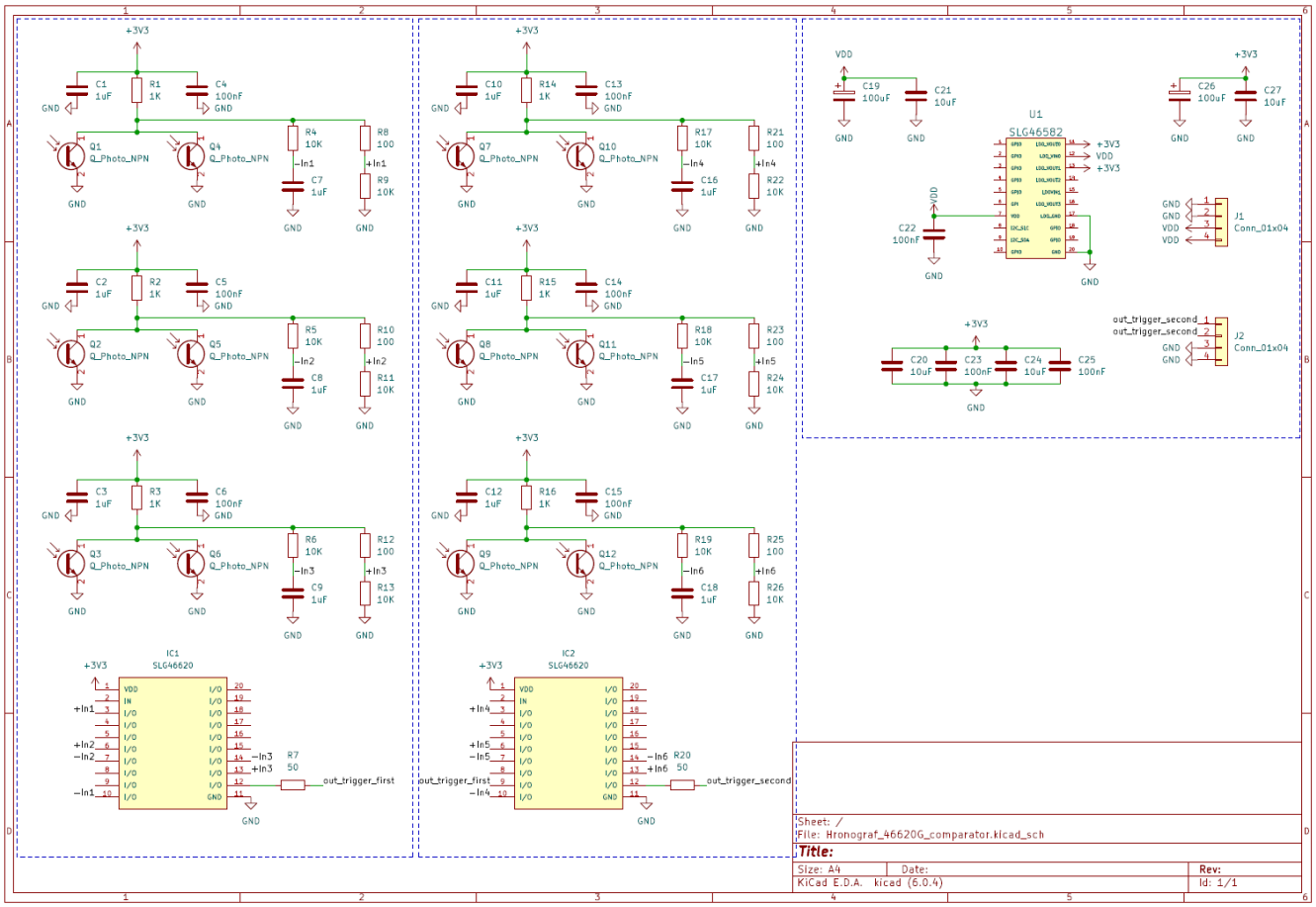

Figure 3: Optical Sensor Schematic

The scheme can be divided into 3 main parts. The first two parts consist of 6 infrared phototransistors divided into 3 groups of 2 pieces, and SLG46620V. They are responsible for detecting the moment flight of the object through the sensor. The Trigger output (pin12) of the first SLG46620V is connected to the Trigger_in (pin9) input of the second SLG46620V, and the output of the second (pin12) via a 50-Ohm line is connected to the main board. The third part consists of SLG46582 and is responsible for powering the sensor. The 3D model of the PCB sensor design is shown in Figure 4 (In this design the SLG46620G is used for convenience, in your design you can use SLG46620V in a smaller case). Physically, the optical sensor looks as shown in Figure 5.

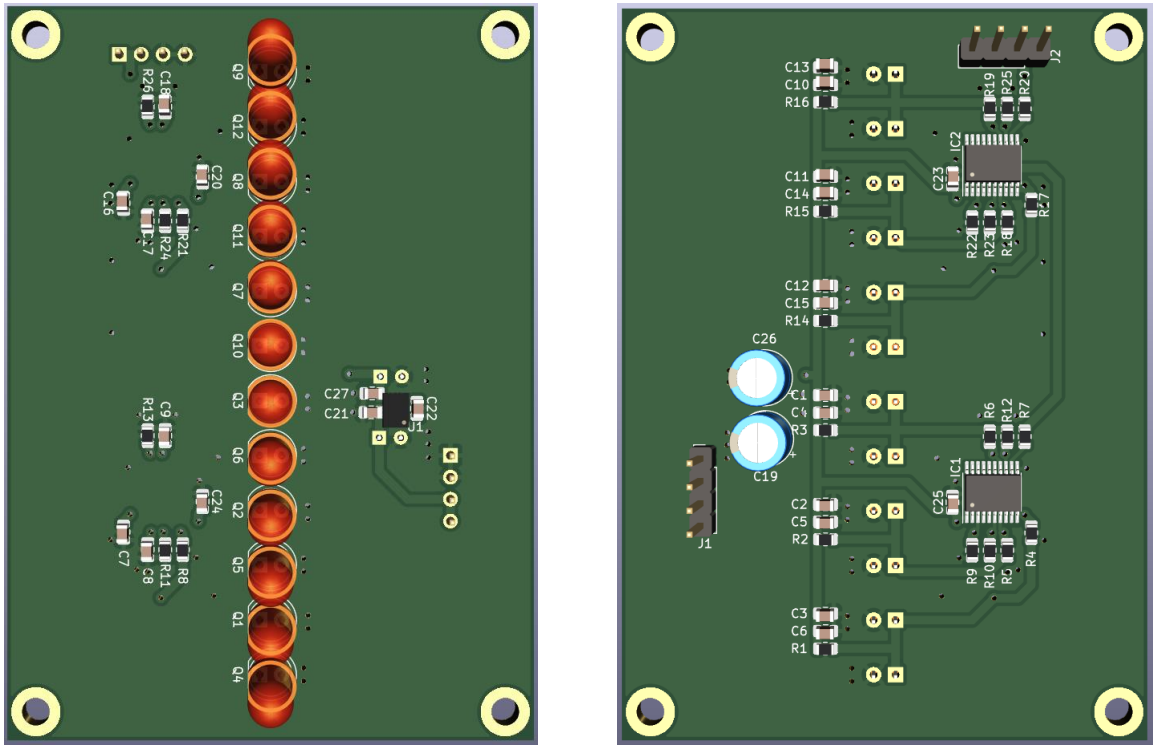

Figure 4: Optical Sensor 3D Model, Top and Bottom View

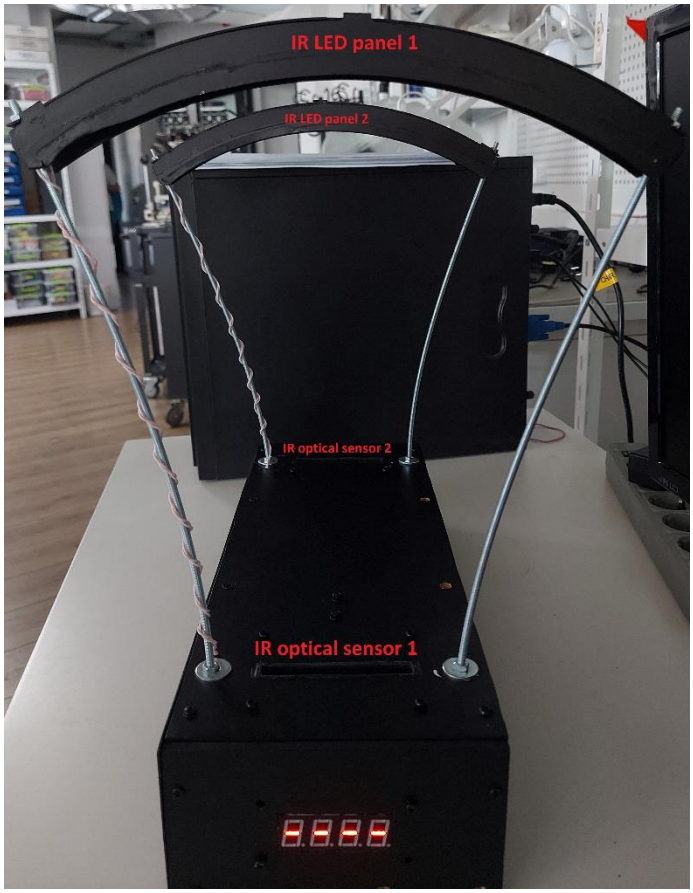

Figure 5: Optical Sensor Physical View

The oscillogram below shows the flight of a bullet with a diameter of 6 mm at a speed of 60 m/s through an optical sensor.

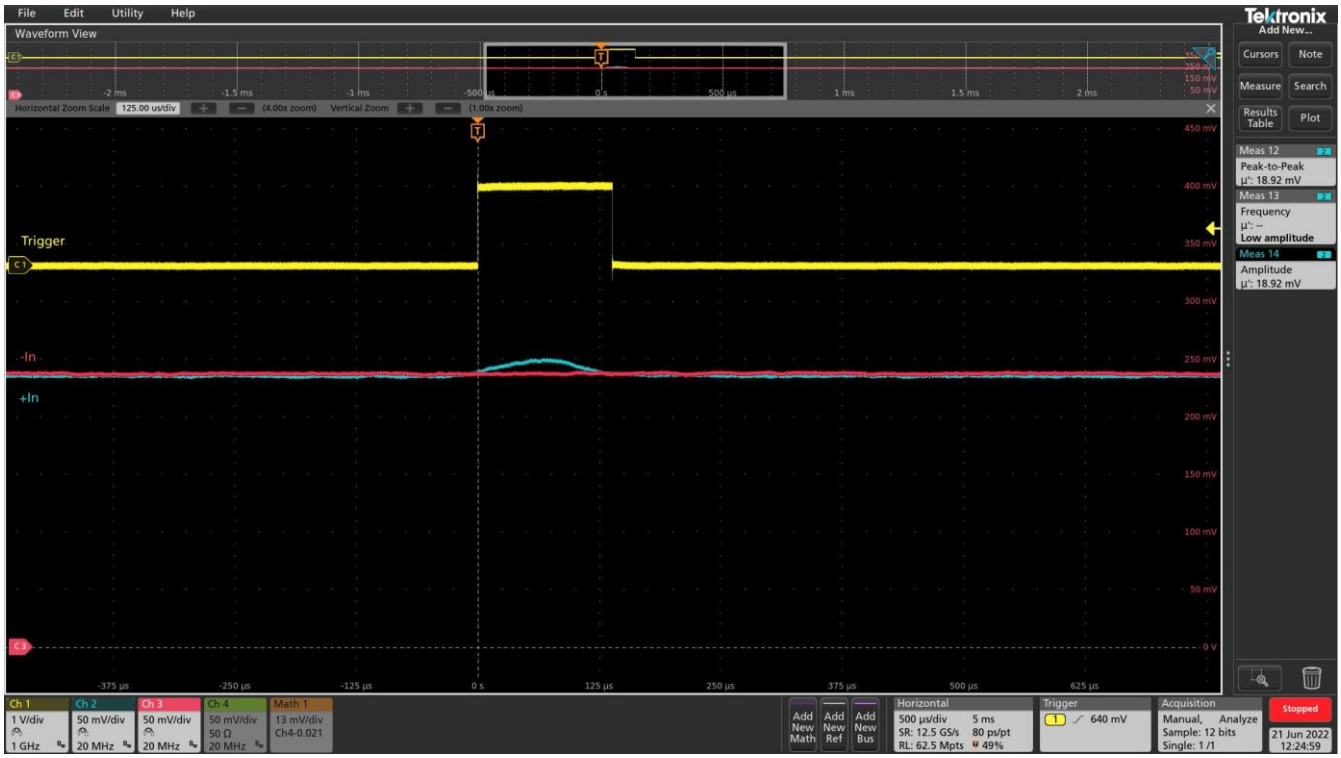

Figure 6: Oscillogram of Optical Sensor Operation, Channel -In and +In DC Coupling

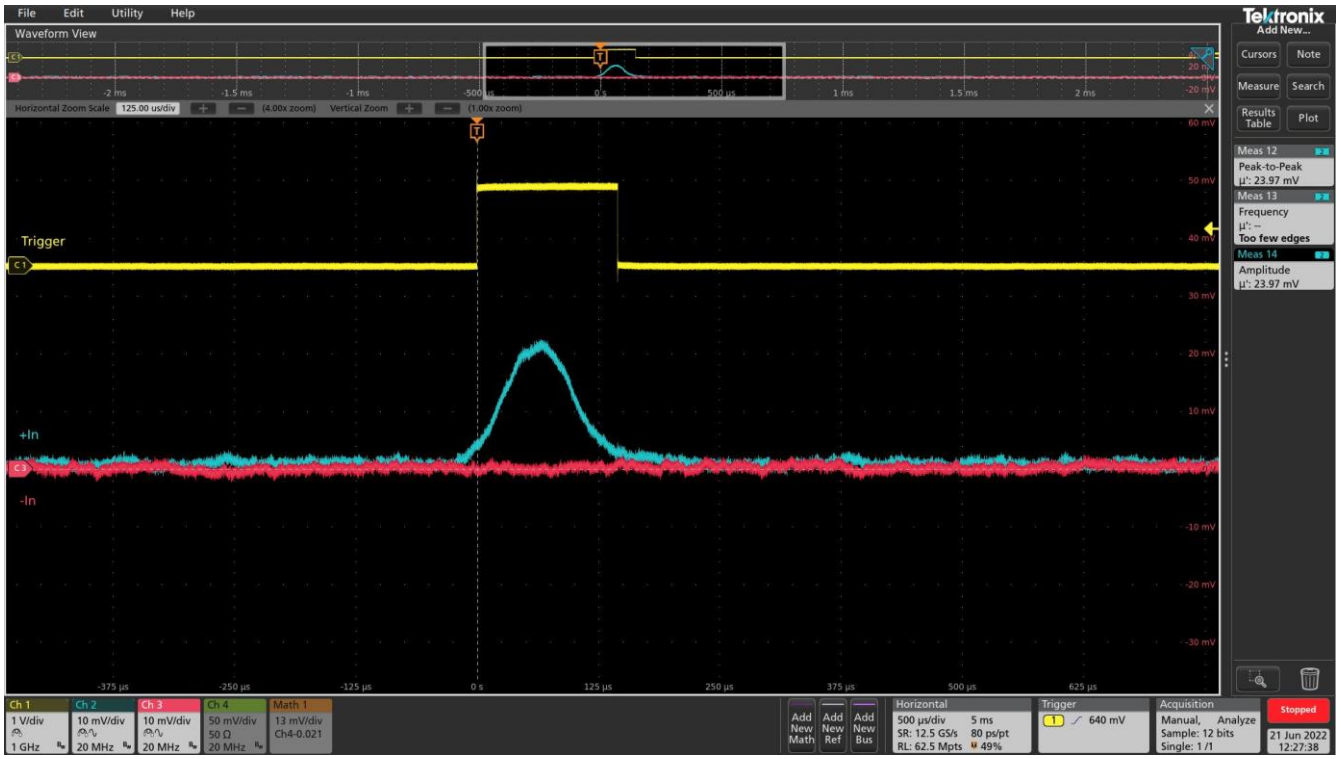

Figure 7: Oscillogram of Optical Sensor Operation, Channel -In and +In AC Coupling

6.1 SLG46582V Design

Since the SLG46582V is only used as an LDO to power the sensor, accordingly, its configuration is quite simple. It is only necessary to configure LDO0 as shown in Figure 8.

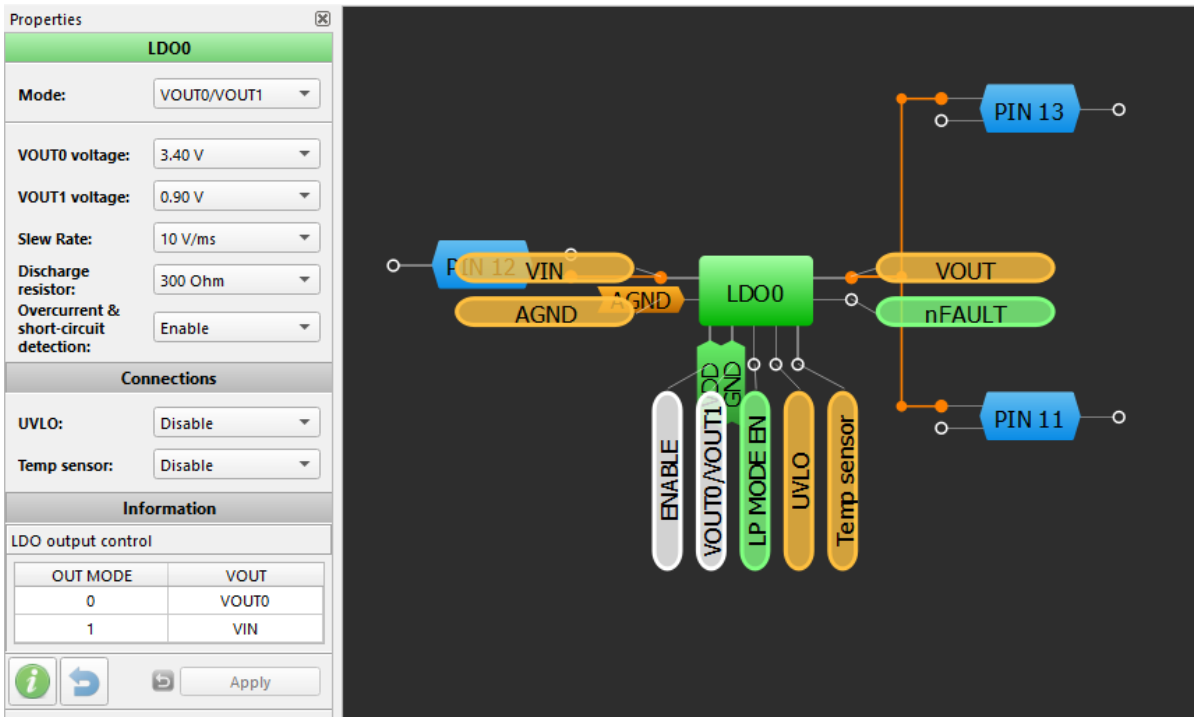

Figure 8: LDO0 Configuration for SLG46582V

6.2 SLG46620V Design

The configuration for SLG46620V is shown in Figure 9.

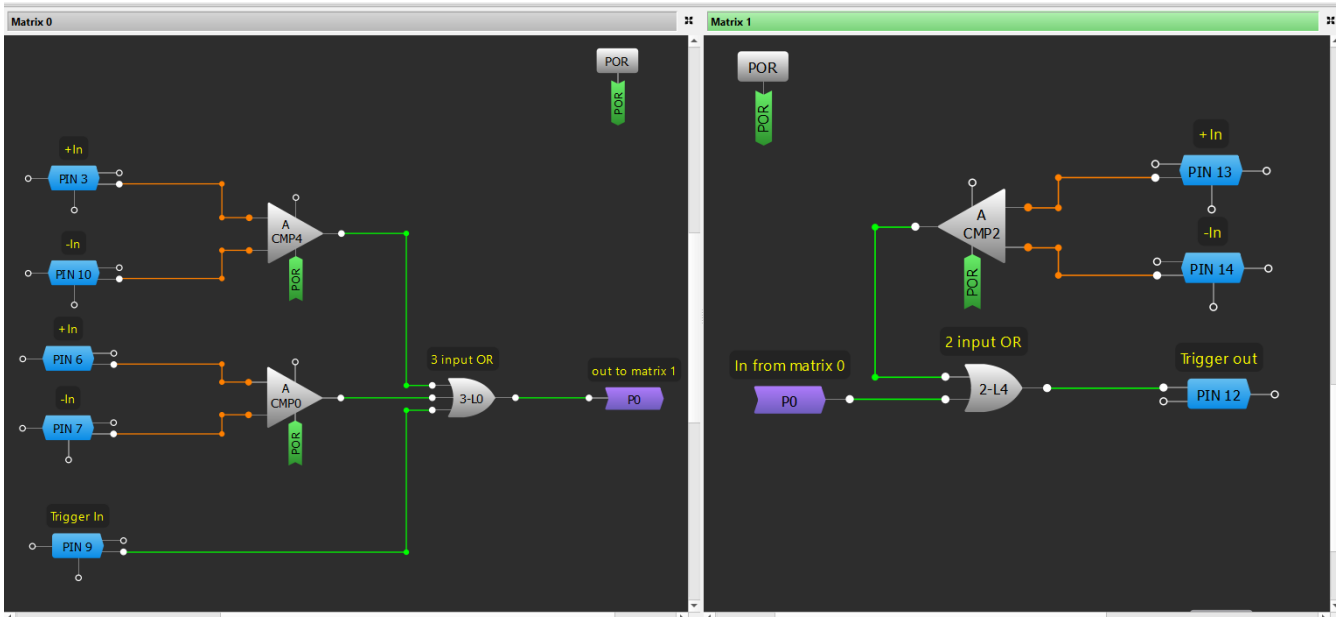

Figure 9: Configuration SLG46620V

The logic of the SLG46620V works as follows, logic '1' appears at the output Trigger out (pin12) only when one or more comparators are triggered, or when a logic '1' comes to the input Trigger in (pin9) from another SLG46620V.

The configuration of the input pins for all comparators is the same and is shown in Figure 10.

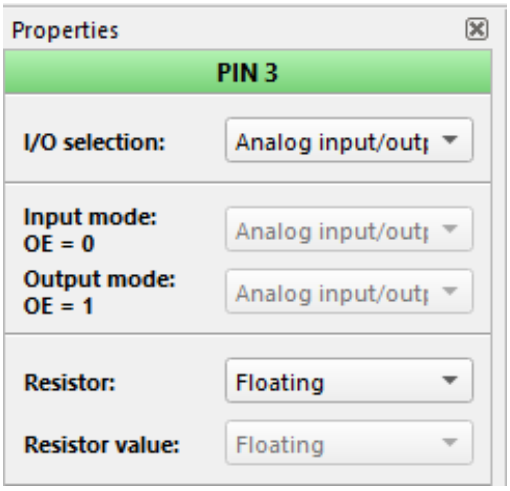

Figure 10: Configuration of the Comparator Input Pins

The configuration of the comparators is shown in Figure 11.

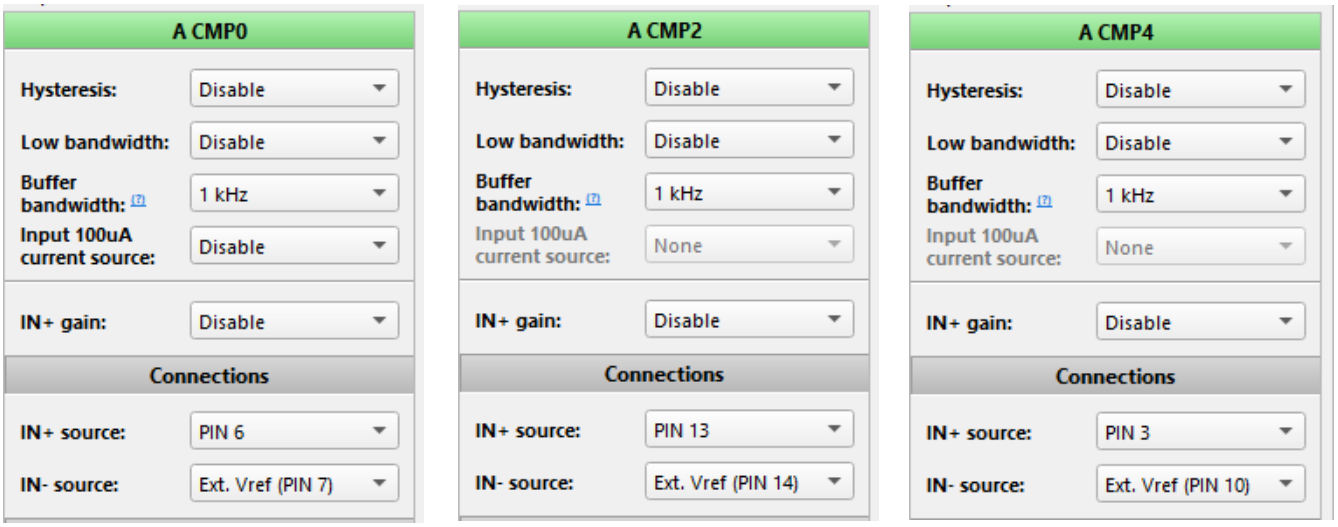

Figure 11: ACMP Configuration

The configurations of the input Trigger in (pin9) and output Trigger out (pin12) are shown in Figure 12.

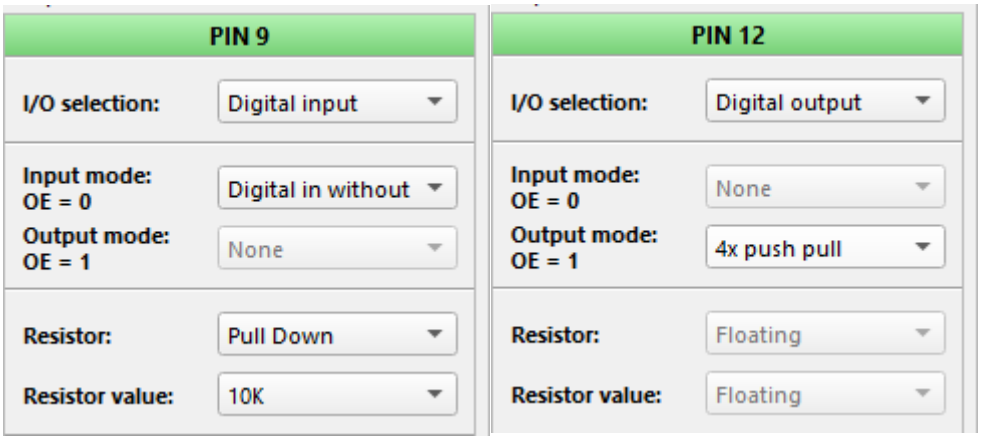

Figure 12: Trigger in and Trigger out Pin Configuration

The Trigger in input has a 10K Ohm pull-down resistor. It is necessary so that in case this pin is not connected anywhere, the sensor does not false positives. This allows you to use the same design for both SLG46620V chips. The Trigger out output pin is configured as a 4x push-pull for driving a 50-Ohm line.

The signal from the POR block (Power On Reset) is connected to the Enable input of all ACMP. After the power supply after 4 microseconds, logic '1' will be set at the output of the POR block, and all ACMPs will be activated.

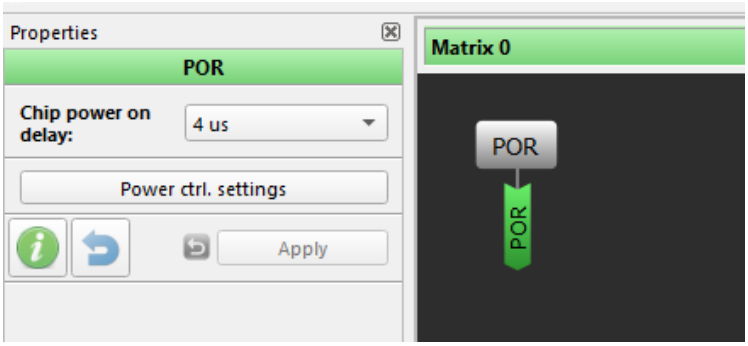

Figure 13: POR Block Configuration

7. Speed Measure Main Board Design

The scheme of the main board is shown in Figure 14.

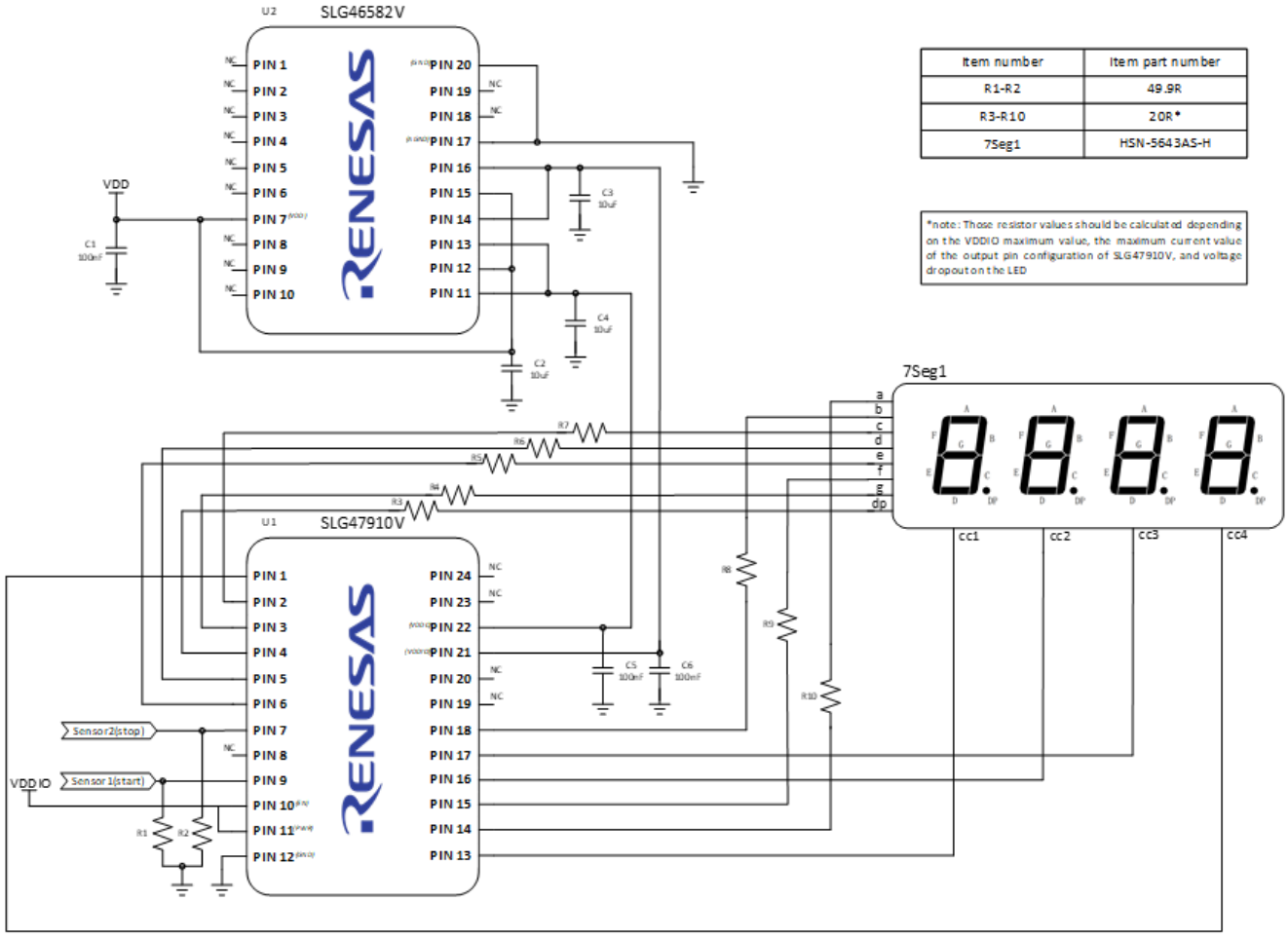

Figure 14: Main Board Schematic

The circuit consists of SLG46582V, SLG47910V (ForgeFPGA), and a seven-segment indicator. SLG46582V is used as LDO to power SLG47910V, Vddc = 1.2V and Vddio = 2.2V. The FPGA measures the time between triggering sensors, calculates the speed, and outputs the result to a seven-segment indicator. The sensors are connected to the FPGA via a 50-Ohm line to reduce signal transmission distortion. The measurement result is displayed with a step of 0.1 m/s when measuring speeds up to 1000 m/s and with a step of 1 m/s at speeds above 1000 m/s. After the measurement, the FPGA waits for 2s and is ready for the next one. The measurement result are displayed for about 60 seconds, after which 4 dashes will be displayed. If the time between triggering sensors is less than the time equivalent to a speed of 1500 m/s, four 0 will be displayed. If after receiving the start signal after 330 ms (equivalent to a speed less than 1 m/s) no stop signal is received, 4 dashes will be displayed and the chronograph will be ready for the next measurement. The 3D model of the PCB design of the main board is shown in Figure 15.

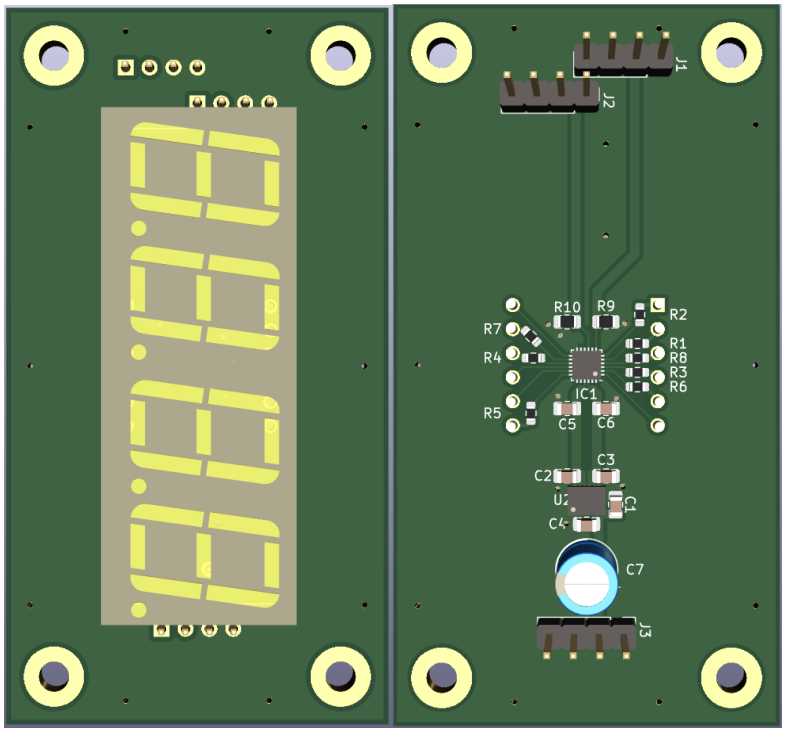

Figure 15: Main board 3D Model, Top and Bottom View

The oscillograms below show the flight of a bullet with a diameter of 6 mm at speeds of 59.5 m/s and 5.4 m/s through optical sensors, as well as the results of velocity measurements.

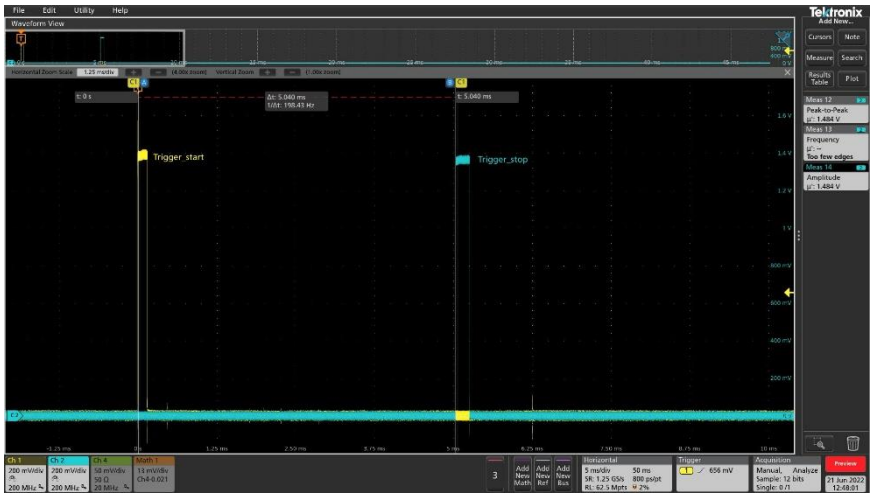

Figure 16: Flight of the Bullet Through the Sensors at a Speed of 59.5 m/s

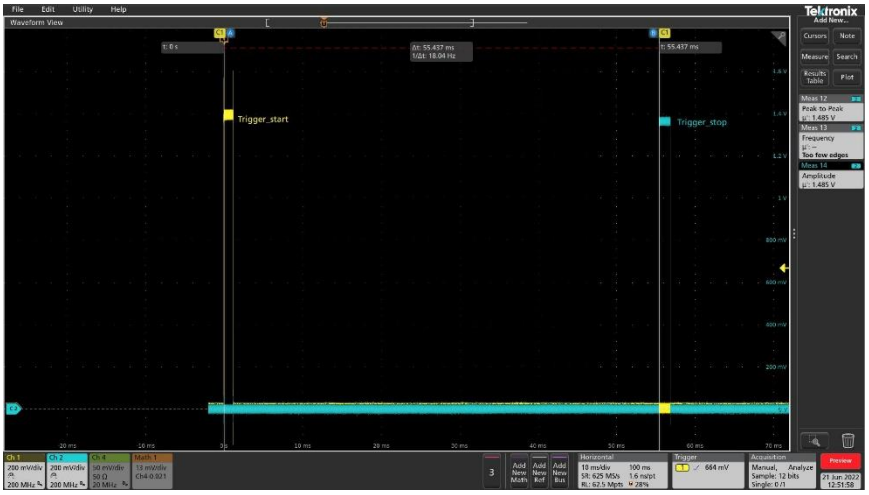

Figure 17: Flight of the Bullet Through the Sensors at a Speed of 5.4 m/s

As can be seen in the first oscillogram Figure 16, the time between the operation of the two sensors is 5.04 ms, ie the speed is 0.3 m (distance between the sensors) /0.00504s = 59.52 m/s.

The time between the operation of the sensors on the second oscillogram is 55.437 ms, ie the speed is 0.3m / 0.055437s = 5.41 m/s.

Figure 18 shows the results of speed measurement for these two cases.

Figure 18: Speed Measurement Results

7.1 SLG46582V Design

In the design for the SLG46582V, you only need to configure two LDOs to power the SLG47910V, Vddc (LDO0) = 1.2V and Vddio (LDO1) = 2.2V. The LDO configuration is shown in Figure 19.

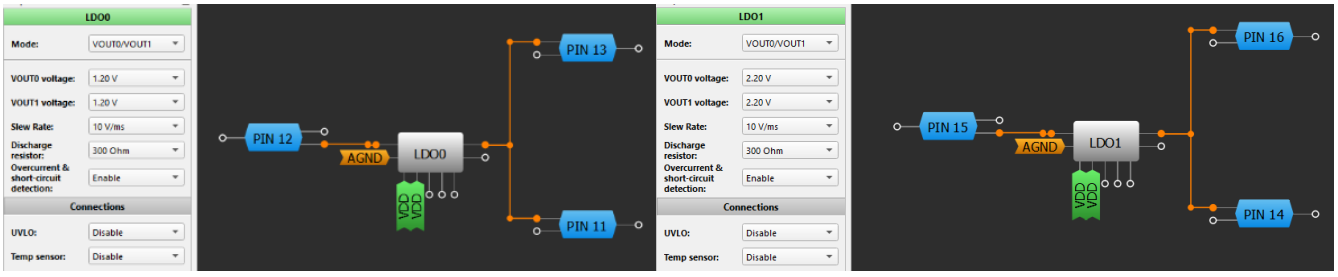

Figure 19: LDO0 and LDO1 Configuration

7.2 SLG47910V Design

The project for SLG47910V consists of 3 modules.

1. TOP – the main module measures the time between the operation of the two sensors, calculates the speed, and sets the control signals for the other modules. The input/output signals are described below.
    - input clk - synchronization input
    - input nreset - inverse reset signal
    - input start - signal from the first optical sensor
    - output start_oe - a signal to configure the start pin as an input
    - input stop - signal from the second optical sensor
    - output stop_oe - a signal to configure the stop pin as input
    - output osc_en - enable signal for the oscillator
    - output osc_mode - a signal to select the mode of operation of the oscillator
    - output reg [13: 0] data_binary - measured speed in binary code
    - output reg enable_BCD_conver - enable for a start converting binary code to binary code decimal
    - output reg lcd_reset - reset signal of the seven-segment indicator.
2. BCD – module for converting 14-bit binary code into 5-bit binary code decimal.
    - input clk – synchronization input
    - input [13:0] binary_in – input data in binary code
    - input enable_conver – enable to start the conversion
    - output reg [3:0] ones - ones in binary code decimal
    - output reg [3:0] tens - tens in BCD
    - output reg [3:0] hundrets - hundrets in BCD
    - output reg [3:0] thousands - thousands in BCD
    - output reg tens_of_thousands - tens_of_thousands in BCD (the maximum value for a 14-bit binary code is 16383, so this bit can only take a value of 0 or 1).
3. Four_digits_7_segment – module for displaying measurement results on a four-bit seven-segment display with a common cathode and dynamic indication. If the input tens_of_thousands is 0, 4 lower digits are displayed (ie the result is displayed with a resolution of 0.1 m / s), otherwise 4 higher digits (resolution is 1 m / s). If the reset signal is 1, 4 dashes will be displayed.
    - input clk - synchronization input
    - input reset – reset signal
    - input tens_of_thousands - tens_of_thousands in BCD
    - input [3:0] thousands - thousands in BCD
    - input [3:0] hundrets - hundrets in BCD
    - input [3:0] tens - tens in BCD
    - input [3:0] ones - ones in BCD
    - output reg [7:0] anods - anodes of the seven-segment indicator
    - output reg [3:0] catods - catods of the seven-segment indicator

The connection diagram of the modules is shown in Figure 20.

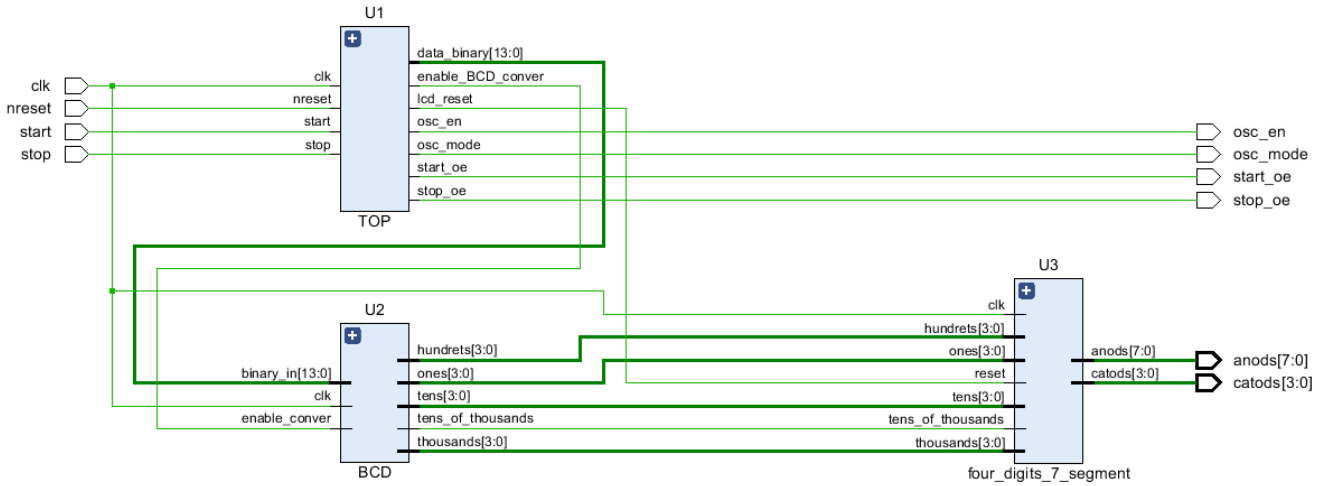

Figure 20: The Connection Diagram of the Modules

Table 1. Correspondence Between Input / Output Signals and Pins

Signal	Pin	Description
clk	OSC_CLK	Frequency 50 MHz from OSC (internal)
nreset	POR	Power on Reset (internal), is set to ‘1’ after power-on delay
start	GPIO16_IN (pin7)	Input signal from first optical sensor
stop	GPIO18_IN (pin9)	Input signal from second optical sensor
osc_en	OSC_CTRL_EN	Enable OSC (internal), always ‘1’
osc_mode	OSC_CTRL_MODE	OSC mode, 1 – 50MHz, 0 – 3.125 MHz (internal), always ‘1’
start_oe	GPIO16_OE (pin7)	configures pin start as input (internal), always ‘1’
stop_oe	GPIO18_OE (pin9)	configures pin stop as input (internal), always ‘1’
anods[7]	GPIO1_OUT (pin14)	Anod a
anods[6]	GPIO2_OUT (pin15)	Anod f
anods[5]	GPIO5_OUT (pin18)	Anod b
anods[4]	GPIO11_OUT (pin2)	Anod g
anods[3]	GPIO12_OUT (pin3)	Anod c
anods[2]	GPIO14_OUT (pin5)	Anod d
anods[1]	GPIO15_OUT (pin6)	Anod e
anods[0]	GPIO13_OUT (pin4)	Anod dpx
catods[3]	GPIO0_OUT (pin13)	4th digit (most significant digit)
catods[2]	GPIO3_OUT (pin16)	3rd digit
catods[1]	GPIO4_OUT (pin17)	2nd digits
catods[0]	GPIO10_OUT (pin1)	1st digit (least significant digit)

Consider in more detail the operation of the TOP module. At the heart of this work is a finite states machine with 7 states. The algorithm of operation of the FSM is shown in Figure 21.

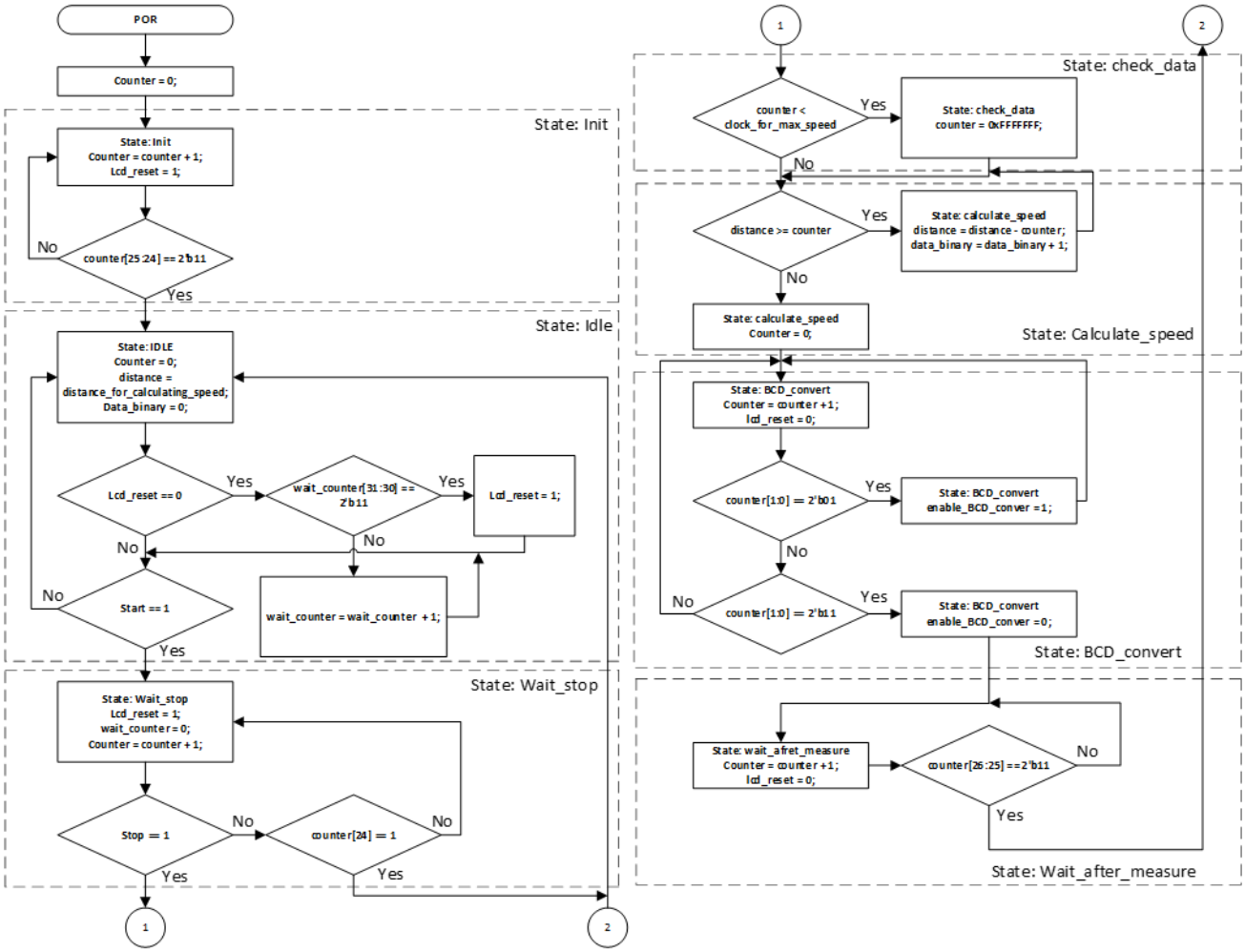

Figure 21: Algorithm of Operation of the FSM

- Init – to this state, the state machine passes after the appearance of a high level at the input nreset. In this state, the counter counts to 50,331,648 (~ 1s) and then goes to the Idle state. This delay is required to initialize the optical sensors after power on.
- Idle – in this state, the FPGA resets the registers to the initial state and waits for the start signal (signal from the first sensor). When the start signal is received, the state machine goes to the wait_stop state. If in this state the signal lcd_reset == 0 (it is set to 0 after the speed measurement is completed) then the counter will start which will count to 3, 221,225,472 (~ 64s) and then set the signal lcd_reset to '1'. That is, after the measurement is completed, the result will be displayed for approximately 64 seconds, after which 4 dashes will be displayed.
- Wait_stop – in this state, the FSM waits for the stop signal and at the same time counts the number of clock pulses that came from the moment of receipt of the start signal. After receiving the stop signal, the FSM goes to the state checked_data. If after 16,777,216 clocks (which at a frequency of 50MHz and a distance between optical sensors of 30 cm is equivalent to a speed less than 1 m/s) the stop signal is not received, FSM goes to Idle state, and the system is ready for the next measurement.
- Check_data – in this state, the FPGA checks whether the time between the operation of the two sensors is greater than the time equivalent to the maximum speed. If the time is less than the counter is set to

0xFFFFFFF. The clock_for_max_speed parameter is equal to the number of clocks that are equivalent to the maximum speed that a chronograph can measure. Its value depends on the frequency of the oscillator and the distance between the sensors. After checking there is a transition to the calculated_speed state.

- Calculate_speed – in this state, the distance between the sensors is divided by time. Because division requires a lot of resources, it has been replaced by subtraction. The measured time is successively subtracted from the distance register until the value of the distance register becomes less than the measured time. The value of the distance register depends on the oscillator frequency and the distance between the sensors.
- BCD_convert – in this state, the enable_BCD_convert signal is sequentially activated and reset, which activates the conversion of binary code to binary code decimal. Also set to 0 signal lcd_reset which activates the output of the result on the seven-segment indicator. After that it goes to the wait_afret_measure state.
- Wait_afret_measure - in this state, FPGA waits for 2s and goes to the idle state. This is necessary so that the chronograph does not react to powder gases and the shock wave from the shot after the bullet has traveled.

Also, the TOP module has 2 parameters that must be set, the frequency of the oscillator and the distance between the sensors. They are set when declaring a module. The Oscillator should be trimmed before building the design or you need to measure the OSC frequency and set this value as parameter OSC_frequency.

```
(* top *) module TOP #(
/* --------------------------- Set hardware parameter --------------------------- */
    parameter OSC_frequency = 50000000,
    parameter distance_between_sensors = 0.3
    )(
    (* iopad_external_pin, clkbuf_inhibit *)    input clk,
```

Figure 22: TOP Module Parameters

The distance parameter is calculated as follows:

$$distance = \text{distance\_between\_sensors} * \text{OSC\_frequency} * 10$$

$$distance\_between\_sensors = distance\ between\ sensors\ in\ meters,$$

$$OSC\_frequency = oscillator\ frequency\ in\ hertz,$$

Additionally, the distance multiplied by the frequency is multiplied by another 10, because the measurement resolution is 0.1 m/s.

The clock_for_max_speed parameter is calculated as follows:

$$clock\_for\_\max\_speed = distanse/15010$$

15010 – the maximum speed that the chronograph can measure is multiplied by 10 (resolution 0.1 m/s).

The Floorplanner tab in the FPGA Editor shows the placement of the CLBs and FFs Figure 23. The resource utilization is shown in the top left corner.

It is also worth noting that the FPGA output pins are used to power the seven-segment indicator. Therefore, to increase the brightness, they were configured as 2x push-pull, Figure 24.

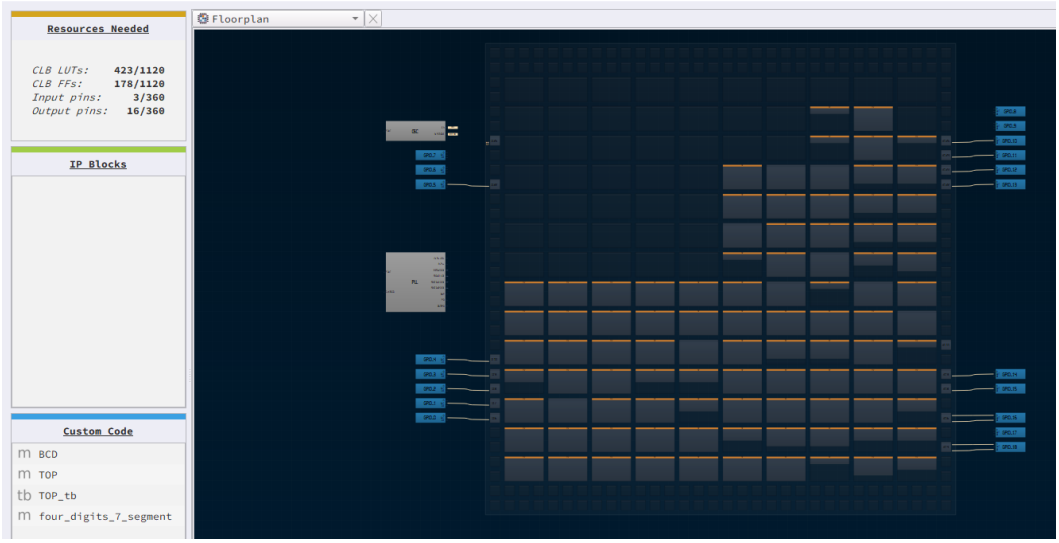

Figure 23: Ballistic Chronograph CLB Utilization

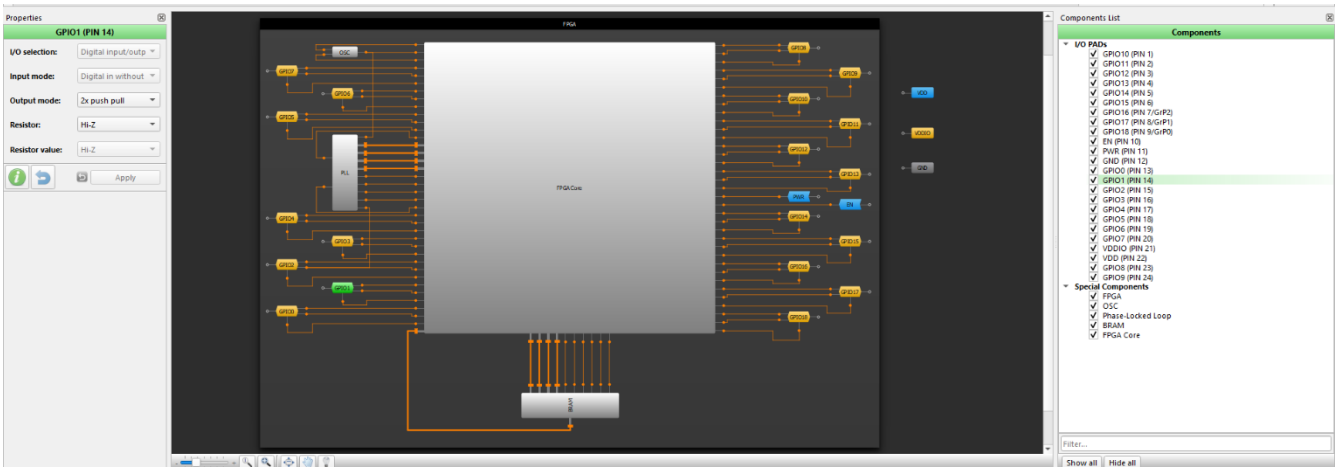

Figure 24: Forge FPGA Window, Output Pin Configuration

7.3 Simulation Results

Also for the TOP module, a testbench is written which simulates the system. Before starting the simulation, you need to replace one line in the Verilog code. Namely, you need to reduce the initial delay in the init state, this is necessary so that the simulation results do not take up much memory.

```
case(state)
init : begin
    counter = counter + 1;
    /* ~1c start delay (if clk == 50Mhz), need for init optical sensor
    * use counter[25:24] == 2'b11 for synthesis
    * use counter[5:4] == 2'b11 for simulation
    */
    if (counter[25:24] == 2'b11)
        state = idle;
end
```

```
case(state)
init : begin
    counter = counter + 1;
    /* ~1c start delay (if clk == 50Mhz), need for init optical sensor
    * use counter[25:24] == 2'b11 for synthesis
    * use counter[5:4] == 2'b11 for simulation
    */
    if (counter[5:4] == 2'b11)
        state = idle;
end
```

Figure 25: On the Left the Init State for Synthesis, on the Right for Simulation

During the simulation, the start and stop signals lasting 100 nanoseconds were transmitted with a delay of 200 microseconds, which at a distance between the sensors of 30 cm is equal to a speed of 1500 m/s. The result of the speed measurement can be seen by looking at the signals tens_of_thousands, thousands, hundreds, tens, ones. Ones are responsible for the least significant digit, tens_of_thousands for the most significant digit. As you can see from the results, the measured speed is 1500.0 m/s.

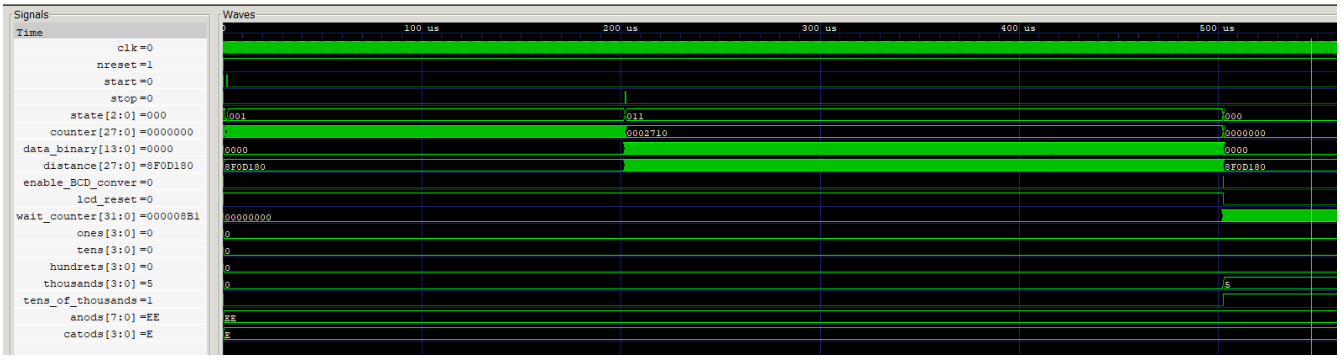

Figure 26: GTWave Simulation Results

8. Prototype Testing

The photo below shows the results of testing a ballistic chronograph.

Figure 27: Cartridge Federal American Eagle CP Solid cal. 22 LR 45 gr. Beginning Speed 295 m/s. Shot from Margolin Pistol

Figure 28: 12 gauge Bullet. Beginning Speed 380 m/s. Shot from a Gun Mossberg Maverick 88 20-inch Barrel

Figure 29: The GGG .223 Rem (5.56x45) Rifled Cartridge with the FMJ bullet (55 gr / 3.56 g). Beginning Speed 980 m/s

Figure 30: Shot from a Gun Ruger AR-556 MPR cal 223 Rem. Air Temperature is +12 C

9. Conclusion

This application note shows how GreenPAK (SLG46620V and SLG46582V) and ForgeFPGA (SLG47910V) products can be used to create a ballistic chronograph with the ability to measure bullet velocities up to 1500m/s. IC GreenPAK and ForgeFPGA demonstrate high efficiency in creating such systems because for a small price we managed to achieve high accuracy and a large dynamic range of speed measurement. This testcase (project link) is available for download. If interested, please contact the ForgeFPGA Business Support Team.

10. Revision History

Revision	Date	Description
1.00	Nov 7, 2022	Initial release.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.