

V850E2/ML4

R01AN1391JJ0100

Rev.1.00

2012.11.01

USB でのマルチメディアカード制御例ソフトウェア

要旨

このアプリケーションノートは、V850E2/ML4 マイクロコントローラ用に作成されたマルチメディアカードドライバに内蔵の USB ファンクション・コントローラ用に作成された MSC(マス・ストレージ・クラス)用サンプルドライバを組み合わせた動作例について説明します。

本アプリケーションノートの内容とソフトウェアは、応用例を説明するもので内容を保証するものではありません。

動作確認デバイス

V850E2/ML4(μ PD70F4022)

目次

1. はじめに	2
2. システム構成	3
3. V850E2/ML4 用サンプルプログラム	4
4. ディスクリプタ	8
5. 関数仕様	11
6. カスタマイズ	12
7. CubeSuite+環境設定	14
8. 動作確認	17
9. 参考ドキュメント	22

1. はじめに

1.1 注意

このアプリケーションノートで使用するサンプルプログラムはあくまで参考用のものであり、当社がこの動作を保証するものではありません。

サンプルプログラムを使用する場合、お客様ののセット上で十分な評価をした上で使用してください。

1.2 対象者

このアプリケーションノートは、V850E2/ML4 マイクロコントローラの機能を理解し、それをういたアプリケーションシステムを開発しようとするユーザを対象とします。

1.3 仕様

USB 対応マルチメディアカードドライバは、次のような特徴があります。

- マス・ストレージ・クラス(以下、MSC)として認識
- セルフ・パワー・デバイスとして動作
- ファイルやフォルダなどのデータをマルチメディアカードに書き込み可能
- マルチメディアカードに書き込んだファイルやフォルダを読み出し可能

1.4 使用機能

- 割り込み機能
- USB ファンクション・コントローラ(USBF)
- クロック同期シリアル・インタフェース H(CSIH)

1.5 適用条件

マイコン	V850E2/ML4
動作周波数	内部クロック : 200MHz
USB クロック	内部クロックと外部クロックを選択可能 内部クロック : 外部 9.6MHz × 内部 20 逡倍 ÷ 4 分周 (48 MHz) または 外部 7.2 MHz × 内部 20 逡倍 ÷ 3 分周 (48 MHz) 外部クロック : USBCLK 端子へ入力 (fUSB = 48 MHz)
MMC クロック	Identification Mode : 343.8KHz Data Transfer Mode : 16.5MHz
統合開発環境	ルネサス エレクトロニクス製 CubeSuite+ V1.02.01
C コンパイラ	ルネサス エレクトロニクス製 CubeSuite+付属 CX V1.21
対応 OS	Windows 7, Vista, XP

1.6 関連アプリケーションノート

- V850E2/ML4 マイクロコンピュータ USB マルチファンクション動作例 (R01AN1037JJ)
- V850E2/ML4 マルチメディアカード SPI モード対応デバイスドライバ : 導入ガイド (R01AN1026JJ)

2. システム構成

サンプルプログラムを利用するシステム構成を下図に示します。

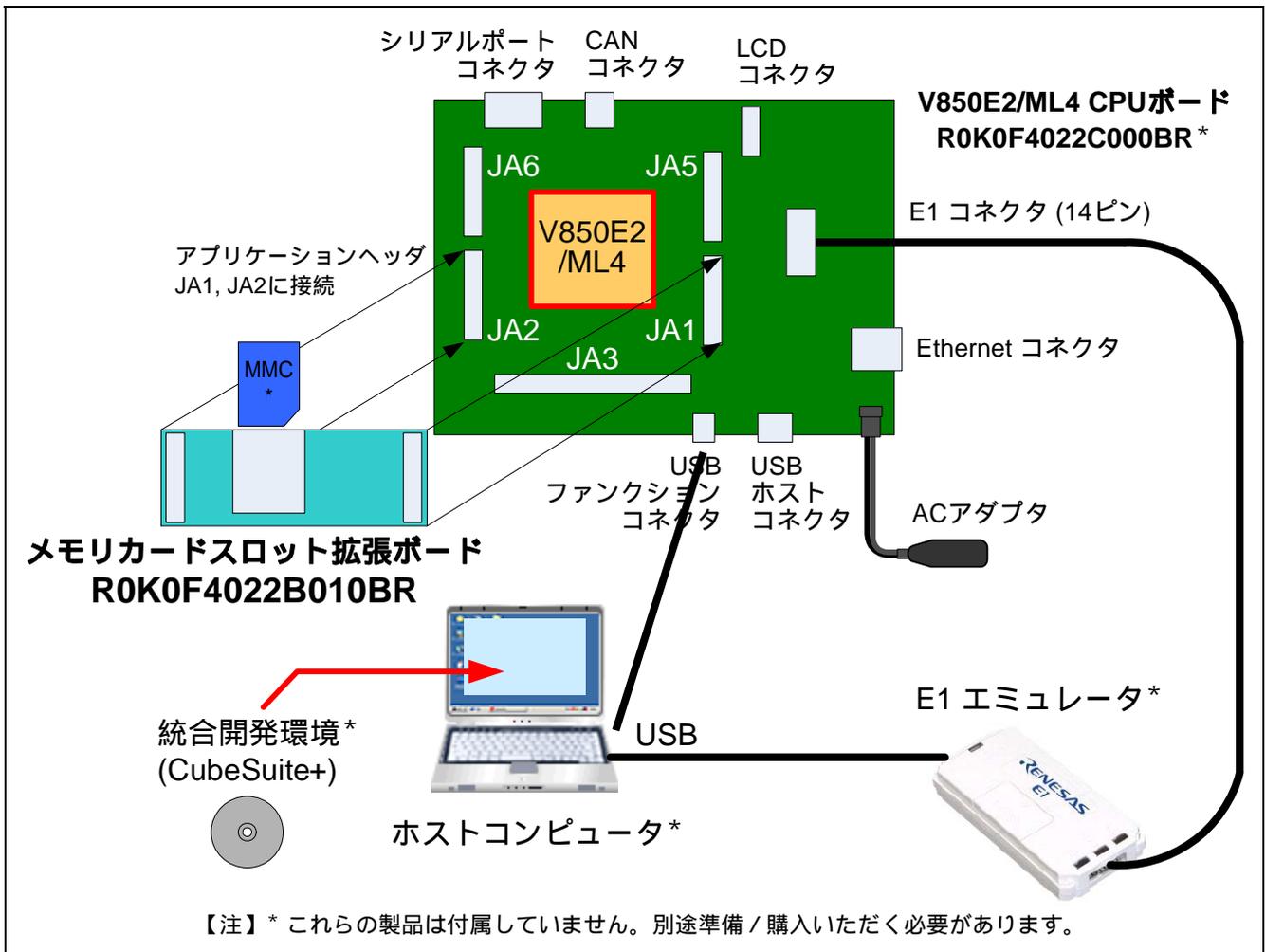


図 2.1 システム構成

3. V850E2/ML4 用サンプルプログラム

3.1 フォルダ構成

サンプルプログラムのフォルダ構成を下表に記載します。

表 3.1 フォルダ構成(CubeSuite+版)

フォルダ名	概要
prj	CubeSuite+用プロジェクト格納フォルダ
src	ソースファイル格納フォルダ

V850E2/ML4 用サンプルプログラムは、メイン処理、マルチメディアカードドライバ、USB MSC ドライバから構成します。USB MSC ドライバは、内蔵メモリをストレージ用データ領域として扱いますが、内蔵メモリの代わりにマルチメディアカードをストレージ用データ領域として扱うように変更しています。

ソースコードは、src フォルダに格納されています。下表にファイル構成を記載します。

表 3.2 ソースコードファイル構成(CubeSuite+版)

フォルダ名	ファイル名	概要
/	cstart.asm	ブートストラップ
	main.c	メインルーチン
	main.h	main.c 関数プロトタイプ宣言
	macrodriver.h	状態制御定義ヘッダファイル
	port.h	CSIH 用ポート設定定義
	serial.c	CSIH 動作制御モジュール
	serial.h	CSIH 動作制御モジュール定義
	serial_user.c	CSIH ユーザ設定モジュール
/usb	usbf850.c	USB 初期化、エンドポイント制御、バルク転送、コントロール転送
	usbf850.h	usbf850.c 関数プロトタイプ宣言
	scsi_cmd.c	SCSI コマンド処理
	usbf850_storage.c	MSC 固有処理
	usbf850_storage.h	usbf850_storage.c 関数プロトタイプ宣言
	usbf850_desc.h	ディスクリプタ定義
	scsi.h	SCSI 関連マクロ定義
	reg_v850e2ml4.h	USB 用レジスタ定義
	usbf850_errno.h	エラーコード定義
	usbf850_types.h	ユーザ型宣言
/mmc	r_mmc.h	マルチメディアカードドライバ共通定義
	r_mmc_io.c	マルチメディアカードドライバ SPI モード I/O モジュール
	r_mmc_io.h	
	r_mmc_mmc.c	マルチメディアカードドライバ SPI モード カード プロトコルモジュール
	r_mmc_sub.c	マルチメディアカードドライバ SPI モード カード プロトコル・サブモジュール
	r_mmc_sub.h	
	r_mmc_spi.c	マルチメディアカードドライバ SPI モード SPI 通信モジュール
r_mmc_usr.c	マルチメディアカードドライバ SPI モードユー ザ API	
/mmc/V850E2_ML4	r_mmc_sfr.h	マルチメディアカードドライバ SFR 定義
	r_mmc_user_config.h	ユーザ定義ファイル
/common	r_mtl_com.c	共通関数 (ログ記録)
	r_mtl_com2.h	共通関数の各種定義
	r_mtl_endi.c	共通関数 (エンディアン関連)
	r_mtl_mem.c	共通関数 (標準ライブラリ関数)
	r_mtl_str.c	共通関数 (標準ライブラリ関数)
	r_mtl_tim.c	共通関数 (ソフトウェアループタイマ)
	r_mtl_tim.h	共通関数 (ソフトウェアループタイマ) の各種定 義
	r_stdint.h	型定義ファイル
/common/V850E2_66MHz	r_mtl_com.h	共通関数ヘッダファイル

3.2 処理の流れ

本サンプルプログラムの全体処理フローを図 3.1 に示します。各初期化処理を実行後、メインループ処理に入ります。

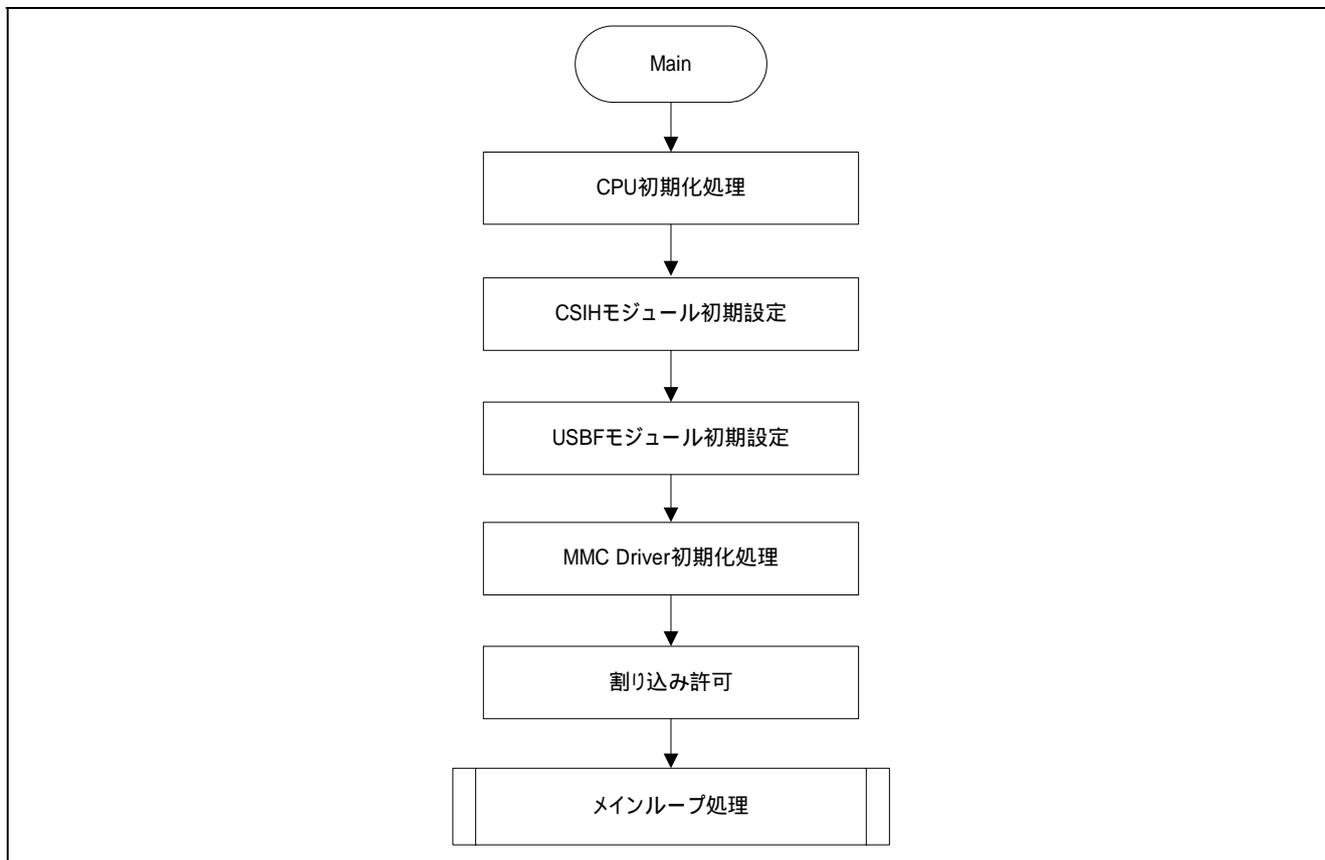


図 3.1 全体処理フロー

メインループ処理を図 3.2 に示します。

メインループでは、

- ・メディア交換処理
- ・USB MSC の Bulk Only Transport プロトコル処理

の 2 つの処理を行います。

メディア交換処理では、メディア未挿入状態 挿入状態に変化したときにマルチメディアカードマウント処理を実行しています。

USB MSC の Bulk Only Transport プロトコルは、

CBW 受信処理

CBW コマンド解析処理

SCSI コマンド処理 (マルチメディアカードへのアクセス)

CSW 送信処理

という処理の流れになります。

各処理の詳細は、USB MSC ドライバ編アプリケーションノートを参照下さい。

CBW 受信処理()は、割り込み処理として実装されています。CBW を受信すると割り込み処理によりフラグがセットされます(cbw_in_cbw = USB_CBW_PROCESS)。

メインループでこのフラグを監視し、フラグがセットされると の処理を実行します。

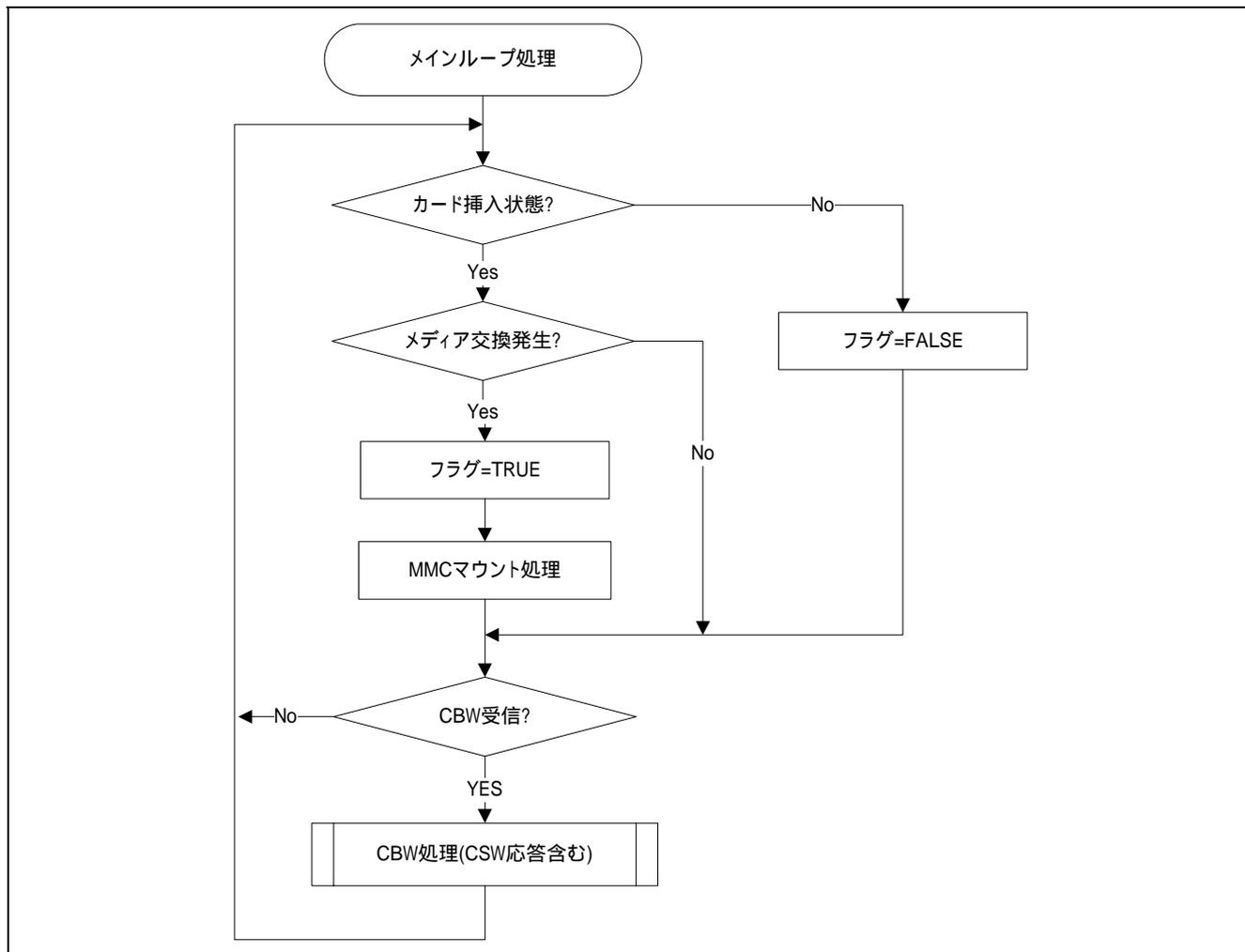


図 3.2 メインループ処理

4. ディスクリプタ

V850E2/ML4 用サンプルプログラムの USB 通信で使用するディスクリプタは、以下の通りです。

4.1 デバイスディスクリプタ

表 4.1 デバイスディスクリプタ

フィールド	サイズ (バイト)	設定値	説明
bLength	1	0x12	ディスクリプタのサイズ：18 バイト
bDescriptorType	1	0x01	ディスクリプタの種類：デバイス
bcdUSB	2	0x0200	USB 仕様リリース番号：USB 2.0
bDeviceClass	1	0x00	クラス・コード：なし
bDeviceSubClass	1	0x00	サブクラス・コード：なし
bDeviceProtocol	1	0x00	プロトコル・コード：固有プロトコル未使用
bMaxPacketSize0	1	0x40	Endpoint0 の最大パケット・サイズ：64
idVendor	2	0x045B	ベンダ ID：Renesas Electronics
idProduct	2	0x0228	プロダクト ID：V850E2/ML4 USB でのマルチメディアカード制御例ソフトウェア
bcdDevice	2	0x0001	デバイスのリリース番号：第 1 版
iManufacturer	1	0x00	製造者を表すストリング・ディスクリプタへのインデックス：0
iProduct	1	0x00	製品を表すストリング・ディスクリプタへのインデックス：0
iSerialNumber	1	0x01	デバイスの製造番号を表すストリング・ディスクリプタへのインデックス：1
bNumConfigurations	1	0x01	コンフィギュレーションの数：1

注：ベンダ ID とプロダクト ID はお客様のシステムに合わせてください。

4.2 コンフィグレーションディスクリプタ

表 4.2 コンフィグレーションディスクリプタ

フィールド	サイズ (バイト)	設定値	説明
bLength	1	0x09	ディスクリプタのサイズ：9 バイト
bDescriptorType	1	0x02	ディスクリプタの種類：コンフィギュレーション
wTotalLength	2	0x0020	コンフィギュレーション、インタフェース、およびエンドポイント・ディスクリプタの総バイト数：32 バイト
bNumInterfaces	1	0x01	このコンフィギュレーションが持つインタフェースの数：1
bConfigurationValue	1	0x01	このコンフィギュレーションの識別番号：1
iConfiguration	1	0x00	このコンフィギュレーションを記述するストリング・ディスクリプタへのインデックス：0
bmAttributes	1	0xC0	このコンフィギュレーションの特徴：セルフ・パワー、リモート・ウェイクアップなし
bMaxPower	1	0x1B	このコンフィギュレーションの最大消費電流：54 mA

4.3 インタフェースディスクリプタ

表 4.3 インタフェースディスクリプタ

フィールド	サイズ (バイト)	設定値	説明
bLength	1	0x09	ディスクリプタのサイズ: 9 バイト
bDescriptorType	1	0x04	ディスクリプタの種類: インタフェース
bInterfaceNumber	1	0x00	このインタフェースの識別番号: 0
bAlternateSetting	1	0x00	このインタフェースに対するオルタナティブ設定の有無: なし
bNumEndpoints	1	0x02	このインタフェースが持つエンドポイントの数: 2
bInterfaceClass	1	0x08	クラス・コード: マス・ストレージ・クラス
bInterfaceSubClass	1	0x06	サブクラス・コード: SCSI transparent command set
bInterfaceProtocol	1	0x50	プロトコル・コード: Bulk Only Transport プロトコル
iInterface	1	0x00	このインタフェースを記述するstring・ディスクリプタへのインデックス: 0

4.4 エンドポイントディスクリプタ

エンドポイントのアドレスはヘッダファイル(usb850.h)の下記の define 定義を有効/無効にすることで切り替えが可能です(デフォルトは有効)。

```
#define USE_EP_BK11
#define USE_EP_BK01
```

表 4.4 エンドポイントの組み合わせ

	Bulk IN	Bulk OUT
define を有効	EP1	EP2
define を無効	EP3	EP4

表 4.5 EP1(Bulk IN)のエンドポイントディスクリプタ

フィールド	サイズ (バイト)	設定値	説明
bLength	1	0x07	ディスクリプタのサイズ: 7 バイト
bDescriptorType	1	0x05	ディスクリプタの種類: エンドポイント
bEndpointAddress	1	0x81	このエンドポイントの転送方向: IN 方向 このエンドポイントのアドレス: 1
bmAttributes	1	0x02	このエンドポイントの転送タイプ: バルク
wMaxPacketSize	2	0x0040	この転送の最大パケット・サイズ: 64 バイト
bInterval	1	0x00	このエンドポイントのポーリング間隔: 0 ms

表 4.6 EP2(Bulk OUT)のエンドポイントディスクリプタ

フィールド	サイズ (バイト)	設定値	説明
bLength	1	0x07	ディスクリプタのサイズ：7 バイト
bDescriptorType	1	0x05	ディスクリプタの種類：エンドポイント
bEndpointAddress	1	0x02	このエンドポイントの転送方向：OUT 方向 このエンドポイントのアドレス：2
bmAttributes	1	0x02	このエンドポイントの転送タイプ：バルク
wMaxPacketSize	2	0x0040	この転送の最大パケット・サイズ：64 バイト
bInterval	1	0x00	このエンドポイントのポーリング間隔：0 ms

4.5 スtringディスクリプタ

表 4.7 スtringディスクリプタ(String 0)

フィールド	サイズ (バイト)	設定値	説明
bLength	1	0x04	ディスクリプタのサイズ：4 バイト
bDescriptorType	1	0x03	ディスクリプタの種類：String
bString	2	0x09,0x04	言語コード：(U.S)

表 4.8 スtringディスクリプタ(String 1)

フィールド	サイズ (バイト)	設定値	説明
bLength	1	0x1A	ディスクリプタのサイズ：4 バイト
bDescriptorType	1	0x03	ディスクリプタの種類：String
bString	24	-	シリアル番号：022708065010

5. 関数仕様

マルチメディアカードドライバの提供する API 関数は以下です。

表 5.1 マルチメディアカードドライバ API 関数

関数名	説明
R_mmc_Init_Driver	マルチメディアカードドライバ初期化処理
R_mmc_Init_Slot	マルチメディアカード初期化処理
R_mmc_Detach	マルチメディアカードドライバ停止処理
R_mmc_Read_Data	マルチメディアカードからデータを読み出す
R_mmc_Write_Data	マルチメディアカードへデータを書き込む
R_mmc_Chk_Detect	メディア挿入状態をチェックする
R_mmc_Get_MmcInfo	メディアの情報(容量など)を取得する

マルチメディアカードにアクセスするために USB MSC ドライバの以下の関数を変更しています。アクセス対象が仮想ディスク(内蔵メモリ)からマルチメディアカードに変更されていますが、関数名や引数等に変更はありません。関数仕様は、USB MSC ドライバ編アプリケーションノートを参照下さい。

表 5.2 マルチメディアカードドライバ API 関数を使用する関数

関数名	説明
ata_read6	R_mmc_Read_Data()を使用し、メディアからデータをリードします。
ata_read10	R_mmc_Read_Data()を使用し、メディアからデータをリードします。
ata_write6	R_mmc_Write_Data()を使用し、メディアにデータをライトします。
ata_write10	R_mmc_Write_Data()を使用し、メディアにデータをライトします。
ata_verify	R_mmc_Read_Data()を使用し、メディアからデータをリードします。
ata_write_verify	R_mmc_Write_Data()を使用し、メディアにデータをライトします。
ata_read_format_capacities	R_mmc_Get_MmcInfo()を使用し、メディア容量を取得します。
ata_read_capacity	R_mmc_Get_MmcInfo()を使用し、メディア容量を取得します。
ata_test_unit_ready	R_mmc_Chk_Detect()を使用し、メディア挿入/未挿入状態を取得します。
全 ata_xxxxxx 関数共通 (一部例外あり)	R_mmc_Chk_Detect()を使用し、カード未挿入状態の場合はエラー応答します。

6. カスタマイズ

6.1 ディスクリプタの内容

初期化処理時に USB ドライバが USB ファンクション・コントローラに登録するデータ（「4. ディスクリプタ」参照）が "usbf850_desc.h" ファイルに定義されています。このファイル内の値を実際のアプリケーションでの使用法に合わせて書き換えることで、サンプルプログラムを通してターゲット・デバイスの属性などの情報を設定できます。

また、istring ディスクリプタには任意の情報を登録できます。サンプルプログラムではシリアル番号を定義していますので、適宜書き換えてください。

```

:
/* 0 : Language Code*/
DSTR(LangString, 2, (0x09,0x04));
/* 1 : Serial Number*/
USTR(SerialString, 12, ('0','2','2','7','0','8','0','6','5','0','1','0'));
:

```

図 6.1 "usbf850_desc.h"のistring ディスクリプタ設定部

6.2 ベンダ名やプロダクト名の設定

"scsi_cmd.c" ファイルに定義されている INQUIRY コマンドの応答値を編集することで、ディスク・ドライブのベンダ名やプロダクト名として表示される名前を変更できます。

"scsi_cmd.c" ファイルの "INQUIRY_TABLE" は下図のように記述されています。

```

1  UINT8  INQUIRY_TABLE[INQUIRY_LENGTH]={
2      0x00,                                /*Qualifier, device type code*/
3      0x80,                                /*RMB, device type modification child*/
4      0x02,                                /*ISO Version, ECMA Version, ANSI Version*/
5      0x02,                                /*AENC, TrmIOP, response data form*/
6      0x1F,                                /*addition data length*/
7      0x00,0x00,0x00,                    /*reserved*/
8      'R','e','n','e','s','a','s',' ',    /*vender ID*/                               <1>
9      'S','t','o','r','a','g','e','F','n','c','D','r','i','v','e','r',
                                           /*product ID*/                               <2>
10     '0','.','0','1'
11 };

```

図 6.2 "scsi_cmd.c" ファイルの "INQUIRY_TABLE" の記述

8 行目<1>でベンダ名、9 行目<2>でプロダクト名を定義しています。ベンダ名には 8 バイト（半角 8 文字）、プロダクト名には 16 バイト（半角 16 文字）の文字列を指定できます。

データ送信時は各文字を ASCII コードの数値にして送信します。このため、ASCII コードでデコードできない文字を使用した場合、正常に表示されません。

"INQUIRY_TABLE"で指定したベンダ名とプロダクト名はデバイス マネージャのディスク ドライブ名として表示されます。

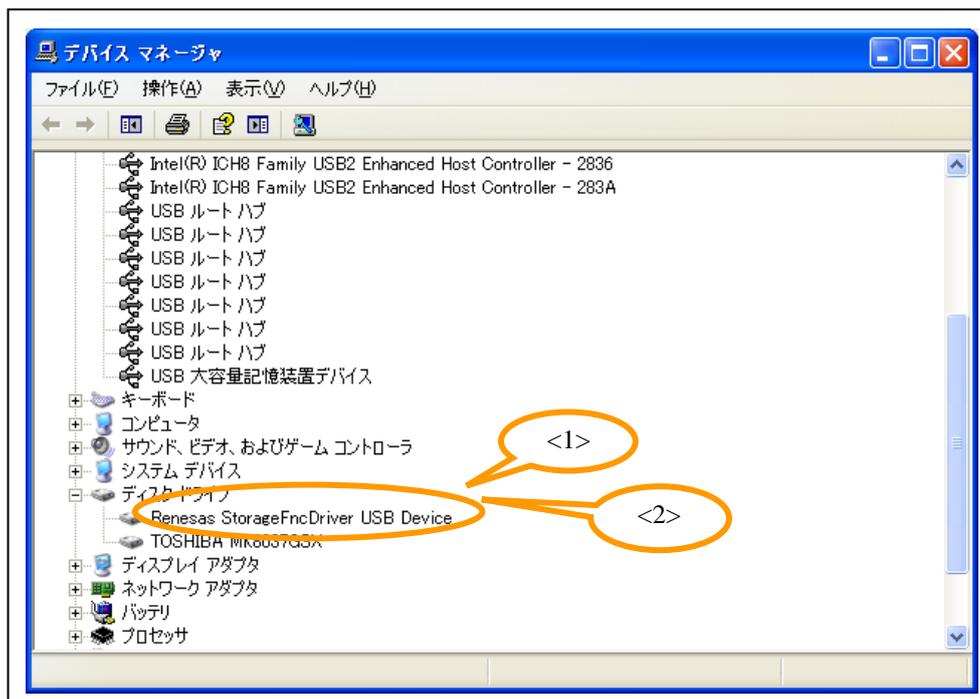


図 6.3 デバイス マネージャの表示

7. CubeSuite+環境設定

7.1 インストール手順

(1) 展開

サンプルソフトウェアの提供ファイル一式をフォルダ構成を変えずに任意のディレクトリに展開します。

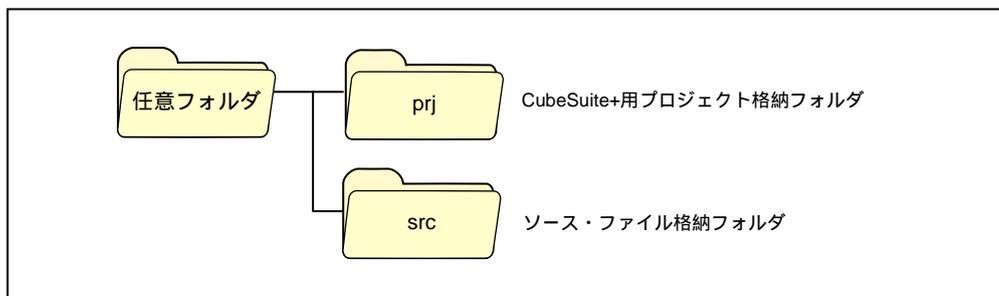


図 7.1 フォルダ構成

(2) ワークスペースの設定

- CubeSuite+を起動し、「ファイル」メニューから「ファイルを開く」を選択します。

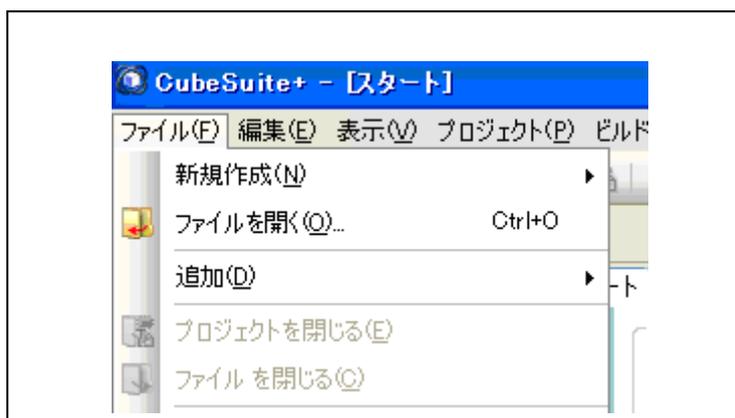


図 7.2 CubeSuite+メニュー選択

- 「ファイルを開く」ダイアログが開きます。サンプルプログラムを格納したディレクトリの「prj」フォルダにある CubeSuite+用プロジェクト・ファイルを指定します。

(3) ビルド・ツールの設定

- CubeSuite+の「プロジェクト・ツリー」から「CX(ビルド・ツール)」を選択し、プロパティを表示します。



図 7.3 ビルド・ツール選択

- 「バージョン選択」の項目を選択し、「使用するコンパイラ・パッケージのバージョン」の項を「使用するコンパイラ・パッケージのバージョン」に設定します。

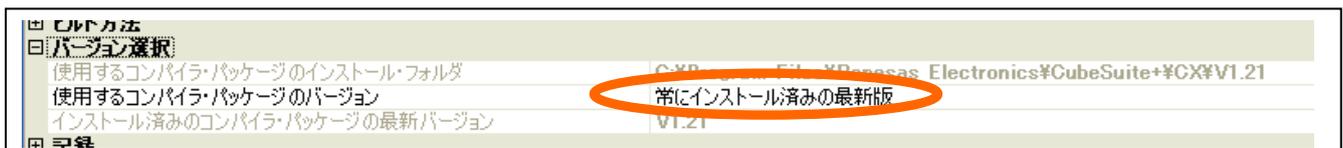


図 7.4 コンパイラ・パッケージ設定

- 「プロジェクト・ツリーより「V850E2M E1(デバッグ・ツール)」を選択し、右クリックメニューから「使用するデバッグ・ツール」 「V850E2 E1」を選択します。

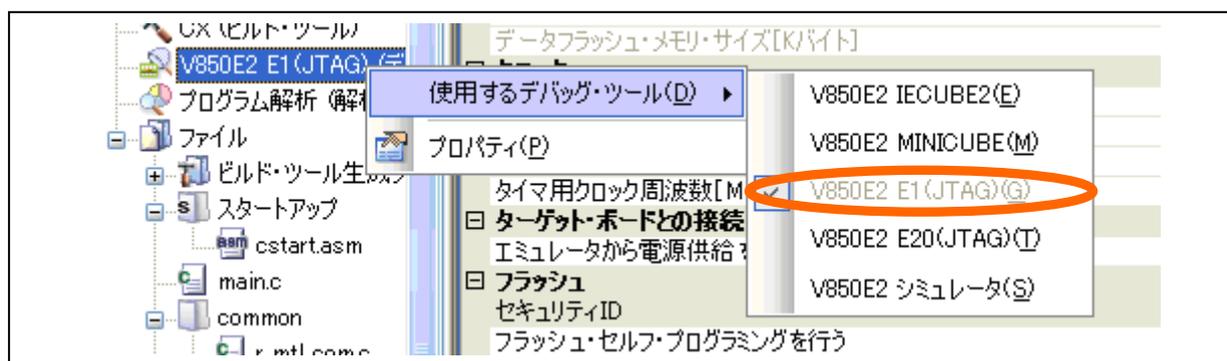


図 7.5 デバッグ・ツール選択

7.2 CubeSuite+環境デバッグ

7.2.1 ロード・モジュール生成

ターゲット・デバイスにプログラムを書き込むには、C言語やアセンブリ言語で記述されたファイルをCコンパイラなどで変換してロード・モジュールを生成します。

CubeSuite+では、「ビルド」メニューから「ビルド・プロジェクト」を選択すると、ロード・モジュールが生成されます。

7.2.2 ロードと実行

生成したロード・モジュールをターゲットに書き込んで（ロード）実行させます。

(1) ロード・モジュールの書き込み

- 「デバッグ」メニューから「デバッグ・ツールヘダウンロード」を選択してデバッガを起動します。
- デバッグ・ツールを介して、ロード・モジュールのダウンロードが開始されます。

(2) プログラムの実行

CubeSuite+の  ボタンを押下します。または「デバッグ」メニューから「実行」を選択します。

8. 動作確認

8.1 ジャンパ設定

評価ボードの電源を入れる前に、ジャンパ設定を確認してください。

(1) USB インタフェース部

V850E2/ML4 CPU 評価ボード(R0K0F4022C000BR)のジャンパを以下のように設定してください。

ジャンパ	設定	備考
JP1	1-2	VBUS を有効にします
JP4	1-2	P2_4 による D+制御を有効にします
JP9	2-3	P2_10/CSI1F_CS4 を JA1-23 に接続します。

(2) マルチメディアカードインタフェース部

メモリカードスロット拡張ボード(R0K0F4022B010BR)のジャンパ設定は以下です。本サンプルプログラムは、ソフトウェアによるカード電源供給制御を実装していない為、必ず 1-2 ショートに設定してください。

ジャンパ	設定	備考
JP1	1-2	カード電源供給制御を CD 信号で行う

8.2 動作確認手順

ここではサンプルプログラムを CubeSuite+環境で実行した後、実行結果を確認する手順を示します。

(1) USB mini-B コネクタとの接続

V850E2/ML4 CPU ボードの USB mini-B コネクタとホスト・マシンの USB ポートを USB ケーブルで接続します。

(2) ホストドライバのインストール

USB B コネクタとの接続に使用するドライバは、Windows 標準のマス・ストレージ・クラス用ホストドライバを使用します。サンプルプログラムを実行している状態でホスト・マシンに接続すると、自動的にドライバがインストールされます。

(3) USB デバイスの接続確認

Windows のデバイス マネージャを開きます。「USB (Universal Serial Bus) コントローラ」のツリーを展開し、「USB 大容量記憶装置デバイス」が表示されていることを確認します。「ディスク ドライブ」のツリーを展開し、「Renesas StorageFncDriver USB Device」が表示されていることを確認します。

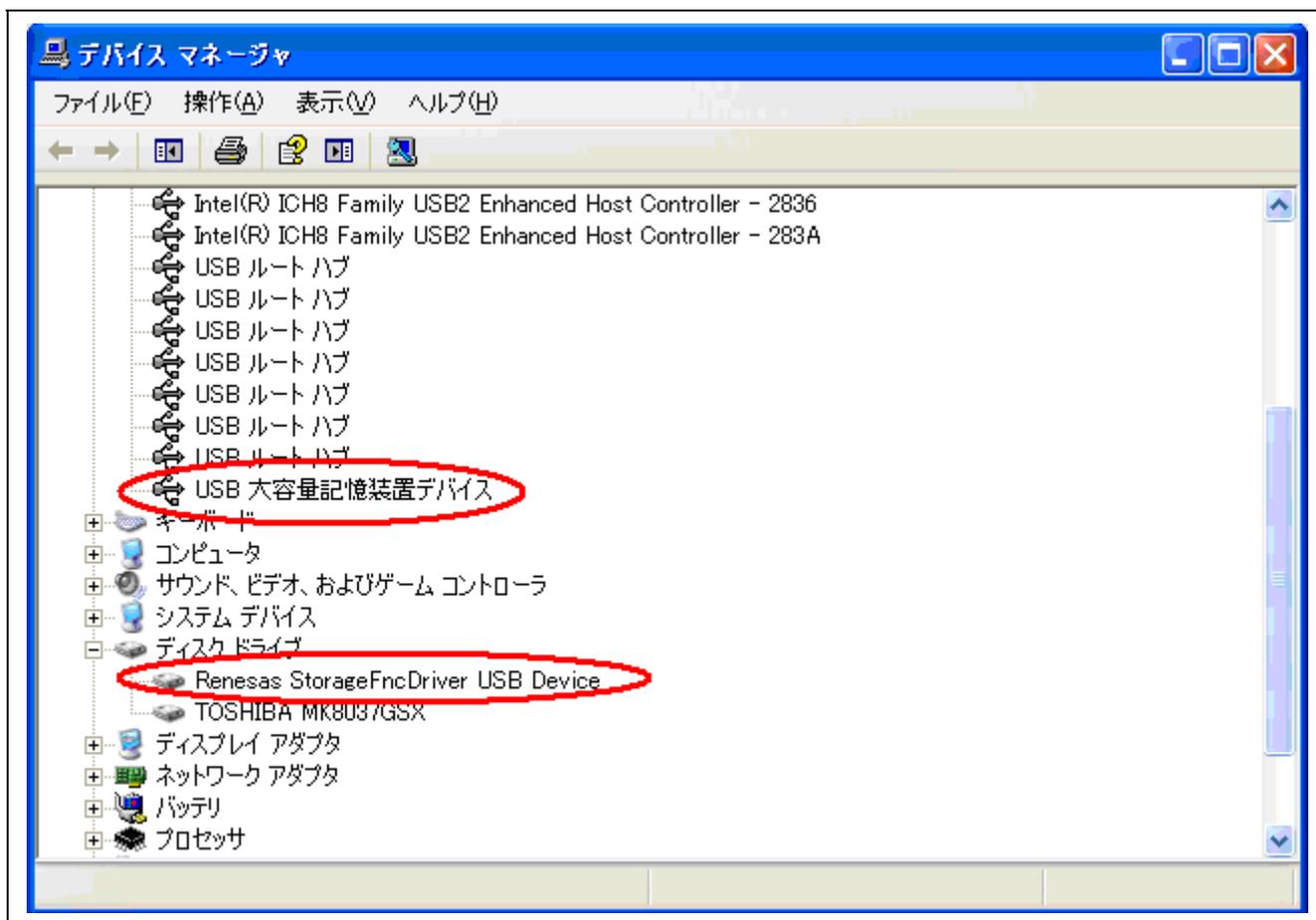


図 8.1 デバイスマネージャ確認

(4) ファイルアクセス

リムーバブルディスクに対するファイルの書き込みと読み出しを確認します。

- ローカル・ディスク上に TEST.txt ファイルと Test フォルダを用意します。

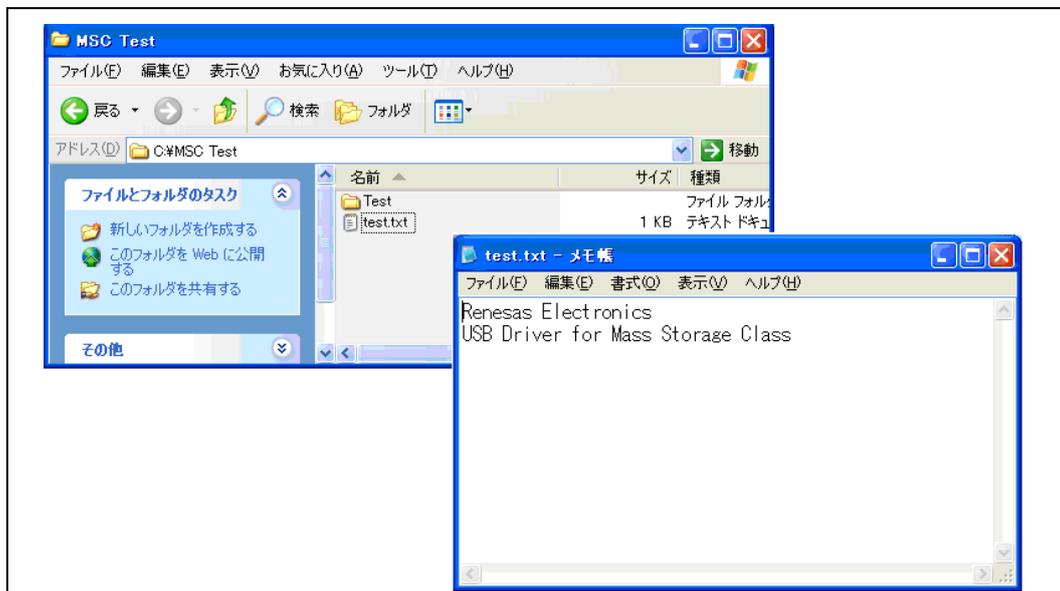


図 8.2 テスト用フォルダとテストデータファイル

- 「マイ コンピュータ」のリムーバブル ディスクを開き、ローカル・ディスクからリムーバブルディスクへ TEST.txt ファイルをコピーします。

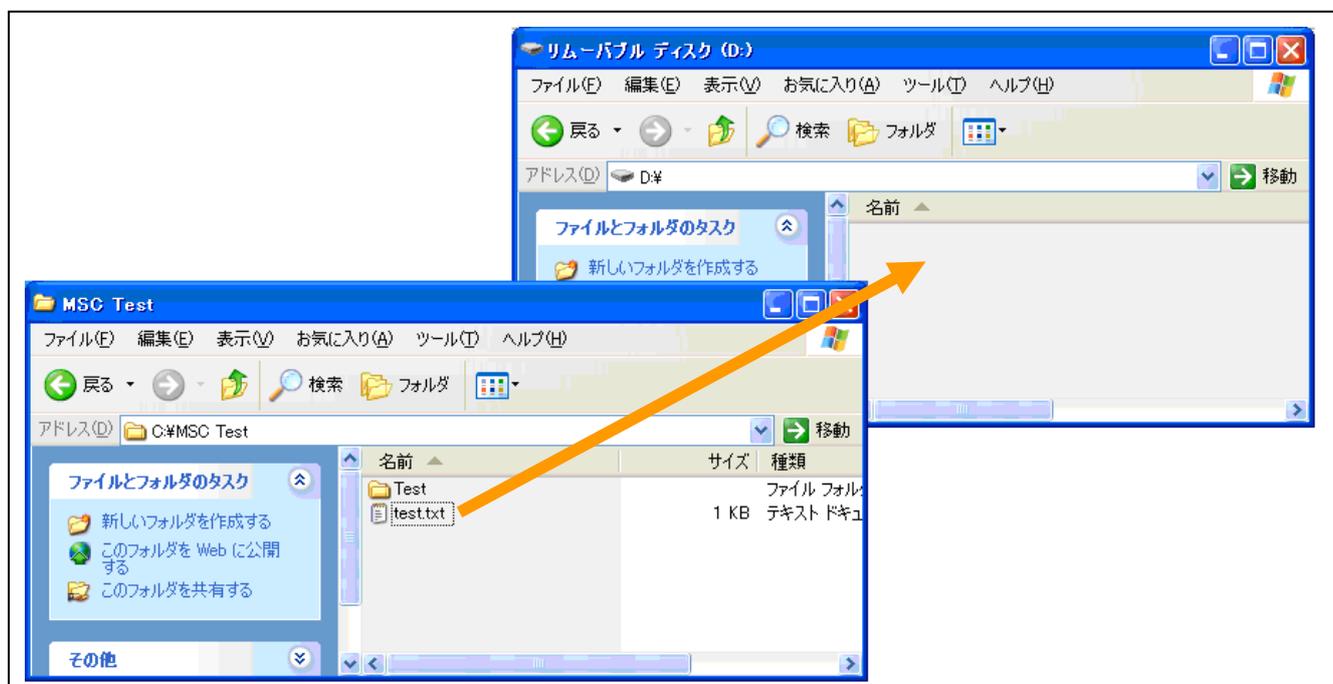


図 8.3 テストデータファイルのコピー

- ローカル・ディスクの Test フォルダを開き、リムーバブル ディスクから Test フォルダへ TEST.txt ファイルをコピーします。

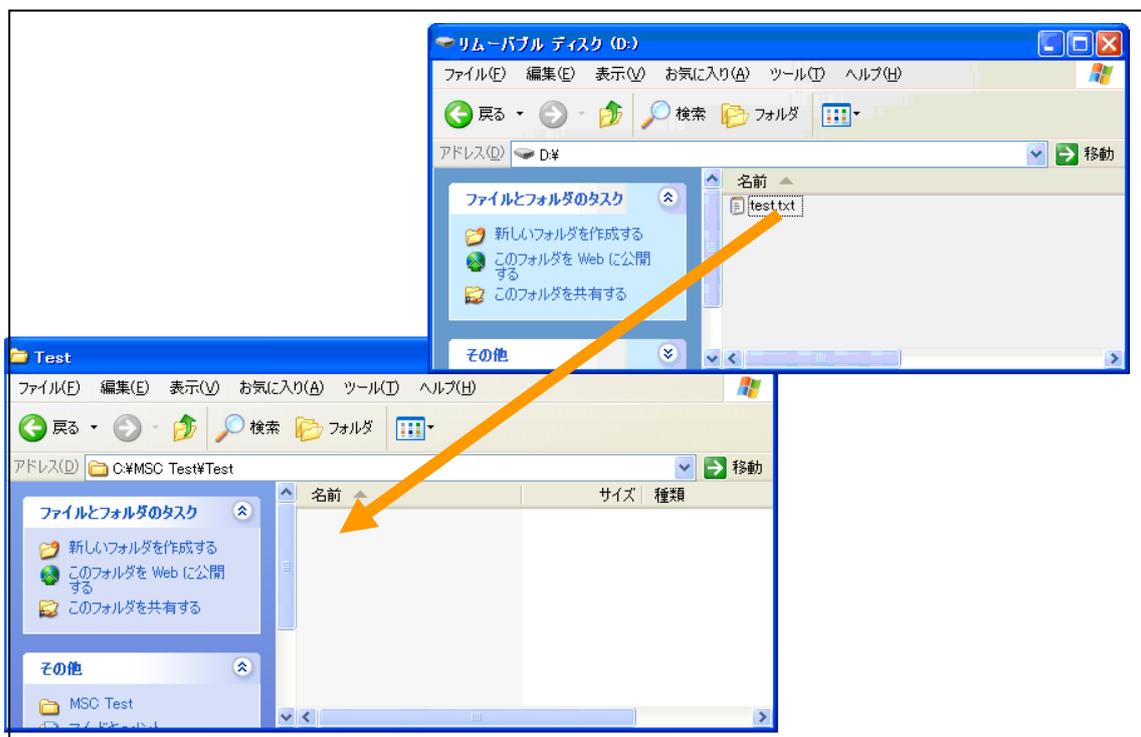


図 8.4 テストデータファイルの再コピー

- Test フォルダの TEST.txt ファイルを開き、ローカル・ディスク上に用意した TEST.txt ファイルと内容が同じであることを確認します。

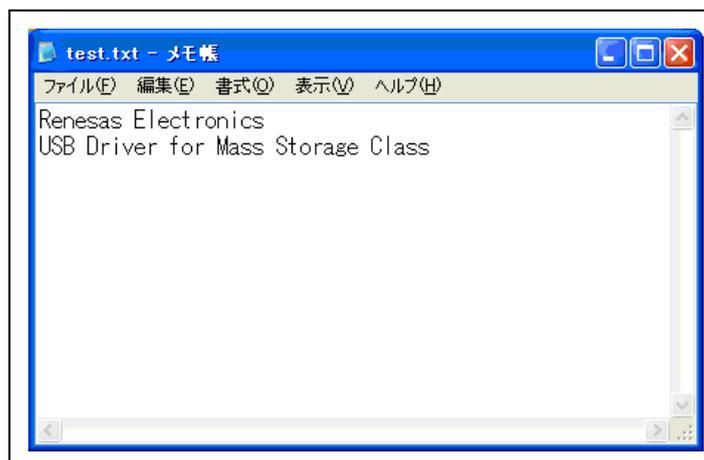


図 8.5 テストデータファイルの確認

- USB 切断後にマルチメディアカードを取り出し、PC で内容を確認します(マルチメディアカードスロットのある PC を使用してください)。マルチメディアカードに TEST.txt が書き込まれていることを確認します。

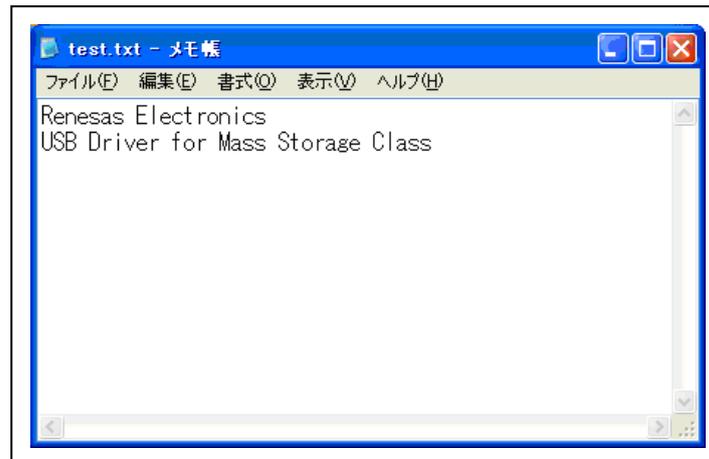


図 8.6 マルチメディアカードのテストデータファイルの確認

9. 参考ドキュメント

- ハードウェアマニュアル

[1] V850E2/ML4 ユーザーズマニュアル ハードウェアマニュアル (R01UH0262JJ)

(最新版はルネサスエレクトロニクスのホームページから入手してください)

[2] V850E2/ML4 CPU ボード ユーザーズマニュアル(R20UT0778JJ)

[3] メモリカードスロット拡張ボード ユーザーズマニュアル(R20UT2111JJ)

- ソフトウェアマニュアル

V850E2M ユーザーズマニュアル アーキテクチャ編 (R01US0001JJ)

(最新版はルネサスエレクトロニクスのホームページから入手してください)

ホームページとサポート窓口

- ルネサス エレクトロニクスホームページ
<http://japan.renesas.com/>
- お問い合わせ先
<http://japan.renesas.com/contact/>

すべての商標および登録商標は、それぞれの所有者に帰属します。

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレスがあります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違っていると、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、
家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交信用信号機器、
防災・防犯装置、各種安全装置等
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍用用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<http://japan.renesas.com/contact/>