

RX ファミリ

R20AN0046JJ0102

SPI モードマルチメディアカードドライバ: 導入ガイド

Rev.1.02

2013.02.20

要旨

本書は、RX ファミリ用 SPI モードマルチメディアカードドライバ V.1.02 Release 00 のソフトウェア構成、使用方法について記述します。

動作確認デバイス

対応マイコン：

- RX610 グループ、RX62N グループ、RX210 グループ、RX630 グループ、RX63N グループ

サンプルプログラム動作環境：

- Renesas Starter Kit for RX610 (型名：R0K556100S000BE)
- Renesas Starter Kit+ for RX62N (型名：R0K5562N0S000BE)
- Renesas Starter Kit for RX210 (型名：R0K505210S000BE)
- Renesas Starter Kit for RX630 (型名：R0K505630S000BE)
- Renesas Starter Kit+ for RX63N (型名：R0K50563NS000BE)

サンプルプログラムの動作確認には、上記環境に加え

「ミドルウェア評価ボード」またはそれに準ずるマルチメディアカードソケット拡張基板が必要です。

「ミドルウェア評価ボード」については以下アプリケーションノートをご参照ください。

- Renesas Starter Kit for RX610 ミドルウェア評価ボードの使用方法(R21AN0004JJ0103)

目次

1. 概要	2
2. プログラム型定義.....	5
3. デバイスドライバ.....	6
4. 設定例.....	12
5. MCU との接続方法と使用する MCU リソース	31
6. アプリケーション作成時の注意事項.....	33
7. サンプルプログラム	37
8. MMC ドライバ改訂履歴	41

1. 概要

1.1 目的

RX ファミリ MCU とマルチメディアカード(以下、MMC と略す。)を SPI モードで通信させるためのインタフェースを提供することを目的としています。

本アプリケーションノートでは、アプリケーションを作成するための情報を提供します。

1.2 機能概要

本ドライバ(以下、MMC ドライバと略す。)は、MMC との通信を RX ファミリ MCU で実現するためのソフトウェアです。

本ソフトウェアは、RX ファミリ MCU に内蔵されている通信機能のうち、シリアルコミュニケーションインタフェース(以下、SCI と略す。)、またはルネサスシリアルペリフェラルインタフェース(以下、RSPI と略す。)を使い、SPI モードでの MMC へのアクセスを実現しています。

MMC ドライバの仕様

- 参照 MMCA 規格 Ver. 3.2
- MMC の SPI モード専用です。
- 1セクタ=512Byte とするブロック型デバイスドライバです。
READ_MULTIPLE_BLOCK コマンドと WRITE_MULTIPLE_BLOCK コマンドを使用しています。
上記2つの MULTIPLE_BLOCK コマンド非対応カードの場合は、READ_SINGLE_BLOCK コマンドと WRITE_SINGLE_BLOCK コマンドで対応します。
- CS 制御による複数デバイスをサポートしています。
- OS 非依存です。

1.3 アプリケーションノート構成

本アプリケーションノートのディレクトリおよびファイルの構成を示します。

ディレクトリ構成、 ディレクトリ名/ファイル名	備考
¥doc <DIR>	ドキュメントディレクトリ
¥ja <DIR>	日本語版ディレクトリ
r20an0046jj0102_rx_mmc.pdf	導入ガイド (本書)
¥en <DIR>	英語版ディレクトリ
r20an0046ej0102_rx_mmc.pdf	Introduction Guide
¥drv <DIR>	MMC ドライバソースプログラム
¥com <DIR>	共通関数のディレクトリ
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, r_mtl_tim.h	共通関数(ソフトウェアループタイマ)、各種定義
r_stdint.h	型定義ヘッダファイル
¥rx200_50MHz <DIR>	RX200 シリーズ 50MHz 動作定義のディレクトリ (CPU に依存した設定用ヘッダファイルがあります。)
¥rx600_100MHz <DIR>	RX600 シリーズ 100MHz 動作定義のディレクトリ
¥rx600_96MHz <DIR>	RX600 シリーズ 96MHz 動作定義のディレクトリ
¥mmc <DIR>	MMC 用デバイスドライバのディレクトリ
r_mmc.h	MMC ドライバ共通定義
r_mmc_mcu_pragma.h	MMC ドライバ コンパイラのプリAGMA宣言ヘッダファイル
r_mmc_io.c r_mmc_io.h	MMC ドライバ SPI モード I/O モジュール
r_mmc_mmc.c	MMC ドライバ SPI モード MMC モジュール
r_mmc_sub.c r_mmc_sub.h	MMC ドライバ SPI モード・サブモジュール
r_mmc_usr.c	MMC ドライバ SPI モード・API ソースプログラム
¥rx <DIR>	RX ファミリ用通信モジュールディレクトリ
r_mmc_sci.c	MMC ドライバ SPI モード SCI を用いた通信モジュール
r_mmc_rspi.c	MMC ドライバ SPI モード RSPI を用いた通信モジュール
r_mmc_rx_str.h	RX ファミリ用構造体定義ヘッダファイル
¥rx210_sci <DIR>	RX210 グループ SCI 用 SFR 定義ディレクトリ (各シリアルインタフェース用の設定用ヘッダファイルがあります。)
¥rx210_rspi <DIR>	RX210 グループ RSPI 用 SFR 定義ディレクトリ
¥rx610_sci <DIR>	RX610 グループ SCI 用 SFR 定義ディレクトリ
¥rx62n_sci <DIR>	RX62N グループ SCI 用 SFR 定義ディレクトリ
¥rx62n_rspi <DIR>	RX62N グループ RSPI 用 SFR 定義ディレクトリ
¥rx630_sci <DIR>	RX630 グループ SCI 用 SFR 定義ディレクトリ
¥rx630_rspi <DIR>	RX630 グループ RSPI 用 SFR 定義ディレクトリ
¥rx63n_sci <DIR>	RX63N グループ SCI 用 SFR 定義ディレクトリ
¥rx63n_rspi <DIR>	RX63N グループ RSPI 用 SFR 定義ディレクトリ
¥sample <DIR>	サンプルプロジェクトディレクトリ

¥MMC_sample_RX210 <DIR>	RX210 用 High-performance Embedded Workshop ディレクトリ
MMC_sample_RX210.hws	ワークスペース
¥MMC_sample_RX610 <DIR>	RX610 用 High-performance Embedded Workshop ディレクトリ
MMC_sample_RX610.hws	ワークスペース
¥MMC_sample_RX62N <DIR>	RX62N 用 High-performance Embedded Workshop ディレクトリ
MMC_sample_RX62N.hws	ワークスペース
¥MMC_sample_RX630 <DIR>	RX630 用 High-performance Embedded Workshop ディレクトリ
MMC_sample_RX630.hws	ワークスペース
¥MMC_sample_RX63N <DIR>	RX63N 用 High-performance Embedded Workshop ディレクトリ
MMC_sample_RX63N.hws	ワークスペース
¥src <DIR>	テスト用サンプルプログラムのディレクトリ
¥apl <DIR>	MMC ドライバ動作検証用のサンプルプログラムディレクトリ
¥drv <DIR>	MMC ドライバソースプログラムディレクトリ
¥bsp <DIR>	スタートアップルーチンプログラムディレクトリ

2. プログラム型定義

本プログラムの整数型の定義は以下のとおりとします。

Datatype	Typedef
unsigned char	uint8_t
unsigned short	uint16_t
unsigned long	uint32_t
signed char	int8_t
signed short	int16_t
signed long	int32_t

3. デバイスドライバ

3.1 ドライバ関数概要

初期化関数

関数名	機能概要
R_mmc_Init_Driver ()	MMC ドライバ初期化処理

デバイス操作関数

関数名	機能概要
R_mmc_Init_Slot()	MMC スロット初期化処理
R_mmc_Detach()	MMC スロット停止処理
R_mmc_Chk_Detect()	MMC 挿入チェック処理

データ・アクセス操作関数

関数名	機能概要
R_mmc_Read_Data()	データ読み出し処理
R_mmc_Write_Data()	データ書き込み処理
R_mmc_Get_MmcInfo()	MMC 情報取得処理

内部使用コマンド一覧

コマンドインデックス	コマンド名
CMD0	GO_IDLE_STATE
CMD1	SEND_OP_COND
CMD9	SEND_CSD
CMD10	SEND_CID
CMD12	STOP_TRANSMISSION
CMD13	SEND_STATUS
CMD17	READ_SINGLE_BLOCK
CMD18	READ_MULTIPLE_BLOCK
CMD24	WRITE_BLOCK
CMD25	WRITE_MULTIPLE_BLOCK
CMD58	READ_OCR
CMD59	CRC_ON_OFF

【注】 未サポート・コマンドに対しては、ユーザ側で対応してください。

3.2 関数詳細

3.2.1 MMC ドライバ初期化処理(R_mmc_Init_Driver)

項目	内容
プロトタイプ	void R_mmc_Init_Driver(void)
引数	なし
説明	MMC ドライバの初期化を行います。 MMC 制御用 SFR の初期化、スロット毎の制御ポートと RAM を初期化します。 本関数はシステム起動時に一度だけ呼び出してください。
戻り値	なし

3.2.2 MMC スロット初期化処理(R_mmc_Init_Slot)

項目	内容
プロトタイプ	int16_t R_mmc_Init_Slot(uint8_t SlotNo)
引数	uint8_t SlotNo : スロット番号
説明	引数で指定されたスロットに対する RAM およびポートの初期化を行います。 また、MMC に対して初期化処理を行います。 本関数はカード挿入検出時に呼び出してください。
戻り値	初期化結果を返します。 MMC_OK : 正常終了 MMC_ERR_PARAM : パラメータエラー MMC_ERR_HARD : ハードウェアエラー MMC_ERR_CRC : CRC エラー MMC_ERR_IDLE : Idle state エラー MMC_ERR_OTHER : その他エラー

3.2.3 MMC スロット停止処理(R_mmc_Detach)

項目	内容
プロトタイプ	int16_t R_mmc_Detach(uint8_t SlotNo)
引数	uint8_t SlotNo : スロット番号
説明	指定スロット MMC 引抜き時の処理を行います。 MMC 制御用 SFR の初期化、制御 port の開放、制御用 RAM の初期化を行います。 本関数はカード引抜き検出時に呼び出してください。
戻り値	引き抜き処理の結果を返します。 MMC_OK : 正常終了 MMC_ERR_PARAM : パラメータエラー

3.2.4 MMC 挿入チェック処理(R_mmc_Chk_Detect)

項目	内容
プロトタイプ	int16_t R_mmc_Chk_Detect(uint8_t SlotNo, uint8_t *pDetSts)
引数	uint8_t SlotNo : スロット番号 uint8_t *pDetSts : MMC 挿入状態格納先バッファポインタ
説明	引数で指定されたスロットに対して MMC 挿入状態のチェックを行います。 戻り値が MMC_OK の場合、MMC 挿入状態格納先バッファ(pDetSts)には MMC 挿入検出端子状態が格納されます。 <ul style="list-style-type: none"> • MMC_TRUE : MMC 挿入検出端子 Active • MMC_FALSE : MMC 挿入検出端子 Non Active この処理内でチャタリングの除去は行いません。 上位にて必要なチャタリングの除去を行ってください。 定周期でのポーリングによるメディアの挿入確認を推奨します。
戻り値	チェック結果を返します。 MMC_OK : 正常終了 MMC_ERR_PARAM : パラメータエラー

3.2.5 データ読み出し処理(R_mmc_Read_Data)

項目	内容
プロトタイプ	int16_t R_mmc_Read_Data(uint8_t SlotNo, uint32_t BlkNo, uint32_t BlkCnt, uint8_t *pData, uint8_t Mode)
引数	uint8_t SlotNo : スロット番号 uint32_t BlkNo : 読み出し開始ブロック番号 uint32_t BlkCnt : 読み出しブロック数 uint8_t *pData : 読み出しデータ格納バッファポインタ uint8_t Mode : 読み出しデータ転送モード
説明	MMC からブロック(512byte)単位でデータの読み出しを行います。 指定ブロックから指定ブロック数分、データを読み出します。 転送モード(Mode)は MMC_MODE_NORMAL(データを引数のデータ格納バッファ(pData)に転送するモード)を選択してください。 MMC からの読み出しは、R_mmc_Get_MmcInfo()関数から渡される MMC 情報のうち、カード種別(MmcInfo.Card)が 'MMC_CARD_UNDETECT' でない場合のみ可能です。 最大ブロック番号は、R_mmc_Get_MmcInfo()関数から渡される 'pMmcInfo.MaxBlkNum' です。 最大ブロック数は、'pMmcInfo.MaxBlkNum'+1 です。
戻り値	読み出し結果を返します。 MMC_OK : 正常終了 MMC_ERR_PARAM : パラメータエラー MMC_ERR_HARD : ハードウェアエラー MMC_ERR_CRC : CRC エラー MMC_ERR_OTHER : その他エラー

3.2.6 データ書き込み処理(R_mmc_Write_Data)

項目	内容
プロトタイプ	int16_t R_mmc_Write_Data(uint8_t SlotNo, uint32_t BlkNo, uint32_t BlkCnt, uint8_t *pData, uint8_t Mode)
引数	uint8_t SlotNo : スロット番号 uint32_t BlkNo : 書き込み開始ブロック番号 uint32_t BlkCnt : 書き込みブロック数 uint8_t *pData : 書き込みデータ格納バッファポインタ uint8_t Mode : 書き込みデータ転送モード
説明	MMC ヘブロック(512byte)単位でデータの書き込みを行います。 指定ブロックから指定ブロック数分、データを書き込みます。 転送モード(Mode)は MMC_MODE_NORMAL(データを引数のデータ格納バッファ(pData)から転送するモード)を選択してください。 MMC への書き込みは、R_mmc_Get_MmcInfo()関数から渡される MMC 情報のうち、カード種別(MmcInfo.Card)が 'MMC_CARD_UNDETECT' でない場合のみ可能です。 最大ブロック番号は、R_mmc_Get_MmcInfo()関数から渡される 'pMmcInfo.MaxBlkNum' です。 最大ブロック数は、'pMmcInfo.MaxBlkNum'+1 です。
戻り値	MMC 情報取得結果を返します。 MMC_OK : 正常終了 MMC_ERR_PARAM : パラメータエラー MMC_ERR_HARD : ハードウェアエラー MMC_ERR_WP : ライトプロテクトエラー MMC_ERR_OTHER : その他エラー

3.2.7 MMC 情報取得処理(R_mmc_Get_MmcInfo)

項目	内容
プロトタイプ	int16_t R_mmc_Get_MmcInfo(uint8_t SlotNo, MMC_INFO* pMmcInfo)
引数	uint8_t SlotNo : スロット番号 MMC_INFO* pMmcInfo : MMC 情報格納バッファポインタ
説明	<p>MMC 情報を返します。 MMC 情報格納バッファ(pMmcInfo)には MMC 情報が格納されます。</p> <ul style="list-style-type: none"> • pMmcInfo.Card : カード種別 <ul style="list-style-type: none"> — MMC_CARD_UNDETECT : カード未検出 — MMC_CARD_MMC : MMC — MMC_CARD_OTHER : その他カード pMmcInfo.WProtect : ライトプロテクト状態 <ul style="list-style-type: none"> — MMC_NO_PROTECT : ライトプロテクト解除 — MMC_W_PROTECT_SOFT : ソフトライトプロテクト • pMmcInfo.MemSize : カード容量(byte) • pMmcInfo.MaxBlkNum : 最大ブロック番号(メディアの最大ブロック番号) <p>'pMmcInfo.MemSize'が 0xFFFFFFFF の場合は、'pMmcInfo.MaxBlkNum'+1 が、メディアの持つブロック数 (1 ブロック=512 バイト) を示します。 カード容量の計算が必要な場合、('pMmcInfo.MaxBlkNum'+1) × 512 で計算してください。</p>
戻り値	書き込み結果を返します。 MMC_OK : 正常終了 MMC_ERR_PARAM : パラメータエラー MMC_ERR_OTHER : その他エラー

3.3 データ構造体

データ構造体を以下に示します。

MMC 情報構造体定義

```
typedef struct {
    uint8_t    Card;           /* Card type                */
    uint8_t    WProtect;      /* Write-protection status  */
    uint32_t   MemSize;       /* Card capacity            */
    uint32_t   MaxBlkNum;     /* The number of the max blocks */
} MMC_INFO;                 /* total 12byte            */
```

3.4 マクロ定義

マクロ定義を以下に示します。

```
/*----- Definitions of return value -----*/
#define MMC_OK                (int16_t)( 0)          /* Successful operation     */
#define MMC_ERR_PARAM        (int16_t)(-1)          /* Parameter error         */
#define MMC_ERR_HARD         (int16_t)(-2)          /* Hardware error          */
#define MMC_ERR_CRC          (int16_t)(-3)          /* CRC error               */
#define MMC_ERR_WP           (int16_t)(-4)          /* Write-protection error  */
#define MMC_ERR_MBLKCMD      (int16_t)(-5)          /* Multi-block command error */
#define MMC_ERR_IDLE         (int16_t)(-6)          /* Idle state error        */
#define MMC_ERR_OTHER        (int16_t)(-7)          /* Other error             */

/*----- Definitions of flag -----*/
#define MMC_TRUE              (uint8_t)0x01          /* Flag "ON"               */
#define MMC_FALSE             (uint8_t)0x00          /* Flag "OFF"              */

/*----- Definition of card type -----*/
#define MMC_CARD_UNDETECT    (uint8_t)0x00          /* Card is not found       */
#define MMC_CARD_MMC         (uint8_t)0x01          /* MMC                     */
#define MMC_CARD_OTHER       (uint8_t)0xFF          /* Other card              */

/*----- Definitions of write-protection status -----*/
#define MMC_NO_PROTECT       (uint8_t)0x00          /* None setting            */
#define MMC_W_PROTECT_SOFT  (uint8_t)0x02          /* Software write-protection*/
```

4. 設定例

4.1 共通関数 r_mtl_XXX の可変データ設定例

各システムのリソースに合わせて設定をする部分です。

設定箇所は、各ファイル中の「`/** SET **/`」というコメントの部分です。

ファイル毎に抜粋を示し、詳細な解説を加えます。

4.1.1 r_mtl_com.h

共通で使用される共通関数のヘッダです。

r_mtl_com.h は MCU およびシステムクロック別に個々に用意しています。

ご使用の環境に合わせて下記の表に示すディレクトリにある r_mtl_com.h をインクルードしてください。

異なる MCU やシステムクロックで動作させる場合は、ユーザでご用意ください。

使用 MCU – システムクロック	インクルードディレクトリ(drv ディレクトリ以降)
RX600 シリーズ - 100MHz	¥com¥rx600_100MHz
RX600 シリーズ - 96MHz	¥com¥rx600_96MHz
RX200 シリーズ - 50MHz	¥com¥rx200_50MHz

(1) 共通アクセス領域(SFR)を定義したヘッダファイル定義

- MCU の機能レジスタの定義がされているヘッダファイルをインクルードします。
主にドライバがポート制御等に使用するため、インクルードする必要があります。
MMC ドライバ使用時は、インクルードしてください。

```

/* In order to use definitions of MCU SFR area,                */
/* include the header file of MCU SFR definition.              */
#include "iodefine.h"          /* definition of MCU SFR      */
*/ /** SET **/
*/ /** SET **/
*/ /** SET **/

```

(2) ループタイマの定義

- ソフトウェア・ループタイマを使用する場合、r_mtl_tim.h をインクルードします。
主にドライバが待ち時間を確保するために、使用します。
ソフトウェア・ループタイマを使用しない場合は、下記インクルードをコメントにしてください。
下記の例は、ソフトウェア・ループタイマを使用する場合の例です。
本ドライバ使用時は、インクルードしてください。また、r_mtl_tim.h からシステムクロックに合ったマクロを定義してください。RX600 シリーズの MCU を 100MHz で動作する場合は
"MTL_TIM_RX600__12_5MHz_noWait_Ix8"を定義します。

```

/* When not using the loop timer, put the following 'include' as comments. */
#define MTL_TIM_RX600__12_5MHz_noWait_Ix8

#include "r_mtl_tim.h"

```

(3) エンディアンタイプ定義

— リトルエンディアン/ビッグエンディアンのどちらかの指定を行います。

RX ファミリの場合は、コンパイラオプションのエンディアンの設定にて指定してください。

```
#if ( (defined(__LIT)) || (!defined(__BIG)) )
#define MTL_MCU_LITTLE          /* Little Endian          */ /** SET **/
#endif
```

(4) 使用する標準ライブラリのタイプの定義

— 使用する標準ライブラリのタイプを定義してください。

下記に示す処理を標準ライブラリで使用する場合は、下記マクロ定義をコメントにしてください。

下記の例は、コンパイラ添付のライブラリを使用しない場合の例です。

```
/* Specify the type of user standard library.                */ /** SET **/
/* When using the compiler-bundled library for the following processes, */ /** SET **/
/* put the following 'define' as comments.                   */ /** SET **/
/* memcmp()/memmove()/memcpy()/memset()/strcat()/strcmp()/strcpy()/strlen() */ /** SET **/
#define MTL_USER_LIB          /* use optimized library          */ /** SET **/
```

4.1.2 r_mtl_tim.h

r_mtl_com.h にて、ループタイマを定義した場合に、インクルードされます。

使用する MCU やクロック、ウェイトに依存します。

環境に合うものがない場合は、ユーザにて定義を作成してください。

```
/* Define the counter value for the timer. */
/* Specify according to the user MCU, clock and wait requirements. */
/*
/* Set the reference value to 10% more than the actual calculated value. */
/*=====*/

/*=====*/
#ifdef MTL_TIM_RX600__12_5MHz_noWait_Ix8
/* Setting for 12.5MHz no wait Ix8 = 100MHz(Compile Option "-optimize=2 -size")*/
#define MTL_T_250NS      5      /* loop Number of 250ns */ /** SET **/
#define MTL_T_500NS     11     /* loop Number of 500ns */ /** SET **/
#define MTL_T_1US       24     /* loop Number of 1us   */ /** SET **/
#define MTL_T_2US       49     /* loop Number of 2us   */ /** SET **/
#define MTL_T_4US       99     /* loop Number of 4us   */ /** SET **/
#define MTL_T_5US      124     /* loop Number of 5us   */ /** SET **/
#define MTL_T_10US     249     /* loop Number of 10us  */ /** SET **/
#define MTL_T_20US     499     /* loop Number of 20us  */ /** SET **/
#define MTL_T_30US     749     /* loop Number of 30us  */ /** SET **/
#define MTL_T_50US    1249     /* loop Number of 50us  */ /** SET **/
#define MTL_T_100US   2499     /* loop Number of 100us */ /** SET **/
#define MTL_T_200US   4999     /* loop Number of 200us */ /** SET **/
#define MTL_T_300US   7499     /* loop Number of 300us */ /** SET **/
#define MTL_T_400US   9999     /* loop Number of 400us */ /** SET **/
#define MTL_T_1MS    24999     /* loop Number of 1ms   */ /** SET **/
#endif

#ifdef MTL_TIM_RX600__12_0MHz_noWait_Ix8
/* Setting for 12.0MHz no wait Ix8 = 96MHz(Compile Option "-optimize=2 -size")*/
(省略)
#endif

#ifdef MTL_TIM_RX200__50_0MHz_noWait
/* Setting for 50.0MHz no wait (Compile Option "-optimize=2 -size")*/
(省略)
#endif
```

4.2 MMC ドライバ可変データ設定例

各システムのリソースに合わせて設定をする部分です。

設定箇所は、各ファイル中の「`/** SET **/`」というコメントの部分です。

ファイル毎に抜粋を示し、詳細な解説を加えます。

4.2.1 r_mmc.h(ドライバ共通定義)

(1) デバイス数、デバイス番号の定義

— メモリカードのスロット数を定義してください。

```

/* Define number of required card slots. (1-N slots) */
/* Define slot number in accordance with the number of card slots to be connected. */
/*-----*/
/* Define number of slots (devices). */
#define MMC_SLOT_NUM      1                /* 1slots                */ /** SET **/

/* Define slot number. */
#define MMC_SLOT0        0                /* Slot 0                */ /** SET **/
#define MMC_SLOT1        1                /* Slot 1                */ /** SET **/

```

(2) SPI モード マルチブロックコマンド非対応カードの場合の定義

— デフォルトでは、`MULTIPLE_BLOCK` コマンド非対応カードの場合は、`READ_SINGLE_BLOCK` コマンドと `WRITE_SINGLE_BLOCK` コマンドで対応する設定のため、このままの設定を推奨します。

```

/* When use the card which does not support a multi-block command, please define it.*/
/* Use single block commands in the case of the card which does not support multiple block*/
/* commands. */
#define MMC_SBLK_CMD      /* Support single block commands */ /** SET **/

```

(3) 対象メディアの定義

— `MMC_SUPPORT_MMC` を定義してください。

```

/*-----*/
/* Please define the media to support. */
/*-----*/
#define MMC_SUPPORT_MMC      /* MMC                */ /** SET **/

```

4.2.2 r_mmc_user_config.h(MCU 個別定義)

(1) 使用 MCU および通信モジュールの選択

r_mmc_user_config.h は MCU および通信モジュール別に個々に用意しています。

ご使用の環境に合わせて下記の表に示すディレクトリにある mmc_user_config.h をインクルードしてください。

使用 MCU - 通信モジュール	インクルードディレクトリ(drv ディレクトリ以降)
RX610 - SCI	¥mmc¥rx¥rx610_sci
RX62N - SCI	¥mmc¥rx¥rx62n_sci
RX62N - RSPI	¥mmc¥rx¥rx62n_rsipi
RX210 - SCI	¥mmc¥rx¥rx210_sci
RX210 - RSPI	¥mmc¥rx¥rx210_rsipi
RX630 - SCI	¥mmc¥rx¥rx630_sci
RX630 - RSPI	¥mmc¥rx¥rx630_rsipi
RX63N - SCI	¥mmc¥rx¥rx63n_sci
RX63N - RSPI	¥mmc¥rx¥rx63n_rsipi

(2) MCU のピン数による定義

— RX62N-SCI を使用する場合:

- 100pin または 85pin パッケージを使用する場合、MMC_SCI_PACKAGE_100PIN_OR_LESS マクロを定義して下さい。

— RX62N-RSPI を使用する場合:

- 85pin パッケージを使用する場合、MMC_RSPI_PACKAGE_85PIN マクロを定義して下さい。

[RX62N-SCI 使用時]

```
/* When Use 100pin or 85pin MCU package, MMC_SCI_PACKAGE_100PIN_OR_LESS macro must
define.*/
//#define MMC_SCI_PACKAGE_100PIN_OR_LESS                                /** SET **/
```

[RX62N-RSPI 使用時]

```
/* When Use 85pin MCU package, MMC_RSPI_PACKAGE_85PIN macro must define.*/
//#define MMC_RSPI_PACKAGE_85PIN                                       /** SET **/
```


(3) 通信モジュールのチャネル番号および端子選択の定義

MMC_SCI_CHANNEL マクロ(SCI 使用時)、または MMC_RSPI_CHANNEL マクロ(RSPI 使用時)で使用する通信モジュールのチャネル番号を定義してください。

— RX62N の SCI を使用する場合:

- 定義したチャネル番号によっては使用する端子を定義する必要があります。対応表を参照して MMC_SCI_PIN マクロに定義する値を定義してください。

— RX62N の RSPI を使用する場合:

- 対応表を参照して MMC_RSPI_PIN マクロに値を定義してください。

— RX210、RX630、RX63N の SCI を使用する場合:

- 定義したチャネル番号によっては使用する端子を定義する必要があります。対応表を参照して MMC_SCI_RXD_PIN マクロ(RxD 端子)、MMC_SCI_SCK_PIN マクロ(SCK 端子)、MMC_SCI_TXD_PIN マクロ(TxD 端子)に値を定義してください。

— RX210、RX630、RX63N の RSPI を使用する場合:

- 定義したチャネル番号によっては使用する端子を定義する必要があります。対応表を参照して MMC_RSPI_RXD_PIN マクロ(MISO 端子)、MMC_RSPI_SCK_PIN マクロ(RSPCK 端子)、MMC_RSPI_TXD_PIN マクロ(MOSI 端子)に値を定義してください。

[RX610- SCI 使用時]

```

/* SCI Channel Select (0 or 1 or 2 or 3 or 4 or 5 or 6) */
/*
    ||      Select Port      |
+-----+-----+-----+
MMC_SCI_ ||  RxD  |  SCK  |  TxD  |
CHANNEL ||  Select |  Select |  Select |
Value   ||  port  |  port  |  port  |
+-----+-----+-----+
    0    || P21  | P22  | P20  |
(=Use SCI0) ||      |      |      |
+-----+-----+-----+
    1    || P25  | P27  | P26  |
(=Use SCI1) ||      |      |      |
+-----+-----+-----+
    2    || P12  | P11  | P13  |
(=Use SCI2) ||      |      |      |
+-----+-----+-----+
    3    || P16  | P15  | P17  |
(=Use SCI3) ||      |      |      |
+-----+-----+-----+
    4    || P05  | P03  | P04  |
(=Use SCI4) ||      |      |      |
+-----+-----+-----+
    5    || PC6  | PC5  | PC7  |
(=Use SCI5) ||      |      |      |
+-----+-----+-----+
    6    || P01  | P02  | P00  |
(=Use SCI6) ||      |      |      |
+-----+-----+-----+*/
#define MMC_SCI_CHANNEL      1                               /** SET **/

```

[RX62N-SCI 使用時]

```

/* SCI Channel Select (0 or 1 or 2 or 3 or 5 or 6) */
#define MMC_SCI_CHANNEL      2                /** SET **/

/* When MMC_SCI_CHANNEL defines 1 or 2 or 3 or 6 ,
   MMC_SCI_PIN must define 'A' or 'B' or 'C'. */
/*


| MMC_SCI_         | MMC_SCI_        | Select Port |        |        |
|------------------|-----------------|-------------|--------|--------|
| CHANNEL          | PIN             | RxD         | SCK    | TxD    |
| Value            | Value           | Select      | Select | Select |
|                  |                 | port        | port   | port   |
| 0<br>(=Use SCI0) | (Don't<br>care) | P21         | P22    | P20    |
| 1<br>(=Use SCI1) | 'A'             | P30         | P27    | P26    |
|                  | 'B'             | PF2         | PF1    | PF0    |
| 2<br>(=Use SCI2) | 'A'             | P12         | P11    | P13    |
|                  | 'B'             | P52         | P51    | P50    |
|                  | 'C'             | P12         | P51    | P13    |
| 3<br>(=Use SCI3) | 'A'             | P16         | P15    | P17    |
|                  | 'B'             | P25         | P24    | P23    |
| 5<br>(=Use SCI5) | (Don't<br>care) | PC2         | PC1    | PC3    |
| 6<br>(=Use SCI6) | 'A'             | P01         | P02    | P00    |
|                  | 'B'             | P33         | P34    | P32    |


*/
#define MMC_SCI_PIN          'A'                /** SET **/

```

[RX62N-RSPI 使用時]

```
/* RSPI Channel Select (0 or 1)*/
```

```
#define MMC_RSPI_CHANNEL 1
```

```
/** SET **/
```

```
/* RSPI PIN select ( 'A' or 'B' ) */
```

```
/*
```

		Select Port		
MMC_SCI_ CHANNEL Value	MMC_RSPI_ PIN Value	RxD Select port	SCK Select port	TxD Select port
0 (=Use RSPI0)	'A'	PC7	PC5	PC6
	'B'	PA7	PA5	PA6
1 (=Use RSPI1)	'A'	P30	P27	P26
	'B'	PE7	PE5	PE6

```
#define MMC_RSPI_PIN 'B'
```

```
/** SET **/
```

[RX210-SCI 使用時]

```

/* SCI Channel Select (0 or 1 or 5 or 6 or 8 or 9 or 12) */
#define MMC_SCI_CHANNEL      9                               /** SET **/

/* When MMC_SCI_CHANNEL macro defines 1 or 5 or 6 ,
   MMC_SCI_SCK_PIN and MMC_SCI_RXD_PIN and MMC_SCI_TXD_PIN macro must define. */
/*
Select  ||                               Select Port                               |
Channel +-+-----+-----+-----+-----+-----+-----+-----+-----+
        ||           RxD           ||           SCK           ||           TxD           ||
-----+-----+-----+-----+-----+-----+-----+-----+
MMC_SCI_ || MMC_SCI_ |           || MMC_SCI_ |           || MMC_SCI_ |           ||
CHANNEL || RXD_PIN  | Select  || SCK_PIN  | Select  || TXD_PIN  | Select  ||
Value   || Value   | port   || Value   | port   || Value   | port   ||
-----+-----+-----+-----+-----+-----+-----+-----+
0       || (Don't  | P21   || (Don't  | P22   || (Don't  | P20   ||
(=Use SCI0) || care) |      || care) |      || care) |      ||
-----+-----+-----+-----+-----+-----+-----+-----+
1       || 'A'    | P15   || 'A'    | P17   || 'A'    | P16   ||
(=Use SCI1) || 'B'    | P30   || 'B'    | P27   || 'B'    | P26   ||
-----+-----+-----+-----+-----+-----+-----+-----+
5       || 'A'    | PA2   || 'A'    | PA1   || 'A'    | PA4   ||
(=Use SCI5) || 'B'    | PA3   || 'B'    | PC1   || 'B'    | PC3   ||
        || 'C'    | PC2   || 'C'    | PC4   || -      | -     ||
-----+-----+-----+-----+-----+-----+-----+-----+
6       || 'A'    | P33   || 'A'    | P34   || 'A'    | P32   ||
(=Use SCI6) || 'B'    | PB0   || 'B'    | PB3   || 'B'    | PB1   ||
-----+-----+-----+-----+-----+-----+-----+-----+
8       || (Don't  | PC6   || (Don't  | PC5   || (Don't  | PC7   ||
(=Use SCI8) || care) |      || care) |      || care) |      ||
-----+-----+-----+-----+-----+-----+-----+-----+
9       || (Don't  | PB6   || (Don't  | PB5   || (Don't  | PB7   ||
(=Use SCI9) || care) |      || care) |      || care) |      ||
-----+-----+-----+-----+-----+-----+-----+-----+
12      || (Don't  | PE2   || (Don't  | PE0   || (Don't  | PE1   ||
(=Use SCI12) || care) |      || care) |      || care) |      ||
-----+-----+-----+-----+-----+-----+-----+-----+
*/
#define MMC_SCI_RXD_PIN      'A'                               /** SET **/
#define MMC_SCI_SCK_PIN     'A'                               /** SET **/
#define MMC_SCI_TXD_PIN     'A'                               /** SET **/

```

[RX210-RSPI 使用時]

```

/* RSPI Channel Select ( 0 only ) */
#define MMC_RSPI_CHANNEL    0

/* RSPI PIN select ('A' or 'B' or 'C') */
/*
Select   ||                               Select Port                               |
Channel  +-+-----+-----+-----+-----+-----+-----+-----+-----+
          ||           MISO           ||           RSPCK           ||           MOSI           |
-----+-----+-----+-----+-----+-----+-----+-----+
MMC_RSPI_ || MMC_RSPI_ |           || MMC_RSPI_ |           || MMC_RSPI_ |           |
CHANNEL  || RXD_PIN  | Select  || SCK_PIN  | Select  || TXD_PIN  | Select  |
Value    || Value    | port    || Value    | port    || Value    | port    |
=====+=====+=====+=====+=====+=====+=====+=====+
0        || 'A'     | P17    || 'A'     | PA5    || 'A'     | P16    |
(=Use RSPI0) +-+-----+-----+-----+-----+-----+-----+-----+
          || 'B'     | PA7    || 'B'     | PB0    || 'B'     | PA6    |
          +-+-----+-----+-----+-----+-----+-----+-----+
          || 'C'     | PC7    || 'C'     | PC5    || 'C'     | PC6    |
-----+-----+-----+-----+-----+-----+-----+
#define MMC_RSPI_RXD_PIN      'B'                /** SET **/
#define MMC_RSPI_SCK_PIN     'A'                /** SET **/
#define MMC_RSPI_TXD_PIN     'B'                /** SET **/

```

[RX630-SCI 使用時]

/* SCI Channel Select (from 0 to 12) */

#define MMC_SCI_CHANNEL 1

/** SET **/

/* When MMC_SCI_CHANNEL defines 0 or 1 or 2 or 3 or 4 or 5 or 6 or 9,
MMC_SCI_SCK_PIN and MMC_SCI_RXD_PIN and MMC_SCI_TXD_PIN must define. */

/*

Select Channel	Select Port					
	RxD		SCK		TxD	
MMC_SCI_CHANNEL Value	MMC_SCI_RXD_PIN Value	Select port	MMC_SCI_SCK_PIN Value	Select port	MMC_SCI_TXD_PIN Value	Select port
0	'A'	P21	'A'	P22	'A'	P20
(=Use SCI0)	'B'	P33	'B'	P34	'B'	P32
1	'A'	P15	'A'	P17	'A'	P16
(=Use SCI1)	'B'	P30	'B'	P27	'B'	P26
	'C'	PF2	'C'	PF1	'C'	PF0
2	'A'	P12	'A'	P11	'A'	P13
(=Use SCI2)	'B'	P52	'B'	P51	'B'	P50
3	'A'	P16	'A'	P15	'A'	P17
(=Use SCI3)	'B'	P25	'B'	P24	'B'	P23
4	'A'	PB0	'A'	PB3	'A'	PB1
(=Use SCI4)	'B'	PK4	'B'	P70	'B'	PK5
5	'A'	PA2	'A'	PA1	'A'	PA4
(=Use SCI5)	'B'	PA3	'B'	PC1	'B'	PC3
	'C'	PC2	'C'	PC4	-	-
6	'A'	P33	'A'	P34	'A'	P32
(=Use SCI6)	'B'	PB0	'B'	PB3	'B'	PB1
	'C'	P01	'C'	P02	'C'	P00
7	(Don't care)	P92	(Don't care)	P91	(Don't care)	P90
(=Use SCI7)						
8	(Don't care)	PC6	(Don't care)	PC5	(Don't care)	PC7
(=Use SCI8)						

9	'A'	PB6	'A'	PB5	'A'	PB7	
(=Use SCI9)	'B'	PK3	'B'	P60	'B'	PK2	
10	(Don't care)	P81	(Don't care)	P80	(Don't care)	P82	
(=Use SCI10)	(Don't care)	P76	(Don't care)	P75	(Don't care)	P77	
11	(Don't care)	PE2	(Don't care)	PE0	(Don't care)	PE1	
(=Use SCI11)	(Don't care)	PE2	(Don't care)	PE0	(Don't care)	PE1	
12	(Don't care)	PE2	(Don't care)	PE0	(Don't care)	PE1	
(=Use SCI12)	(Don't care)	PE2	(Don't care)	PE0	(Don't care)	PE1	

*/

```
#define MMC_SCI_RXD_PIN      'C'          /** SET **/
#define MMC_SCI_SCK_PIN     'C'          /** SET **/
#define MMC_SCI_TXD_PIN     'C'          /** SET **/
```

[RX630-RSPI 使用時]

```

/* RSPI Channel Select ( 0 or 1 or 2 ) */
#define MMC_RSPI_CHANNEL1                                     /** SET **/

/* When MMC_RSPI_CHANNEL macro defines 0 or 1,
   MMC_RSPI_RXD_PIN and MMC_RSPI_SCK_PIN and MMC_RSPI_TXD_PIN macro must define.
*/
/*
Select   ||                               Select Port                               |
Channel  +-+-----+-----+-----+-----+-----+-----+-----+-----+
          ||           MISO           ||           RSPCK           ||           MOSI           |
-----+-----+-----+-----+-----+-----+-----+-----+
MMC_RSPI_ || MMC_RSPI_ |           || MMC_RSPI_ |           || MMC_RSPI_ |           |
CHANNEL  || RXD_PIN  | Select   || SCK_PIN  | Select   || TXD_PIN  | Select   |
Value    || Value    | port     || Value    | port     || Value    | port     |
-----+-----+-----+-----+-----+-----+-----+-----+
0        || 'A'      | P17      || 'A'      | PA5      || 'A'      | P16      |
(=Use RSPI0) +-+-----+-----+-----+-----+-----+-----+-----+
          || 'B'      | PA7      || 'B'      | PB0      || 'B'      | PA6      |
          +-+-----+-----+-----+-----+-----+-----+-----+
          || 'C'      | PC7      || 'C'      | PC5      || 'C'      | PC6      |
-----+-----+-----+-----+-----+-----+-----+-----+
1        || 'A'      | P30      || 'A'      | P27      || 'A'      | P26      |
(=Use RSPI1) +-+-----+-----+-----+-----+-----+-----+-----+
          || 'B'      | PE3      || 'B'      | PE1      || 'B'      | PE2      |
          +-+-----+-----+-----+-----+-----+-----+-----+
          || 'C'      | PE7      || 'C'      | PE5      || 'C'      | PE6      |
-----+-----+-----+-----+-----+-----+-----+-----+
2        || (Don't   | PD2      || (Don't   | PD3      || (Don't   | PD1      |
(=Use RSPI2) +-+-----+-----+-----+-----+-----+-----+-----+
          || care)    |          || care)    |          || care)    |          |
-----+-----+-----+-----+-----+-----+-----+-----+
#define MMC_RSPI_RXD_PIN           'C'                                     /** SET **/
#define MMC_RSPI_SCK_PIN           'C'                                     /** SET **/
#define MMC_RSPI_TXD_PIN           'C'                                     /** SET **/

```


[RX63N-SCI 使用時]

/* SCI Channel Select (from 0 to 12) */

#define MMC_SCI_CHANNEL 1

/** SET **/

```
/* When MMC_SCI_CHANNEL defines 0 or 1 or 2 or 3 or 5 or 6 ,
   MMC_SCI_SCK_PIN and MMC_SCI_RXD_PIN and MMC_SCI_TXD_PIN must define. */
/*
```

Select Channel	Select Port					
	RxD		SCK		TxD	
MMC_SCI_CHANNEL Value	MMC_SCI_RXD_PIN Value	Select port	MMC_SCI_SCK_PIN Value	Select port	MMC_SCI_TXD_PIN Value	Select port
0 (=Use SCI0)	'A'	P21	'A'	P22	'A'	P20
	'B'	P33	'B'	P34	'B'	P32
1 (=Use SCI1)	'A'	P15	'A'	P17	'A'	P16
	'B'	P30	'B'	P27	'B'	P26
	'C'	PF2	'C'	PF1	'C'	PF0
2 (=Use SCI2)	'A'	P12	'A'	P11	'A'	P13
	'B'	P52	'B'	P51	'B'	P50
3 (=Use SCI3)	'A'	P16	'A'	P15	'A'	P17
	'B'	P25	'B'	P24	'B'	P23
4 (=Use SCI4)	(Don't care)	PB0	(Don't care)	PB3	(Don't care)	PB1
5 (=Use SCI5)	'A'	PA2	'A'	PA1	'A'	PA4
	'B'	PA3	'B'	PC1	'B'	PC3
	'C'	PC2	'C'	PC4	-	-
6 (=Use SCI6)	'A'	P33	'A'	P34	'A'	P32
	'B'	PB0	'B'	PB3	'B'	PB1
	'C'	P01	'C'	P02	'C'	P00
7 (=Use SCI7)	(Don't care)	P92	(Don't care)	P91	(Don't care)	P90
8 (=Use SCI8)	(Don't care)	PC6	(Don't care)	PC5	(Don't care)	PC7
9 (=Use SCI9)	(Don't care)	PB6	(Don't care)	PB5	(Don't care)	PB7

```

-----+-----+-----+-----+-----+-----+-----+
    10   || (Don't | P81   || (Don't | P80   || (Don't | P82   |
(=Use SCI10) || care) |      || care) |      || care) |      |
-----+-----+-----+-----+-----+
    11   || (Don't | P76   || (Don't | P75   || (Don't | P77   |
(=Use SCI11) || care) |      || care) |      || care) |      |
-----+-----+-----+-----+-----+
    12   || (Don't | PE2   || (Don't | PE0   || (Don't | PE1   |
(=Use SCI12) || care) |      || care) |      || care) |      |
-----+-----+-----+-----+-----+*/
#define MMC_SCI_RXD_PIN      'C'          /** SET **/
#define MMC_SCI_SCK_PIN      'C'          /** SET **/
#define MMC_SCI_TXD_PIN      'C'          /** SET **/

```

[RX63N-RSPI 使用時]

```

/* RSPI Channel Select ( 0 or 1 or 2 ) */
#define MMC_RSPI_CHANNEL 0                                /** SET **/

/* When MMC_RSPI_CHANNEL macro defines 0 or 1,
   MMC_RSPI_RXD_PIN and MMC_RSPI_SCK_PIN and MMC_RSPI_TXD_PIN macro must define. */
/*
Select   ||                               Select Port                               |
Channel  +-+-----+-----+-----+-----+-----+-----+-----+-----+
          ||           MISO           ||           RSPCK           ||           MOSI           |
-----+-----+-----+-----+-----+-----+-----+-----+
MMC_RSPI_ || MMC_RSPI_ |           || MMC_RSPI_ |           || MMC_RSPI_ |           |
CHANNEL   || RXD_PIN  | Select  || SCK_PIN  | Select  || TXD_PIN  | Select  |
Value     || Value   | port   || Value   | port   || Value   | port   |
-----+-----+-----+-----+-----+-----+-----+-----+
0         || 'A'     | P17    || 'A'     | PA5    || 'A'     | P16    |
(=Use RSPI0) +-+-----+-----+-----+-----+-----+-----+-----+
          || 'B'     | PA7    || 'B'     | PB0    || 'B'     | PA6    |
          || 'C'     | PC7    || 'C'     | PC5    || 'C'     | PC6    |
-----+-----+-----+-----+-----+-----+-----+-----+
1         || 'A'     | P30    || 'A'     | P27    || 'A'     | P26    |
(=Use RSPI1) +-+-----+-----+-----+-----+-----+-----+-----+
          || 'B'     | PE3    || 'B'     | PE1    || 'B'     | PE2    |
          || 'C'     | PE7    || 'C'     | PE5    || 'C'     | PE6    |
-----+-----+-----+-----+-----+-----+-----+-----+
2         || (Don't  | PD2    || (Don't  | PD3    || (Don't  | PD1    |
(=Use RSPI2) || care)  |        || care) + |        || care)  |        |
-----+-----+-----+-----+-----+-----+-----+-----+
#define MMC_RSPI_RXD_PIN          'C'                                /** SET **/
#define MMC_RSPI_SCK_PIN          'C'                                /** SET **/
#define MMC_RSPI_TXD_PIN          'C'                                /** SET **/

```

(3)使用制御ポートの定義

— DETECT(カード検出) 信号および CS(カードセレクト)信号を接続回路に合わせて定義してください。

```
/*-----*/
/* Define the control port.                               */
/*-----*/
#define MMC_CS0_PORTNO      7      /* CS0 Port No. */      /** SET **/
#define MMC_CS0_BITNO      0      /* CS0 Bit No. */      /** SET **/

#define MMC_DETECT0_PORTNO 7      /* DETECT0 Port No. */ /** SET **/
#define MMC_DETECT0_BITNO 1      /* DETECT0 Bit No.  */ /** SET **/

#if (MMC_SLOT_NUM > 1)
#define MMC_CS1_PORTNO      /* CS1 Port No. */      /** SET **/
#define MMC_CS1_BITNO      /* CS1 Bit No. */      /** SET **/

#define MMC_DETECT1_PORTNO /* DETECT1 Port No. */ /** SET **/
#define MMC_DETECT1_BITNO /* DETECT1 Bit No.  */ /** SET **/
#endif /* #if (MMC_SLOT_NUM > 1) */
```

(4) 通信タイムアウト検出処理の定義

— 通信中のタイムアウト検出処理を省略することが出来ます。

省略する場合、MMC_NOCHK_TIMEOUT マクロを定義してください。定義した場合は処理速度が向上するメリットがありますが、通信機能に異常が発生した場合にプログラムが停止する可能性があるデメリットがあります。

省略しない場合は、タイムアウト時間を設定してください。

- MMC_T_SCI_WAIT(SCI 使用時)、MMC_T_RSPI_WAIT(RSPI 使用時)で時間の単位を設定します。設定するマクロは r_mtl_tim.h から選択してください。
- MMC_SCI_TX_WAIT(SCI 使用時)、MMC_RSPI_TX_WAIT(RSPI 使用時)マクロでデータ送信時のタイムアウト時間を定義してください。
- MMC_SCI_RX_WAIT(SCI 使用時)、MMC_RSPI_RX_WAIT(RSPI 使用時)マクロでデータ受信時のタイムアウト時間を定義してください。
- 各タイムアウト時間マクロの設定値は (タイムアウト時間 / 単位)となります。

```

/*-----*/
/* Macro "MMC_NOCHK_TIMEOUT" omits detecting timeout during communication. */
/* If user omits detecting timeout, please define this macro. */
/* If this macro is defined, processing speed would be increased. */
/*-----*/
#define MMC_NOCHK_TIMEOUT /* No Check Communication Timeout */ /** SET **/

/*-----*/
/* If MMC_NOCHK_TIMEOUT would be not defined, please set timeout time.
/* MMC_T_SCI_WAIT is unit of measuring timeout.
/* Please select value from "r_mtl_tim.h"
/* Please set value of (timeout time/unit) to MMC_SCI_TX_WAIT(transmitting)
/* and MMC_SCI_RX_WAIT(receiving).
/*-----*/
#define MMC_T_SCI_WAIT (uint32_t)MTL_T_250NS
/* SCI transmit&receive completion waiting polling time */ /** SET **/

#define MMC_SCI_TX_WAIT (uint32_t)200000
/* SCI transmission completion waiting time 200000 * 250ns = 50ms *//** SET **/
#define MMC_SCI_RX_WAIT (uint32_t)200000
/* SCI receive completion waiting time 200000 * 250ns = 50ms*/ /** SET **/

```

(5) 使用リソースの定義

— MCU に応じて、最適化した MCU リソースの組み合わせを定義しています。

以下の定義のうち、1つ定義してください。

```

/*-----*/
/* Define the combination of the MCU's resources. */
/*-----*/
#define MMC_OPTION_1 /* SCI */ /** SET **/
//#define MMC_OPTION_2 /* SCI + CRC calculation circuit */ /** SET **/
//#define MMC_OPTION_3 /* SCI + DTC */ /** SET **/
//#define MMC_OPTION_4 /* SCI + CRC + DTC */ /** SET **/

```

(6) 通信ボーレートの定義

— 通信ボーレートを定義してください。

使用する SIO の定義を修正する場合は、各々の SIO レジスタに応じた SFR 設定が必要になりますので、MCU のデータシートを参照し、システムに応じた SFR 設定を行ってください。

特に、通信速度設定に関して、Identification mode/Data Transfer mode それぞれのカード仕様書の tODLY を満たすように設定する必要があります。

さらに、カード仕様書では、Identification mode では、 $tOD (100\text{KHz} \leq tOD \leq 400\text{KHz})$ 、Data Transfer mode では、 $tPP (0.1\text{MHz} \leq tPP \leq 20\text{MHz}(*))$ を満たすように設定する必要があります。

なお、tOD, tPP は、本システムにおいて、クロック周波数を示すことになります。

MCU により、サポート可能な SIO のクロック周波数が変わりますので、MCU のデータシートを参照してください。

- MMC_UBRG_IDENTIFICATION マクロには Identification mode 時のクロック設定を設定してください。
- SCI を使用する場合、MMC_UBRG_IDENTIFICATION マクロの値は BRR レジスタに設定されます。
- RSPI を使用する場合、MMC_UBRG_IDENTIFICATION マクロの値は SPBR レジスタに設定されます。
- MMC_UBRG_D_TRANSFER マクロには Transfer mode 時のクロック設定を定義してください。
- SCI を使用する場合、MMC_UBRG_D_TRANSFER マクロの値は BRR レジスタに設定されます。
- RSPI を使用する場合、MMC_UBRG_D_TRANSFER マクロの値は SPBR レジスタに設定されます。
- MMC_CLK_D_TRANSFER マクロには Transfer mode 時のクロック周波数を定義してください。

RX610 での例：

```

/*-----*/
/* Define the value of the bit rate register according to a communication baud rate.          */
/* Set the frequency of CLK to 6MHz or less.                                              */
/* The possible maximum transfer frequency of CLK is depends on hardware circuit         */
/* and MCU conditions.                                                                    */
/* Refer to MCU hardware manual/memory card specifications and specify the buad rate.     */
/* When operating card with SPI mode,                                                    */
/* specify the following two definitions of Identification mode and Data Transfer mode.     */
/* Specify the definition to meet tODLY of both Identification mode and Data Transfer mode. */
/* In addition, meet tOD (100KHz <= tOD <= 400KHz) at Identification mode                */
/* and tPP (0.1MHz <= tPP <= 20MHz ) at Data Transfer mode.                             */
/* The maximum frequency depends on MCU type.                                           */
/*                                                                                          */
/*BRR = (PCLK / (8 * 2 ^ (2n - 1) * B)) - 1
/*PCLK: Operating frequency [MHz]
/*B   : Bit rate [bit/s]
/*n   : Determined by the SMR settings shown in the following table.
/*
/*  CKS1 | CKS0 | n
/* -----+-----+-----
/*  0   |  0   |  0
/*  0   |  1   |  1
/*  1   |  0   |  2
/*  1   |  1   |  3
/*-----*/
/* PCLK = 50MHz, n=0 */
#define MMC_UBRG_IDENTIFICATION (uint8_t)0x1f /* BRR identification mode setting*/ /** SET **/
/*                                     +------+ 391KHz          */ /** SET **/
#define MMC_UBRG_D_TRANSFER (uint8_t)0x01 /* BRR data Transfer mode setting */ /** SET **/
/*                                     +------+ 6.25MHz         */ /** SET **/
#define MMC_CLK_D_TRANSFER (uint32_t)6250000 /* Data Transfer mode clock frequency *//** SET **/

```

5. MCU との接続方法と使用する MCU リソース

5.1 使用する MCU リソース

本プログラムは、以下の制御を行っています。

データの入出力を、クロック同期形シリアル I/O(内部クロックを使用)で、制御します。

クロック同期形シリアル I/O を割り当てる際には、高速動作させるため CMOS 出力可能な端子割り当てと CMOS 出力設定をしてください。

送信制御は、送信バッファの空きを検出して、割り込み使用せずに送信割り込み要求ビットを利用しています。したがって、割り込み関連を以下のように設定しています。

- 割り込み優先レベルを、レベル 0(割り込み禁止)に設定。
- MMC の CS#端子を MCU の Port に接続し、MCU 汎用ポート出力で制御する。

使用するリソース	RX610	RX62N	RX630	RX63N	RX210
DTC	○	○	○	○	○
CRC 演算回路	○	○	○	○	○
CS#用ポート 1本/Card	◎	◎	◎	◎	◎
カード検出用ポート 1本/Card	◎	◎	◎	◎	◎
電源制御用ポート 1本/Card	◎	◎	◎	◎	◎

◎ : 必須

○ : 使用することを推奨。(RX ファミリ固有のリソースを使う場合、高速化が可能)

したがって、SCI クロック同期モードまたは RSPI の機能を持つ RX ファミリ MCU であれば、動作可能です。

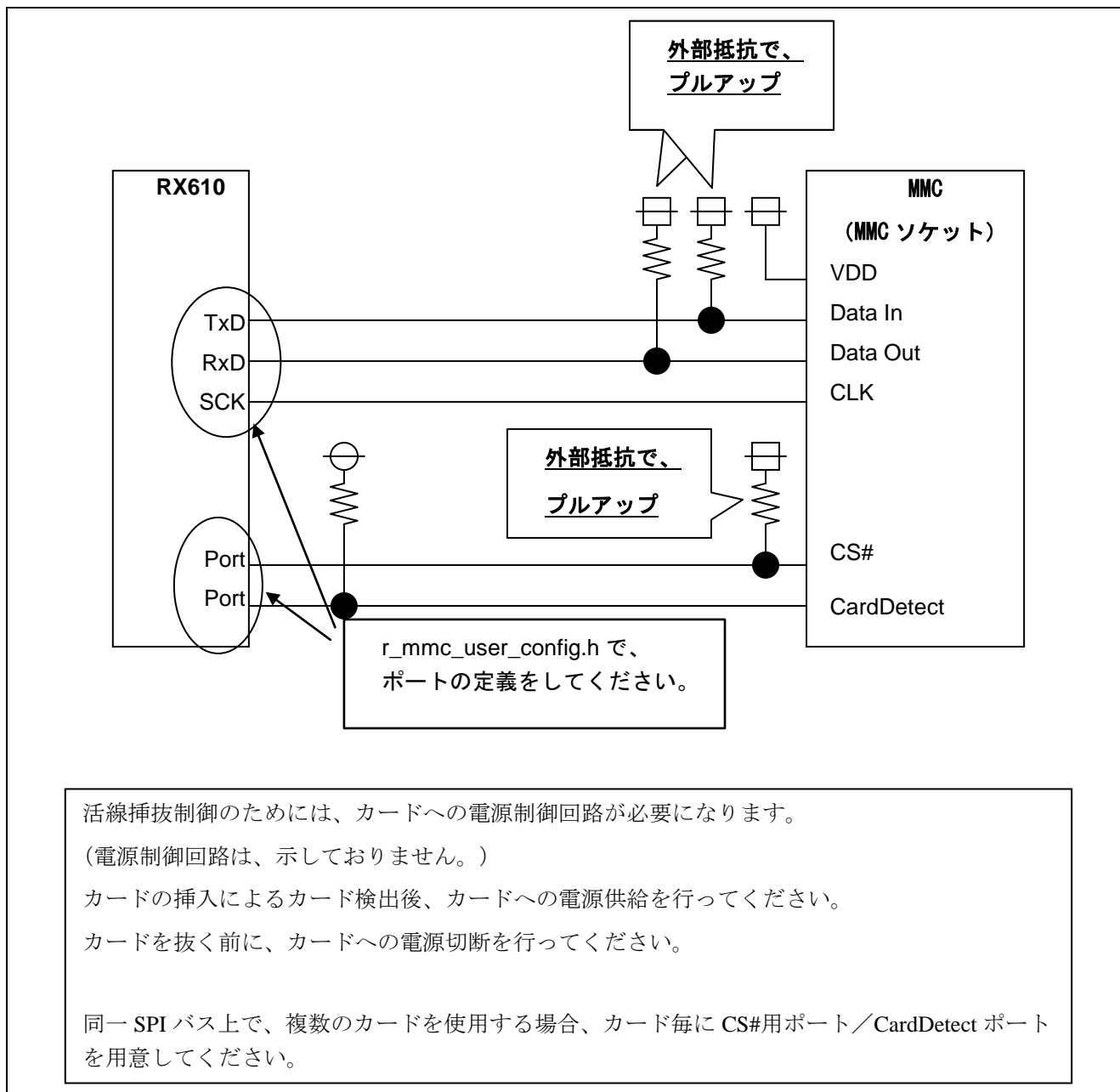
使用するリソース	RX610	RX62N	RX630	RX63N	RX210
SCI クロック同期	○	○	○	○	○
RSPI	-	○	○	○	○

○ : 使用可能

5.2 MCU との接続方法

RX610 との接続例を示します。

他 RX ファミリ MCU の場合であっても、同様の接続となります。



6. アプリケーション作成時の注意事項

6.1 使用上の注意事項

- 使用にあたっては、ハードウェアに合わせてソフトウェアを設定してください。
- メモリカードを非アクティブにし、MCU-メモリカード間の信号を Hi-z にし、デバイスへの電源供給を停止させた後、メモリカードを抜いてください。動作中にデバイスを抜いた場合、デバイスが壊れる可能性があります。
- 活線挿抜対応の回路を実現していない場合、活線挿抜を行うと電源等が不安定になり MCU がリセット状態になる可能性があります。

6.2 組み込み時の注意事項

6.2.1 開発環境

弊社の開発環境を以下に示します。

ユーザアプリケーション開発時は以下のバージョンより新しいものをご使用下さい。

[ソフトウェアツール]

統合開発環境

High Performance Embedded Workshop Version 4.09.00.007

C コンパイラ

C/C++ compiler package for RX family V.1.02 Release 01

[デバッグツール]

エミュレータデバッガ

RX E1/E20 Emulator Debugger

エミュレータソフトウェア

RX E1/E20 Emulator software V.1.03.00

[評価機材]

Renesas Starter Kit for RX610 (型名 : R0K556100S000BE)

Renesas Starter Kit+ for RX62N (型名 : R0K5562N0S000BE)

Renesas Starter Kit for RX210 (型名 : R0K505210S000BE)

Renesas Starter Kit for RX630 (型名 : R0K505630S000BE)

Renesas Starter Kit+ for RX63N (型名 : R0K50563NS000BE)

サンプルプログラムの動作確認には、上記環境に加え

「ミドルウェア評価ボード」またはそれに準ずるマルチメディアカードソケット拡張基板が必要です。

「ミドルウェア評価ボード」については以下アプリケーションノートをご参照ください。

- Renesas Starter Kit for RX610 ミドルウェア評価ボードの使用法(R21AN0004JJ0103)

6.2.2 インクルードするファイル

MMC ドライバを組み込む場合は、`r_mtl_com.h` と `r_mmc.h` をインクルードしてください。

`r_mtl_com.h` を先にインクルードする必要があります。

6.2.3 DTC を使用する場合の注意事項

r_mmc_user_config.h の設定で MMC_OPTION_3 マクロまたは MMC_OPTION_4 :を定義した場合は、DTC を使用してシリアル通信を行います。

- ・DTC を使用してカードにアクセスする場合、RAM 領域に BDTCTBL セクションを 4096 バイト境界に配置する必要があります。

6.2.4 RX610-SCI,または RX62N-SCI を使用する場合の注意事項

RX610-SCI,または RX62N-SCI の MMC ドライバ SFR 定義ヘッダファイルを組み込む場合、ROM 領域に CMMCBITCHG_1 セクションを 256 バイト境界に配置する必要があります。

6.2.5 チャネル選択および端子選択について

MMC ドライバは下記のパッケージに対応しています。

RX610 グループ : 176 ピン,144 ピンパッケージ

RX62N, RX621 グループ : 176 ピン 145 ピン/144 ピン,100 ピン,85 ピンパッケージ

RX210 グループ : 100 ピン,80 ピン,64 ピン,48 ピンパッケージ

RX630 グループ : 177/176 ピン,145/144 ピン, ,100 ピン,80 ピンパッケージ

RX63N, RX631 グループ : 177/176 ピン,145/144 ピン,100 ピン,64 ピン,48 ピンパッケージ

6.2.6 動作確認済みチャネルおよび端子選択

弊社が動作確認した設定チャネルおよび端子選択の内容を以下に示します。

使用 MCU - 通信モジュール	RX610 SCI	RX62N SCI	RX62N RSPI	RX630 SCI	RX630 RSPI
チャネル番号	1	2	1	9	0
SIO-RxD 端子のポート番号	P30	P12	PE7	PB6	PA7
SIO-CLK 端子のポート番号	P27	P11	PE5	PB5	PA5
SIO-TxD 端子のポート番号	P26	P13	PE6	PB7	PA6
CS 端子のポート番号	P70	P90	P90	PD1	PD1
CardDetect 端子のポート番号	P71	P91	P91	PD0	PD0

使用 MCU - 通信モジュール	RX63N SCI	RX63N RSPI	RX210 SCI	RX210 RSPI
チャネル番号	7	0	9	0
SIO-RxD 端子のポート番号	P92	PA7	PB6	PA7
SIO-CLK 端子のポート番号	P91	PA5	PB5	PA5
SIO-TxD 端子のポート番号	P90	PA6	PB7	PA6
CS 端子のポート番号	P97	P97	P50	P50
CardDetect 端子のポート番号	P96	P96	P51	P51

6.3 ROM/RAM/スタックサイズ

MMC ドライバが使用する ROM/RAM サイズおよびスタックサイズは以下のとおりです。

最適化レベル=2、サイズ優先時の値です。各サイズはシステムの設定により変化します。

ROM/RAM サイズ

SCI、RSPI 以外で 使用している リソース	分類(セクション)	サイズ [単位 : kByte]				
		RX610 (SCI)	RX62N (SCI)	RX62N (RSPI)	RX210 (SCI)	RX210 (RSPI)
-	ROM(セクション P,C, CBITCHGTBL_1)	6.6	6.6	7.0	6.4	7.0
	RAM(セクション B)	0.1	0.1	0.1	0.1	0.1
CRC	ROM(セクション P,C, CBITCHGTBL_1)	6.0	6.0	6.1	5.9	6.3
	RAM(セクション B)	0.1	0.1	0.1	0.1	0.1
DTC	ROM(セクション P,C, CBITCHGTBL_1)	7.0	7.0	7.2	6.8	7.4
	RAM(セクション B,BDTCTBL)	1.1	1.1	1.1	1.1	1.1
CRC + DTC	ROM(セクション P,C, CBITCHGTBL_1)	6.2	6.4	6.5	6.2	6.7
	RAM(セクション B,BDTCTBL)	1.1	1.1	1.1	1.1	1.1

SCI、RSPI 以外で 使用している リソース	分類(セクション)	サイズ [単位 : kByte]			
		RX630 (SCI)	RX630 (RSPI)	RX63N (SCI)	RX63N (RSPI)
-	ROM(セクション P,C, CBITCHGTBL_1)	6.3	7.0	6.3	7.0
	RAM(セクション B)	0.1	0.1	0.1	0.1
CRC	ROM(セクション P,C, CBITCHGTBL_1)	5.9	6.3	5.9	6.3
	RAM(セクション B)	0.1	0.1	0.1	0.1
DTC	ROM(セクション P,C, CBITCHGTBL_1)	6.8	7.4	6.8	7.4
	RAM(セクション B,BDTCTBL)	1.1	1.1	1.1	1.1
CRC + DTC	ROM(セクション P,C, CBITCHGTBL_1)	6.2	6.7	6.2	6.7
	RAM(セクション B,BDTCTBL)	1.1	1.1	1.1	1.1

スタックサイズ

関数名	RX610
R_mmc_Init_Driver	24
R_mmc_Init_Slot	128
R_mmc_Detach	24
R_mmc_Chk_Detect	12
R_mmc_Read_Data	136
R_mmc_Write_Data	144

6.4 カードの挿抜検出に関する注意事項

カードのコネクタにあるカード検出端子を使用することで、`R_mmc_Chk_Detect()`を使ってカード挿抜の検出が可能です。したがって、カード検出のために定周期でのポーリングによるカードの挿入確認を推奨しません。

また、通信中においてカードが抜かれた状態になった場合、コマンドに対するレスポンス異常となり、結果的にドライバがエラーを返します。

最もケアしなければならない項目として、通信中に一瞬挿抜が行われた場合が挙げられます。

以下のような場合、ドライバがエラーを返さない可能性があります。

- 周期内の抜挿はドライバでは検出できず、カードからレスポンス異常が無い場合は正常動作を行います。
- 書き込み中に一瞬抜挿すると、ドライバが書き込み完了と認識してしまう可能性があります。これは、書き込み Busy 信号解除は `DataIn` 端子の "H" で検出する仕様であるためです。(DataIn 端子はプルアップされています。)

ハードウェア割り込み制御やポーリング周期の見直し等でシステムに適した方法で解決してください。

6.5 ポートの割り当てとカードの挿抜に関するポートの Hi-z 化処理の注意事項

- カードの挿入においては、カードの `CS#`, `DataIn`, `DataOut`, `CLK` 信号を Hi-z 状態にし、カードを挿入してください。その後、カードへの電源を供給してください。
- カードの抜去においては、カードへの電源供給停止後、カードの `CS#`, `DataIn`, `DataOut`, `CLK` 信号を Hi-z 状態にしてからカードを抜いてください。
- カードの `CS#`, `DataIn`, `DataOut`, `CLK` は、MCU の `SIO`、ポート端子に割り当てられていますが、そのポートが他のリソースに割り当てられている場合が想定されますので、本ドライバでは Hi-z 処理を行っておりません。したがって、カードの挿抜においては上位側で MCU 端子の Hi-z 処理をお願いします。

7. サンプルプログラム

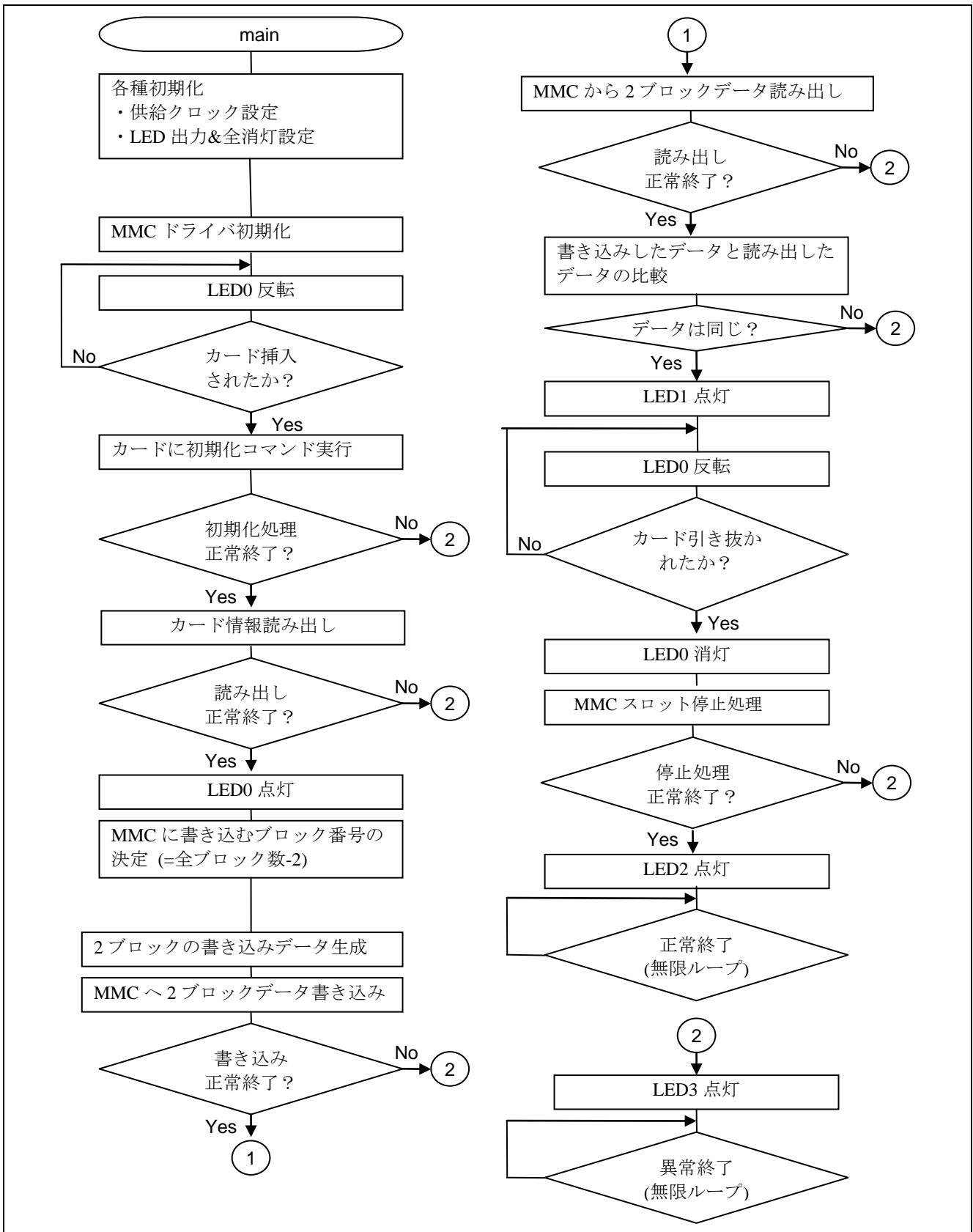
ここではサンプルプログラムについて説明します。

7.1 処理内容

- MMC を挿入すると MMC 内のメモリの最終 2 ブロックを使用してデータの読み書きと各データの比較を行います。
- 上記を最終ブロックまで実行した後、カードを抜いて動作終了です。
- プログラムの進行状況および結果を LED で表示します。表示の内容は下表を参照してください。

LED0	消灯 … カードが抜かれている 点灯 … カードが挿されている 点滅 … カードの引き抜きまたは挿入の要求
LED1	点灯 … カードへの読み書き処理終了
LED2	点灯 … プログラム正常終了
LED3	点灯 … プログラム異常終了

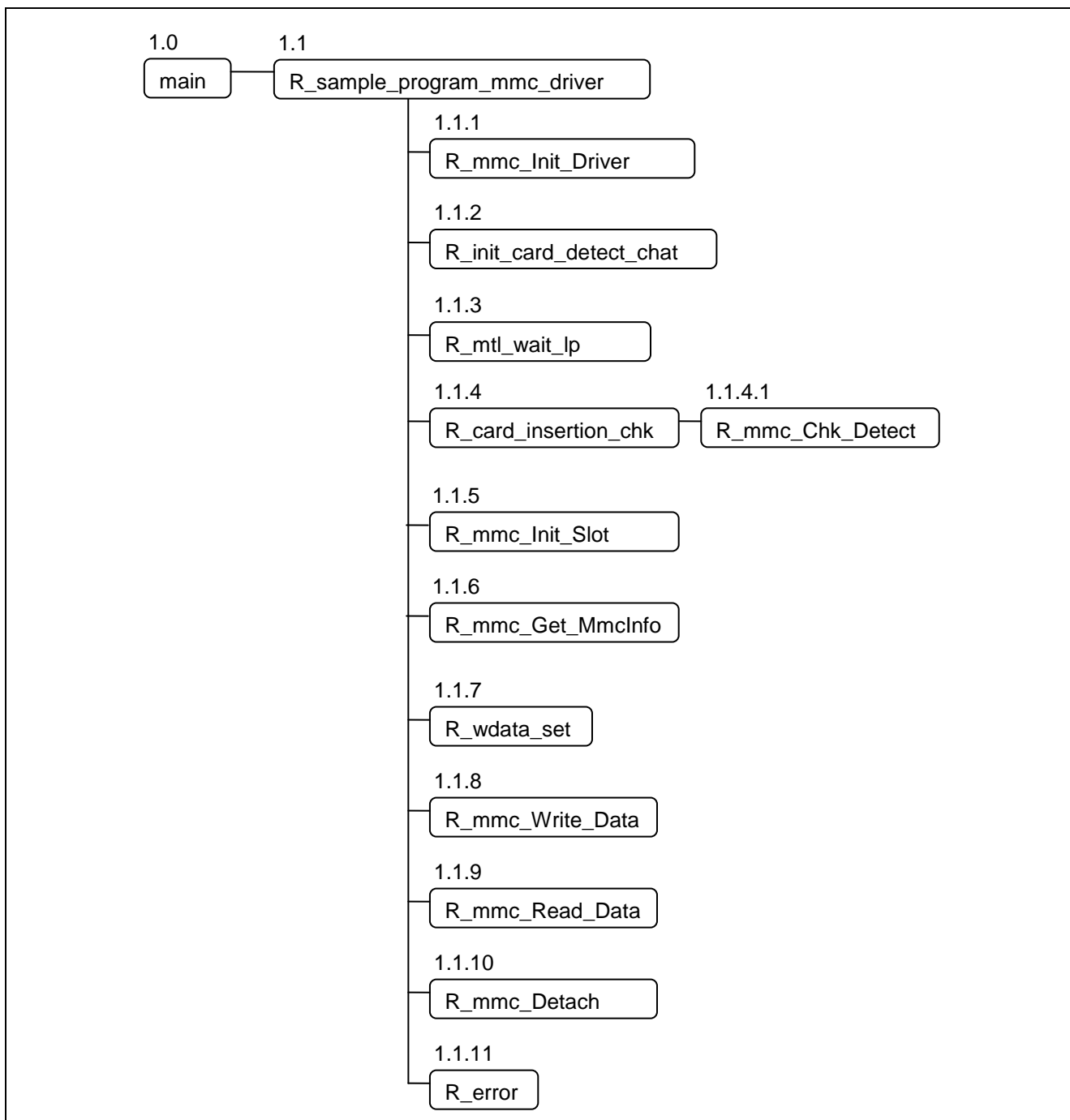
7.2 処理の流れ



7.3 関数一覧

No	関数名	概略
1.0	main	MCU のクロックおよび RSK の LED を初期化し、サンプルプログラムを実行します。
1.1	R_sample_program_mmc_driver	MMC が挿入されるとカードに対してランダムな値を読み書きし、それぞれのデータを比較します。
1.1.1	R_mmc_Init_Driver	MMC ドライバ処理の初期化をします。 - API 関数です。
1.1.2	R_init_card_detect_chat	カード挿抜検出用のメモリを初期化します。
1.1.3	R_mtl_wait_lp	次の端子確認のための時間待ち処理です。
1.1.4	R_card_insertion_chk	カードの挿抜をチェックします。
1.1.4.1	R_mmc_Chk_Detect	カードの挿抜状態を返します。 - API 関数です。
1.1.5	R_mmc_Init_Slot	カードが挿入された場合にカードに対して初期化処理を実行します。 - API 関数です。
1.1.6	R_mmc_Get_MmcInfo	カード情報を取得します。 - API 関数です。
1.1.7	R_wdata_set	MMC に書き込むデータを作成します。
1.1.8	R_mmc_Write_Data	MMC にデータを書き込みます。 - API 関数です。
1.1.9	R_mmc_Read_Data	MMC からデータを読み出します。 - API 関数です。
1.1.10	R_mmc_Detach	カードが引き抜かれた場合にスロットに対して停止処理を行います。 - API 関数です。
1.1.11	R_error	異常を検出したときに実行される関数です。

7.4 関数チャート



8. MMC ドライバ改訂履歴

Ver	変更点	リリース日時
1.02	<ul style="list-style-type: none">● RX630、RX63N の SCI、RSPI に対応しました。● RX210-SCI で SCI1 を選択したときに通信できない問題を修正しました。● DTC を使用した送信処理を実装しました。● DTC を使用した受信でのオーバランエラー検出処理を追加しました。● RX62N-SCI のコンフィグで 100 ピン以下のパッケージ用のマクロ「MMC_SCI_PACKAGE_100PIN_OR_LESS」を追加しました。● RX62N-RSPI のコンフィグで 85 ピンパッケージ用のマクロ「MMC_RSPI_PACKAGE_85PIN」を追加しました。	2013/02/20
1.01	<ul style="list-style-type: none">● RX62N、RX210 の SCI、RSPI に対応しました。● R_mmc_Read_Data 関数の BlkCnt 引数を 32 ビットに変更しました。● R_mmc_Write_Data 関数の BlkCnt 引数を 32 ビットに変更しました。● ユーザが定義するシステムの設定方法を変更しました。	2012/09/01
1.00	新規リリース	2011/3/31

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。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2011.02.21	—	初版発行
1.01	2012.09.01	—	RX ファミリ用 SPI モードマルチメディアカードドライバ V.1.01 Release 00 に合わせてリリース
1.02	2013.02.20	—	RX ファミリ用 SPI モードマルチメディアカードドライバ V.1.02 Release 00 に合わせてリリース

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くと、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）、交通用信号機器、
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、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いません。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事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、お客様にて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/>