RX Family

SPI mode MultiMediaCard Driver: Introduction Guide

R20AN0046EJ0102
Rev.1.02
Feb 20, 2013

Introduction

This manual shows the software configuration of SPI mode MultiMediaCard Driver for the RX family V.1.02 Release 00 and how to use it.

Target Device

Support microcomputer:

- RX610 Group, RX62N Group, RX210 Group, RX630 Group, RX63N Group

Operating environment of Sample program:

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

To confirm sample program running, it is necessary for these Renesas Starter Kit to add MMC socket expansion board called “Middleware evaluation board” or like this.

Please refer to the Application note.

— Renesas Starter Kit for RX610 Middleware evaluation board usage. (R21AN0004EJ0103).

Contents

1. Summary

1.1 Purpose

The purpose is to provide an interface that connects RX family MCU to MultiMediaCard (hereafter referred to as "MMC") in SPI mode.

This manual provides information to create the application.

1.2 Function Description

This driver (hereafter referred to as "MMC driver") is software that enables communication with MMC by RX family.

This software achieves accessing to MMC using SPI mode in 3-lines serial array unit that is RX Family peripherals. Serial Communication Interface (SCI) or Renesas Serial Peripheral Interface (RSPI).

MMC driver

- Reference MMCA System Specifications; Version 3.2
- This is only used in MMC SPI mode
- This is a block type device driver that defines one sector as 512Byte.
  The commands of READ_MULTIPLE_BLOCK and WRITE_MULTIPLE_BLOCK are used.
  As for cards that not support aforesaid two MULTIPLE_BLOCK commands, it is operated by commands of READ_SINGLE_BLOCK and WRITE_SINGLE_BLOCK.
- It supports multiple devices controlled by CS signals.
- It is independent of OS.

1.3 Application Note Structure

This table shows directory and file structure.

Directory Configuration <Directory name> ,File name	Reference
\doc <DIR>	**Document Directory**
\ja <DIR>	**Japanese Document Directory**
r20an0046jj0102_rx_mmc.pdf	Apprication Note
\en <DIR>	**English Document Directory**
r20an0046ej0102_rx_mmc.pdf	Apprication Note (This manual)
\drv <DIR>	**MMC Driver Source Program Directory**
\com <DIR>	**Common Function Directory**
r_mtl_com.c	Common Function (logging function)
r_mtl_com2.h	Common Header file
r_mtl_endi.c	Common Function (endian function)
r_mtl_mem.c	Common Function (standard library function)
r_mtl_str.c	Common Function (standard library function)
r_mtl_tim.c r_mtl_tim.h	Common Function (software loop timer)
r_stdint.h	Data type header file
\rx200_50MHz <DIR>	**Definitions Directory for RX200 series 50MHz Operation** There is the header file for that depends on CPU.
\rx600_100MHz <DIR>	**Definitions Directory for RX600 series 100MHz Operation**
\rx600_96MHz <DIR>	**Definitions Directory for RX600 series 96MHz Operation**
\ mmc <DIR>	**Directory of Device Driver for MMC**
r_mmc.h	Common Header File
r_mmc_mcu_pragma.h	Compiler's pragma declaration header file
r_mmc_io.c r_mmc_io.h	I/O Module for SPI Mode
r_mmc_mmc.c	MMC Module for SPI Mode
r_mmc_sub.c r_mmc_sub.h	Sub Module for SPI Mode
r_mmc_usr.c	API for SPI Mode
\rx	Communication module for RX Family
r_mmc_sci.c	SCI Module for SPI Mode
r_mmc_rspi.c	RSPI Module for SPI Mode
r_mmc_rx_str.h	RX Family structure definition header file
\rx210_sci <DIR>	**Individual Definitions of SFR Directory for RX210 SCI Module** There is the header file for that each serial interfaces.
\rx210_rspi <DIR>	**Individual Definitions of SFR Directory for RX210 RSPI Module**
\rx610_sci <DIR>	**Individual Definitions of SFR Directory for RX610 SCI Module**
\rx62n_sci <DIR>	**Individual Definitions of SFR Directory for RX62N SCI Module**
\rx62n_rspi <DIR>	**Individual Definitions of SFR Directory for RX62N RSPI Module**
\rx630_sci <DIR>	**Individual Definitions of SFR Directory for RX630 SCI Module**
\rx630_rspi <DIR>	**Individual Definitions of SFR Directory for RX630 RSPI Module**
\rx63n_sci <DIR>	**Individual Definitions of SFR Directory for RX63N SCI Module**
\rx63n_rspi <DIR>	**Individual Definitions of SFR Directory for RX63N RSPI Module**
\sample <DIR>	**Sample Workspace Directory**
\MMC_sample_RX210 <DIR>	**High-performance Embedded Workshop Directory for RX210**
MMC_sample_RX210.hws	Workspace
\MMC_sample_RX610 <DIR>	**High-performance Embedded Workshop Directory for RX610**
MMC_sample_RX610.hws	Workspace

	\MMC_sample_RX62N <DIR>	**High-performance Embedded Workshop Directory for RX62N**
	MMC_sample_RX610.hws	Workspace
	\MMC_sample_RX630 <DIR>	**High-performance Embedded Workshop Directory for RX630**
	MMC_sample_RX630.hws	Workspace
	\MMC_sample_RX63N <DIR>	**High-performance Embedded Workshop Directory for RX63N**
	MMC_sample_RX63N.hws	Workspace
	\src <DIR>	**Sample program directory for test**
	\apl **<DIR>**	**Sample program directory for MMC driver test**
	\drv **<DIR>**	**Source code directory for MMC driver**
	\bsp **<DIR>**	**Source code directory for startup routine**

2. Program type definitions

This section gives the details about the type definitions used in the program.

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. Device Driver

3.1 Driver function details

Initialization function

Function Name	Function description
R_mmc_Init_Driver ()	Slot initialization process

Function of device control

Function Name	Function description
R_mmc_Init_Slot()	Slot initialization process
R_mmc_Detach()	Slot stop process
R_mmc_Chk_Detect()	Insertion check process

Data access control function

Function Name	Function description
R_mmc_Read_Data()	Data reading process
R_mmc_Write_Data()	Data writing process
R_mmc_Get_MmcInfo()	MMC information obtaining process

Command list of internal use

Command index	Name of command
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

Note: User needs to respond to unsupported commands

3.2 Function details

3.2.1 Initialization process of driver (R_mmc_Init_Driver)

clause	detail
Function Name	void R_mmc_Init_Driver(void)
Argument	None
Function	Initialize driver. Initialize SFR for card control. The following process is done in every slot. (1) Open card control port. (2) Initialize card control RAM. Execute once when the system starts up.
Return Value	None

3.2.2 Initialization of card slot (R_mmc_Init_Slot)

clause	detail
Function Name	int16_t R_mmc_Init_Slot(uint8_t SlotNo)
Argument	uint8_t SlotNo : Slot number
Function	Initialize card slot. Initialize card control RAM. Initialization of card. Execute when card insertion is detected.
Return Value	Returns initialization result. MMC_OK : Successful operation MMC_ERR_PARAM : Parameter error MMC_ERR_HARD : Hardware error MMC_ERR_CRC : CRC error MMC_ERR_IDEL : Idle state error MMC_ERR_OTHER : Other error

3.2.3 Card slot stop process (R_mmc_Detach)

clause	detail
Function Name	int16_t R_mmc_Detach(uint8_t SlotNo)
Argument	uint8_t SlotNo : Slot number
Function	Process when removing card from designated slot. -Initialize card control SFR. -Open card control port. -Initialize card control RAM. Execute when card removal is detected.
Return Value	Returns removal result. MMC_OK : Successful operation MMC_ERR_PARAM : Parameter error

3.2.4 Card insertion checking process (R_mmc_Chk_Detect)

clause	detail
Function Name	int16_t R_mmc_Chk_Detect(uint8_t SlotNo, uint8_t* pDetSts)
Argument	uint8_t SlotNo : Slot number uint8_t *pDetSts : Buffer pointer for card insertion condition
Function	Check the condition of card being inserted. If returns "MMC_OK", The port status of card detecting will be in buffer 'pDetSts'. — MMC_TRUE :The port status of card detecting is active — MMC_FALSE:The port status of card detecting Non is non-active Cannot remove chattering in this process. Remove chattering in upper system if needed. Recommend confirming card insertion by periodic polling.
Return Value	Returns the check result. MMC_OK : Successful operation MMC_ERR_PARAM : Parameter error

3.2.5 Data reading process (R_mmc_Read_Data)

clause	detail
Function Name	int16_t R_mmc_Read_Data(uint8_t SlotNo, uint32_t BlkNo, uint32_t BlkCnt, uint8_t *pData, uint8_t Mode)
Argument	uint8_t SlotNo : Slot number uint32_t BlkNo : Block number to start readout uint32_t BlkCnt : Number of readout blocks uint8_t *pData : Pointer to the area where the data which is read must be stored uint8_t Mode : Transfer mode of reading data
Function	Readout the data from card by block (512byte) Readout the data in the designated number of blocks from the designated block. Choose MMC_MODE_NORMAL(transfers data to the designated buffer 'pData'.) in "Mode". The readout from MMC is possible among MMC information handed from R_mmc_Get_MmcInfo() function only when card classification (MmcInfo.Card) is not 'MMC_CARD_UNDETECT'. Maximum block number is 'pMmcInfo.MaxBlkNum' from the "R_mmc_Get_MmcInfo()" function. Maximum number of blocks is 'pMmcInfo.MaxBlkNum' +1.
Return Value	Returns the result of reading. MMC_OK : Successful operation MMC_ERR_PARAM : Parameter error MMC_ERR_HARD : Hardware error MMC_ERR_CRC : CRC error MMC_ERR_OTHER : Other error

3.2.6 Data writing process (R_mmc_Write_Data)

clause	detail	
Function Name	int16_t R_mmc_Write_Data(uint8_t SlotNo, uint32_t BlkNo, uint32_t BlkCnt, uint8_t *pData, uint8_t Mode)	
Argument	uint8_t SlotNo	: Slot number
	uint32_t BlkNo	: Block number to start writing
	uint32_t BlkCnt	: Number of writing blocks
	uint8_t *pData	: Pointer to the area where the data which is written must be stored
	uint8_t Mode	: Transfer mode of writing data
Function	Write the data to card by block (512byte). Write the data in the designated number of blocks to the designated block. Choose MMC_MODE_NORMAL (This is a mode that transfers data from the designated buffer 'pData') in "Mode". The transfers to MMC is possible among MMC information handed from R_mmc_Get_MmcInfo() function only when card classification (MmcInfo.Card) is not 'MMC_CARD_UNDETECT'. Maximum block number is 'pMmcInfo.MaxBlkNum' from the "R_mmc_Get_MmcInfo()" function. Maximum number of blocks is 'pMmcInfo.MaxBlkNum' +1.	
Return Value	Returns the result of writing.	
	MMC_OK	: Successful operation
	MMC_ERR_PARAM	: Parameter error
	MMC_ERR_HARD	: Hardware error
	MMC_ERR_WP	: Write-protection error
	MMC_ERR_OTHER	: Other error

3.2.7 Card information obtaining process (R_mmc_Get_MmcInfo)

clause	detail
Function Name	int16_t R_mmc_Get_MmcInfo(uint8_t SlotNo, MMC_INFO* pMmcInfo)
Argument	uint8_t SlotNo : Slot number MMC_INFO *pMmcInfo : Buffer pointer for card information
Function	It returns MMC information. The buffer 'pMmcInfo' holds card information. • pMmcInfo.Card : Card types — MMC_CARD_UNDETECT : Card not detected — MMC_CARD_MMC : MMC — MMC_CARD_OTHER : Other card • pMmcInfo.WProtect : Write-protection status — MMC_NO_PROTECT : Write-protection cancel — bit1: MMC_W_PROTECT_SOFT : Software write-protection pMmcInfo.MemSize : Card capacity(byte) pMmcInfo.MaxBlkNum : Maximum block number of the media When 'pMmcInfo.MemSize' is 0xFFFFFFFF, 'pMmcInfo.MaxBlkNum' +1 inidicates the number of the media and the size is ('pMmcInfo.MaxBlkNum'+1)*512.
Return Value	Returns the result of obtaining card information. MMC_OK : Successful operation MMC_ERR_PARAM : Parameter error MMC_ERR_OTHER : Other error

3.3 Data Structure

Data structure is showed as follow.

Definition of Card Information Data Structure

```
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 Definitions

Definitions are showed as follow.

```
/*------------ 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. Setup Examples

4.1 r_mtl_XXX : Variable Data Setup Example

This section is for setting the resources of each user system

The setting should be made in the [/**SET**/] comment of each file.

An excerpt of each file is provided with detailed comments.

4.1.1 r_mtl_com.h

This file is a common header file.

r_mtl_com.h is prepared for each MCU and system clock settings.

Please select using directory for your environment.

If these are not suitable for your environment, please make directory and setting files for your environment.

MCU - System clock	Include Directory (drv directory)
RX600 Series - 100MHz	\com\rx600_100MHz
RX600 Series - 96MHz	\com\rx600_96MHz
RX200 Series - 50MHz	\com\rx200_50MHz

(1) Define the SFR header file

— Include the header file that contains the definition of the MCU function registers.
This file must be included due to device drivers accessing ports designated in it.
The following is an example of the setting when not using the Renesas MCU SFR header file.
When using MMC driver, include the SFR header file.

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

(2) Define the software loop timer

— When using the loop timer, include following header file.
The loop timer process is used for waiting duration of driver.
The following is an example of the setting when using the software loop timer.
And please define the macro that is suitable for your system clock in r_mtl_tim.h. In case, running RX600 in 100MHz, "MTL_TIM_RX600__12_5MHz_noWait_Ix8" should be defined.

```
/* 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) Define Endian type

— Set the following definitions to configure the File System to the Endian type in the user system.
In the case of RX family, please appoint it by the setting of the endian of the compiler option.

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

(4) Specify type of user standard library

— Specify the type of standard library in the user system.

When using the library bundled with the compiler for the processes stated below, add the listed define definitions as comments.

When using the optimized library, define the optimized library.

```
/* 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

When including r_mtl_tim.h, it is enable.

The value depends on clock frequency and wait of MCU.

Set the software timer to be used for internal operations.

If there are no "define" that is suitable for user system, user has to make own "define".

```
/* 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")*/
 (omit)
#endif

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

4.2 MMC Driver : Variable Data Setup Example

This section is for setting the resources of each user system.

The setting should be made in the [/**SET**/] comment of each file.

An excerpt of each file is provided with detailed comments.

4.2.1 r_mmc.h(Common header file)

(1) Define number of slots (devices) and slot number

— Specify number of slots (devices) and slot number.

```
/*  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) Define use of single block commands with SPI mode

— Do not make any changes.

```
/* 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) Define card type

— Define MMC_SUPPORT_MMC.

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

4.2.2 r_mmc_user_setting.h (User Definition Header file)

(1) Selecting MCU and Communication module

r_mmc_user_config.h is prepared for each MCU and communication module. Please include r_mmc_user_config.h in directory that is suitable for user system.

MCU – Communication Module	Include Directory (drv directory)
RX610 - SCI	\mmc\rx\rx610_sci
RX62N - SCI	\mmc\rx\rx62n_sci
RX62N - RSPI	\mmc\rx\rx62n_rspi
RX210 - SCI	\mmc\rx\rx210_sci
RX210-RSPI	\mmc\rx\rx210_rspi
RX630 - SCI	\mmc\rx\rx630_sci
RX630 - RSPI	\mmc\rx\rx630_rspi
RX63N - SCI	\mmc\rx\rx63n_sci
RX63N - RSPI	\mmc\rx\rx63n_rspi

(2) Definition for difference of numbers of MCU pins

— In case, RX62N-SCI:

- When user uses 100pin or 85pin package, please define MMC_SCI_PACKAGE_100PIN_OR_LESS macro

— In case, RX62N-RSPI:

- When user uses 85pin package, please define MMC_RSPI_PACKAGE_85PIN macro

[In case, RX62N-SCI]

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

[In case, RX62N-RSPI]

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

(3) “define” for channel number of communication unit and terminal

Please define MMC_SCI_CHANNEL macro(if you use SCI), or MMC_RSPI_CHANNEL macro(if you use RSPI) with the channel number.

— In case, RX62N-SCI:
- It is often necessary to define using pins. Please refer to the table below, and please define the value for MMC_SCI_PIN macro.

— In case, RX62N-RSPI:
- Please refer to the table below, and please define the value for MMC_RSPI_PIN macro.

— In case, RX210-SCI or RX630-SCI or RX63N-SCI:
- It is often necessary to define using pins. Please refer to the table below, and please define the value for MMC_SCI_RXD_PIN macro (RxD pin) and MMC_SCI_SCK_PIN macro (SCK pin) and MMC_SCI_TXD_PIN macro (TxD pin).

— In case, RX210-RSPI or RX630-RSPI or RX63N-RSPI:
- It is often necessary to define using pins. Please refer to the table below, and please define the value for MMC_RSPI_RXD_PIN macro (MISO pin) and MMC_RSPI_SCK_PIN macro (RSPCK pin) and MMC_RSPI_TXD_PIN macro (MOSI pin).

[When user uses RX610 SCI]

```
/* 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 **/
```

[When user uses RX62N SCI]

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

[When user uses RX62N RSPI]

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

/* RSPI PIN select  ( 'A' or 'B' ) */
/*
                          ||    Select Port                  |
------------+-----------++--------------------------------+
  MMC_SCI_  | MMC_RSPI_ ||    RxD   |   SCK    |   TxD    |
  CHANNEL   | PIN       ||  Select  |  Select  |  Select  |
   Value    |  Value    ||   port   |   port   |   port   |
============+===========++==========+==========+==========+
     0      | 'A'       || PC7      | PC5      |  PC6     |
            |-----------++----------+----------+----------+
(=Use RSPI0)| 'B'       || PA7      | PA5      |  PA6     |
------------+-----------++----------+----------+----------+
     1      | 'A'       || P30      | P27      |  P26     |
            |-----------++----------+----------+----------+
(=Use RSPI1)| 'B'       || PE7      | PE5      |  PE6     |
------------+-----------++----------+----------+----------+*/
#define MMC_RSPI_PIN    'B'                             /** SET **/
```

[When user uses RX210 SCI]

```
/* 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 **/
```

[When user uses RX210 RSPI]

```
/* 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 **/
```

[When user uses RX630 SCI]

```
/* 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 (=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)	'A'	PB0	'A'	PB3	'A'	PB1
	'B'	PK4	'B'	P70	'B'	PK5
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      || 'A'      | PB6      || 'A'      | PB5      || 'A'      |  PB7     |
            ++----------|----------++----------|----------++----------|----------+
(=Use SCI9) || 'B'      | PK3      || 'B'      | P60      || 'B'      |  PK2     |
------------++----------+----------++----------+----------++----------+----------+
    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 **/
```

[When user uses RX630 RSPI]

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

[When user uses RX63N SCI]

```
/* 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	'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	(Don't care)	PB0	(Don't care)	PB3	(Don't care)	PB1
(=Use SCI4)						
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	(Don't care)	PB6	(Don't care)	PB5	(Don't care)	PB7
(=Use SCI9)						

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

[When user uses RX63N RSPI]

```
/* 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) Define control ports

— Please define the macro for DETECT(detecting card insertion) pins or CS(card select) pins suitable for user's circuit.

```
/*-----------------------------------------------------------------------*/
/*  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) Definition of detecting communication timeout

— This macro can omit detecting timeout during communication.
If user omits detecting timeout, please define “MMC_NOCHK_TIMEOUT”. If this macro is defined, processing speed would be increased, but there is a possibility program stops when abnormal communication status occurs. If user does not omit this macro definition, please set time to this macro.

- Setting time unit using MMC_T_SCI_WAIT (when user uses SCI) or MMC_T_RSPI_WAIT (when user uses RSPI) macro. Please select setting macro from r_mtl_tim.h.
- Please define transmit timeout time using MMC_SCI_TX_WAIT macro (when user uses SCI) or MMC_RSPI_TX_WAIT macro (when user uses RSPI).
- Please define reception timeout time using MMC_SCI_RX_WAIT macro (when user uses SCI) or MMC_RSPI_RX_WAIT macro (when user uses RSPI).
- Setting values of each timeout time macro are [timeout time/unit].

```
/*-----------------------------------------------------------------------*/
/* 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) Define resources

— The data transfer depends on MCU resource for use with.
Select one of the following for use as your system.

```
/*-------------------------------------------------------------------------*/
/*  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) Define bit rate

— As for transfer speed setting, it is necessary to meet tODLY of both Identification mode and Data Transfer mode in spec.
In addition, it is necessary to meet tOD (100kHz <= tOD <= 400kHz) at Identification mode and tPP (0.1MHz <= tPP <= 20MHz (*)) at Data Transfer mode.
The frequency of tOD and tPP mean the frequency of SCLK in this device driver.
As for maximum frequency, make a confirmation of each MCU datasheet.

- MMC_UBRG_IDENTIFICATION macro
  This macro sets transfer clock when user uses “Identification mode”.
  In case user uses SCI, MMC_UBRG_IDENTIFICATION macro value is set to BRR register.
  In case user uses RSPI, MMC_UBRG_IDENTIFICATION macro value is set to SPBR register.
- MMC_UBRG_D_TRANSFER macro
  This macro sets transfer clock when user uses “Transfer mode”.
  In case user uses SCI, MMC_UBRG_D_TRANSFER macro value is set to BRR register.
  In case user uses RSPI, MMC_UBRG_D_TRANSFER macro value is set to SPBR register.
- MMC_CLK_D_TRANSFER macro
  This macro sets transfer clock frequency when user uses “Transfer mode”.

Example of RX610:

```
/*-----------------------------------------------------------------------------------------------*/
/* 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. Method for connecting to MCU and MCU resource for use with

5.1 MCU resource for use with

This software controls as follows:

Data input/output is controlled by clock synchronous serial I/O (internal clock).

Allocate CMOS output port and set CMOS output of the clock synchronous serial I/O in order to perform high-speed processing.

Please do CMOS output setting.

The transmission control detects the space of the transmission buffer, and use a transmission interrupt request bit without using an interrupt. Therefore, I set it about an interrupt as follows.

- Interrupt level is level 0 (interrupt is prohibited in it).
- Connect Card CS# pin to RX Port and control it by RX general port setting.

Resources	RX610	RX62N	RX210	RX630	RX63N
DTC	R	R	R	R	R
CRC calculation circuit	R	R	R	R	R
Port for CS#: 1port/Card	M	M	M	M	M
Port for Card detection: 1port/Card	M	M	M	M	M
Port for Power Control: 1port/Card	M	M	M	M	M

M: mandatory
R: recommended (high-speed processing is enabled when unique resource of RX is used)

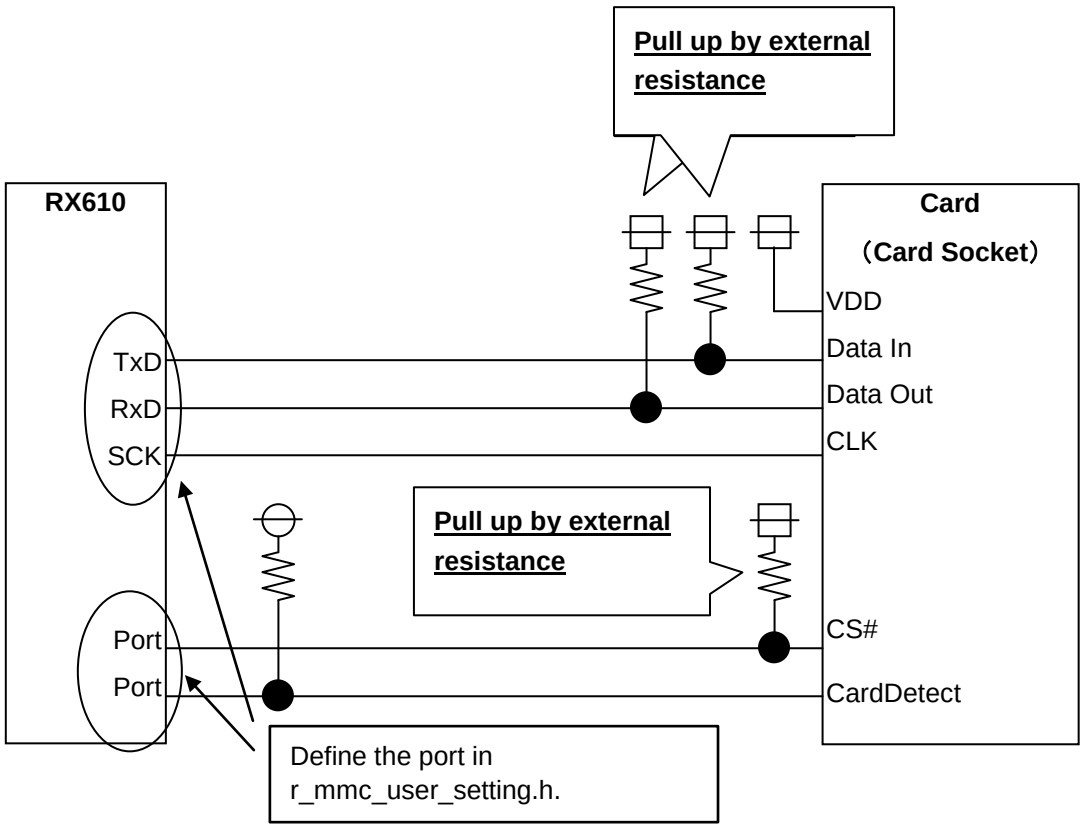
Therefore, RX family MCU in SCI Clocked Synchronous mode or RSPI can be operated.

Resource	RX610	RX62N	RX210	RX630	RX63N
SCI	available	available	available	available	available
RSPI	-	available	available	available	available

5.2 Method for connecting to MCU

An example of connecting to RX610 is showed.

In case of other RX family MCUs, the same connection is made



Power control circuit for card is required to control inserting and removing.

(power control circuit is not described).

Supply power to the card after detection of insertion.

Cut off the power supply to a card before removing.

In case of using multiple cards in the same SPI bus, assigne CS# port/CardDetect port for each card.

6. Notes for Application Development

6.1 Notes for use

- Configure the software according to the hardware.
- Remove card after deactivation, setting signals between MCU and card into Hi-z and cutting off power supply to card.
- In case that insertion/removable circuit is not realized, inserting/removing card might cause the power source to be unstable and reset MCU.

6.2 Notes for embedding

6.2.1 Development environment

Requirement items

When user develops, choose newer version than below.

[Software]

-Integrated Development Environment

High Performance Embedded Workshop Version 4.09.00.007

-C compiler

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

[Debug tools]

-Emulator debugger

RX E1/E20 Emulator Debugger

-Emulator software

RX E1/E20 Emulator Sorftware V.1.03.00

[Board]

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

To confirm sample program running, it is necessary for these Renesas Starter Kit to add MMC socket expansion board called “Middleware evaluation board” or like this.

Please refer to the Application note.

— Renesas Starter Kit for RX610 Middleware evaluation board usage. (R21AN0004EJ0103).

6.2.2 Files for includding

Include "r_mmc.h" and "r_mtl_com.h" when embedding this driver.

It is necessary for include to do "r_mtl_com.h" first.

6.2.3 Notice of DTC using

The setting of r_mmc_user_config.h has MMC_OPTION_3 macro or MMC_OPTION_4, this setting uses DTC for serial communication.

・In case, this option is active, it is necessary to allocate DBTCTBL section to 4096 byte aligned RAM area.

・In case, embed RX610-SCI or RX62N-SCI MMC driver SFR definition header file, it is necessary to allocate CBITCHGTBL_1 section to 256 byte aligned ROM area.

6.2.4 Notice for Channel selection and Pin selection.

MMC driver corresponds MCU packages below.

RX610 group: 176,144 pin package

RX62N,RX621 group: 176,155/144,100,85 pin package

RX210 group: 100,80,64,48 pin package

RX630 group: 177/176,145/144,100,80 pin package

RX63N,RX631 group: 177/176,145/144,100,64,48 pin package

6.2.5 Confirmed Channel and Pin selection

We have confirmed these Channels and Pins selection settings.

MCU – Communication Module	RX610 SCI	RX62N SCI	RX62N RSPI	RX630 SCI	RX630 RSPI
Channel No.	1	2	1	9	0
SIO-RxD Port No.	P30	P12	PE7	PB6	PA7
SIO-CLK Port No.	P27	P11	PE5	PB5	PA5
SIO-TxD Port No.	P26	P13	PE6	PB7	PA6
CS Port No.	P70	P90	P90	PD1	PD1
CardDetect Port No.	P71	P91	P91	PD0	PD0

MCU – Communication Module	RX63N SCI	RX63N RSPI	RX210 SCI	RX210 RSPI
Channel No.	7	0	9	0
SIO-RxD Port No.	P92	PA7	PB6	PA7
SIO-CLK Port No.	P91	PA5	PB5	PA5
SIO-TxD Port No.	P90	PA6	PB7	PA6
CS Port No.	P97	P97	P50	P50
CardDetect Port No.	P96	P96	P51	P51

6.3 ROM size / RAM size / Stack size

MMC driver requires ROM/RAM/Stack size as below.

Compile condition is optimize level =2, optimize method = size. Each size would change with user settings.

ROM/RAM size

Resources excepting serial I/O	ROM/RAM (Section)	Size [unit : kByte]				
		RX610 (SCI)	RX62N (SCI)	RX62N (RSPI)	RX210 (SCI)	RX210 (RSPI)
None	ROM (section P,C, CBITCHGTBL_1)	6.6	6.6	7.0	6.4	7.0
	RAM (section B)	0.1	0.1	0.1	0.1	0.1
CRC	ROM (section P,C, CBITCHGTBL_1)	6.0	6.0	6.1	5.9	6.3
	RAM (section B)	0.1	0.1	0.1	0.1	0.1
DTC	ROM (section P,C, CBITCHGTBL_1)	7.0	7.0	7.2	6.8	7.4
	RAM (section B,BDTCTBL)	1.1	1.1	1.1	1.1	1.1
CRC + DTC	ROM (section P,C, CBITCHGTBL_1)	6.2	6.4	6.5	6.2	6.7
	RAM (section B,BDTCTBL)	1.1	1.1	1.1	1.1	1.1

Resources excepting serial I/O	ROM/RAM (Section)	Size [unit : kByte]			
		RX630 (SCI)	RX630 (RSPI)	RX63N (SCI)	RX63N (RSPI)
None	ROM (section P,C, CBITCHGTBL_1)	6.3	7.0	6.3	7.0
	RAM (section B)	0.1	0.1	0.1	0.1
CRC	ROM (section P,C, CBITCHGTBL_1)	5.9	6.3	5.9	6.3
	RAM (section B)	0.1	0.1	0.1	0.1
DTC	ROM (section P,C, CBITCHGTBL_1)	6.8	7.4	6.8	7.4
	RAM (section B,BDTCTBL)	1.1	1.1	1.1	1.1
CRC + DTC	ROM (section P,C, CBITCHGTBL_1)	6.2	6.7	6.2	6.7
	RAM (section B,BDTCTBL)	1.1	1.1	1.1	1.1

Stack size

Function	Stack size
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
R_mmc_Get_MmcInfo	24

6.4 Notes on insertion/removal of the card

Enable to detect the insertion or removal of the card using the function "R_mmc_Chk_Detect()" with the card detection pins, which comes with the card connector.

Therefore, it is recommended to detect the insertion or removal of the card by software polling.

The driver returns an error when the card is removed in a data transmission eventually.

However, the driver may not return an error when the card is removed momentary in a data transmission in case of the following conditions:

- Data transmission is operated properly when no response error from the card occurs, because the driver cannot detect the insertion or removal of the card by software polling.
- The driver may recognize the completion of writing to the card when the card is inserted or removed momentary in writing stage. This is because of the specification that a writing completion will be detected by "H" signal of DetaIn pin. DataIn pin is pulled up.

Please avoid this problem by the system hardware such as a hardware interrupt control and polling period time etc.

6.5 Note of the Hi-z setting processing of the port about the exclusion and adding of the card

In the insertion of the card, Please insert card after having set CS#, DataIn, DataOut, and Clock terminal in Hi-z. Please supply the power supply to a card afterwards.

In the extraction of the card, after the power supply supply stop to a card, after the power supply supply stop to a card, please extract a card, after having set CS#, DataIn, DataOut, and Clock terminal in Hi-z.

The CS#, DataIn, DataOut, CLK terminal of the card is assigned to SIO or a port terminal of MCU, but does not process Hi-z by this driver because the case that the port is assigned to other resources is expected.

Therefore, please make the Hi-z processing of the MCU terminal in the high rank side in the exclusion and adding of the card.

7. Sample program usage

This section explains the sample program for MMC driver usage.

7.1 Outline

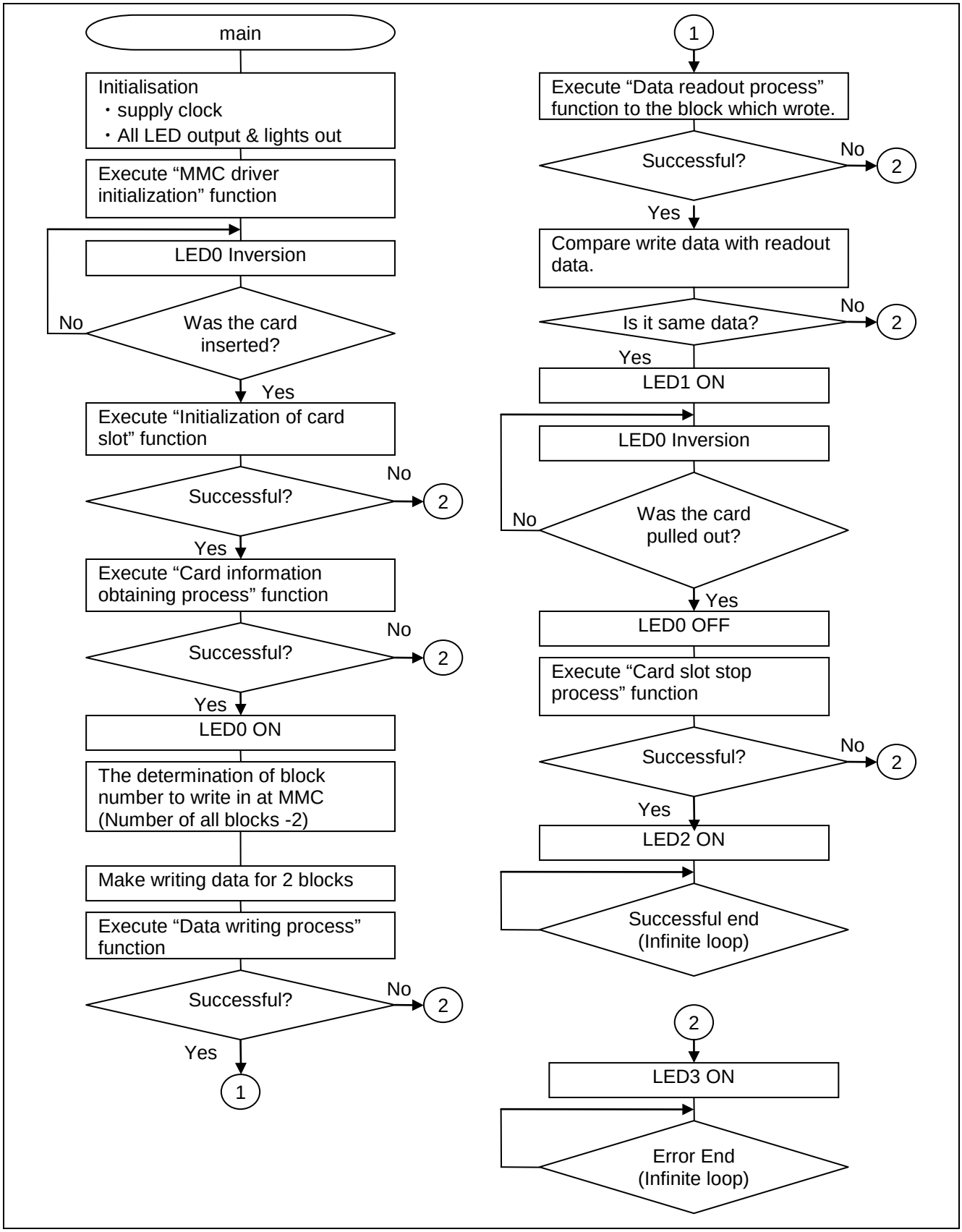
After having inserted MMC, using last 2 blocks of the memory in the MMC, compares it with the reading and writing of data.

After having executed the above, extracts a card, and it is the movement end.

Displays the progress of the program and a result with LED. Please refer to a list shown below for the contents of the indication.

LED0	ON	: A card is put.
	OFF	: A card is inserted.
	BLINK	: Require the drawing of the card or the insertion.
LED1	ON	: Write/Read Execution Successful.
LED2	ON	: Program Successful.
LED3	ON	: Program Error.

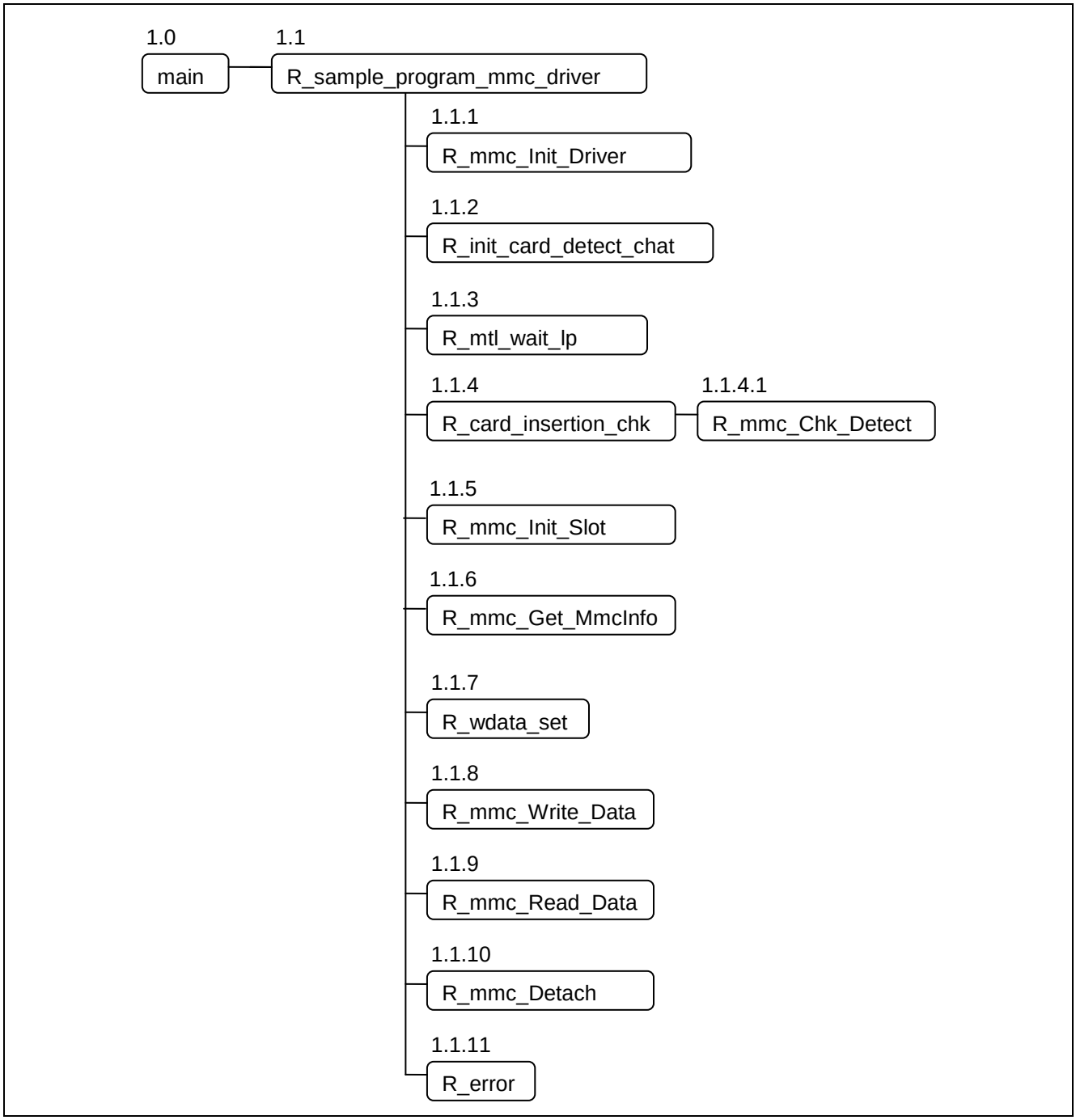
7.2 Flow


main
Initialisation
・supply clock
・All LED output & lights out
Execute “MMC driver initialization” function
LED0 Inversion
No
Was the card inserted?
Yes
Execute “Initialization of card slot” function
Successful?
No
2
Yes
Execute “Card information obtaining process” function
Successful?
No
2
Yes
LED0 ON
The determination of block number to write in at MMC (Number of all blocks -2)
Make writing data for 2 blocks
Execute “Data writing process” function
Successful?
No
2
Yes
1
1
Execute “Data readout process” function to the block which wrote.
Successful?
No
2
Yes
Compare write data with readout data.
Is it same data?
No
2
Yes
LED1 ON
LED0 Inversion
No
Was the card pulled out?
Yes
LED0 OFF
Execute “Card slot stop process” function
Successful?
No
2
Yes
LED2 ON
Successful end (Infinite loop)
2
LED3 ON
Error End (Infinite loop)

7.3 Function list

No	Function Name	Outline
1.0	main	Initialize MCU clock and LED of RSK, and execute sample program.
1.1	R_sample_program_mmc_driver	When MMC is inserted, reads and writes a random value to a card and compares each.
1.1.1	R_mmc_Init_Driver	Initializes the MMC driver processing. - This is API function.
1.1.2	R_init_card_detect_chat	Initializes memory for the card detection.
1.1.3	R_mtl_wait_lp	It is time waiting processing before confirming the next terminal state.
1.1.4	R_card_insertion_chk	Checks a having card or not state.
1.1.4.1	R_mmc_Chk_Detect	Returns a having card or not state. - This is API function.
1.1.5	R_mmc_Init_Slot	When a card is inserted, I execute initialization processing for the card. - This is API function.
1.1.6	R_mmc_Get_MmcInfo	Gets card information. - This is API function.
1.1.7	R_wdata_set	Makes data to write in at MMC.
1.1.8	R_mmc_Write_Data	Writes data to a MMC. - This is API function.
1.1.9	R_mmc_Read_Data	Reads data from a MMC. - This is API function.
1.1.10	R_mmc_Detach	When a card is pulled up, I initialize it for a slot. - This is API function.
1.1.11	R_error	Error handling function

7.4 Function chart



8. MMC Driver Information

Ver	change	Release date
1.02	• Added support RX630-SCI, RX630-RSPI, RX63N-SCI, RX63N-RSPI • Corrected wrong code for RX210-SCI with SCI1 selection. • Added using DTC for sending. • Added detecting overrun error using DTC for receiving. • Added settings for RX62N-SCI for less 100pin package. • Added settings for RX62N-RSPI for 85 pin package.	Feb.20.13
1.01	• Added support RX62N-SCI, RX62N-RSPI, RX210-SCI, RX210-RSPI • Changed R_mmc_Read_Data() BlkCnt argument type to 32 bit • Changed R_mmc_Write_Data() BlkCnt argument type to 32 bit • Changed method of user settings.	Sep.01.12
1.00	first release	Mar.31.11

Website and Support

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

Inquiries
http://www.renesas.com/contact/

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Feb.21.2011	-	First edition issued
1.01	Sep.01.2012	-	Release with SPI mode MultiMediaCard Driver for the RX Family V.1.01 Release 00
1.02	Feb.20.2013	-	Release with SPI mode MultiMediaCard Driver for the RX Family V.1.02 Release 00

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.
   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.
5. Differences between Products
   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 2.2