RH850/U2B

Using eMMC Storage

Introduction

RH850/U2B series provide a Multimedia Card Interface (MMCA) for external eMMC memory.

This document provides a general introduction of the eMMC standards, MMCA configuration flows and the sample software.

Target Device

This document is intended to describe the MMCA sample SW on RH850/U2B series.

In this document, the RH850/U2B device R7F702Z21∗ is employed to implement the example application. Still, the concept described in this document applies also to other members of the RH850/U2B series.

Disclaimer

Renesas Electronics does not warrant the information included in this document. You are fully responsible for incorporation of these circuits, software, and information in the design of your equipment and system. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.

Contents

1. Background

Since much more complicated automotive applications and the corresponding memory extension are required for the future, RH850/U2B devices provide a Multimedia Card Interface for external eMMC memory.

This application note contains a general description of the eMMC standards, typical eMMC operations, related software procedures, MMCA module settings etc., and introduces the sample software and its software / hardware setups.

In this document, Section 2 includes the reference documents. For some detailed information or special use cases, please use this application note in combination with the references.

Section 3 lists of the register abbreviations in this document.

Section 4 provides a description about the hardware setups and software environment for the MCU applications with eMMC storage.

Section 5 introduces the eMMC standards with which the MCU is compliant. This section contains the operation modes, bus configuration, state machines, commands and response, card registers etc.

Section 6 and Section 7 describe individually the port and clock setups for the eMMC interface (MMCA) on MCU.

Section 8 provides the details of software flows and the related MCU register settings for the applications using eMMC storage.

Section 9 explains the structure of the related sample software.

2. Reference Documents

2.1 RH850/U2B Group User's Manual: Hardware

The Hardware User's Manual provides information about the functional and electrical behavior of the device.

At the release time of this document the following manual version is available:

- RH850/U2B User's Manual (Rev.1.00): R01UH0923EJ0100

2.2 RH850/X2X User's Manual: Main Board

The Main Board User's Manual provides information about the X2X main board and peripheral circuits.

At the release time of this document the following manual version is available:

RH850/X2X User's Manual: Main Board (Rev.1.00): R20UT4459ED0100

2.3 eMMC standards

The MMCA module in target device is compliant with JEDEC Standard JESD84-A441 and the JEDEC Standards with later versions.

Renesas provides also an extension board Y-RH850-EMMC-SFMA-EXT-BRD with eMMC. The eMMC device on this application board is the Swissbit SFEM4096B1EA1TO-I-GE-111-E02 device of the EM-26 Series, or the SFEM032GB1EA1TO-I-LF-111-STD device of the EM-20 Series. Both of the eMMC devices are compliant with JEDEC Standard JESD84-B50.

Therefore, the application note and sample code are based on the following documents:

- JESD84-A441, JEDEC Standard - Embedded Multi-Media Card (e•MMC) Electrical Standard (4.41)
- JESD84-B50, JEDEC Standard - Embedded Multi-Media Card (e•MMC) Electrical Standard (5.0)

Section 5.6 provides a comparation between these two standard versions.

3. Definitions and Abbreviations

For the purposes of this publication, the following abbreviations for common terms apply:

- **eMMC:** embedded multimedia card
- **MMCA:** multimedia card interface A
- **Host:** the MCU device with multimedia card interface
- **Slave:** the multimedia card
- **Block:** basic data transfer unit
- **Stuff bit:** fill 0 bits to ensure fixed length frames for commands and responses
- **Erase:** group erase operation which does not require actual physical NAND erase
- **TRIM:** erase operation to write blocks instead of erase groups
- **Sanitize:** physically remove data from the unmapped user address space

Additionally, the abbreviation of registers, which are mentioned in this application note is listed in Table 3-1.

Table 3-1 Symbols of Registers

Register Category	Register Name	Symbol
eMMC Card Registers*1	Card Identification Number Register	CID
	Relative Card Address Register	RCA
	Driver Stage Register	DSR
	Card Specific Data Register	CSD
	Operation Conditions Register	OCR
	Extended Card Specific Data Register	EXT_CSD
MCU Port Registers*2	Port Mode Register	PMn
	Port Mode Control Register	PMCn
	Port Mode Control Register	PIPCn
	Port Function Control Register	PFCn
	Port Function Control Expansion Register	PFCEn
	Port Function Control Additional Expansion Register	PFACEn
	Port Universal Characteristic Control Register	PUCCn
	Port Drive Strength Control Register	PDSCn
	Port Keycode Protection	PKCPROT
	Port Write Enable Register	PWE
MCU Clock Registers*2	Module Standby Register for MMCA	MSR_MMCA
	Module Standby Register Key Code Protection Register	MSRKCPROT
MCU MMCA Module Registers*2	MMCAn Command Setting Register	MMCAnCE_CMD_SET
	MMCAn Argument Register	MMCAnCE_ARG
	MMCAn Argument Register for Automatically-Issued CMD12	MMCAnCE_ARG_CMD12
	MMCAn Command Control Register	MMCAnCE_CMD_CTRL
	MMCAn Transfer Block Setting Register	MMCAnCE_BLOCK_SET
	MMCAn Clock Control Register	MMCAnCE_CLK_CTRL
	MMCAn Buffer Access Configuration Register	MMCAnCE_BUF_ACC
	MMCAn Response Register 3	MMCAnCE_RESP3
	MMCAn Response Register 2	MMCAnCE_RESP2

	MMCAn Response Register 1	MMCAnCE_RESP1
	MMCAn Response Register 0	MMCAnCE_RESP0
	MMCAn Response Register for Automatically-Issued CMD12	MMCAnCE_RESP_CMD12
	MMCAn Data Register	MMCAnCE_DATA
	MMCAn Boot Operation Setting Register	MMCAnCE_BOOT
	MMCAn Interrupt Flag Register	MMCAnCE_INT
	MMCAn Interrupt Enable Register	MMCAnCE_INT_EN
	MMCAn Status Register 1	MMCAnCE_HOST_STS1
	MMCAn Status Register 2	MMCAnCE_HOST_STS2
	MMCAn Software Reset Register	MMCAnCE_SWRESA

Notes: 1. For details of the eMMC card registers, please refer to Section 5.7.

2. The general MCU register configurations are mentioned in Section 6 to 8.
   For further information, please refer to the Hardware User's Manual *R01UH0923EJxxxx* of the device.

4. Software and Hardware Tools

This section contains the information about tools which are used to implement the MMCA application.

4.1 Development Tools

The compiler used for the sample code is Renesas Electronics CS+.

The required version of the CS+ compiler for U2B devices is V8.09.00 or later versions.

4.2 Emulator E2

The Renesas E2 Emulator is required for the debugging.

4.3 RH850/U2B Evaluation Boards

The following evaluation boards are used to implement the MMCA application.

4.3.1 PiggyBack Board

The PiggyBack Board Y-RH850-U2B-292PIN-PB-T1-V1 D018567_06_V02 carries the target device. The SW in this document is performed on the RH850/U2B device R7F702Z21* within this piggyback board.

4.3.2 Main Board

Renesas provides the main board RH850-X2X-MB-T1-V1 for the RH850/X2X microcontrollers as easy to use platform to evaluate the devices. This main board should be used in conjunction with a piggyback board.

4.3.3 Extension eMMC Board

Renesas provides also the extension board Y-RH850-EMMC-SFMA-EXT-BRD with eMMC.

This board is described in the RH850/X2X Main Board User's Manual *R20UT4459EDxxxx Section 4.9 'eMMC/SFMA Module'*.

5. eMMC Standards

The MMCA module in the target device is compliant with the JEDEC Standard JESD84-A441*Note and the JEDEC Standards with later versions.

The description in this section is based on the eMMC standard version 4.41 and version 5.0.

5.1 System Overview

Figure 5-1 shows the general eMMC system overview. The eMMC specification covers the behavior of the interface and the device controller.

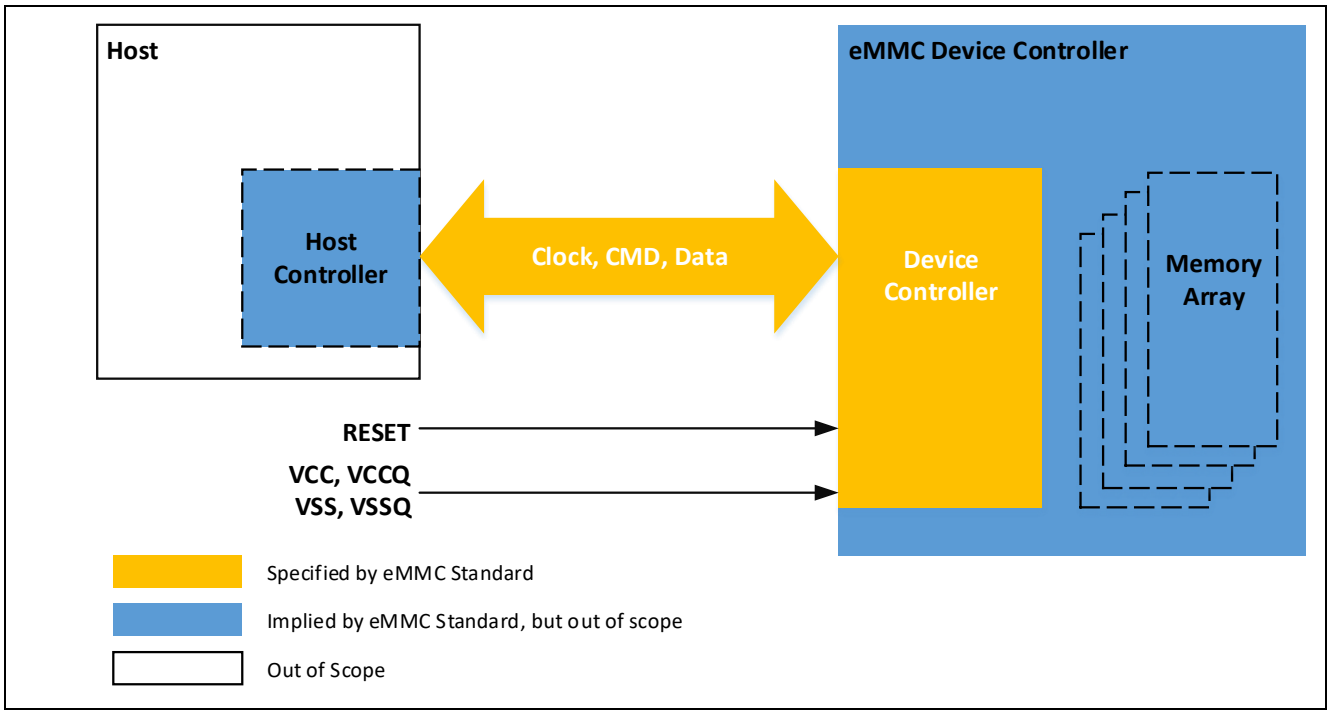


Figure 5-1 eMMC System Overview

The eMMC device supports the following features:

- For read-only, read/write and I/O cards.
- Maintains card support for three different data bus width modes: 1-bit (default), 4-bit, and 8-bit.
- Includes definition for a higher than 2GB density of memories.
- Includes password protection of data.
- Supports basic file formats for high data interchangeability.
- Includes application specific commands.
- Enables correction of memory field errors.
- Has built-in write protection features for the boot and user areas, which may be permanent, power-on, or temporary.
- Includes a simple erase mechanism.
- Maintains full backward compatibility with previous MultiMediaCard systems.
- Supports multiple command sets.
- Provides a possibility for the host to make sudden power failure safe-update operations for the data content.
- Enhanced power saving method by introducing a sleep functionality.
- Introduces Boot Operation Mode to provide a simple boot sequence method.
- Introduces Secure Erase & Trim to enhance data security.
- Signed access to a Replay Protected Memory Block.

5.2 Operation Modes

The eMMC devices has several operation modes:

- Inactive mode
  The device will enter inactive mode if either the device operating voltage range or access mode is not valid. The device can also enter inactive mode with the GO_INACTIVE_STATE command (CMD15). The device will reset to *Pre-idle* state with each power cycle.
- Boot mode
  The device will be in boot mode after a power cycle, the reception of CMD0 with an argument of 0xF0F0F0F0 or the assertion of a hardware reset signal.
- Card-identification mode
  The device will be in card-identification mode after the boot operation mode is finished or if the host and /or the device does not support the boot operation mode. The device will remain in this mode until the SET_RCA command (CMD3) is received.
- Data transfer mode
  The device will enter the data transfer mode once an RCA is assigned to it. The host will enter the data transfer mode after identifying the device on the bus.
- Interrupt mode
  The host and the device enter and exit the interrupt mode simultaneously. In interrupt mode there is no data transfer. The only message allowed is an interrupt service request from the device or the host.

5.3 Bus Protocol and Speed Mode

After a power-on reset, the host must initialize the device by a special message-based eMMC bus protocol. The eMMC bus operations always contain a command token and a response token, some operations have a data token. For the detailed coding scheme of the tokens, please refer to the corresponding eMMC standards.

The default eMMC transfer frequency is up to 26MHz, with the maximal transfer speed of 26MB/s.

If the maximal MMCA clock frequency for RH850/U2B series of 40 MHz is required, the eMMC device should be switched to high speed mode for this higher eMMC clock frequency.

To switch the device into high speed mode, the host should use CMD6 to set the bytes [185] HS_TIMING from register EXT_CSD register to 1_B.

5.4 State Machines

Table 5-1 shows the dependencies between bus modes, operation modes and device states. Each device state is associated with one bus mode and one operation mode.

Table 5-1 Overview of Device States and Bus modes

Device State	Operation Mode	Bus Mode
Inactive State	Inactive mode	Open drain
Pre-Idle State	Boot mode	
Pre-Boot State		
Idle State	Device identification mode	
Ready State		
Identification State		
Stand-by State	Data transfer mode	Push-pull
Sleep State		
Transfer State		
Bus-Test State		

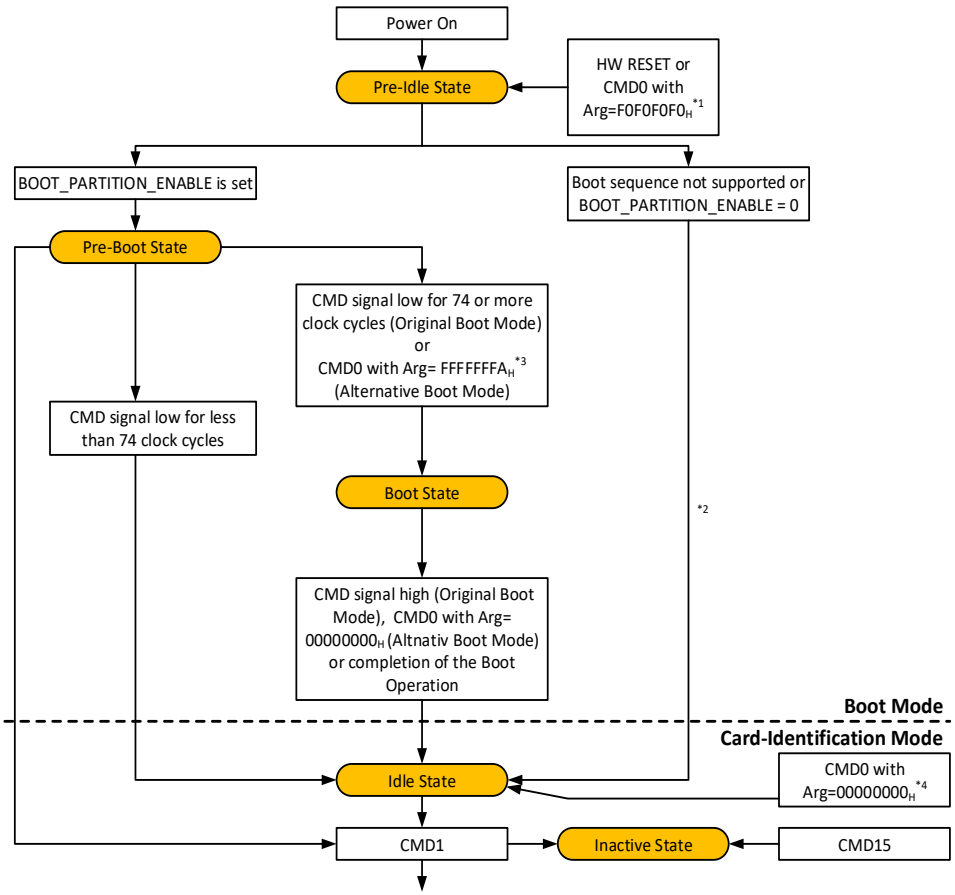
Sending-data State		
Receive-data State		
Programming State		
Disconnect State		
Boot State	Boot mode	
Wait-IRQ State	Interrupt mode	Open drain

5.4.1 Boot Mode

5.4.1.1 Alternative Boot Operation

The boot function described in this section is for devices with JEDEC Standard 4.4 and later. The compliant devices for the RH850/U2B MMCA module must show “1” on bit 0 in the Extended CSD byte [228].

Figure 5-2 shows the state diagram of the boot mode.



Notes: 1. If a hardware reset signal is detected or CMD0 with Arg=F0F0F0F0$_H$ is sent to the eMMC card, the card switches into Pre-Idle State from all states except in inactive state, the operation mode switches from any other modes to Boot Mode.

2. When the device is in alternate boot and all the data has been read from the boot partition, the device should automatically go to Idle State, but the host is still required to issue CMD0 with Arg=00000000$_H$.
3. Within 1 second after CMD0 with the argument of FFFFFFFA$_H$ is issued, the slave starts to send the first boot data to the master on the data token.
4. If CMD0 with Arg=00000000$_H$ is sent to MMC card, the card switches into Idle State from all states except in inactive state, the operation mode switches from any other modes to Card- Identification Mode.

Figure 5-2 Boot Mode

If boot acknowledge is enabled[*1], the slave must send the acknowledge pattern 010_B to the master within 50ms after the CMD0 with the argument of 0xFFFFFFFA is received, so that the master can recognize that the slave is operating. If boot acknowledge is disabled, the slave will not send out the acknowledge pattern 010_B. The master can terminate the boot mode by issuing CMD0. If the master issues CMD0 in the middle of a data transfer, the slave must terminate the data transfer or the acknowledge pattern within 2 clock cycles[*2].

If the master terminates boot mode between consecutive blocks, the slave must release the data line(s) within 2 clock cycles. Boot operation will be terminated when all contents of the enabled boot data are sent to the master. After boot operation is executed, the slave shall be ready for CMD1 operation and the master needs to start a normal MMC initialization sequence by sending CMD1.

If the CMD line is held LOW for less than 74 clock cycles after power-up before CMD1 is issued, or the master sends any normal MMC command other than CMD1 and CMD0 with argument 0xFFFFFFFA before initiating boot mode, the slave does not respond and will be locked out of boot mode until the next power cycle and the Idle State is entered. When BOOT_PARTITION_ENABLE bits are set and the master sends CMD1, the slave must enter Card-Identification Mode and respond to the command.

If the slave does not support boot operation mode, or the BOOT_PARTITION _ENABLE bit is cleared, the slave automatically enters the Idle State after power-on.

Notes: 1. Set BOOT_ACK bit, referring to Table 5-2.
2. One data cycle and end bit cycle, for detailed information, please refer to JESD84-A441.

5.4.1.2 Configuration of Boot Partition, Bus Width and Data Access

The boot partition, bus width and data access can be configured in the EXT_CSD register. The host can set the eMMC register by sending CMD6.

There are two partition regions. The minimum size of each boot partition is 128KB. The boot partition size is calculated as follows, with BOOT_SIZE_MULT from Extended CSD register byte [226]:

Maximum boot partition size = 128K byte x BOOT_SIZE_MULT

To configure the boot partition and data width, the protection of the boot configuration should be disabled by the EXT_CSD register byte [178]. The boot partition and data access can be individually selected in the byte [179] PARTITION_CONFIG bit [5:3] and bit [2:0]. The bus width can be selected in byte [177] bit [1:0].

Table 5-2 lists the boot mode configuration and corresponding bytes in the EXT_CSD register.

Table 5-2 Details of Boot Mode Configuration in EXT_CSD Register

EXT_CSD Field	Slice	Bit Name	Bit Position	Cell Type	Description	Set Value[*1]
PARTITION_CONFIG	[179]	BOOT_ACK	6	R/W/E	No boot acknowledge sent	**0_B**
					Boot acknowledge sent during boot operation	1_B
		BOOT_PARTITION_ENABLE	5 to 3	R/W/E	Boot not enabled	**000_B**
					Boot partition 1 enabled for boot	001_B
					Boot partition 2 enabled for boot	010_B
					User area enabled for boot	111_B
		PARTITON_ACCESS	2 to 0	R/W/E_P	No access to boot partition	**000_B**
					R/W boot partition 1	001_B
					R/W boot partition 2	010_B
					R/W Replay Protected Memory Block	011_B
					Access to General Purpose partition 1	100_B

					Access to General Purpose partition 2	101_B
					Access to General Purpose partition 3	110_B
					Access to General Purpose partition 4	111_B
BOOT_CONFIG_PROT	[178]	PERM_BOOT_CONFIG_PROT	4	R/W	Protection disabled	**0_B**
					Protection of boot configuration (PARTITION_CONFIG and BOOT_BUS_CONDITIONS) is permanently enabled	1_B
		PWR_BOOT_CONFIG_PROT	0	R/W/C_P	Protection disabled	**0_B**
					Protection of boot configuration (PARTITION_CONFIG and BOOT_BUS_CONDITIONS) from at this point until next power cycle or next H/W reset operation is enabled	1_B
BOOT_BUS_CONDITIONS	[177]	BOOT_MODE	4 to 3	R/W/E	Use single data rate + backward compatible timings in boot operation	**00_B**
					Use single data rate + high speed timings[*2] in boot operation mode	01_B
					Use dual data rate in boot operation	10_B
		RESET_BOOT_BUS_WIDTH	2	R/W/E	Reset bus width to ×1, single data rate and backward compatible timings after boot operation	**0_B**
					Retain BOOT_BUS_WIDTH and BOOT_MODE values after boot operation	1_B
		BOOT_BUS_WIDTH	1 to 0	R/W/E	×1 (sdr) or ×4 (ddr) bus width in boot operation mode	**00_B**
					×4 (sdr/ddr) bus width in boot operation mode	01_B
					×8 (sdr/ddr) bus width in boot operation mode	10_B
BOOT_WP	[173]	B_PWR_WP_DIS[*3]	6	R/W/C_P	Master is permitted to set B_PWR_WP_EN	0_B
					Disable B_PWR_WP_EN	1_B
		B_PREM_WP_DIS[*4]	4	R/W	Master is permitted to set B_PREM_WP_EN	0_B
					Disable B_PREM_WP_EN	1_B
		B_PREM_WP_EN[*5]	2	R/W	Boot region is not permanently write protected	0_B
					Boot region is permanently write protected	1_B
		B_PWR_WP_EN[*6]	0	R/W/C_P	Boot region is not power-on write protected	0_B
					Enable Power-On Period write protection to the boot area	1_B

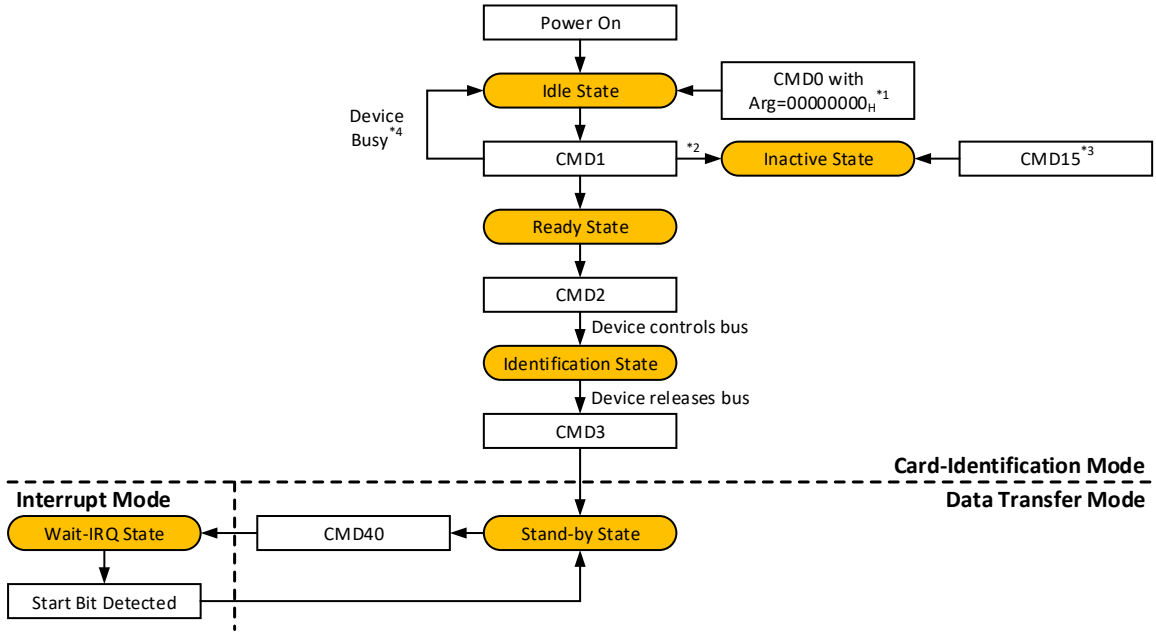
Notes: 1. The bolded values are the default settings after reset.

2. EXT_CSD register byte [228] bit 2 tells the master if the high-speed timing during boot is supported by the device.
3. This bit must be zero if B_PWR_WP_EN is set.
4. This bit must be zero if B_PERM_WP_EN is set. This bit has no impact on the setting of CSD register byte [13].
5. This bit must be zero if B_PERM_WP_DIS is set. This bit only indicates if permanent protection has been set specifically for the boot region. This bit may be zero if the whole card is permanently protected using CSD register byte [13].
6. This bit must be zero if B_PWR_WP_DIS (bit 6) is set.

5.4.2 Card-Identification Mode

In card identification mode, the host resets the card, validates the operation voltage range and access mode, identifies the card and assigns a Relative Card Address (RCA) to the card on the bus. All data communication in the Card Identification Mode uses the command line (CMD) only.

Figure 5-3 shows the state machine for card-identification mode.



Notes: 1. If CMD0 with Arg=00000000_H is sent to the MMC card, the card switches into Idle State from all states except from inactive state, and the operation mode switches from any other modes to Card-Identification Mode.
2. If the host cannot use a sector access mode (>2GB) in card, or the voltage range is not compatible, the card switches into Inactive State.
3. If CMD15 is sent to the MMC card, the card switches into Inactive State from all states except in inactive state, the operation mode switches from any other modes to Card-Identification Mode. The CMD15 is used when the host explicitly wants to deactivate a device.
   To access a device already in Inactive State, a hard reset must be done by switching the power supply off and back on.
4. If the slave is still working on its power-up/reset procedure and is not ready for communication, the host can get the state by reading the card power up status bit in the response of CMD1. In this case the host must repeat CMD1 until the card power up status bit is cleared.
   During the initialization procedure, the host is not allowed to change the operating voltage range or access mode setting. Such changes shall be ignored by the device. If there is a real change in the operating conditions, the host must reset the device and restart the initialization procedure.

Figure 5-3 Card-Identification Mode

The device moves into idle state within the following cases:

- After receiving a CMD0 with Arg=00000000_H GO_IDLE_STATE.
- After completing the boot operation.
- After a CMD line low for less than 74 clocks in pre-boot state.
- After power up if the device is not boot enabled.

After the bus is activated, the host will request the eMMC card to send its valid operation conditions. The response to CMD1 is the wired-and operation on the condition restrictions of all slaves in the system. Incompatible devices are sent into inactive state.

The host then issues the broadcast command CMD2, asking all devices for its unique CID numbers to identify the device. All unidentified devices, which are ready, simultaneously start sending their CID numbers serially, while bit-wise monitoring their outgoing bit stream. The devices which successfully sends its full CID number to the host goes into identification state.

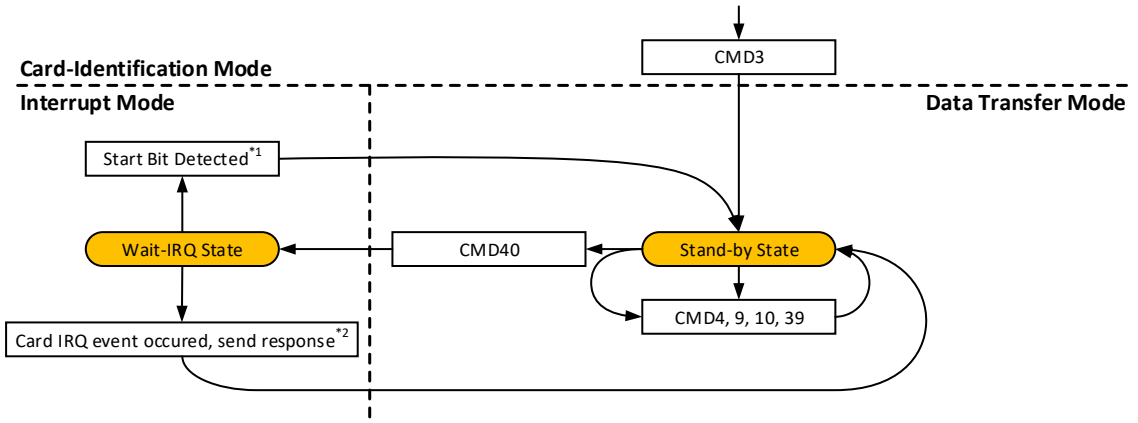
Then the host issues CMD3 to assign to this device a relative device address, which is shorter than CID and which will be used to address the device in the future data transfer mode. Once the RCA is received, the device state changes to the stand-by state, and the device does not react to further identification cycles.

The host repeats the identification process, until it receives a response to its identification command. The time-out condition to recognize completion of the identification process is the absence of a start bit for more than 5 clock cycles after sending CMD2.

5.4.3 Interrupt Mode

The interrupt mode enables the host to grant the transmission allowance to the slaves simultaneously. This mode reduces the polling load for the host and hence, the power consumption of the system, while maintaining adequate responsiveness of the host to a card request for service.

Figure 5-4 shows the transformation among the states in interrupt mode.



Notes: 1. Start bit '0' is detected on bus, and no internal interrupt events from the eMMC card occur.
2. An internal interrupt trigger event of the card occurs. After the interrupt, the response is received by the host, the host returns to Stand-By State.
If the host wants to terminate the interrupt mode before an interrupt response is received, it can generate a CMD40 response using the reserved RCA address 0000_H by himself, the card switches back to Stand-By State.

Figure 5-4 Interrupt Mode

5.4.4 Data Transfer Mode

5.4.4.1 Data Transfer Mode

The card specific data, e.g. block length, storage capacity, clock rate, etc., can be issued in data transfer mode; and the communication between the host and slave is also included in this mode.

Figure 5-5 shows the state machine of data transfer mode.

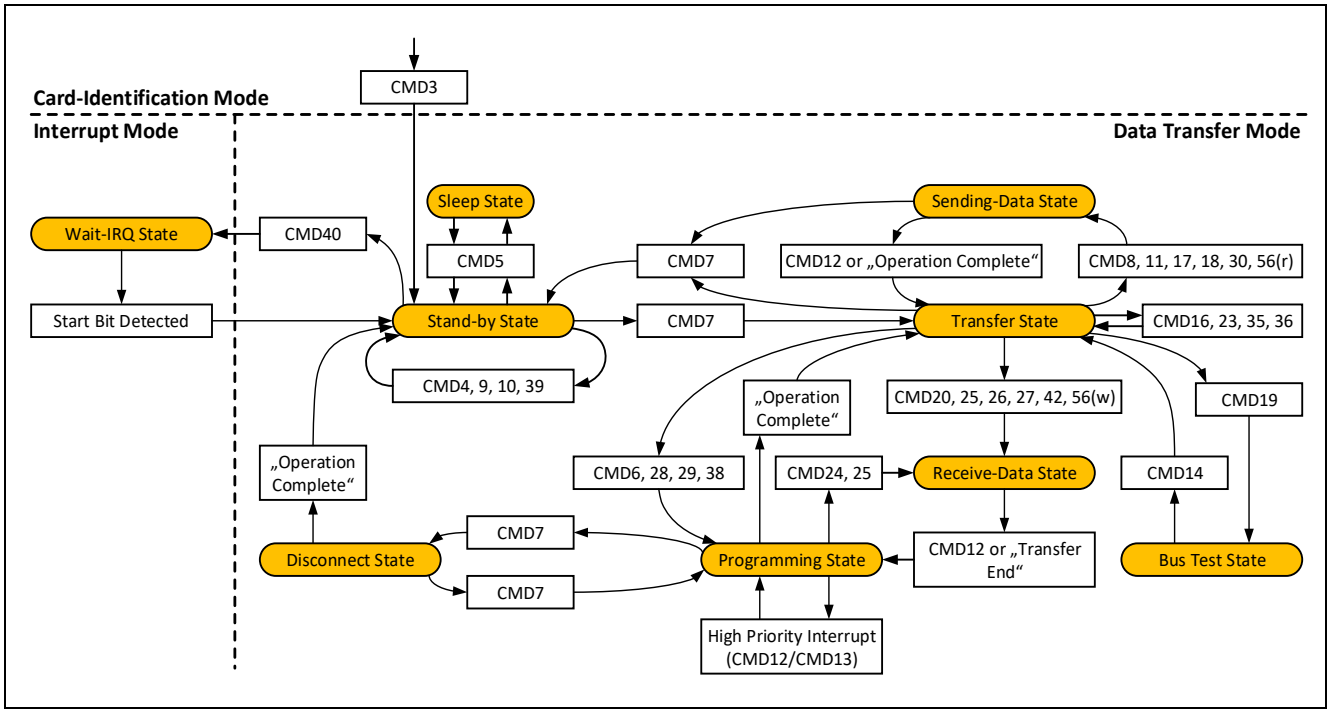


Figure 5-5 Data Transfer Mode

5.4.4.2 Command sets and extended settings

The eMMC card operates in a given command set, by default, after a power cycle, a reset by CMD0 with an argument of 0x00000000 or after a boot operation, using a single data line DAT0. The host can change the active command set by issuing the SWITCH command CMD6 with a certain access mode*Note. The supported command sets are defined in the EXT_CSD register.

The EXT_CSD register is divided into two segments, a properties segment and a modes segment. The properties segment contains information about the device capabilities, the modes segment reflects the current selected mode of the device.

The host reads the EXT_CSD register by issuing the CMD8, SEND_EXT_CSD command. The device sends the EXT_CSD register as a block of data, 512 bytes long. Any reserved, or write-only field, reads as ‘0’. The host can write the Modes segment of the EXT_CSD register by issuing a SWITCH command and setting one of the access modes. All three modes access and modify one of the EXT_CSD bytes, pointed to by the Index field.

The commands, their corresponding argument settings and responses are described in Section 5.5.

Note: In the Command set mode, the details of the access mode settings can be found in Table 5-3.

5.5 Commands and response

The MMCA module of the RH850/U2B devices are compliant with the JEDEC Standard JESD84-A441. All the commands are compliant also with the later JEDEC Standards, e.g. JESD84-B50.

5.5.1 Commands

All commands and responses are sent over the CMD line of the eMMC bus. The command transmission always starts with the left bit corresponding to the command codeword.

All commands have a fixed code length of 48 bits, including a start bit “0”, a transmission bit “1”, a command index, an argument, a CRC7 and an end bit “1”. On RH850/U2B devices (host), the command sequence is configured in the register MMCAnCE_CMD_SET.

Table 5-3 lists the commands with the corresponding argument, response and register settings in the MCU.

Table 5-3 Commands Settings for RH850/U2B Device

CMD	Setting Value of MMCAnCE_CMD_SET Register	Response	Argument	Abbreviation	Command Description
CMD0	00000000_H	—	[31:0] 00000000_H	GO_IDLE_STATE	Reset the card to idle state.
		—	[31:0] $F0F0F0F0_H$	GO_PRE_IDLE_STATE	Reset the card to pre-idle state.
	$400A0X0Y_H$[*1]	—	[31:0] $FFFFFFFA_H$	BOOT_INITIATION	Initiate alternative boot operation.
CMD1	01405000_H	R3	[31] Set to 0 [30:0] OCR bit 30 to 0	SEND_OP_COND	Ask the card, in idle state, to send its Operating Conditions Register contained in the response on the CMD line.
CMD2	02806000_H	R2	[31:0] stuff bits	ALL_SEND_CID	Ask the card to send its CID number on the CMD line.
CMD3	03400000_H	R1	[31:16] RCA [15:0] stuff bits	SET_RELATIVE_ADDR	Assign relative address RCA to the card.
CMD4	04000000_H	—	[31:16] DSR [15:0] stuff bits	SET_DSR	Program the DSR of the card.
CMD5	05600000_H	R1b	[31:16] RCA [15] Sleep/Awake [14:0] stuff bits	SLEEP_AWAKE	Toggle the card between Sleep state and Standby state.
CMD6	06400000_H	R1	[31:26] Set to 0 [25:24] Access[*4] [23:16] Index[*4] [15:8] Value [7:3] Set to 0 [2:0] CMD Set	SWITCH	Switch the mode of operation of the selected card or modifies the EXT_CSD registers.
	06600000_H	R1b			
CMD7	07400000_H	R1	[31:16] RCA [15:0] stuff bits	SELECT/DESELECT_CARD[*5]	Toggle a card between the stand-by and transfer states or between the programming and disconnect states. In both cases the card is selected by its own relative address and gets deselected by any other address; address 0 deselects the card.
	07600000_H	R1b			

CMD8	0848000X$_H$[*2]	R1	[31:0] stuff bits	SEND_EXT_CSD	The card sends its EXT_CSD register as a block of data.
CMD9	09806000$_H$	R2	[31:16] RCA [15:0] stuff bits	SEND_CSD	Addressed card sends its card-specific data CSD on the CMD line.
CMD10	0A806000$_H$	R2	[31:16] RCA [15:0] stuff bits	SEND_CID	Addressed card sends its card identification CID on CMD the line.
CMD12	0C400000$_H$	R1	[31:16] RCA3 [15:1] stuff bits [0] HPI	STOP_TRANSMISSION	Force the card to stop transmission. If HPI flag is set, the device shall interrupt its internal operations in a well-defined timing.
	0C600000$_H$	R1b			
CMD13	0D400000$_H$	R1	[31:16] RCA [15:1] stuff bits [0] HPI	SEND_STATUS	Addressed card sends its status register. If HPI flag is set the device shall interrupt its internal operations in a well-defined timing.
	0D600000$_H$	R1b			
CMD14	0E48040X$_H$[*3]	R1	[31:0] stuff bits	BUSTEST_R[*6*7]	A host reads the reversed bus testing data pat tern from a card.
CMD15	0F000000$_H$	—	[31:16] RCA [15:0] stuff bits	GO_INACTIVE_STATE	Set the card to inactive state.
CMD16	10400000$_H$	R1	[31:0] block length	SET_BLOCKLEN[*7*10]	Set the block length (in bytes) for all following block read/write commands. Default block length is specified in the CSD.
CMD17	1148000X$_H$[*2]	R1	[31:0] data address	READ_SINGLE_BLOCK[*8]	Read a block of the size selected by the SET_BLOCKLEN command.
CMD18	124A000X$_H$[*2]	R1	[31:0] data address	READ_MULTIPLE_BLOCK[*8]	**Pre-Defined:** Continuously transfer data blocks from card to host until the requested number of data blocks is transmitted.
	124B000X$_H$[*2]				**Open-ended:** Continuously transfer data blocks from card to host until interrupted by a stop command.
CMD19	134C010X$_H$[*2]	R1	[31:0] stuff bits	BUSTEST_W[*6*7]	A host sends the bus test data pattern to a card.
CMD23	17400000$_H$	R1	[31] Reliable Write Request[*11] [30:16] set to 0 [15:0] number of blocks	SET_BLOCK_COUNT	Define the number of read/write blocks and the reliable write parameter.
CMD24	184C000X$_H$[*2]	R1	[31:0] data address	WRITE_SINGLE_BLOCK[*9]	Write a block of the size selected by the SET_BLOCKLEN command.
CMD25	194E000X$_H$[*2]	R1	[31:0] data address	WRITE_MULTIPLE_BLOCK[*9]	**Pre-Defined:** Continuously write blocks of data until the requested number of blocks received.
	194F000X$_H$[*2]				**Open-ended:** Continuously write blocks of data until a STOP_TRANSMISSION follows.

CMD26	1A4C000X$_H$[*2]	R1	[31:0] stuff bits	PROGRAM_CID[*9]	Programming of the card identification register. This command shall be issued only once. The card contains hardware to prevent this operation after the first programming. Normally this command is reserved for the manufacturer.
CMD27	1B4C000X$_H$[*2]	R1	[31:0] stuff bits	PROGRAM_CSD [*9]	Programming of the programmable bits of the CSD.
CMD28	1C600000$_H$	R1b	[31:0] data address	SET_WRITE_PROT	If write protection feature is provided, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the CSD register.
CMD29	1D600000$_H$	R1b	[31:0] data address	CLR_WRITE_PROT	If the card provides write protection features, this command clears the write protection bit of the addressed group.
CMD30	1E48000X$_H$[*2]	R1	[31:0] write protect data address	SEND_WRITE_PROT[*8]	If the card provides write protection features, this command asks the card to send the status of the write protection bits.
CMD31	1F48000X$_H$[*2]	R1	[31:0] write protect data address	SEND_WRITE_PROT_TYPE	Send the type of write protection that is set for the different write protection groups.
CMD35	23400000$_H$	R1	[31:0] data address	ERASE_GROUP_START[*10]	Set the address of the first erase group within a range to be selected for erase.
CMD36	24400000$_H$	R1	[31:0] data address	ERASE_GROUP_END[*10]	Set the address of the last erase group within a continuous range to be selected for erase.
CMD38	26600000$_H$	R1b	[31] Secure request [30:16] set to 0 [15] Force garbage collect request [14:1] set to 0 [0] Identify write block for erase	ERASE	Erase all previously selected write blocks according to argument bits.

CMD39	27400000_H	R4	[31:16] RCA [15:15] register write flag [14:8] register address [7:0] register data	FAST_IO	Used to write and read 8-bit register data fields. The command addresses a card and a register and provides the data for writing if the write flag is set. The R4 response contains data read from the addressed register if the write flag is cleared to 0. This command accesses application dependent registers which are not defined in the Multimedia Card standard.
CMD40	28400000_H	R5	[31:0] stuff bits	GO_IRQ_STATE	Set the system into interrupt mode. No IRQ defined for the device, the device returns standby state when start bit “0” is detected.
	$284000C0_H$				Set the system into interrupt mode. IRQ defined for the device, the device returns standby state by card IRQ event.
CMD42	$2A4C000X_H$[*2]	R1	[31:0] stuff bits.	LOCK_UNLOCK[*7]	Used to set/reset the password or lock/unlock the card. The size of the data block is set by the SET_BLOCK_LEN command.
CMD55	37400000_H	R1	[31:16] RCA [15:0] stuff bits	APP_CMD	Indicate to the card that the next command is an application specific command rather than a standard command.
CMD56	$3848000X_H$[*2]	R1	[31:1] stuff bits. [0]: RD/WR	GEN_CMD	Used to get a data block[*12] from the card for general purpose / application specific commands.
	$384C000X_H$[*2]				Used to transfer a data block[*12] to the card the card for general purpose / application specific commands.

Notes: 1. This command is used for alternative boot operation. In this table, it is ordered by CMD0 because of the command number; In RH850/U2B Hardware User’s Manual *R01UH0923EJxxxx*, this command is called “Boot operation” in *Table 18.30 ‘Setting Values of MMCAnCE_CMD_SET’*.
X=$00X0_B$, the bit “X” enable/disables the boot acknowledge on host. If the boot acknowledge is enabled, the corresponding configuration to the card register EXT_CSD should be done. For details, please refer to *Section 5.4.1.2 ‘Configuration of Boot Partition, Bus Width and Data Access’*.
Y=$00XX_B$, the last 2 bits select the data bus width.

2. X=$00XX_B$, the last 2 bits select the data bus width. If the non-default bus width is selected, the connected eMMC device should also be configured to the same bus width in the extended CSD register.
3. X=$10XX_B$, the last 2 bits select the data bus width. If the non-default bus width is selected, the connected eMMC device should also be configured to the same bus width in the extended CSD register.
4. The EXT_CSD register access mode can be selected in bits [25:24] Access with the following values:
   - 00_B, Command Set Mode, the command set is changed according to the argument [2:0] CMD Set field.
   - 01_B, Set Bits Mode, the bits in the pointed byte are set according to “1” in Value field.

- 10_B, Clear Bits Mode, the bits in the pointed byte are cleared according to “1” in Value field.
- 11_B, Write Byte Mode, the Value field is written into the pointed byte.

The bits [23:16] Index specify the pointed byte.

5. If the host switches another device from stand-by state to transfer state, using CMD7, the programming operation will be not terminated. The device will switch to disconnect state and release the DAT0 line.
   A device can be reselected while in the disconnect state, using CMD7. In this case the device will move to the programming state and reactivate the busy indication.
6. To get a true result for the bus testing procedure, it is recommended to set up the clock frequency used for data transfer.
7. These commands are not allowed once the device is configured to operate in dual data rate mode and shall not be executed but regarded as illegal commands.
8. All data read commands can be aborted any time by the stop command CMD12. The data transfer will be terminated, and the device will return to the transfer state.
   No read commands are allowed while the device is programming.
9. All data write commands can be aborted any time by the stop command CMD12. The write commands must be stopped prior to deselecting the device by CMD7.
10. Parameter set commands are not allowed while the device is programming.
11. If reliable write is enabled for the pre-defined multi-block write, the old data pointed to by a logical address must remain unchanged until the new data written to same logical address has been successfully programmed. The data addressed by operation will never contain undefined data, even if unexpected reset or power loss occurs during the programming.
12. The size of the data block shall be set by the SET_BLOCK_LEN command.

5.5.2 Response

All responses are sent via the command line CMD. The response transmission always starts with the left bit corresponding to the response codeword.

The code length depends on the response type. A response always starts with a start bit “0”, followed by the transmission bit “0”. The responses except for the type R3 are protected by a CRC7. Every response codeword is terminated by the end bit “1”.

At the RH850/U2B series, the information bits of the responses can be read from the registers MMCAnCE_RESP3 to 0.

There are five types of responses, which is mentioned in Table 5-3:

- R1/R1b, Normal response command.

  The code length is 48 bits, in which the bit 39 to 8 contains the card status after the corresponding commands. This 32-bit status can be read in the MMCAnCE_RESP0 register.

  R1b is identical to R1 with an optional busy signal transmitted on the data DAT0. For details of the card status please refer to Table 5-4.

- R2, CID/CSD register.

  The code length is 136 bits. Bit 127 to 1 includes the CID or CSD register value including internal CRC7. These values are stored in the registers MMCAnCE_RESP3 to 0.

- R3, OCR register.

  The code length is 48 bits. The content of the OCR register is sent in bit 39 to 8, this 32-bit value is transmitted after the OCR related commands and stored in the MMCAnCE_RESP0 register.

- R4, Fast I/O.

  The code length is 48 bits. The bits 39 to 8 is the argument field, which contains the RCA number, status bit, as well as the register configuration. This field is stored in the MMCAnCE_RESP0 register.

- R5, Interrupt request.

  The code length is 48 bits. The bits 39 to 8 is the argument field, which includes the RCA number. This field can be stored in the MMCAnCE_RESP0 register.

Table 5-4 Card Status

Bit Name	Bit Position	Description	Value
ADDRESS_OUT_OF_RANGE	31	No address range error occurs.	0_B
		Invalid address range is configured or accessed by read/write.	1_B
ADDRESS_MISALIGN	30	No address misalign error occurs.	0_B
		Address misalign is configured or accessed by read/write.	1_B
BLOCK_LEN_ERROR	29	No block length error occurs.	0_B
		The argument of CMD6 or the pre-defined block length is invalid.	1_B
ERASE_SEQ_ERROR	28	No sequence error of erase command occurs.	0_B
		Error occurs in sequence of erase command.	1_B
ERASE_PARAM	27	No erase parameter error occurs.	0_B
		Invalid selection of erase parameters is done.	1_B
WP_VIOLATION	26	No write protection violation error occurs.	0_B
		Error occurs by programing protected block.	1_B
CARD_IS_LOCKED	25	Card is unlocked.	0_B
		Card is locked.	1_B
LOCK_UNLOCK_FAILED	24	No lock/unlock error occurs.	0_B
		A sequence or password error is detected by lock/unlock card command.	1_B
COM_CRC_ERROR	23	No CRC error occurs.	0_B
		The CRC check of previous command failed.	1_B
ILLEGAL_COMMAND	22	No illegal commands are discovered.	0_B
		Illegal command sent for the card state.	1_B
CARD_ECC_FAILED	21	No card ECC error occurs.	0_B
		Card internal ECC was applied but failed to correct the data.	1_B
CC_ERROR	20	No error occurs.	0_B
		A card error NOT related to the host command occurs.	1_B
ERROR	19	No error occurs.	0_B
		A generic error related to execution of the last host command occurs.	1_B
CID/CSD_OVERWRITE	16	No error occurs.	0_B
		• The CID register has been already written and cannot be overwritten. • Or the read only section of the CSD does not match the card content. • Or an attempt to reverse the copy or permanent write protection bits was made.	1_B
WP_ERASE_SKIP	15	No error occurs.	0_B
		Only partial address space was erased due to existing write protected blocks.	1_B
ERASE_RESET	13	No error occurs.	0_B
		Erase sequence was cleared before executing because a no-erase-sequence command was received.	1_B
CURRENT_STATE	12 to 9	Idle State	0000_B
		Ready State	0001_B

		Identification State	0010_B
		Stand-by State	0011_B
		Transfer State	0100_B
		Sending-data State	0101_B
		Receive-data State	0110_B
		Programming State	0111_B
		Disconnect State	1000_B
		Bus-Test State	1001_B
		Sleep State	1010_B
READY_FOR_DATA	8	The bus is NOT ready for data transfer.	0_B
		The bus is ready for data transfer.	1_B
SWITCH_ERROR	7	No error occurs.	0_B
		The card did not switch to the expected mode as requested by the SWITCH command.	1_B
URGENT_BKOPS	6	No error occurs.	0_B
		The card needs to perform background operations urgently. The host can check EXT_CSD field BKOPS_STATUS for the detailed level.	1_B
APP_CMD	5	Application-specific command is disabled.	0_B
		Application-specific command is enabled.	1_B

For timings of the commands and responses, please refer to JEDEC Standard JESD84-A441 or later versions.

5.6 The JEDEC Standard Comparation (eMMC 4.41 vs. eMMC 5.0)

Table 5-5 contains the difference between eMMC 4.41 and eMMC 5.0.

Normally, the cards with the later standard version are compliant with the earlier versions. The additional functions lead to the different register structure and configuration on the card. This difference is described in Section 5.7.

Table 5-5 Functional Difference of JESD84-A441 and JESD84-B50

Functions	eMMC Version 4.41	eMMC Version 5.0
SDR/DDR	SDR and DDR	SDR and DDR
Bus Speed Mode	• Backward Compatibility with legacy MMC card • High Speed	• Backward Compatibility with legacy MMC card • High Speed • HS200 • HS400
Hardware Reset	Available	Available
Partitioning	• General Purpose • Enhanced User Data	• General Purpose • Enhanced User Data • Extended Partition Attribute
Security	• Erase • Trim • Secure Erase • Secure Trim	• Erase • Trim • Sanitize • Secure Erase • Secure Trim
Discard	Available	Available
High Priority Interrupt	Available	Available
Background Operation	Available	Available

Write Reliability	Available	Available
Enhanced Reliable Write	Available	Available
Packed Command	Available	Available
Data Tag	−	Available
Dynamic Capacity	−	Available
Context Management	−	Available
Real Time Clock Info	−	Available
Thermal Spec.	−	Available
Power Off Notification	−	Available
Boot Area Protection	Available	Available
Large Sector Size	Available	Available
Cache Memory	−	Available
Field Firmware Update	−	Available
Production State Awareness	−	Available
Secure Removal Type	−	Available
Device Health Report	−	Available

5.7 eMMC registers

The following information registers in Table 5-6 provide a possibility to read the card status or configure the eMMC card.

As listed in the table, the OCR, CID and CSD registers carry the card/content specific information. The RCA and DSR registers store the configuration parameters for the card. The EXT_CSD registers contains the card or transmission information, as well as the configuration parameters. The values of these registers can be accessed or is required by the related commands.

Table 5-6 eMMC registers

Register Name	Width in bytes	Description	Related Commands
CID	16	Card Identification number, a card individual number for identification.	CMD2, 10, 26
RCA	2	Relative Card Address, is the card system address, dynamically assigned by the host during initialization.	CMD3, 5, 7, 9, 10, 12, 13, 15, 39, 55
DSR	2	Driver Stage Register, to configure the card's output drivers.	CMD4
CSD	16	Card Specific Data, information about the card operation conditions.	CMD9, 27
OCR	4	Operation Conditions Register. Used by a special broadcast command to identify the voltage type of the card.	CMD1
EXT_CSD	512	Extended Card Specific Data. Contains information about the card capabilities and selected modes.	CMD6, 8

For detailed register fields, size and configuration, please refer to the JEDEC Standard JESD84-A441 *Section 8 'Card Registers'*, or the related section in later JEDEC Standard versions.

Comparing eMMC4.41 and eMMC5.0, the register structures are almost the same for the CID, RCA, DSR, CSD, and OCR registers.

The extended CSD EXT_CSD register contains most differences:

- The non-reserved areas keep the same structure, some bytes contain more selections in eMMC5.0, e.g. HS_TIMING.
- The reserved areas of this register in eMMC 4.41 are used for the parameter or configuration of security, FFU etc. in eMMC 5.0.

6. Port Setup for MMCA Module

This section provides a description of the port setups for MMCA application.

6.1 External Input / Output

Table 6-1 lists the external input / output pins of the MMCA module on the RH850/U2B device, which are used for an eMMC application and are connected to the card.

Table 6-1 External Input / Output and the Corresponding Pin Function of MMCA

Signal Name	Description	Corresponding PIN	Port Number	Port Function	I/O Direction
MMCA0CLK	MMCA Clock Signal	P13	0	Alternative function 5*1	Output
MMCA0CMD	MMCA Command/Response Line	P14	5	Alternative function 8*2	Input / Output
MMCA0DAT0	MMCA Data Line	P13	1	Alternative function 5	
MMCA0DAT1			2		
MMCA0DAT2			3		
MMCA0DAT3		P14	1	Alternative function 7	
MMCA0DAT4			0		
MMCA0DAT5			2		
MMCA0DAT6			3		
MMCA0DAT7			4	Alternative function 3	
$\overline{\text{RESETOUT}}$	Reset output to reset external devices at the same time as a reset is generated inside the MCU	P27	0	Special I/O Function*3	Output

Notes: 1. The corresponding pins operate in alternative mode software I/O control, the pins operate as alternative functions, and the input / output direction is selected by setting PMn.PMn_m bit to 1 / 0. To configure the operation mode, the PMCn, PIPCn, PFCEAEn, PFCAEn, PFCEn, and PFCn registers should be configured. Table 6-2 provides an overview of the register settings. For detailed information, please refer to the HW User's Manual *R01UH0923EJxxxx Section 2 'Pin Functions'*.

2. The corresponding pins operate in alternative mode direct I/O control, the pins operate as alternative functions, and the input / output direction is selected by the peripheral function. To configure the operation mode, the PMCn, PIPCn, PFCEAEn, PFCAEn, PFCEn, and PFCn registers should be configured. Table 6-3 provides an overview of the register settings. For detailed information, please refer to the HW User's Manual *R01UH0923EJxxxx Section 2 'Pin Functions'*.
3. The pin drives out low level to external device during and after reset which is effective for any kind of resets (except DeepSTOP Reset, Module Reset and JTAG Reset), and keeps driving out low level until the related registers are changed by user program.

Table 6-2 Alternative Function Selection (Software I/O Control)

Alternative-Function	Bit Name					
	PMCn_m	PIPCn_m	PFCEAEn_m	PFCAEn_m	PFCEn_m	PFCn_m
Alternative function mode 1	1	0	0	0	0	0
Alternative function mode 2						1
Alternative function mode 3					1	0
Alternative function mode 4						1
Alternative function mode 5				1	0	0
Alternative function mode 6						1
Alternative function mode 7					1	0
Alternative function mode 8						1
Alternative function mode 9			1	0	0	0
Alternative function mode 10						1
Alternative function mode 11					1	0
Alternative function mode 12						1
Alternative function mode 13				1	0	0
Alternative function mode 14						1
Alternative function mode 15					1	0
Alternative function mode 16						1

Table 6-3 Alternative Function Selection (Direct I/O Control)

Alternative-Function	Bit Name					
	PMCn_m	PIPCn_m	PFCEAEn_m	PFCAEn_m	PFCEn_m	PFCn_m
Alternative function mode 1	1	1	0	0	0	0
Alternative function mode 2						1
Alternative function mode 3					1	0
Alternative function mode 4						1
Alternative function mode 5				1	0	0
Alternative function mode 6						1
Alternative function mode 7					1	0
Alternative function mode 8						1
Alternative function mode 9			1	0	0	0
Alternative function mode 10						1
Alternative function mode 11					1	0
Alternative function mode 12						1
Alternative function mode 13				1	0	0
Alternative function mode 14						1
Alternative function mode 15					1	0
Alternative function mode 16						1

6.2 Port Output Buffer Drive Strength

Generally, the ports need more current to keep the signal integrity at high frequencies. Therefore, it is recommended to set all the output pins to the highest drive strength. In this case it is the MMCA outputs drive strength 2, using the PUCCn and PDSCn registers.

For details, please refer to the RH850/U2B Hardware User's Manual *R01UH0923EJxxxx Table 2.37 'Port Output Buffer Drive Strength Selection'*.

6.3 Port Write Protection

Before writing to the registers above, the write process must be enabled for port by setting the port key code protection register PKCPROT and port write enable register PWE.

For details, please refer to the RH850/U2B Hardware User's Manual *R01UH0923EJxxxx*.

7. Clock Setup for MMCA Module

The communication and register access clock for the MMCA module is CLK_HSB with the maximum frequency of 80 MHz.

For detailed information of clock supply and the corresponding settings, please refer to the RH850/U2B Hardware User's Manual *R01UH0923EJxxxx*.

To enable the clock connections to the peripherals, the bits MSR_MMCA.MS_MMCA_0 should be cleared to 0.

Before writing to this register, the write protection must be disabled by setting $A5A5A501_H$ to the key code protection register MSRKCPROT. If the write protection is required, please write $A5A5A500_H$ to register MSRKCPROT after the clock setup.

8. SW Procedure and Module Setup for Typical eMMC Operations

This section provides a description of the software procedure and the related configurations. In this section, the host relates to the RH850/U2B device.

For the detailed information of the registers which are mentioned in this section refer to the RH850/U2B Hardware User's Manual *R01UH0923EJxxxx Section 20.3 'Registers' (Multi Media Card Interface)*.

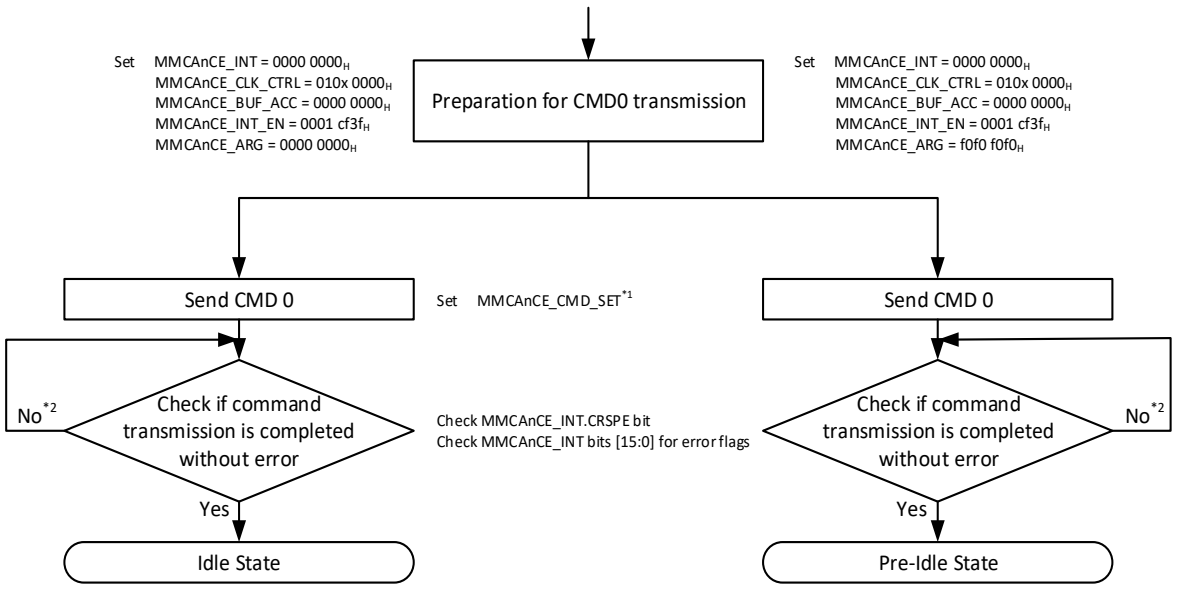
8.1 Card Reset

The host may reset the card by the operations listed in Table 8-1.

Table 8-1 Card Reset Operations

Host Operations		Reset Type	State after Reset
Switch power off and back on	Boot enabled	Hardware	Pre-Idle State
	Boot disabled		Idle State
Output a HW reset signal to the card			Pre-Idle State
Send CMD0	With argument 00000000_H	Software	Idle State
	With argument $F0F0F0F0_H$		Pre-Idle State

Figure 8-1 shows the software reset procedure by sending CMD0.

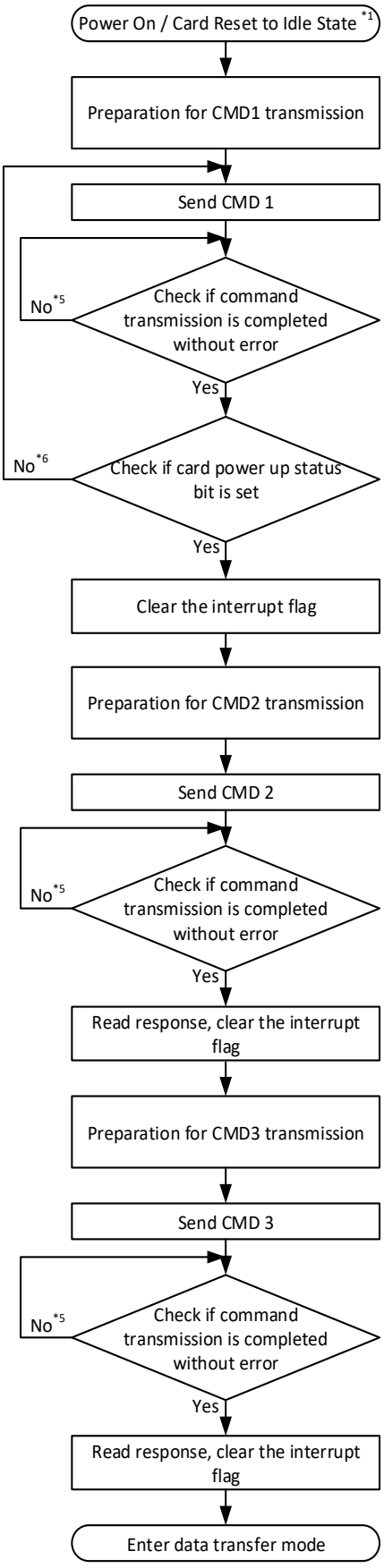


Notes: 1. For detailed register settings, please refer to Table 5-3 in *Section 5.5.1 'Commands'*.
2. If the MMCAnCE_INT.CRSPE bit is not set within a defined time, or any error flag is set, the flow should go to an error/timeout processing.
For details of the timeout setting, please refer to RH850/U2B Hardware User's Manual *R01UH0923EJxxxx Section 20.3.7 'MMCAnCE_CLK_CTRL — MMCAn Clock Control Register'*.
For details of the error flags, please refer to RH850/U2B Hardware User's Manual *R01UH0923EJxxxx Section 20.3.13 'MMCAnCE_INT — MMCAn Interrupt Flag Register'*.

Figure 8-1 Software Flow of Card Reset with CMD0

8.2 Card Identification

Regarding to the state machines on Section 5.4, after power up or card reset to idle state, the card should be identified before entering the data transfer mode. The host sends CMD 1, 2 and 3 to ensure the power up status, check the identification information and assign a short card number to the connected device for the further application. Figure 8-2 shows the host software procedure for the card identification.



Notes: 1. For the detailed information and procedure, please refer to the *Section 8.1 'Card Reset'*.
2. The clock in identification mode must be maximal 400 kHz. The clock divider here provides the

minimal setting for the maximal clock frequency for this mode.
The MMCA clock output from the host can be calculated according to the RH850/U2B Hardware User's Manual *Section 20.3.7 'MMCAnCE_CLK_CTRL — MMCAn Clock Control Register'*.

3. According to the Table 5-3, the argument is set to the OCR register value without its bit [31].
   The value here is related to the device SFEM4096B1EA1TO-I-GE-111-E02 from the Swissbit EM-26 Series on the Renesas extension eMMC/SFMA Board. If another device is connected in the application, please check the OCR register value in the device data sheet.
4. For detailed register settings, please refer to the Table 5-3 in *Section 5.5.1 'Commands'*.
5. If the MMCAnCE_INT.CRSPE bit is not set within a defined time, or any error flag is set, the flow should go to an error/timeout processing.
   For details of the timeout setting, please refer to the RH850/U2B Hardware User's Manual *R01UH0923EJxxxx Section 20.3.7 'MMCAnCE_CLK_CTRL — MMCAn Clock Control Register'*.
   For details of the error flags, please refer to the RH850/U2B Hardware User's Manual *Section R01UH0923EJxxxx 18.3.13 'MMCAnCE_INT — MMCAn Interrupt Flag Register'*.

Figure 8-2 Software Flow of Card Identification

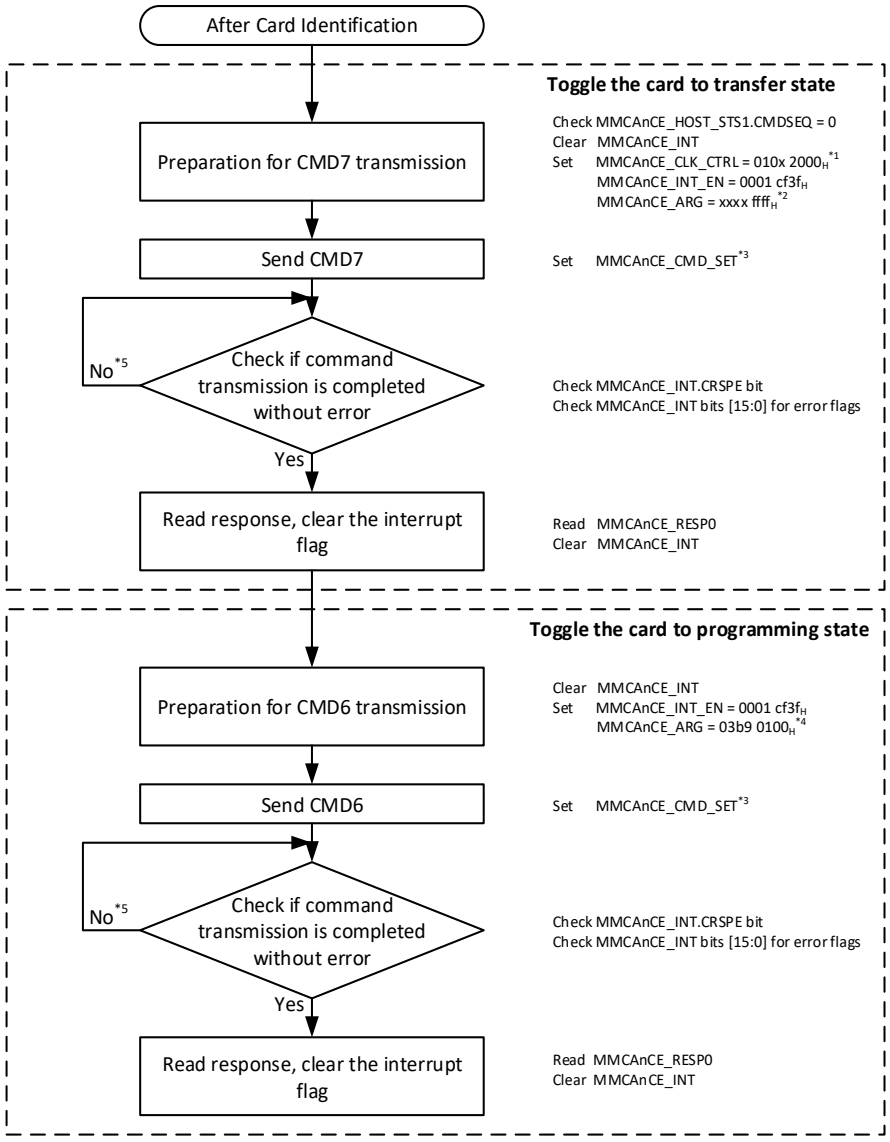
8.3 Configuration Card Register and Basic Settings

According to Section 5.4, the operations such as register configuration, settings of transfer, data transmission etc., can be carried out in Card-Identification Mode or after a card is identified.

Most of the parameters of the card configuration and transmission settings are contained in the card registers.
Some of the register configuration can be made by accessing the registers using a simple command with corresponding arguments, such as setting RCA and DSR registers.
The other registers, such as CID, CSD and EXT_CSD, which contain larger data width, which need a data transmission or a programming process to write, can be configured/programmed after the card is toggled by a CMD7 into transfer state. The flow to toggle the card to transfer state is contained in the first dashed box of Figure 8-3, the figure shows an example to configure the high-speed mode in the EXT_CSD register on the card. For other configurations to the EXT_CSD register, e.g. boot settings and command set settings etc., the configuration and procedure is similar, while other corresponding arguments are required. To program the CID or CSD registers, the related CMD26 or CMD27 should be executed after the card enters the transfer state.

There are also some basic settings without writing the card registers, which are required before certain command is sent, such as the block information settings for data transfer, and the start/end address settings for the erase operation.
Figure 8-4 shows the flow of these basic settings.

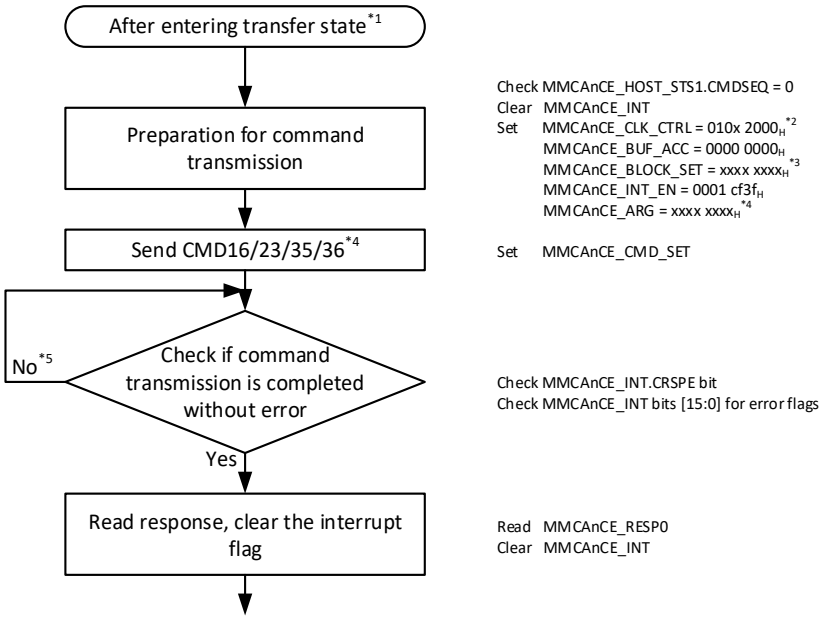
For the detailed setting procedure, which is not described in this section, please refer to the state machine diagramed in Figure 5-5 and commands settings described in Table 5-3.



Notes: 1. After the card finished the identification and entered the date transfer mode, the maximal allowed clock for the transmission is updated to the default frequency of 26MHz.
Considering the MMCA peripheral clock and its clock divider setting on RH850/U2B devices, in this case, the maximal communication clock of the module can be set to 20 MHz here, thus, the minimum setting of the clock divider “x” here should be 1_H.

2. “xxxx” should be filled with the RCA number of the card which is required from the host. This number is set by CMD3, for details, please refer to Figure 8-2.
3. For detailed register settings, please refer to Table 5-3 in *Section 5.5.1 ‘Commands’*.
4. The access mode, pointed byte, setting value and command set can be configured in the 32-bit argument before the CMD 6 is sent.
Here 03B9 0100_H is used to switch the default transfer speed mode to high-speed mode.
5. If the MMCAnCE_INT.CRSPE bit is not set within a defined time, or any error flag is set, the flow should go to an error/timeout processing.
For details of the timeout setting, please refer to RH850/U2B Hardware User’s Manual *R01UH0923EJxxxx Section 20.3.7 ‘MMCAnCE_CLK_CTRL — MMCAn Clock Control Register’*.
For details of the error flags, please refer to RH850/U2B Hardware User’s Manual *Section R01UH0923EJxxxx 18.3.13 ‘MMCAnCE_INT — MMCAn Interrupt Flag Register’*.

Figure 8-3 Software Flow to Set High-Speed Mode



Notes: 1. The card might be toggled by CMD7 to transfer state after identification, abort by CMD12, or returned from any “operation complete”, for details, please refer to Figure 5-5.

2. If high-speed mode is not set, the minimum setting of the clock divider should be 1_H here for “x”, enabling 20MHz, the maximal transfer frequency the host can provide.
   If high-speed mode is set, the minimum setting of the clock divider should be 0_H here for “x”, enabling 40MHz, the maximal transfer frequency the host can provide.
3. To set the block length for any data transfer operation, or block count for pre-defined multi-block data transfer, the CMD16 or CMD23 should be sent to card.
   On the host side, the register MMCA_BLOCK_SET should be configured, it contains the block length setting in bit 15 to 0; and the block count setting in bit 31 to 16.
4. CMD16 and CMD23 set the block parameters, CMD 35 and CMD36 set the start and end address to erase.
   For detailed register settings and corresponding arguments, please refer to Table 5-3 in *Section 5.5.1 ‘Commands’*.
5. If the MMCAnCE_INT.CRSPE bit is not set within a defined time, or any error flag is set, the flow should go to an error/timeout processing.
   For details of the timeout setting, please refer to RH850/U2B Hardware User’s Manual *R01UH0923EJxxxx Section 20.3.7 ‘MMCAnCE_CLK_CTRL — MMCAn Clock Control Register’*.
   For details of the error flags, please refer to RH850/U2B Hardware User’s Manual *Section R01UH0923EJxxxx 18.3.13 ‘MMCAnCE_INT — MMCAn Interrupt Flag Register’*.

Figure 8-4 Software Flow for Block or Erase Settings

8.4 Read Card Register

The host can read the card register to get card information for further operations.

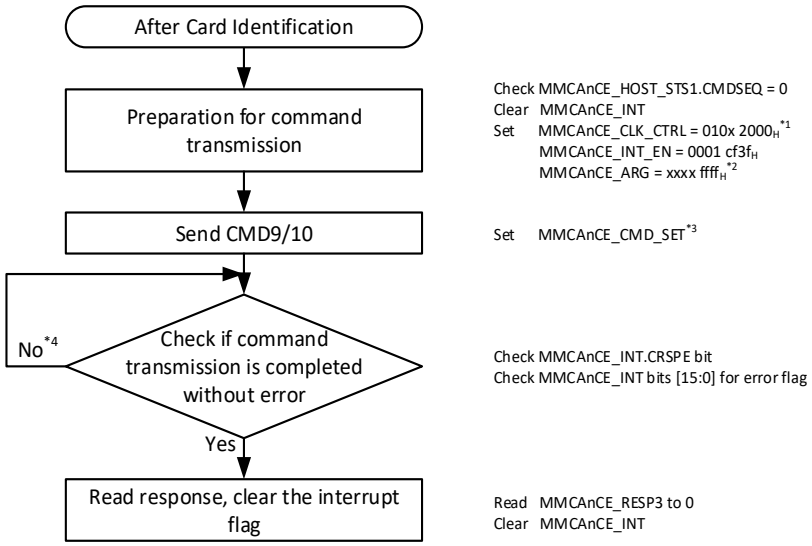
The read procedure is different depending on the target register:

- For registers like CID and CSD, the register value is transferred on the command line, and stored in the response registers MMCAnCE_RESP3 to 0.

  Figure 8-5 provides the flow of this read operation.

- For register EXT_CSD, the register contains 512 bytes, therefore, its value is transferred on the data line.

  Figure 8-6 describes the flow to read the EXT_CSD register.

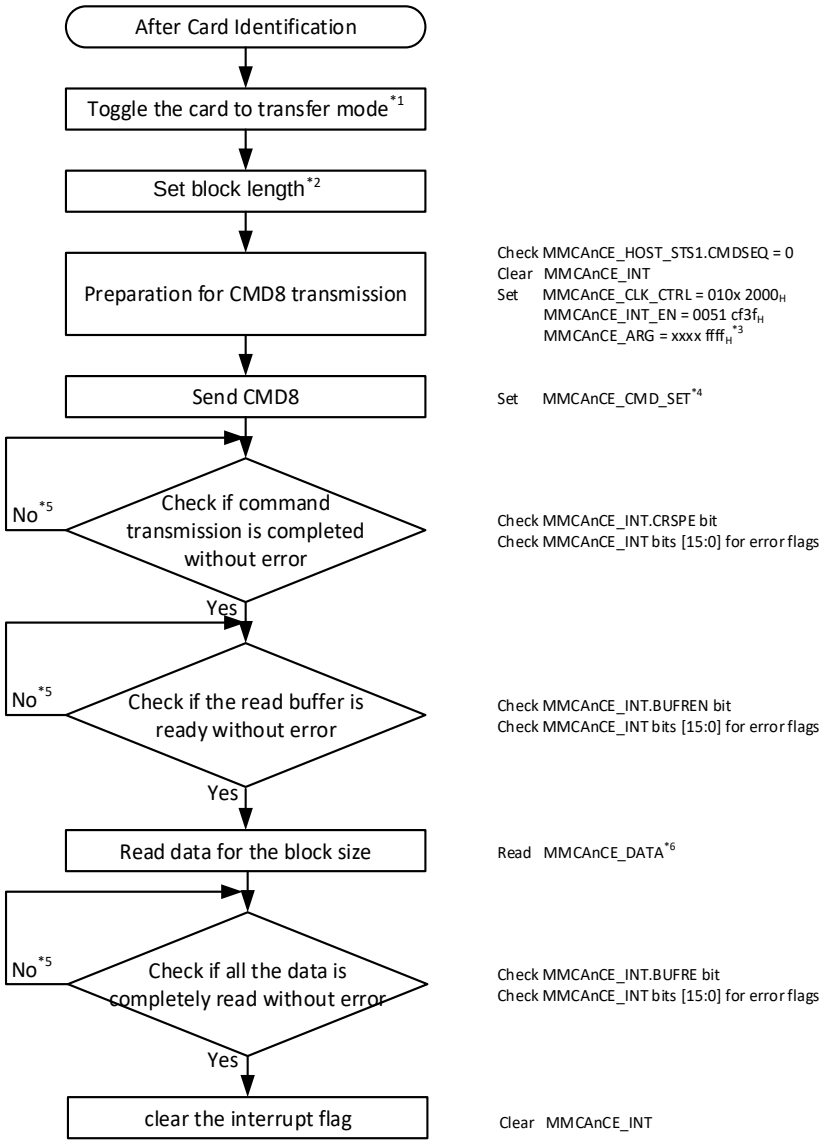


Notes: 1. After the card finished identification and entered the date transfer mode, the maximal allowed clock for the transmission is updated to the default frequency of 26MHz.
Considering the MMCA peripheral clock and its clock divider setting on RH850/U2B devices, in this case, the maximal communication clock the module can be set is 20 MHz here, thus, the minimum setting of the clock divider "x" here should be 1_H.
2. "xxxx" should be filled with the RCA number of the card which is required from the host. This number is set by CMD3, for details, please refer to Figure 8-2.
3. For detailed register settings, please refer to Table 5-3 in *Section 5.5.1 'Commands'*.
4. If the MMCAnCE_INT.CRSPE bit is not set within a defined time, or any error flag is set, the flow should go to an error/timeout processing.
For details of the timeout setting, please refer to RH850/U2B Hardware User's Manual *R01UH0923EJxxxx Section 20.3.7 'MMCAnCE_CLK_CTRL — MMCAn Clock Control Register'*.
For details of the error flags, please refer to RH850/U2B Hardware User's Manual *Section R01UH0923EJxxxx 18.3.13 'MMCAnCE_INT — MMCAn Interrupt Flag Register'*.

Figure 8-5 Software Flow to Read CID / CSD register

The value of CID or CSD register is transferred by a 17-byte response, the contained 128 information bits are stored within the response registers of the host:

- MMCAnCE_RESP3 stores the bit 127 to 96 of the response.
- MMCAnCE_RESP2 stores the bit 95 to 64 of the response.
- MMCAnCE_RESP1 stores the bit 63 to 32 of the response.
- MMCAnCE_RESP0 stores the bit 31 to 0 of the response, in which bit 0 is an always-"1" end bit, from bit 1 on starts the contents of the CID / CSD register.

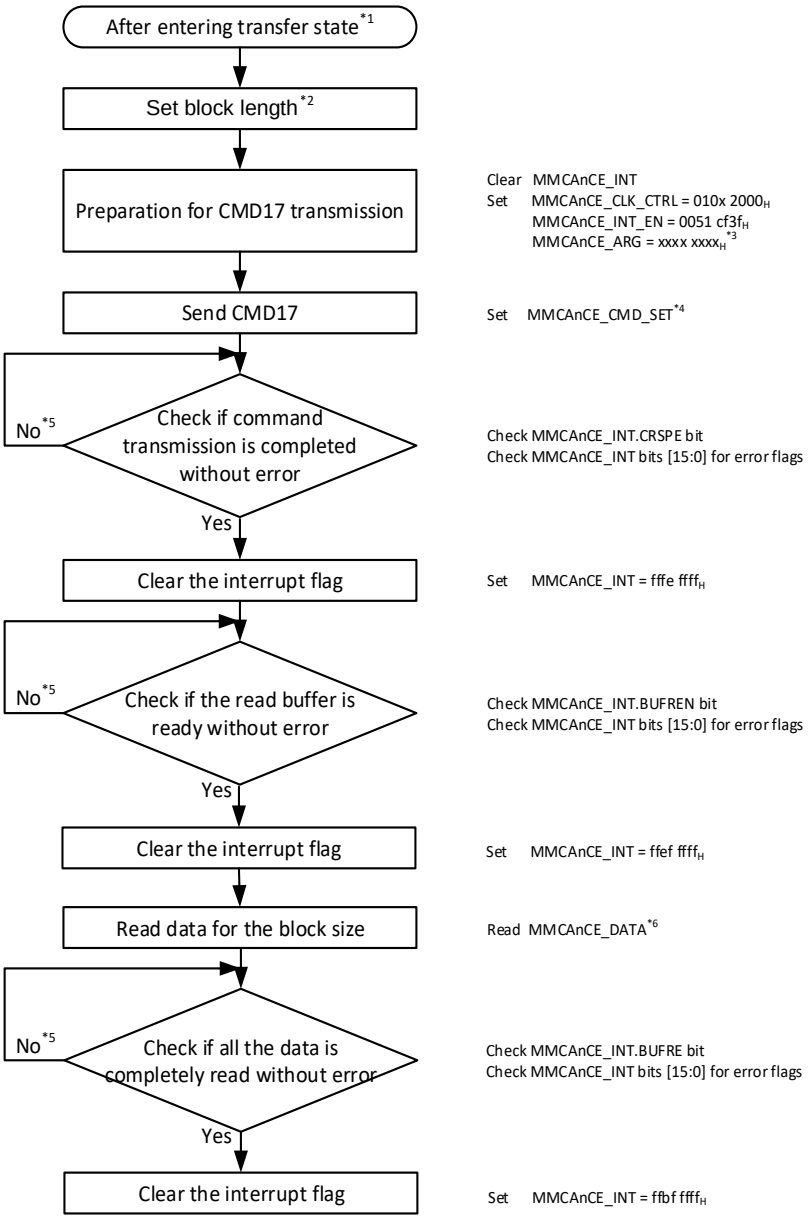


Notes: 1. For details, please refer to the dashed box “toggle the card to transfer state” in Figure 8-3.
2. For details, please refer to the Figure 8-4. It is recommended to set the block length as 200_H to read the complete register.
3. “xxxx” should be filled with the RCA number of the card which is required from the host. This number is set by CMD3, for details, please refer to Figure 8-2.
4. For detailed register settings, please refer to Table 5-3 in *Section 5.5.1 ‘Commands’*.
5. If the MMCAnCE_INT.CRSPE bit is not set within a defined time, or any error flag is set, the flow should go to an error/timeout processing.
For details of the timeout setting, please refer to RH850/U2B Hardware User’s Manual *R01UH0923EJxxxx Section 20.3.7 ‘MMCAnCE_CLK_CTRL — MMCAn Clock Control Register’*.
For details of the error flags, please refer to RH850/U2B Hardware User’s Manual *Section R01UH0923EJxxxx 18.3.13 ‘MMCAnCE_INT — MMCAn Interrupt Flag Register’*.
6. To read the complete register, the block length is set to 200_H, this means that the buffer register MMCAnCE_DATA must be read for 128 times.

Figure 8-6 Software Flow to Read EXT_CSD register

8.5 Single-Block Read

The host can read the data on card with CMD17, if only a single block is required. Figure 8-7 shows the software for single-block read.



Notes: 1. The card might be toggled by CMD7 to transfer state after identification, abort by CMD12, or returned from any “operation complete”, for details, please refer to Figure 5-5.

2. For details, please refer to the Figure 8-4.
   Basically, the size can be set as 1 to 512 bytes on the host side.
   When the Renesas extension eMMC board Y-RH850-EMMC-SFMA-EXT-BRD is used for the application, it is recommended to set the block length as 200_H, because the card on board doesn’t support to read partial block.
   If another card is connected to the circuit, please refer to the corresponding data sheet.
3. The argument specifies the block address on the card to read.
   When the Renesas extension eMMC board Y-RH850-EMMC-SFMA-EXT-BRD is used for the application, please set the sector address here.
4. For detailed register settings, please refer to Table 5-3 in *Section 5.5.1 ‘Commands’*.
5. If the MMCAnCE_INT.CRSPE bit is not set within a defined time, or any error flag is set, the flow

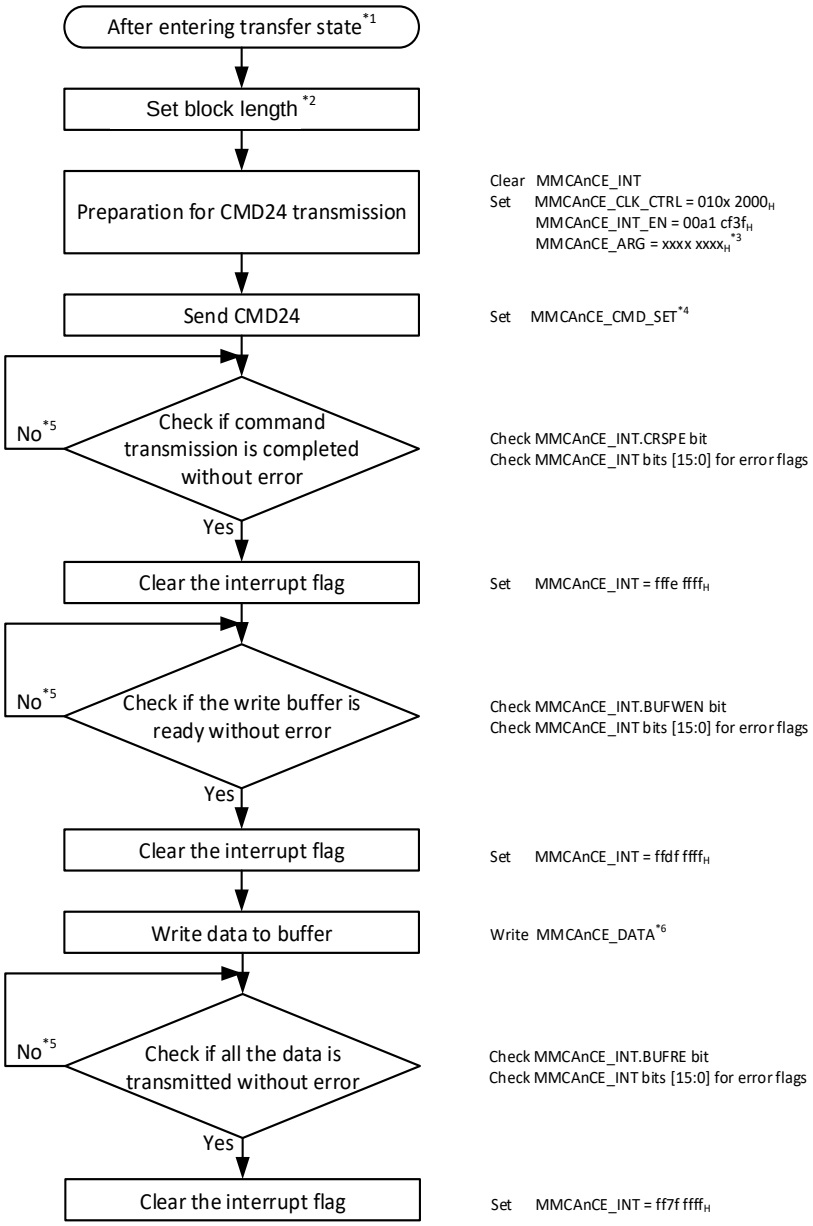
should go to an error/timeout processing.
For details of the timeout setting, please refer to RH850/U2B Hardware User's Manual *R01UH0923EJxxxx Section 20.3.7 'MMCAnCE_CLK_CTRL — MMCAn Clock Control Register'*.
For details of the error flags, please refer to RH850/U2B Hardware User's Manual *Section R01UH0923EJxxxx 18.3.13 'MMCAnCE_INT — MMCAn Interrupt Flag Register'*.

6. In case that the block length is set to 200_H, the register MMCAnCE_DATA must be read for 128 times.

Figure 8-7 Software Flow of Single-Block Read

8.6 Single-Block Write

The host can write data to the card with CMD24, if only a single block is required. Figure 8-8 shows the software for single-block write.



Notes: 1. The card might be toggled by CMD7 to transfer state after identification, abort by CMD12, or returned from any "operation complete", for details, please refer to Figure 5-5.

2. For details, please refer to the Figure 8-4.
   Basically, the size can be set as 1 to 512 bytes on the host side.
   When the Renesas extension eMMC board Y-RH850-EMMC-SFMA-EXT-BRD is used for the application, it is recommended to set the block length as 200_H, because the card on board doesn't support to write a partial block.
   If another card is connected to the circuit, please refer to the corresponding data sheet.
3. The argument specifies the block address to write to.
   When the Renesas extension eMMC board Y-RH850-EMMC-SFMA-EXT-BRD is used for the application, please set the sector address here.
4. For detailed register settings, please refer to Table 5-3 in *Section 5.5.1 'Commands'*.
5. If the MMCAnCE_INT.CRSPE bit is not set within a defined time, or any error flag is set, the flow should go to an error/timeout processing.
   For details of the timeout setting, please refer to RH850/U2B Hardware User's Manual *R01UH0923EJxxxx Section 20.3.7 'MMCAnCE_CLK_CTRL — MMCAn Clock Control Register'*.
   For details of the error flags, please refer to RH850/U2B Hardware User's Manual *Section R01UH0923EJxxxx 18.3.13 'MMCAnCE_INT — MMCAn Interrupt Flag Register'*.
6. In case that the block length is set to 200_H, the register MMCAnCE_DATA must be written for 128 times.

Figure 8-8 Software Flow of Single-Block Write

8.7 Multi-Block Read

It is also possible for the host to read a quantity of blocks, this is called multi-block read, which can be executed by CMD18.

For multi-block read, every block has the length of 512 bytes.

If the number of the blocks to read is defined before CMD18 is sent, the host can count and read the defined quantity of blocks. This is pre-defined multi-block read.

In case the number is not defined for the blocks to read for CMD18, the address of the last byte to read can be used to define the end of transmission. On the host, if the automatic CMD12 issuance is enabled, the host issues automatically a CMD12 to the card to abort the process after the last block is read. This is open-ended multi-block read.

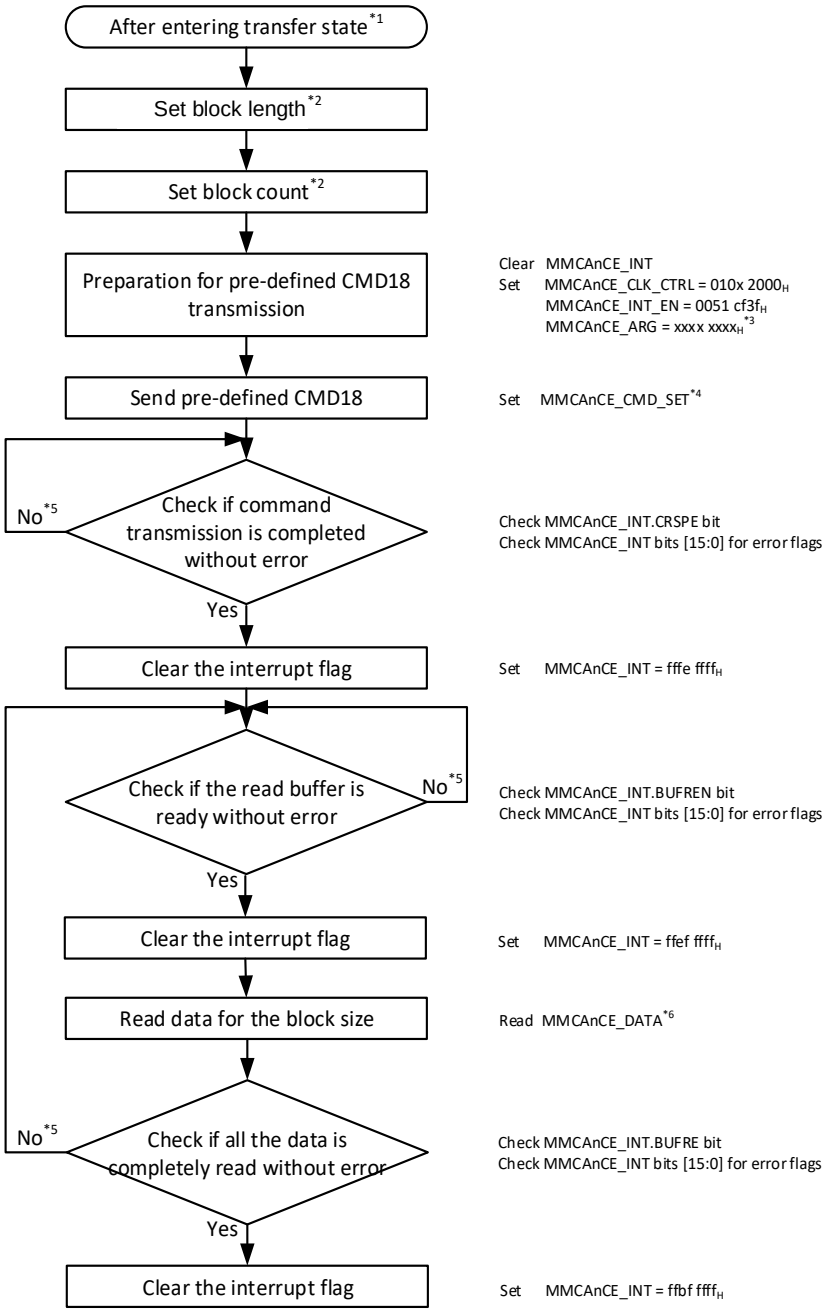
The configuration on the host side and the software flow for pre-defined and open-ended multi-block read is individually described in Section 8.7.1 and Section 8.7.2.

8.7.1 Pre-defined

For pre-defined multi-block read, the automatic CMD12 issuance is disabled and the host command setting register is configured with pre-defined CMD18. The setting value of this MMCAnCE_CMD_SET register can be found in Table 5.3.

Figure 8-9 shows the flow of the pre-defined multi-block read.

After the multi-block read command CMD18 is transmitted to the card, the read buffer is always prepared to read the next coming block (512 bytes), until all the data blocks are sent to the host. The card data is transmitted to the host and stored in the read buffer. Reading the MMCAnCE_DATA register accesses the read buffer, for each block the register must be read for 128 times. After a block is completely read, the read buffer starts to prepare for the next block.



Notes: 1. The card might be toggled by CMD7 to transfer state after identification, abort by CMD12, or returned from any “operation complete”, for details, please refer to Figure 5-5.

2. For details, please refer to the Figure 8-4.
   On the host side, the transfer block size must be set to 512 bytes for multi-block transfer, and the number of blocks for transfer can be set as any value except 0.
3. The argument specifies the address of the first block to be read.
   When the Renesas extension eMMC board Y-RH850-EMMC-SFMA-EXT-BRD is used for the application, please the set sector address here.
4. For detailed register settings, please refer to Table 5-3 in *Section 5.5.1 ‘Commands’*.
5. If the MMCAnCE_INT.CRSPE bit is not set within a defined time, or any error flag is set, the flow should go to an error/timeout processing.
   For details of the timeout setting, please refer to RH850/U2B Hardware User’s Manual *R01UH0923EJxxxx Section 20.3.7 ‘MMCAnCE_CLK_CTRL — MMCAn Clock Control Register’*.
   For details of the error flags, please refer to RH850/U2B Hardware User’s Manual *Section R01UH0923EJxxxx 18.3.13 ‘MMCAnCE_INT — MMCAn Interrupt Flag Register’*.

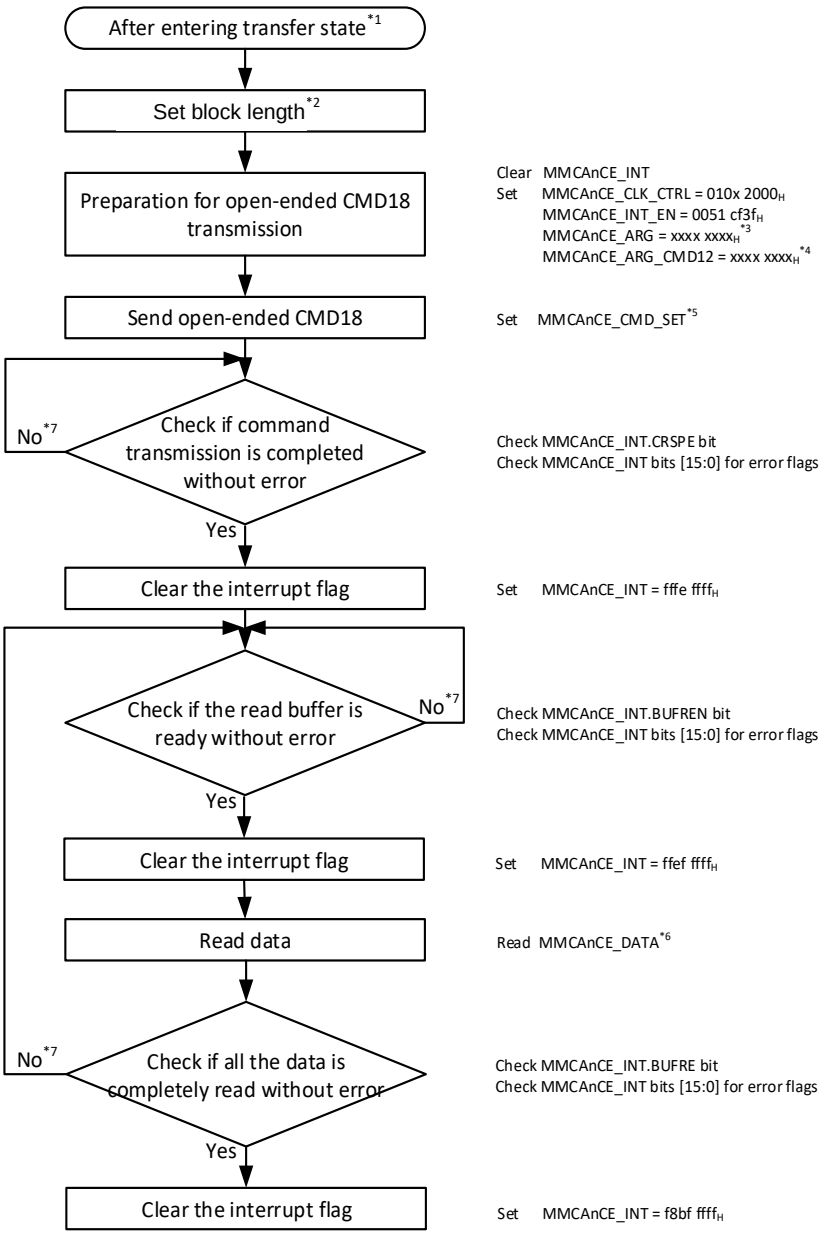
6. For each block, the register MMCAnCE_DATA must be read for 128 times.

Figure 8-9 Software Flow of Pre-Defined Multi-Block Read

8.7.2 Open-ended

For open-ended multi-block read, the automatic CMD12 issuance is enabled and the host command setting register is configured with open-ended CMD18. The setting value of this MMCAnCE_CMD_SET register can be found in Table 5.3.

Figure 8-10 shows the flow of the open-ended multi-block read.



Notes: 1. The card might be toggled by CMD7 to the transfer state after identification, abort by CMD12, or returned from any “operation complete”, for details, please refer to Figure 5-5.

2. For details, please refer to the Figure 8-4.
On the host side, the transfer block size must be set to 512 bytes for multi-block transfer, and the number of blocks for transfer can be set as any value larger than the number of blocks between the first address and the last address: MMCAnCE_ARG_CMD12 - MMCAnCE_ARG + 1.

3. The argument specifies the address of the first block to be read.
   When the Renesas extension eMMC board Y-RH850-EMMC-SFMA-EXT-BRD is used for the application, please set the sector address here.
4. The register MMCAnCE_ARG_CMD12 specifies the address of the last block to be read.
5. For detailed register settings, please refer to Table 5-3 in *Section 5.5.1 'Commands'*.
6. If the MMCAnCE_INT.CRSPE bit is not set within a defined time, or any error flag is set, the flow should go to an error/timeout processing.
   For details of the timeout setting, please refer to RH850/U2B Hardware User's Manual *R01UH0923EJxxxx Section 20.3.7 'MMCAnCE_CLK_CTRL — MMCAn Clock Control Register'*.
   For details of the error flags, please refer to RH850/U2B Hardware User's Manual *Section R01UH0923EJxxxx 18.3.13 'MMCAnCE_INT — MMCAn Interrupt Flag Register'*.
7. For each block, the register MMCAnCE_DATA must be read for 128 times.

Figure 8-10 Software Flow of Open-Ended Multi-Block Read

8.8 Multi-Block Write

Similar to the read process, the host can also write a quantity of blocks once to the card using multi-block write command CMD25.

For multi-block write, every block has the length of 512 bytes.

If the number of the blocks to write is defined before CMD25 is sent, the host can count and write the defined quantity of blocks to the card. This is pre-defined multi-block write.

In case the number is not defined for the blocks to be written for CMD25, the address on the card of the last byte to be written can be used to define the end of transmission. On the host, if the automatic CMD12 issuance is enabled, the host issues automatically a CMD12 to the card to abort the process after the address for last block is written. This is open-ended multi-block read.

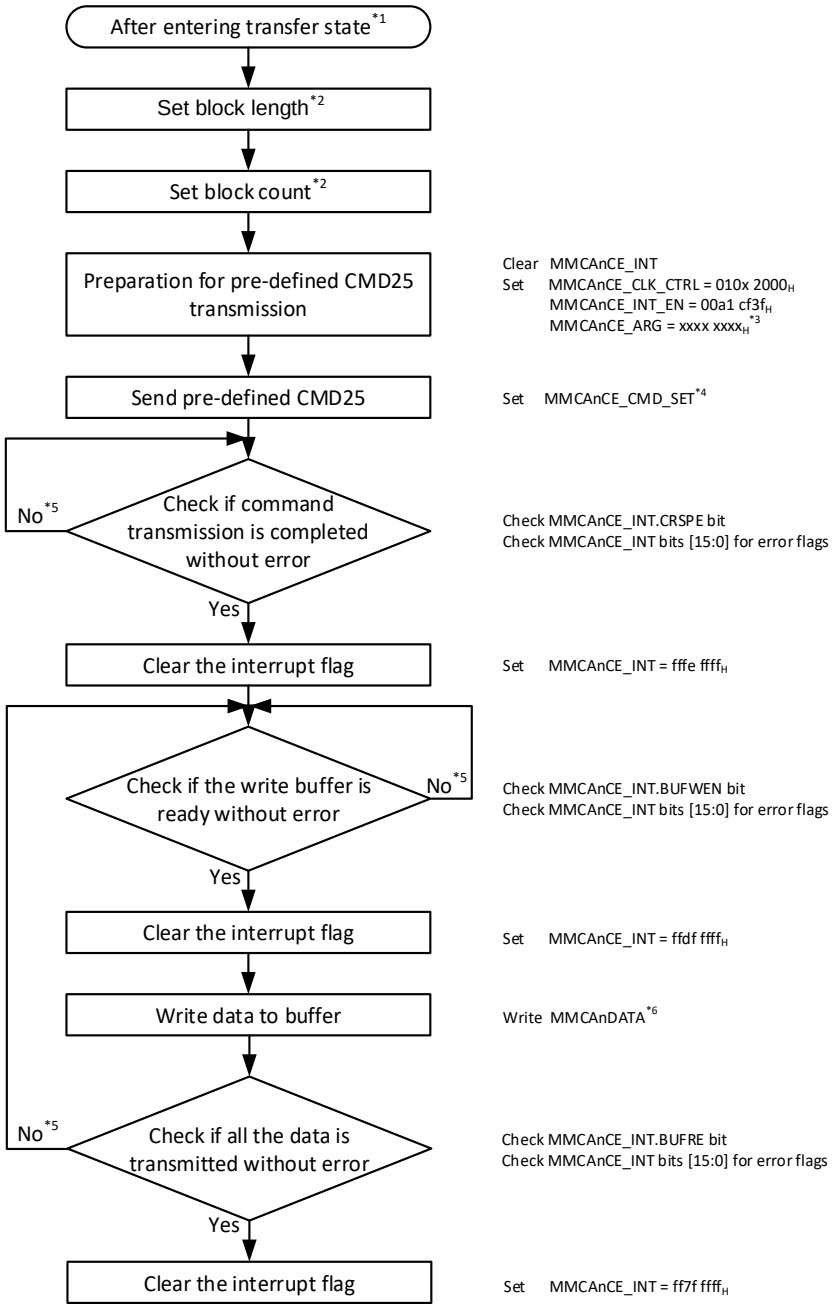
The configuration on host side and the software flow for the pre-defined and the open-ended multi-block write is individually described in Section 8.8.1 and Section 8.8.2.

8.8.1 Pre-defined

For the pre-defined multi-block write, the automatic CMD12 issuance is disabled and the host command setting register is configured with pre-defined CMD25. The setting value of this MMCAnCE_CMD_SET register can be found in Table 5.3.

Figure 8-11 shows the flow of the pre-defined multi-block write.

After the multi-block write command CMD25 is transmitted to the card, the write buffer is always prepared to get the next coming block (512 bytes), until all the data blocks are sent to the card. The data is written to the MMCAnCE_DATA register to access the write buffer, for each block 128 times. After a complete block is sent, the write buffer starts to prepare for the next block.



Notes: 1. The card might be toggled by CMD7 to transfer state after identification, abort by CMD12, or returned from any “operation complete”, for details, please refer to Figure 5-5.

2. For details, please refer to the Figure 8-4.
   On the host side, the transfer block size must be set to 512 bytes for multi-block transfer, and the number of blocks for transfer can be set as any value except 0.

3. The argument specifies the address of the first block to be written.
   When the Renesas extension eMMC board Y-RH850-EMMC-SFMA-EXT-BRD is used for the application, please set the sector address here.

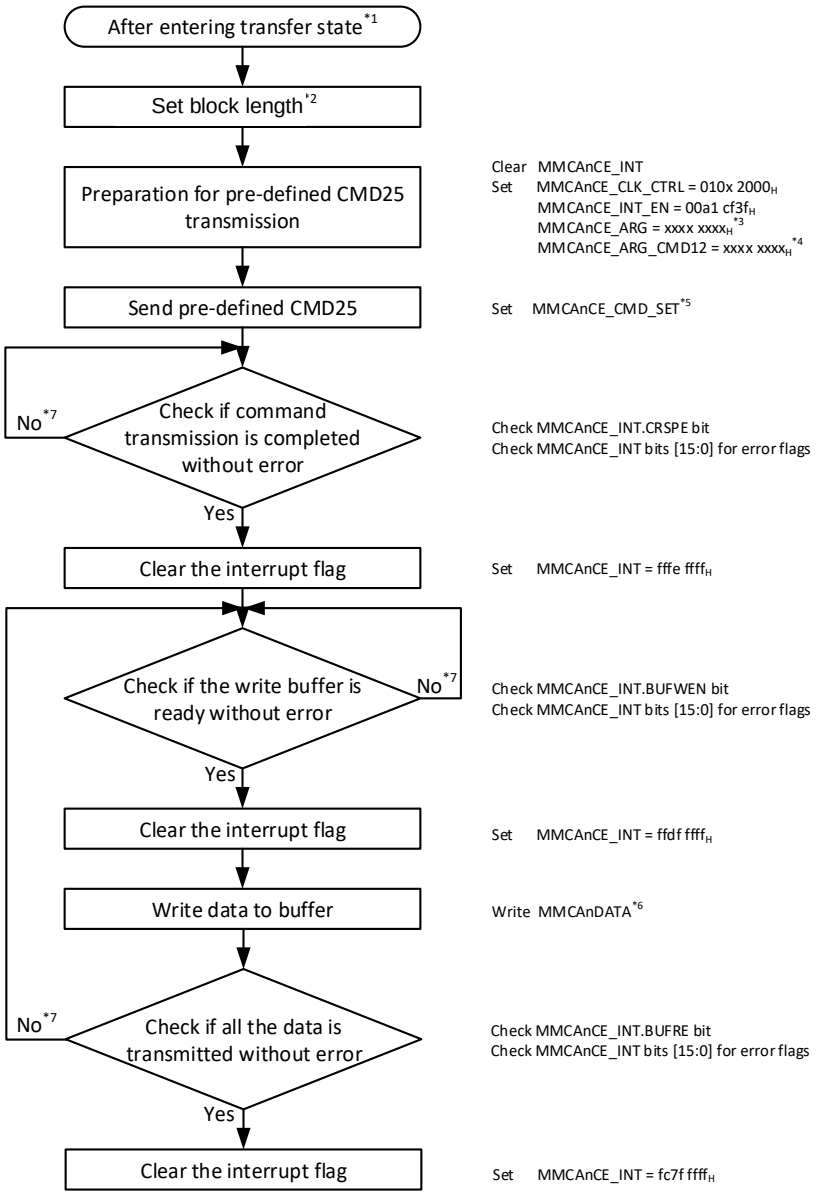
4. For detailed register settings, please refer to Table 5-3 in *Section 5.5.1 ‘Commands’*.

5. If the MMCAnCE_INT.CRSPE bit is not set within a defined time, or any error flag is set, the flow should go to an error/timeout processing.
   For details of the timeout setting, please refer to RH850/U2B Hardware User’s Manual *R01UH0923EJxxxx Section 20.3.7 ‘MMCAnCE_CLK_CTRL — MMCAn Clock Control Register’*.
   For details of the error flags, please refer to RH850/U2B Hardware User’s Manual *Section R01UH0923EJxxxx 18.3.13 ‘MMCAnCE_INT — MMCAn Interrupt Flag Register’*.

6. For each block, the register MMCAnCE_DATA must be written for 128 times.

Figure 8-11 Software Flow of Pre-Defined Multi-Block Write

8.8.2 Open-ended



Notes: 1. The card might be toggled by CMD7 to transfer state after identification, abort by CMD12, or returned from any “operation complete”, for details, please refer to Figure 5-5.
2. For details, please refer to the Figure 8-4.
On the host side, the transfer block size must be set to 512 bytes for multi-block transfer, and the number of blocks for transfer can be set as any value larger than the number of blocks between the first address and the last address: MMCAnCE_ARG_CMD12 - MMCAnCE_ARG + 1.
3. The argument specifies the address of the first block to be written.
When the Renesas extension eMMC board Y-RH850-EMMC-SFMA-EXT-BRD is used for the application, please set sector address here.
4. The register MMCAnCE_ARG_CMD12 specifies the address of the last block to be written.
5. For detailed register settings, please refer to Table 5-3 in *Section 5.5.1 ‘Commands’*.
6. If the MMCAnCE_INT.CRSPE bit is not set within a defined time, or any error flag is set, the flow should go to an error/timeout processing.

For details of the timeout setting, please refer to RH850/U2B Hardware User's Manual *R01UH0923EJxxxx Section 20.3.7 'MMCAnCE_CLK_CTRL — MMCAn Clock Control Register'*.
For details of the error flags, please refer to RH850/U2B Hardware User's Manual *Section R01UH0923EJxxxx 18.3.13 'MMCAnCE_INT — MMCAn Interrupt Flag Register'*.

7. For each block, the register MMCAnCE_DATA must be written for 128 times.

Figure 8-12 Software Flow of Open-Ended Multi-Block Write

8.9 Erase

The host can erase a large area of the card, the area to erase is defined by a start address and an end address via CMD35 and CMD36, like is shown in Figure 8-4.

After the basic parameters are set to the card, the host can send CMD38 to execute the erase operation. The detailed settings for CMD38 are listed in Table 5-3.

Normally an erase unit is large, therefore, it is recommended to set the end address larger than the sum of the start address and an erase unit size.

The erase unit size can be calculated by the following equation:

size of erasable unit = (ERASE_GRP_SIZE + 1) * (ERASE_GRP_MULT + 1)

in which ERASE_GRP_SIZE and ERASE_GRP_MULT are 2 parameters in the card CSD register.

For the Renesas extension eMMC board Y-RH850-EMMC-SFMA-EXT-BRD, the card on board has an erase unit of 1024 bytes.

8.10 Boot

Boot operation is a read operation of the boot information in boot mode.

The boot information might be contained in one of the two boot partitions or user area, this boot information should be written before the boot mode is carried out. The host can select the memory partitions for boot by setting byte [179] bit [5:3] of the card EXT_CSD register, if the memory partition is selected, the boot mode will be automatically executed after power on or reset operation.

The data size that the host can read during the boot operation can be calculated as:

128kB × BOOT_SIZE_MULT

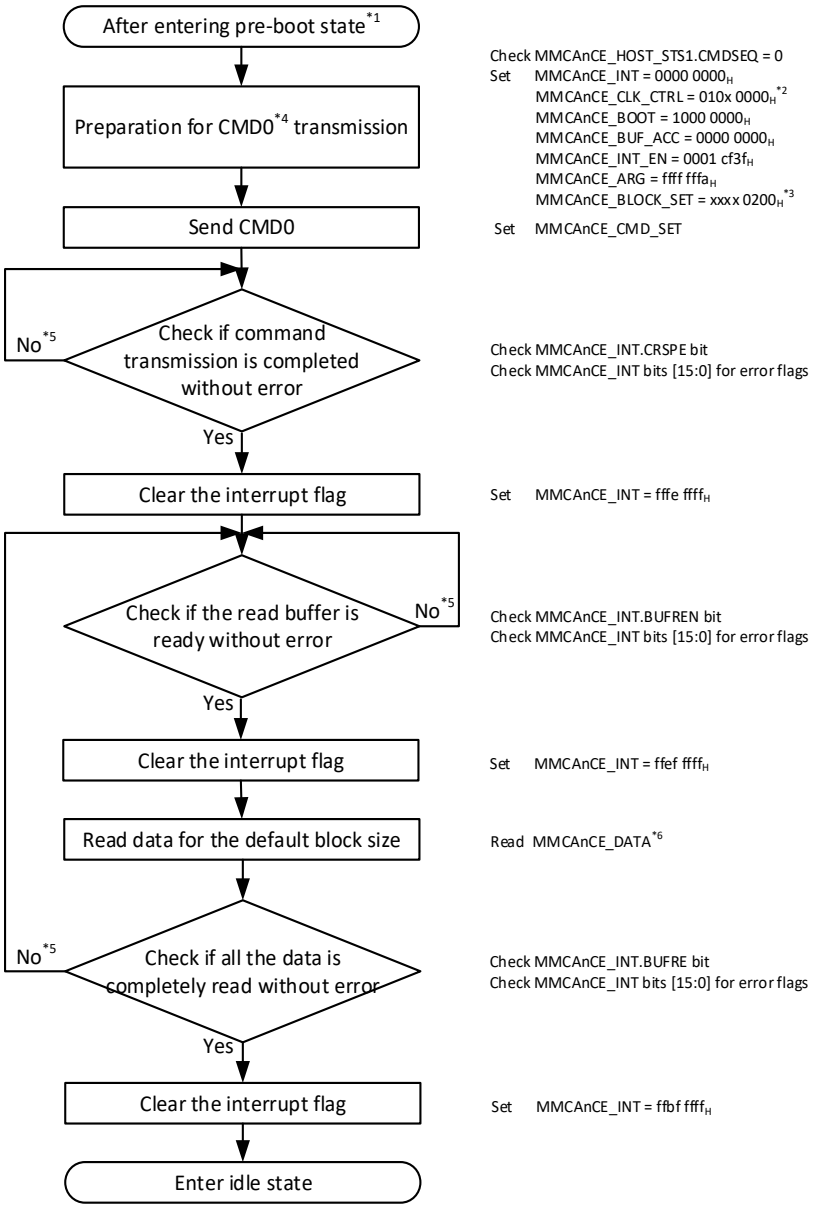
In which BOOT_SIZE_MULT is byte [226] in register EXT_CSD.

For the Renesas extension eMMC board Y-RH850-EMMC-SFMA-EXT-BRD, the card on board has a boot partition size of 4MB.

Any configuration for boot mode can be done in data transfer mode by writing the EXT_CSD register, such as the protection of the boot configuration, the data access, the boot bus conditions and the boot acknowledge configuration. It means, the host must enter the data transfer mode to specify the boot parameters, so that the card enters the pre-boot state from the next pre-idle state. The card goes to pre-idle state after power on, or switches from all other states to pre-idle state except inactive state after a hardware reset or CMD0 with argument $F0F0F0F0_H$.

The detailed configuration is described in this document, *Section 5.4.1.2 'Configuration of Boot Partition, Bus Width and Data Access'*.

Figure 8-13 shows the flow for the boot procedure after the card enters pre-boot state. After the specified memory partition is completely read, the card enters idle state.



Notes: 1. Regarding to Figure 5-2, the card enters pre-boot state from pre-idle state if the BOOT_PARTITION_ENABLE is set.
2. The clock divider can be set minimal as 7_H for the maximal clock frequency 400kHz here.
3. Basically, the size can be set as 1 to 512 bytes on the host side.
If the Renesas extension eMMC board Y-RH850-EMMC-SFMA-EXT-BRD is used for the application, it is recommended to set the block length as 200_H, because the card on board doesn't support to read partial block.
If other card is connected to the circuit, please refer to the corresponding data sheet.
4. Please refer to Table 5-3 in *Section 5.5.1 'Commands'* for BOOT_INITIATION.
5. If the MMCAnCE_INT.CRSPE bit is not set within a defined time, or any error flag is set, the flow should go to an error/timeout processing.
For details of the timeout setting, please refer to RH850/U2B Hardware User's Manual *R01UH0923EJxxxx Section 20.3.7 'MMCAnCE_CLK_CTRL — MMCAn Clock Control Register'*.
For details of the error flags, please refer to RH850/U2B Hardware User's Manual *Section R01UH0923EJxxxx 18.3.13 'MMCAnCE_INT — MMCAn Interrupt Flag Register'*.
6. In case that the block length is set to 200_H, the register MMCAnCE_DATA must be read for 128 times.

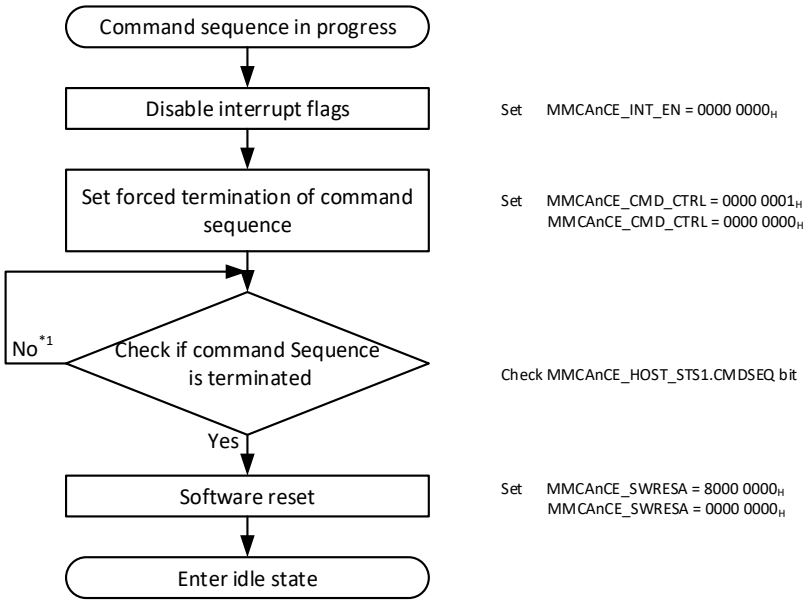
Figure 8-13 Software Flow of Boot

8.11 Forcible Termination

The MMCA module for eMMC interface on RH850/U2B device may not stop when an error or timeout occurs, the MMCAnCE_HOST_STS1.CMDSEQ bit is still indicating a command sequence in progress. In this case, the forcible termination and software reset can be executed before issuing the next command.

Figure 8-14 shows the flow of forcible termination.

Please notice that the data for transmission or received data that had been stored in the buffers at the time of error occurrence is not guaranteed.



Notes: 1. Wait until MMCAnCE_HOST_STS1.CMDSEQ bit is cleared.
For details, please refer to RH850/U2B Hardware User's Manual *R01UH0923EJxxxx Section 20.3.15 'MMCAnCE_HOST_STS1 — MMCAn Status Register 1'*.

Figure 8-14 Software Flow of Forcible Termination

9. Description of Sample Software

The corresponding sample software of this document can be found in the attachment file: *r01an6849 ejxxxx-rh850u2b-mmca sample code.*

It is based on the software tools and hardware setups described in Section 2.

The sample software is executed from core 0, it contains the configuration of different MMCA functions, ports configuration, clock configuration and the main function.

9.1 main_pe0.c

The location of this file is in the folder: …\ *r01an6849ejxxxx-rh850u2b-mmca sample code* *src*.

The sample software is for core0. The main_pe0.c contains the main function of this application.

The variable arrays for read and write operation are 8-bit unsigned data arrays: data_rd[512][4] and data_wr[512][4];

The main function calls other local / global functions to implement the application. Some of these functions are included in main_pe0.c:

- Dedicated module configuration: R_MMCA_CfgPort ().
- Result comparison: MMCA_TestChk ().

The sample software uses 1-bit bus width for the data line transfer.

9.2 MMCA.c and MMCA.h

The location of this file is in folder: …\ *r01an*6849*ejxxxx-rh850u2b-mmca sample code* *src.*
MMCA.c contains the configuration for the basic operations of eMMC related application:

- Read CSD register: loc_ReadRg_Csd (),
  Read EXT_CSD register: MMCA_ReadRg_Ext ().
- Set the block length to the card: loc_setblkLen (uint32_t blklen, uint32_t blknr).
- Set the data pattern to be written to the card: loc_setblkBuf (uint32_t blklen, uint32_t blknr).
- Select / deselect the card or switch the card between stand-by and transfer state: MMCA_SelectCard ().
- Set the high speed mode for the bus: MMCA_SetHighSpeed ().
- Card identification: MMCA_Identification ().
- Single-Block Read: MMCA_SingleBlockRead (uint32_t blklen, uint32_t addr).
- Single-Block Write: MMCA_SingleBlockWr (uint32_t blklen, uint32_t addr).
- Pre-Defined Multi-Block Read: MMCA_MultiBlockRead (uint32_t blknr, uint32_t addr).
  Open-Ended Multi-Block Read: MMCA_MultiBlockRead_12 (uint32_t addr, uint32_t addr_12).
- Pre-Defined Multi-Block Write: MMCA_MultiBlockWr (uint32_t blknr, uint32_t addr).
  Open-Ended Multi-Block Write: MMCA_MultiBlockWr_12 (uint32_t addr, uint32_t addr_12).
- Erase card data: MMCA_EraseData (uint32_t addr_str, uint32_t addr_end).

MMCA.h is the header file of MMCA.c, in which the card dedicated parameters and command settings are defined.

9.3 port_init.c

The location of this file is in folder: …\ *r01an6849ejxxxx-rh850u2b-mmca sample code* *src*.

port_init.c contains the configuration for port functions:

- Set port: port_init ().

Revision History

Rev.	Date	Description	
		Page	Summary
0.10	Mar 31, 2023	All	Release
1.00	Jan 31, 2024	5	Changed revision of reference document.
		41	Modified the referenced figure number on *8.8.1 Pre-defined*.

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance 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 the 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.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. 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 shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended 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; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or 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 damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, 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 and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall 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 shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

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

Contact information

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