RX Family, RL78 Family, 78K0R/Kx3-L

Micron Technology M25P Series

Serial Flash memory Control Software

R01AN0566EJ0106
Rev.1.06
Mar 31, 2016

Introduction

This application note explains how to control Micron Technology, Inc. M25P series SPI Serial Flash memory and how to use the sample code for the products.

This sample code lies in a higher-level layer of the software for controlling a Serial Flash memory as a slave device.

Also available for your reference is the separate software (clock synchronous single-master control software) which lies in the lower-level layer of the software for controlling the SPI mode of the individual MCUs serving as master devices, so please obtain this from the following URL as well. In addition, when a new microcontroller is added to the clock synchronous single-master control software, update of this application note may not be in time. Refer to 'Clock Synchronous Single Master Control Software (Lower-level layer of the software)' information in the following URL for the combination information on the latest supported microcontroller and its single-master control software.

- SPI/QSPI Serial Flash Memory, QSPI Serial Phase Change Memory Driver

  http://www.renesas.com/driver/spi_serial_flash

Target Device

Serial Flash memory: Micron Technology, Inc. M25P series SPI Serial Flash memory

MCU used for checking the operation of the sample code:

RX600 series	: RX610, RX62N, RX63N, RX63T group (using the SCI)
	: RX62N, RX63N, RX63T group (using the RSPI)
RX200 series	: RX210, RX21A, RX220 group (using the SCI)
	: RX210, RX21A, RX220 group (using the RSPI)
RX100 series	: RX111 (using the SCI)
	: RX111 (using the RSPI)
78K0R/Kx	: 78K0R/KE3-L (using the SAU)
RL78/G1x	: RL78/G14, RL78/G1C (using the SAU)
RL78/L1x	: RL78/L12, RL78/L13, RL78/L1C (using the SAU)

When applying the contents of this application note to other series of microcomputers or memory devices, make necessary modifications to and make extensive evaluations of the sample code according to the specifications for the microcomputer or memory to be used.

Contents

1. Specifications

The SPI Serial Flash memory control program controls the Micron Technology, Inc. M25P series SPI Serial Flash memory devices using a Renesas Electronics' MCU.

A clock synchronous single-master control program that is specific to the individual MCU is separately required.

Table 1-1 summarizes the peripheral devices to be used and their uses. Figure 1 illustrates a sample configuration.

The major functions of the SPI Serial Flash memory control program are summarized below.

- The SPI Serial Flash memory control program is a block-type device driver that assumes a Renesas Electronics MCU as the master device and a Micron Technology, Inc. M25P series SPI Serial Flash memory as a slave device.
- It controls the devices in the SPI mode using the MCU's internal serial communication function (clock synchronous mode). It can use no more than one user-configured serial I/O channel.
- The SPI Serial Flash memory control program can control a maximum of two SPI Serial Flash memory devices of the same type.
- The communication rate is user-programmable (fixed speed).
- Both big endian and little endian modes are supported (dependent on the MCU in use)

Table 1-1 Peripheral Devices Used and their Uses

Peripheral Device	Use
MCU internal serial communication function (Clock synchronous mode)	Communication with SPI slave devices using the serial communication function (clock synchronous mode) 1 channel (required)
Port	For SPI slave device select control signals. As many ports as there are SPI slave devices in use are necessary (required).

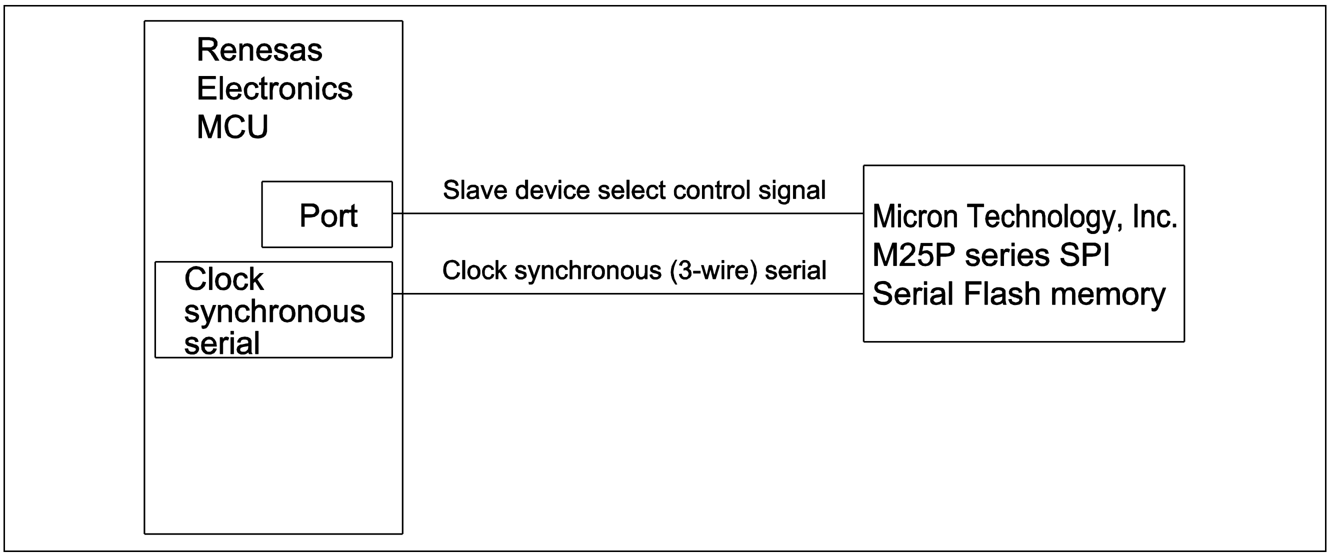

Renesas Electronics MCU
Port
Clock synchronous serial
Slave device select control signal
Clock synchronous (3-wire) serial
Micron Technology, Inc. M25P series SPI Serial Flash memory

Figure 1 Sample Configuration

2. Conditions for Checking the Operation of the SPI Serial Flash memory Control Software

The sample code described in this application note has been confirmed to run normally under the operating conditions given below.

2.1 RX Family

(1) **RX610 SCI**

Table 2-1 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RX610 group (program ROM 2 MB/RAM 128 KB)
Operating frequency (microcomputer)	ICLK: 100 MHz, PCLK: 50 MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics High-performance embedded Workshop Version 4.07.00.007
C compiler	Renesas Electronics RX family C/C++ compiler package (Toolchain 1.0.0.0)
	Compiler options: The default settings for the integrated development environment are used.
Endian	Big endian/Little endian
Version of the sample code	Ver.2.00
Software used for evaluation	Clock synchronous single-master control software using the SCI for the RX610, Ver.2.00
Evaluation board used	Renesas Starter Kit for the RX610

(2) **RX62N RSPI**

Table 2-2 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RX62N group (program ROM 512KB/RAM 96KB)
Operating frequency (microcomputer)	ICLK: 96MHz, PCLK: 48MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics High-performance embedded Workshop Version 4.09.00.007
C compiler	Renesas Electronics RX family C/C++ compiler package (Toolchain 1.0.1.0)
	Compiler options: The default settings (Optimize Level:2, Optimize for size) for the integrated development environment are used.
Endian	Big endian/Little endian
Version of the sample code	Ver.2.01
Software used for evaluation	RX62N Group Clock synchronous single-master control software using the RSPI, Ver.2.03
Evaluation board used	Renesas Starter Kit for the RX62N

(3) **RX62N SCI**

Table 2-3 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RX62N group (program ROM 512KB/RAM 96KB)
Operating frequency (microcomputer)	ICLK: 96MHz, PCLK: 48MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics High-performance embedded Workshop Version 4.09.00.007
C compiler	Renesas Electronics RX family C/C++ compiler package (Toolchain 1.0.1.0)
	Compiler options: The default settings (Optimize Level:2, Optimize for size) for the integrated development environment are used.
Endian	Big endian/Little endian
Version of the sample code	Ver.2.01
Software used for evaluation	RX62N Group Clock synchronous single-master control software using the SCI, Ver.2.01
Evaluation board used	Renesas Starter Kit for the RX62N

(4) **RX63N RSPI**

Table 2-4 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RX63N group (program ROM 1MB/RAM 128KB)
Operating frequency (microcomputer)	ICLK: 96MHz, PCLK: 48MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics High-performance embedded Workshop Version 4.09.00.007
C compiler	Renesas Electronics RX family C/C++ compiler package (Toolchain 1.2.1.0)
	Compiler options: The default settings (Optimize Level:2, Optimize for size) for the integrated development environment are used.
Endian	Big endian/Little endian
Version of the sample code	Ver.2.02
Software used for evaluation	RX210, RX21A, RX220, RX63N, RX63T Group Clock synchronous single-master control software using the RSPI, Ver.2.04
Evaluation board used	Renesas Starter Kit for the RX63N

(5) **RX63N SCI**

Table 2-5 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RX63N group (program ROM 1MB/RAM 128KB)
Operating frequency (microcomputer)	ICLK: 96MHz, PCLK: 48MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics High-performance embedded Workshop Version 4.09.00.007
C compiler	Renesas Electronics RX family C/C++ compiler package (Toolchain 1.2.1.0)
	Compiler options: The default settings (Optimize Level:2, Optimize for size) for the integrated development environment are used.
Endian	Big endian/Little endian
Version of the sample code	Ver.2.02
Software used for evaluation	RX210, RX21A, RX220, RX63N, RX63T Group Clock synchronous single-master control software using the SCI, Ver.2.01
Evaluation board used	Renesas Starter Kit for the RX63N

(6) **RX63T RSPI**

Table 2-6 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RX63T group (program ROM 512KB/RAM 48KB)
Operating frequency (microcomputer)	ICLK: 96MHz, PCLK: 48MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics CubeSuite+ V2.00.00
C compiler	Renesas Electronics RX family C/C++ compiler package (Toolchain 2.00.00)
	Compiler options: The default settings (Optimize Level:2, Optimize for size) for the integrated development environment are used.
Endian	Big endian/Little endian
Version of the sample code	Ver.2.02
Software used for evaluation	RX210, RX21A, RX220, RX63N, RX63T Group Clock synchronous single-master control software using the RSPI, Ver.2.04
Evaluation board used	Renesas Starter Kit for the RX63T

(7) **RX63T SCI**

Table 2-7 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RX63T group (program ROM 512KB/RAM 48KB)
Operating frequency (microcomputer)	ICLK: 96MHz, PCLK: 48MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics CubeSuite+ V2.00.00
C compiler	Renesas Electronics RX family C/C++ compiler package (Toolchain 2.00.00)
	Compiler options: The default settings (Optimize Level:2, Optimize for size) for the integrated development environment are used.
Endian	Big endian/Little endian
Version of the sample code	Ver.2.02
Software used for evaluation	RX210, RX21A, RX220, RX63N, RX63T Group Clock synchronous single-master control software using the SCI, Ver.2.01
Evaluation board used	Renesas Starter Kit for the RX63N

(8) **RX210 RSPI**

Table 2-8 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RX210 group (program ROM 512KB/RAM 64KB)
Operating frequency (microcomputer)	ICLK: 50MHz, PCLK: 25MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics High-performance embedded Workshop Version 4.09.00.007
C compiler	Renesas Electronics RX family C/C++ compiler package (Toolchain 1.2.1.0)
	Compiler options: The default settings (Optimize Level:2, Optimize for size) for the integrated development environment are used.
Endian	Big endian/Little endian
Version of the sample code	Ver.2.02
Software used for evaluation	RX210, RX21A, RX220, RX63N, RX63T Group Clock synchronous single-master control software using the RSPI, Ver.2.04
Evaluation board used	Renesas Starter Kit for the RX210

(9) **RX210 SCI**

Table 2-9 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RX210 group (program ROM 512KB/RAM 64KB)
Operating frequency (microcomputer)	ICLK: 50MHz, PCLK: 25MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics High-performance embedded Workshop Version 4.09.00.007
C compiler	Renesas Electronics RX family C/C++ compiler package (Toolchain 1.2.1.0)
	Compiler options: The default settings (Optimize Level:2, Optimize for size) for the integrated development environment are used.
Endian	Big endian/Little endian
Version of the sample code	Ver.2.02
Software used for evaluation	RX210, RX21A, RX220, RX63N, RX63T Group Clock synchronous single-master control software using the SCI, Ver.2.01
Evaluation board used	Renesas Starter Kit for the RX210

(10) **RX21A RSPI**

Table 2-10 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RX21A group (program ROM 512KB/RAM 64KB)
Operating frequency (microcomputer)	ICLK: 50MHz, PCLK: 25MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics High-performance embedded Workshop Version 4.09.00.007
C compiler	Renesas Electronics RX family C/C++ compiler package (Toolchain 1.2.1.0)
	Compiler options: The default settings (Optimize Level:2, Optimize for size) for the integrated development environment are used.
Endian	Big endian/Little endian
Version of the sample code	Ver.2.02
Software used for evaluation	RX210, RX21A, RX220, RX63N, RX63T Group Clock synchronous single-master control software using the RSPI, Ver.2.04
Evaluation board used	HOKUTO DENSHI HSBRX21AP-B

(11) **RX21A SCI**

Table 2-11 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RX21A group (program ROM 512KB/RAM 64KB)
Operating frequency (microcomputer)	ICLK: 50MHz, PCLK: 25MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics High-performance embedded Workshop Version 4.09.00.007
C compiler	Renesas Electronics RX family C/C++ compiler package (Toolchain 1.2.1.0)
	Compiler options: The default settings (Optimize Level:2, Optimize for size) for the integrated development environment are used.
Endian	Big endian/Little endian
Version of the sample code	Ver.2.02
Software used for evaluation	RX210, RX21A, RX220, RX63N, RX63T Group Clock synchronous single-master control software using the SCI, Ver.2.01
Evaluation board used	HOKUTO DENSHI HSBRX21AP-B

(12) **RX220 RSPI**

Table 2-12 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RX220 group (program ROM 256KB/RAM 16KB)
Operating frequency (microcomputer)	ICLK: 32MHz, PCLK: 32MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics High-performance embedded Workshop Version 4.09.00.007
C compiler	Renesas Electronics RX family C/C++ compiler package (Toolchain 1.2.1.0)
	Compiler options: The default settings (Optimize Level:2, Optimize for size) for the integrated development environment are used.
Endian	Big endian/Little endian
Version of the sample code	Ver.2.02
Software used for evaluation	RX210, RX21A, RX220, RX63N, RX63T Group Clock synchronous single-master control software using the RSPI, Ver.2.04
Evaluation board used	Renesas Starter Kit for RX220

(13) **RX220 SCI**

Table 2-13 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RX220 group (program ROM 256KB/RAM 16KB)
Operating frequency (microcomputer)	ICLK: 32MHz, PCLK: 32MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics High-performance embedded Workshop Version 4.09.00.007
C compiler	Renesas Electronics RX family C/C++ compiler package (Toolchain 1.2.1.0)
	Compiler options: The default settings (Optimize Level:2, Optimize for size) for the integrated development environment are used.
Endian	Big endian/Little endian
Version of the sample code	Ver.2.02
Software used for evaluation	RX210, RX21A, RX220, RX63N, RX63T Group Clock synchronous single-master control software using the SCI, Ver.2.01
Evaluation board used	Renesas Starter Kit for RX220

(14) **RX111 RSPI**

Table 2-14 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RX111 Group (Program ROM: 128 KB, RAM: 16 KB)
Operating frequency (microcomputer)	ICLK: 32 MHz, PCLK: 32 MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics CubeSuite+ V2.01.00
C compiler	Renesas Electronics RX family C/C++ compiler package (Toolchain 2.01.00)
	Compiler options: The default settings (Optimize Level: 2, Optimize for size) for the integrated development environment are used.
Endian	Big endian/Little endian
Version of the sample code	Ver. 2.02 R01
Software used for evaluation	RX210, RX21A, RX220, RX63N, RX63T, RX111 Group Clock Synchronous Single Master Control Software Using the RSPI (R01AN1196EJ), Ver. 2.04.R04
Evaluation board used	Renesas Starter Kit for the RX111

(15) **RX111 SCI**

Table 2-15 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RX111 Group (Program ROM: 128 KB, RAM: 16 KB)
Operating frequency (microcomputer)	ICLK: 32 MHz, PCLK: 32 MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics CubeSuite+ V2.01.00
C compiler	Renesas Electronics RX family C/C++ compiler package (Toolchain 2.01.00)
	Compiler options: The default settings (Optimize Level: 2, Optimize for size) for the integrated development environment are used.
Endian	Big endian/Little endian
Version of the sample code	Ver. 2.02 R01
Software used for evaluation	RX210, RX21A, RX220, RX63N, RX63T,RX111 Group Clock Synchronous Single Master Control Software Using the SCI (R01AN1229EJ), Ver. 2.01.R05
Evaluation board used	Renesas Starter Kit for the RX111

2.2 RL78 Family, 78K0R/Kx3-L

(1) **78K0R/Kx3-L SAU**

Table 2-16 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	78K0R/KE3-L (program ROM 64KB/RAM 3KB)
Operating frequency (microcomputer)	Main System Clock: 20MHz CPU/peripheral Hardware Clock: 20MHz Serial Clock: 2.5MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics Project Manager (PM+ Ver.6.31)
C compiler	Renesas Electronics 78K0R C compiler (CC78K0R Ver.2.13) 78K0R Assembler package (RA78K0R Ver.1.33)
	Compiler options: The default settings ("-a. -zp") for the integrated development environment are used.
Emulator	Minicube2
Integrated Debugger	Renesas Electronics Integrated Debugger ID78K0R-QB Ver.3.61
Version of the sample code	Ver.2.01
Software used for evaluation	Clock synchronous single-master control software using the SAU CSI mode for the 78K0R/Kx3-L, Ver.2.00
Evaluation board used	78K0R/KE3-L Target Board (QB-78K0RKE3L-TB)

(2) **RL78/G14 SAU Integrated Development Environment CubeSuite+ (Compiler:CA78K0R)**

Table 2-17 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RL78/G14 (program ROM 256KB/RAM 24KB)
Operating frequency	Main system clock: 32MHz CPU/peripheral hardware clock: 32MHz Serial clock: 4MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics CubeSuite+ V1.01.01a
C compiler, assembler	Renesas Electronics RL78,78K0R C compiler (CA78K0R V1.30)
	Compiler options: The default settings (-qx2) for the integrated development environment are used.
Version of the sample code	Ver.2.02
Software used for evaluation	RL78/G14 Group Clock synchronous single-master control software using CSI mode of serial array unit Ver. 2.03
Evaluation board used	Renesas Starter Kit for RL78/G14

(3) **RL78/G14 SAU Integrated Development Environment CS+ for CC (Compiler:CC-RL)**

Table 2-18 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RL78/G14 (program ROM 256KB/RAM 24KB)
Operating frequency	Main system clock: 32MHz CPU/peripheral hardware clock: 32MHz Serial clock: 4MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics Corporation CS+ for CC V3.02.00
C compiler, assembler	Renesas Electronics Corporation RL78 compiler CC-RL V1.02.00
	Compiler options The default settings (Perform the default optimization(None)) for the integrated development environment are used.
Version of the sample code	Ver. 2.03
Software used for evaluation	RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C Group Clock Synchronous Single Master Control Software Using CSI Mode of Serial Array Unit (R01AN1195EJ0105), version 2.05)
Evaluation board used	Renesas Starter Kit for RL78/G14

(4) RL78/G14 SAU Integrated Development Environment IAR Embedded Workbench

Table 2-19 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RL78/G14 (program ROM 256KB/RAM 24KB)
Operating frequency	Main system clock: 32MHz CPU/peripheral hardware clock: 32MHz Serial clock: 4MHz
Operating voltage	3.3 V
Integrated development environment	IAR Systems IAR Embedded Workbench for Renesas RL78 (Ver.1.30.2)
C compiler, assembler	IAR Systems IAR Assembler for Renesas RL78 (Ver.1.30.2.50666) IAR C/C++ Compiler for Renesas RL78 (Ver.1.30.2.50666)
	Compiler options: The default settings (Low) for the integrated development environment are used.
Version of the sample code	Ver.2.02
Software used for evaluation	RL78/G14 Group Clock synchronous single-master control software using CSI mode of serial array unit Ver. 2.03
Evaluation board used	Renesas Starter Kit for RL78/G14

(5) RL78/G1C SAU Integrated Development Environment CubeSuite+ (Compiler:CA78K0R)

Table 2-20 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RL78/G1C Group (Program ROM:32 KB, RAM:5.5 KB)
Operating frequency (microcomputer)	Main system clock: 24 MHz CPU/peripheral hardware clock: 24 MHz Serial clock: 12 MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics CubeSuite+ V2.01.00
C compiler	Renesas Electronics CubeSuite+ RL78,78K0R C compiler (CA78K0R V1.70)
Version of the sample code	Ver. 2.02 R01
Software used for evaluation	RL78/G14,RL78/G1C, RL78/L12, RL78/L13, RL78/L1C Group Clock Synchronous Single Master Control Software Using CSI Mode of Serial Array Unit (R01AN1195EJ0103), Ver. 2.03
Evaluation board used	Renesas RL78/G1C Target Board QB-R5F10JGC-TB

(6) RL78/G1C SAU Integrated Development Environment CS+ for CC (Compiler:CC-RL)

Table 2-21 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RL78/G1C Group (Program ROM:32 KB, RAM:5.5 KB)
Operating frequency (microcomputer)	Main system clock: 24 MHz CPU/peripheral hardware clock: 24 MHz Serial clock: 12 MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics Corporation CS+ for CC V3.02.00
C compiler	Renesas Electronics Corporation RL78 compiler CC-RL V1.02.00
	Compiler options The default settings (Perform the default optimization(None)) for the integrated development environment are used.
Version of the sample code	Ver. 2.03
Software used for evaluation	RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C Group Clock Synchronous Single Master Control Software Using CSI Mode of Serial Array Unit (R01AN1195EJ0105), version 2.05)
Evaluation board used	Renesas RL78/G1C Target Board QB-R5F10JGC-TB

(7) RL78/G1C SAU Integrated Development Environment IAR Embedded Workbench

Table 2-22 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RL78/G1C Group (Program ROM: 32 KB, RAM: 5.5 KB)
Operating frequency (microcomputer)	Main system clock: 24 MHz CPU/peripheral hardware clock: 24 MHz Serial clock: 12 MHz
Operating voltage	3.3 V
Integrated development environment	IAR Systems IAR Embedded Workbench for Renesas RL78 (Ver.1.30.5)
C compiler	IAR Systems IAR Assembler for Renesas RL78 (Ver. 1.30.4.50715) IAR C/C++ Compiler for Renesas RL78 (Ver. 1.30.5.50715)
	Compiler options: The default settings ("level: low") for the integrated development environment are used.
Version of the sample code	Ver. 2.02 R01
Software used for evaluation	RL78/G14,RL78/G1C, RL78/L12, RL78/L13, RL78/L1C Group Clock Synchronous Single Master Control Software Using CSI Mode of Serial Array Unit (R01AN1195EJ0103), Ver. 2.03
Evaluation board used	Renesas RL78/G1C Target Board QB-R5F10JGC-TB

(8) **RL78/L12 SAU Integrated Development Environment CubeSuite+**

Table 2-23 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RL78/L12 Group (Program ROM: 32 KB, RAM:1.5 KB)
Operating frequency (microcomputer)	Main system clock: 24 MHz CPU/peripheral hardware clock: 24 MHz Serial clock: 6 MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics CubeSuite+ V2.01.00
C compiler	Renesas Electronics CubeSuite+ RL78,78K0R C compiler (CA78K0R V1.70)
Version of the sample code	Ver. 2.02 R01
Software used for evaluation	RL78/G14,RL78/G1C, RL78/L12, RL78/L13, RL78/L1C Group Clock Synchronous Single Master Control Software Using CSI Mode of Serial Array Unit (R01AN1195EJ0103), Ver. 2.03
Evaluation board used	Renesas Starter Kit for RL78/L12

(9) **RL78/L12 SAU Integrated Development Environment IAR Embedded Workbench**

Table 2-24 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RL78/L12 Group (Program ROM: 32 KB, RAM: 1.5 KB)
Operating frequency (microcomputer)	Main system clock: 24 MHz CPU/peripheral hardware clock: 24 MHz Serial clock: 6 MHz
Operating voltage	3.3 V
Integrated development environment	IAR Systems IAR Embedded Workbench for Renesas RL78 (Ver.1.30.5)
C compiler	IAR Systems IAR Assembler for Renesas RL78 (Ver. 1.30.4.50715) IAR C/C++ Compiler for Renesas RL78 (Ver. 1.30.5.50715)
	Compiler options: The default settings ("level: low") for the integrated development environment are used.
Version of the sample code	Ver. 2.02 R01
Software used for evaluation	RL78/G14,RL78/G1C, RL78/L12, RL78/L13, RL78/L1C Group Clock Synchronous Single Master Control Software Using CSI Mode of Serial Array Unit (R01AN1195EJ0103), Ver. 2.03
Evaluation board used	Renesas Starter Kit for RL78/L12

(10) **RL78/L13 SAU Integrated Development Environment CubeSuite+**

Table 2-25 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RL78/L13 Group (Program ROM: 128 KB, RAM: 8 KB)
Operating frequency (microcomputer)	Main system clock: 24 MHz CPU/peripheral hardware clock: 24 MHz Serial clock: 6 MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics CubeSuite+ V2.01.00
C compiler	Renesas Electronics CubeSuite+ RL78,78K0R C compiler (CA78K0R V1.70)
Version of the sample code	Ver. 2.02 R01
Software used for evaluation	RL78/G14,RL78/G1C, RL78/L12, RL78/L13, RL78/L1C Group Clock Synchronous Single Master Control Software Using CSI Mode of Serial Array Unit (R01AN1195EJ0103), Ver. 2.03
Evaluation board used	Renesas Starter Kit for RL78/L13

(11) **RL78/L13 SAU Integrated Development Environment IAR Embedded Workbench**

Table 2-26 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RL78/L13 Group (Program ROM: 256 KB, RAM: 24 KB)
Operating frequency (microcomputer)	Main system clock: 24 MHz CPU/peripheral hardware clock: 24 MHz Serial clock: 6 MHz
Operating voltage	3.3 V
Integrated development environment	IAR Systems IAR Embedded Workbench for Renesas RL78 (Ver.1.30.5)
C compiler	IAR Systems IAR Assembler for Renesas RL78 (Ver. 1.30.4.50715) IAR C/C++ Compiler for Renesas RL78 (Ver. 1.30.5.50715)
	Compiler options: The default settings ("level: low") for the integrated development environment are used.
Version of the sample code	Ver. 2.02 R01
Software used for evaluation	RL78/G14,RL78/G1C, RL78/L12, RL78/L13, RL78/L1C Group Clock Synchronous Single Master Control Software Using CSI Mode of Serial Array Unit (R01AN1195EJ0103), Ver. 2.03
Evaluation board used	Renesas Starter Kit for RL78/L13

(12) **RL78/L1C SAU Integrated Development Environment CubeSuite+**

Table 2-27 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RL78/L1C Group (Program ROM: 256 KB, RAM: 16 KB)
Operating frequency (microcomputer)	Main system clock: 24 MHz CPU/peripheral hardware clock: 24 MHz Serial clock: 6 MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics CubeSuite+ V2.01.00
C compiler	Renesas Electronics CubeSuite+ RL78,78K0R C compiler (CA78K0R V1.70)
Version of the sample code	Ver. 2.02 R01
Software used for evaluation	RL78/G14,RL78/G1C, RL78/L12, RL78/L13, RL78/L1C Group Clock Synchronous Single Master Control Software Using CSI Mode of Serial Array Unit (R01AN1195EJ0103), Ver. 2.03
Evaluation board used	Renesas Starter Kit for RL78/L1C

(13) **RL78/L1C SAU Integrated Development Environment IAR Embedded Workbench**

Table 2-28 Operating Conditions

Item	Description
Memory used for evaluation	Micron Technology, Inc. M25P series SPI Serial Flash memory
Microcomputer used for evaluation	RL78/L1C Group (Program ROM: 256 KB, RAM: 16 KB)
Operating frequency (microcomputer)	Main system clock: 24 MHz CPU/peripheral hardware clock: 24 MHz Serial clock: 6 MHz
Operating voltage	3.3 V
Integrated development environment	IAR Systems IAR Embedded Workbench for Renesas RL78 (Ver.1.30.5)
C compiler	IAR Systems IAR Assembler for Renesas RL78 (Ver. 1.30.4.50715) IAR C/C++ Compiler for Renesas RL78 (Ver. 1.30.5.50715)
	Compiler options: The default settings ("level: low") for the integrated development environment are used.
Version of the sample code	Ver. 2.02 R01
Software used for evaluation	RL78/G14,RL78/G1C, RL78/L12, RL78/L13, RL78/L1C Group Clock Synchronous Single Master Control Software Using CSI Mode of Serial Array Unit (R01AN1195EJ0103), Ver. 2.03
Evaluation board used	Renesas Starter Kit for RL78/L1C

3. Related Application Notes

The applications notes that are related to this application note are listed below. Reference should also be made to those application notes.

- RX610 Group Clock Synchronous Single Master Control Software Using the SCI (R01AN0534EJ)
- RX62N Group Clock Synchronous Single Master Control Software Using the RSPI (R01AN0323EJ)
- 78K0R/Kx3-L Clock Synchronous Single Master Control Software Using the SAU CSI mode (R01AN0708EJ)
- RX62N Group Clock Synchronous Single Master Control Software Using the SCI (R01AN1088EJ)
- RX210, RX21A, RX220, RX63N, RX63T, RX111 Group Clock synchronous single-master control software using the RSPI (R01AN1196EJ)
- RX210, RX21A, RX220, RX63N, RX63T, RX111 Group Clock synchronous single-master control software using the SCI (R01AN1229EJ)
- RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C Group Clock synchronous single-master control software using CSI mode of serial array unit (R01AN1195EJ)

4. Hardware Description

4.1 List of Pins

The following table lists the MCU pins that are used and their uses.

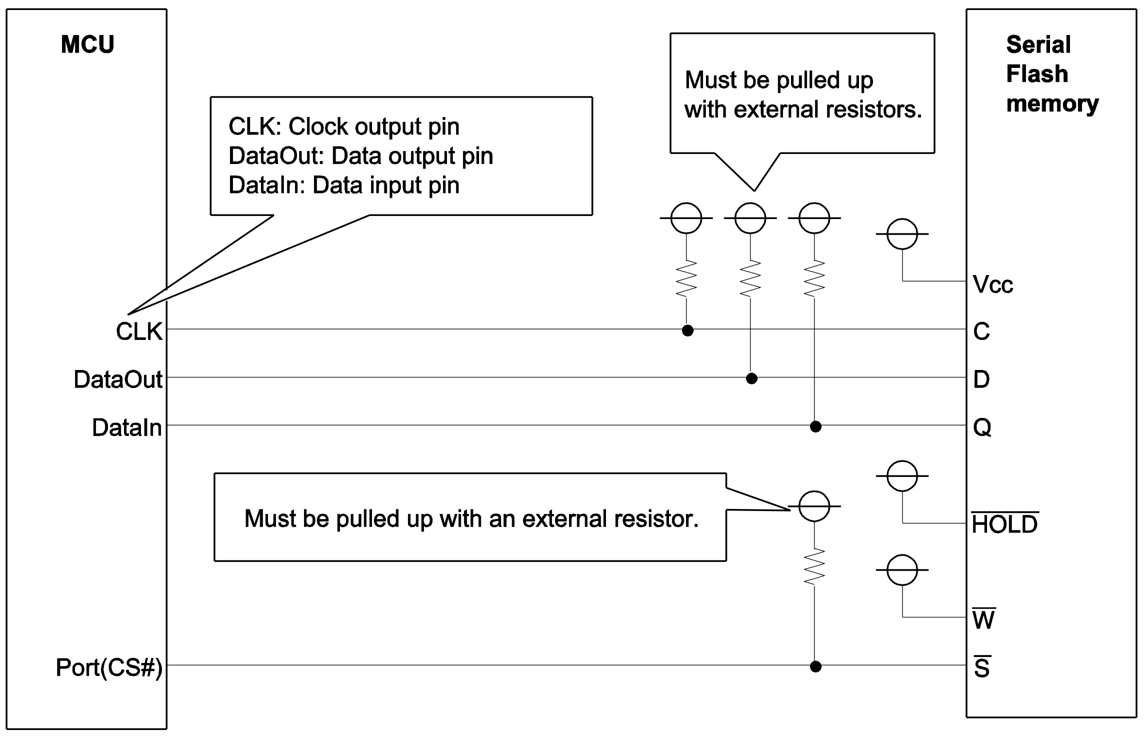
Table 4-1 List of Pins Used

Pin Name	I/O	Description
CLK *1	Output	Clock output
DataOut *1	Output	Master data output
DataIn *1	Input	Master data input
Port(CS#) *1	Output	Storage device select output

Note: *1 In this application note, the pin names CLK, DataIn, DataOut, and Port (CS#) are used in accordance with the pin names used in the sample code.

4.2 Reference Circuit

Figure 2 shows a sample wiring configuration.



• The names of the MCU pins used for serial I/O are dependent on the individual MCU.
• In this application note, pin names CLK, DataIn, DataOut, and Port (CS#) are used in accordance with the sample code.

Figure 2 Sample Wiring Diagram for an MCU and an SPI Slave Device

5. Software Description

5.1 Operation Outline

The MCU's clock synchronous (3-wire) serial communication function is used to realize the control of Serial Flash memory devices.

The sample code explained in this section provides the following control functions:

- Connects the S pin of the SPI slave device to the Port pin of the MCU and controls it as an MCU's general port output (controlled by this sample code).
- Controls the input/output of the data in the clock synchronous mode (using an internal clock). (This sample code uses the MCU-specific clock synchronous single-master control software.)

In this sample code, the byte offset value of the data on the device is made equal to the byte offset value in the source or destination memory as illustrated in the figure below.

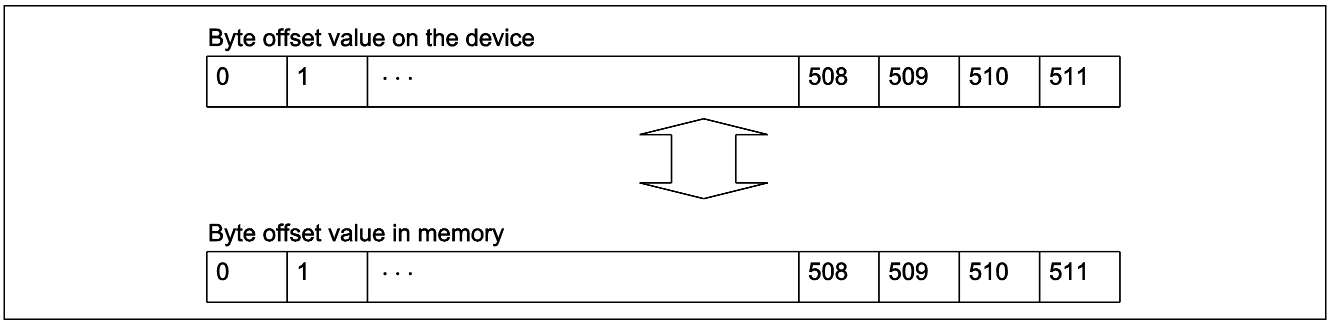
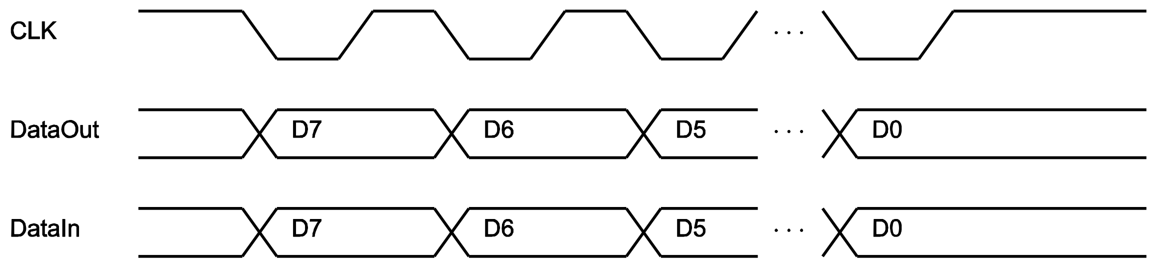


Figure 3 Storage Format of the Transferred Data

5.1.1 Clock Synchronous Mode Timing

The SPI mode 3 (CPOL=1, CPHA=1) timing shown in Figure 4 is used to control the Serial Flash memory.



- MCU->Serial Flash memory transmission: Transmission of transmit data is started on the falling edge of the transfer clock.
- Serial Flash memory->MCU reception: The receive data is taken in on the rising edge of the transfer clock.
- MSB first mode transfer

The level of the CLK pin is held high when no transfer processing is in progress.

Figure 4 Clock Synchronous Mode Timing Setup

For available serial clock frequencies, see the datasheets for the individual MCUs and SPI devices.

5.1.2 Serial Flash memory S Pin Control

The S pin of the Serial Flash memory is connected to the Port pin of the MCU and controlled as an MCU general port output.

The interval between the falling edge of the S (MCU's Port(CS#)) signal of the Serial Flash memory and the falling edge of the C (MCU's CLK) signal of the Serial Flash memory is controlled with software wait processing to account for the Serial Flash memory S setup time.

The interval between the rising edge of the C (MCU's CLK) signal of the Serial Flash memory and the rising edge of the S (MCU's Port (CS#)) signal of the Serial Flash memory is controlled with software wait processing to account for the Serial Flash memory S hold time.

Check the datasheet for the Serial Flash memory in use and set up the software wait times that are appropriate to your system.

5.1.3 Serial Flash memory Instruction Code

The instruction codes listed in the table below are available for controlling the Serial Flash memory. These codes are used to carry out command control of the Serial Flash memory.

Table 5-1 Instruction Set

Instruction	Description	Instruction Format
WREN	Write Enable	0000 0110
WRDI	Write Disable	0000 0100
RDID	Read Identification	1001 1111
RDSR	Read Status-Register	0000 0101
WRSR	Write Status-Register	0000 0001
READ	Read Data Bytes	0000 0011
FAST READ	Read Data Bytes at Higher Speed	0000 1011
PP	Page Program	0000 0010
SE	Sector Erase	1101 1000
BE	Bulk Erase	1100 0111
DP	Deep Power-down	1011 1001
RES	Release from Deep Power-down, and Read Electric Signature	1010 1011
	Release from Deep Power-down	
	Read Electric Signature	

Note: The DP and RES instructions are not supported (dependent on the device in use).

5.1.4 Serial Flash memory Status Register

The status register of the Serial Flash memory can be read and written using dedicated instructions.

See the Serial Flash memory's datasheet for details on the individual status register bits.

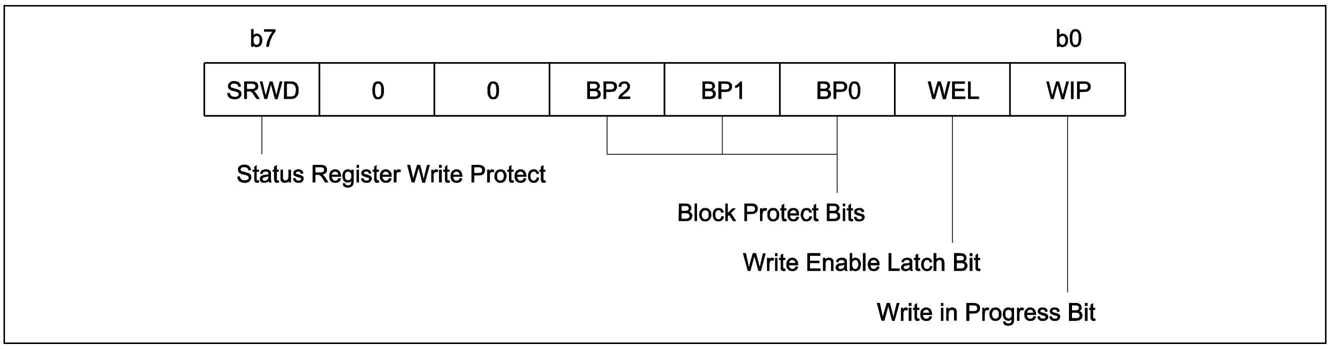


Figure 5 Status Register Configuration

5.2 Software Control Outline

5.2.1 Software Configuration

The sample code ranks in the higher-level layer of the SPI Serial Flash memory control software system (flash memory control software shown in Figure 6).

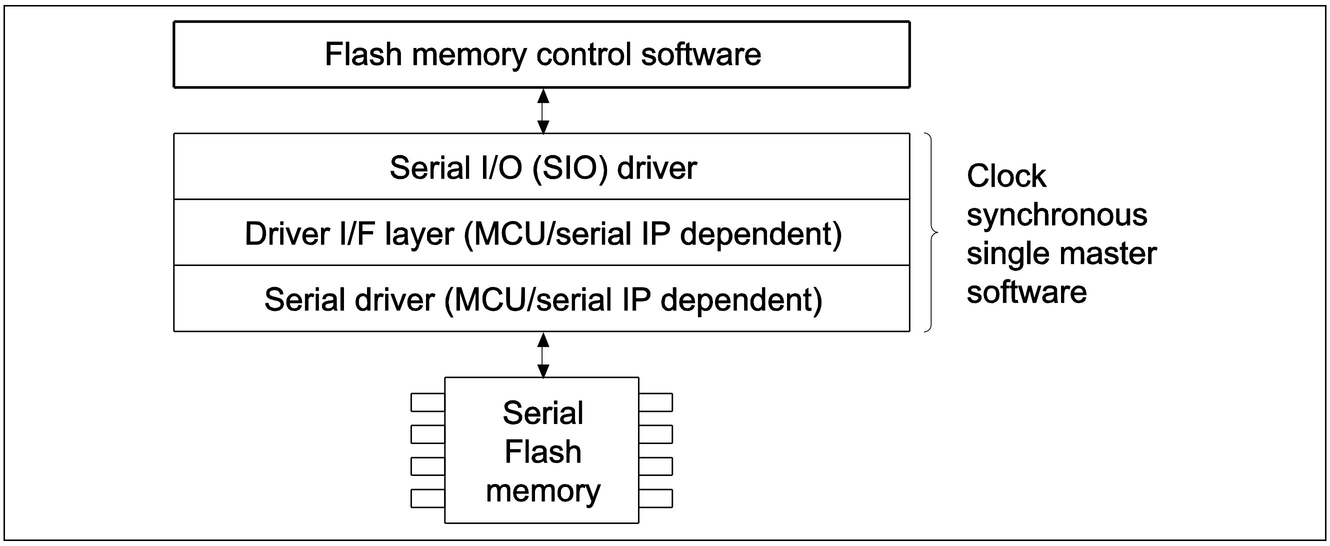

Flash memory control software
Serial I/O (SIO) driver
Driver I/F layer (MCU/serial IP dependent)
Serial driver (MCU/serial IP dependent)
Clock synchronous single master software
Serial Flash memory

Figure 6 Software Configuration

The general control procedure is given below.

(1) Set the Port (CS#) signal low.
(2) Software wait.
(3) Send/receive command/data using the clock synchronous single-master software.
(4) Software wait.
(5) Set the Port (CS#) signal high.

This sample code is made up of the following eight basic routines:

- Chip Select pin initialization
  Set the Port (CS#) pin high.
- Chip Select pin control
  Set the Port (CS#) pin high/low.
- Software wait
  Adjust timing using software wait.
- Serial enabling
  Set the DataIn pin for port input, set the DataOut and CLK pins high, Enable serial I/O and set the bit rate.
- Command transmission
  Send command to the Serial Flash memory.
- Data transmission
  Send data to the Serial Flash memory.
- Data reception
  Receive response/data from the Serial Flash memory.
- Serial disabling
  Disable serial I/O, set the DataIn pin for port input, set the DataOut and CLK pins high.

5.2.2 Chip Select Pin Initialization (R_SPI_FLASH_Init_Port())

This routine sets the Chip Select signal of the slave device high (deactivates the device before operation).

The basic control procedure is as follows, though the actual control procedure varies from MCU to MCU:

(1) Set the port output value to "High" output (to generate a high output when the port configuration is switched to "output").
(2) Set the port for "output."
(3) Set the port output value to "High" output.

5.2.3 Chip Select Pin Control (FLASH_SET_CS())

This routine sets the Chip Select signal of the slave device high or low.

5.2.4 Software Wait (mtl_wait_lp())

This routine controls wait processing using a software loop.

5.2.5 Serial Enabling (R_SIO_Enable())

This routine enables the serial interface using the following procedure:

(1) Sets the DataIn pin to be used for serial I/O for port input and set the DataOut and CLK pins high.
(2) Enables the serial I/O function and switches the DataIn pin for data input, the DataOut pin for data output, and the CLK pin for clock output.
(3) Sets the baud rate (bit rate) to be used for serial I/O.

For details on R_SIO_Enable(), refer to the individual clock synchronous single-master software application note.

5.2.6 Command Transmission (R_SPI_FLASH_Send_Cmd())

This routine sends an instruction code (command) to the Serial Flash memory.

R_SIO_Rx_Data () is used to receive the response to the command.

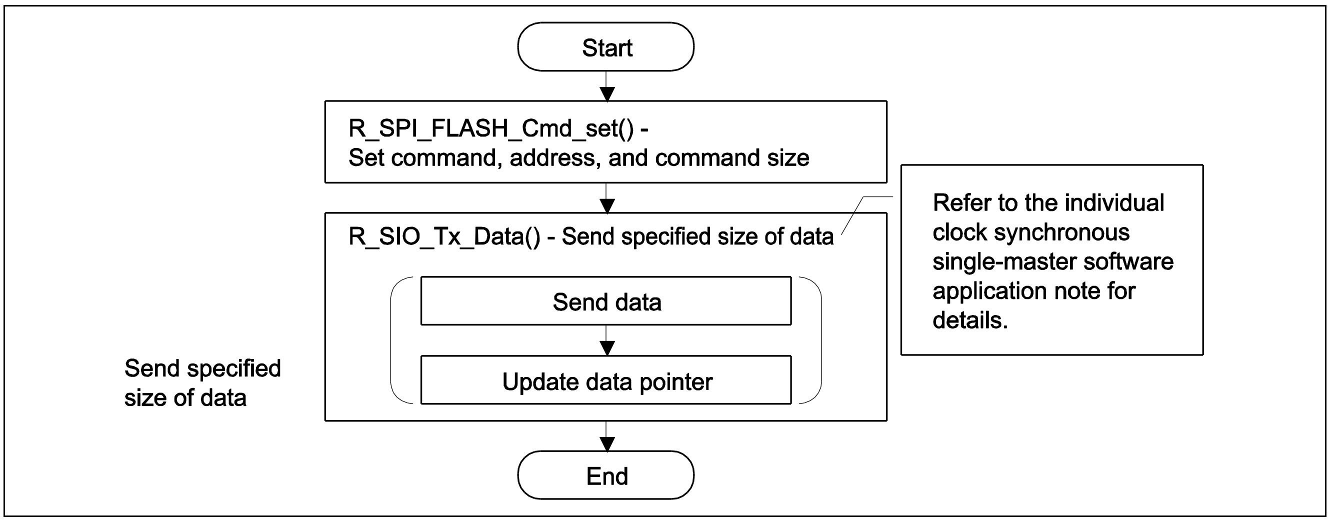

Start
R_SPI_FLASH_Cmd_set() -
Set command, address, and command size
R_SIO_Tx_Data() - Send specified size of data
Send data
Update data pointer
Send specified size of data
Refer to the individual clock synchronous single-master software application note for details.
End

Figure 7 Outline of Command Transmission Processing (R_SPI_FLASH_Send_Cmd())

5.2.7 Data Transmission (R_SIO_Tx_Data ())

This routine sends data using the serial I/O function. It sends an instruction code, address information, write data or the value of the status register.

Refer to the individual clock synchronous single-master software application note for details.

5.2.8 Data Reception (R_SIO_Rx_Data ())

This routine receives data using the serial I/O function. It receives either read data or the value of the status register.

Refer to the individual clock synchronous single-master software application note for details.

5.2.9 Serial Disabling (R_SIO_Disable ())

This routine switches the pin to be used for serial I/O to a port pin and sets the DataIn pin for port input and sets the DataOut and CLK pins high.

Refer to the individual clock synchronous single-master software application note for details.

5.3 Sizes of Required Memory

The sizes of the required memory areas are given below.

5.3.1 RX Family

(1) RX610 SCI

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-2 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1119 bytes (little endian) 1291 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	0 bytes (little endian) 5 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	160 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The above-mentioned memory sizes vary with the endian mode adopted.
The above-mentioned Maximum user stack size is included the stack size of clock synchronous single-master control software which lies in the lower-level layer of the software.

(2) **RX62N RSPI**

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-3 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1119 bytes (little endian) 1291 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	0 bytes (little endian) 5 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	160 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The above-mentioned memory sizes vary with the endian mode adopted.
The above-mentioned Maximum user stack size is included the stack size of clock synchronous single-master control software which lies in the lower-level layer of the software.

(3) **RX62N SCI**

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-4 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1119 bytes (little endian) 1291 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	0 bytes (little endian) 5 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	160 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The above-mentioned memory sizes vary with the endian mode adopted.
The above-mentioned Maximum user stack size is included the stack size of clock synchronous single-master control software which lies in the lower-level layer of the software.

(4) **RX63N RSPI**

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-5 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1119 bytes (little endian) 1291 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	0 bytes (little endian) 5 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	160 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The above-mentioned memory sizes vary with the endian mode adopted.
The above-mentioned Maximum user stack size is included the stack size of clock synchronous single-master control software which lies in the lower-level layer of the software.

(5) **RX63N SCI**

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-6 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1119 bytes (little endian) 1291 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	0 bytes (little endian) 5 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	160 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The above-mentioned memory sizes vary with the endian mode adopted.
The above-mentioned Maximum user stack size is included the stack size of clock synchronous single-master control software which lies in the lower-level layer of the software.

(6) RX63T RSPI

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-7 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1119 bytes (little endian) 1291 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	0 bytes (little endian) 5 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	160 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The above-mentioned memory sizes vary with the endian mode adopted.
The above-mentioned Maximum user stack size is included the stack size of clock synchronous single-master control software which lies in the lower-level layer of the software.

(7) RX63T SCI

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-8 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1119 bytes (little endian) 1291 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	0 bytes (little endian) 5 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	160 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The above-mentioned memory sizes vary with the endian mode adopted.
The above-mentioned Maximum user stack size is included the stack size of clock synchronous single-master control software which lies in the lower-level layer of the software.

(8) RX210 RSPI

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-9 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1119 bytes (little endian) 1291 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	0 bytes (little endian) 5 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	160 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The above-mentioned memory sizes vary with the endian mode adopted.
The above-mentioned Maximum user stack size is included the stack size of clock synchronous single-master control software which lies in the lower-level layer of the software.

(9) RX210 SCI

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-10 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1119 bytes (little endian) 1291 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	0 bytes (little endian) 5 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	160 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The above-mentioned memory sizes vary with the endian mode adopted.
The above-mentioned Maximum user stack size is included the stack size of clock synchronous single-master control software which lies in the lower-level layer of the software.

(10) **RX21A RSPI**

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-11 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1119 bytes (little endian) 1291 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	0 bytes (little endian) 5 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	160 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The above-mentioned memory sizes vary with the endian mode adopted.
The above-mentioned Maximum user stack size is included the stack size of clock synchronous single-master control software which lies in the lower-level layer of the software.

(11) **RX21A SCI**

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-12 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1119 bytes (little endian) 1291 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	0 bytes (little endian) 5 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	160 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The above-mentioned memory sizes vary with the endian mode adopted.
The above-mentioned Maximum user stack size is included the stack size of clock synchronous single-master control software which lies in the lower-level layer of the software.

(12) **RX220 RSPI**

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-13 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1119 bytes (little endian) 1291 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	0 bytes (little endian) 5 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	160 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The above-mentioned memory sizes vary with the endian mode adopted.
The above-mentioned Maximum user stack size is included the stack size of clock synchronous single-master control software which lies in the lower-level layer of the software.

(13) **RX220 SCI**

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-14 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1119 bytes (little endian) 1291 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	0 bytes (little endian) 5 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	160 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The above-mentioned memory sizes vary with the endian mode adopted.
The above-mentioned Maximum user stack size is included the stack size of clock synchronous single-master control software which lies in the lower-level layer of the software.

(14) RX111 RSPI

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-15 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	2,474 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	4+n bytes (little endian) (n denotes the number of devices to be used.)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	180 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The above-mentioned memory sizes vary with the endian mode adopted.
The above-mentioned Maximum user stack size is included the stack size of clock synchronous single-master control software which lies in the lower-level layer of the software.

(15) RX111 SCI

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-16 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	2,474 bytes (little endian)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	4+n bytes (little endian) (n denotes the number of devices to be used.)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	176 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The above-mentioned memory sizes vary with the endian mode adopted.
The above-mentioned Maximum user stack size is included the stack size of clock synchronous single-master control software which lies in the lower-level layer of the software.

5.3.2 RL78 Family, 78K0R/Kx3-L

(1) 78K0R/Kx3-L SAU

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-17 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1855 bytes 1824 bytes	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	0 bytes 6 bytes	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	160 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The above-mentioned Maximum user stack size is included the stack size of clock synchronous single-master control software which lies in the lower-level layer of the software.

(2) RL78/G14 SAU Integrated Development Environment CubeSuite+ (Compiler:CA78K0R)

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-18 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1598 bytes 1638 bytes	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	0 bytes 6 bytes	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	114 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The above-mentioned Maximum user stack size is included the stack size of clock synchronous single-master control software which lies in the lower-level layer of the software.

(3) RL78/G14 SAU Integrated Development Environment CS+ for CC (Compiler:CC-RL)

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-19 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1491 bytes 1389 bytes	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	0 bytes 6 bytes	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	102 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The above-mentioned Maximum user stack size is included the stack size of clock synchronous single-master control software which lies in the lower-level layer of the software.

(4) RL78/G14 Integrated Development Environment IAR Embedded Workbench

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-20 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1804 bytes 1590 bytes	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	0 bytes 6 bytes	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	130 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The maximum user stack size is the stack size for the entire project. It includes the stack of the lower-layer clock synchronous single-master control software.

(5) RL78/L13 SAU Integrated Development Environment CubeSuite+ (Compiler:CA78K0R)

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-21 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	3,237 bytes	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	6 bytes (The number of devices to be used is 1 or 2.)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	114 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The above-mentioned Maximum user stack size is included the stack size of clock synchronous single-master control software which lies in the lower-level layer of the software.

(6) RL78/L13 SAU Integrated Development Environment CS+ for CC (Compiler:CC-RL)

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-22 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	2,884 bytes	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	6 bytes (The number of devices to be used is 1 or 2.)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	102 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The above-mentioned Maximum user stack size is included the stack size of clock synchronous single-master control software which lies in the lower-level layer of the software.

(7) RL78/L13 Integrated Development Environment IAR Embedded Workbench

See chapter 2, Conditions for Checking the Operation of the SPI Serial Flash memory Control Software, for the environment.

Table 5-23 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	2,942 bytes	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
RAM	6 bytes (The number of devices to be used is 1 or 2.)	R_SPI_FLASH_m25p_usr.c R_SPI_FLASH_m25p_io.c
Maximum user stack size	120 bytes	
Maximum interrupt stack size	—	Not interrupt used

Note: The sizes of required memory areas vary with the version and compiler options of the C compiler.
These sizes do not include the size of the memory that is used by the clock synchronous single-master software.
The above-mentioned memory sizes vary with the type of MCU to be used.
The maximum user stack size is the stack size for the entire project. It includes the stack of the lower-layer clock synchronous single-master control software.

5.4 File Configuration

The following table lists the files that are used for the sample code. The table excludes the files that are automatically generated by the integrated development environment.

Table 5-24 File Configuration

\an_r01an0566ej0106_mcu_serial				<DIR>	Folder for the sample code
	r01an0566ej0106_mcu.pdf				Application note
	\source			<DIR>	Folder for storing the programs
		\r_spi_flash_m25p		<DIR>	Folder for the Serial Flash memory control software
			R_SPI_FLASH_m25p.h		Header file
			R_SPI_FLASH_m25p_io.c		I/O module
			R_SPI_FLASH_m25p_io.h		I/O module common definitions
			R_SPI_FLASH_m25p_usr.c		User I/F module
			R_SPI_FLASH_sfr.h.78k0r		Common register definitions (78K0R/Kx3-L)
			R_SPI_FLASH_sfr.h.rl78g14		Common register definitions (RL78/G14)
			R_SPI_FLASH_sfr.h.rl78g1c		Common register definitions (RL78/G1C)
			R_SPI_FLASH_sfr.h.rl78l1c		Common register definitions (RL78/L1C)
			R_SPI_FLASH_sfr.h.rl78l12		Common register definitions (RL78/L12)
			R_SPI_FLASH_sfr.h.rl78l13		Common register definitions (RL78/L13)
			R_SPI_FLASH_sfr.h.rx21a		Common register definitions (RX21A)
			R_SPI_FLASH_sfr.h.rx62n		Common register definitions (RX62N)
			R_SPI_FLASH_sfr.h.rx63n		Common register definitions (RX63N)
			R_SPI_FLASH_sfr.h.rx63t		Common register definitions (RX63T)
			R_SPI_FLASH_sfr.h.rx111		Common register definitions (RX111)
			R_SPI_FLASH_sfr.h.rx210		Common register definitions (RX210)
			R_SPI_FLASH_sfr.h.rx220		Common register definitions (RX220)
			R_SPI_FLASH_sfr.h.rx610		Common register definitions (RX610)
		\sample		<DIR>	Folder for storing the test program
			testmain.c		Sample program for testing

Note: An MCU-specific clock synchronous single-master control program is separately required.

5.5 List of Constants

5.5.1 Return Values

The following table lists the return values that are returned by the sample code.

Table 5-25 Return Values

Constant Name	Value	Description
FLASH_OK	(error_t)(0)	Successful Operation
FLASH_ERR_PARAM	(error_t)(-1)	Parameter Error
FLASH_ERR_HARD	(error_t)(-2)	Hardware Error
FLASH_ERR_OTHER	(error_t)(-7)	Other Error

5.5.2 Command Definitions

The following table lists the command definitions that are used in the sample code.

Table 5-26 Command Definitions

Constant Name	Value	Description
FLASH_CMD_WREN	(uint8_t)0x06	Write Enable
FLASH_CMD_WRDI	(uint8_t)0x04	Write Disable
FLASH_CMD_RDSR	(uint8_t)0x05	Read Status Register
FLASH_CMD_WRSR	(uint8_t)0x01	Write Status Register
FLASH_CMD_READ	(uint8_t)0x0b	Read for Memory Array at Higher Speed
FLASH_CMD_READ	(uint8_t)0x03	Read for Memory Array
FLASH_CMD_WRITE	(uint8_t)0x02	Write for Memory Array
FLASH_CMD_SE	(uint8_t)0xd8	S_Erase for Memory Array
FLASH_CMD_BE	(uint8_t)0xc7	B_Erase for Memory Array
FLASH_CMD_DP	(uint8_t)0xb9	Deep Power Down
FLASH_CMD_RES	(uint8_t)0xab	Release Deep Power Down
FLASH_CMD_RDID	(uint8_t)0x9f	Read Identification

5.5.3 Miscellaneous Definitions

The following table lists miscellaneous definitions that are used in the sample code.

Table 5-27 Miscellaneous Definitions

Constant Name	Value	Description
FLASH_LOG_ERR	1	Log Type: Error
FLASH_TRUE	(uint8_t)0x01	Flag "ON"
FLASH_FALSE	(uint8_t)0x00	Flag "OFF"
FLASH_WP_NONE	(uint8_t)0x00	Write-Protection Status : None setting
FLASH_B_ERASE	(uint8_t)0x00	Erasure Type: Bulk Erase
FLASH_S_ERASE	(uint8_t)0x01	Erasure Type: Sector Erase
FLASH_MEM_SIZE	(uint32_t)2097152	Memory size (in bytes) The value shown to the left is for 16 Mbit memory.
FLASH_SECT_ADDR	(uint32_t)0xffff0000	Sector address mask value on erasing a sector The value shown to the left is for 16 Mbit memory.
FLASH_WPAG_SIZE	(uint32_t)256	Page size (in bytes) The value shown to the left is for 16 Mbit memory.
FLASH_ADDR_SIZE	(uint8_t)3	Address size (in bytes) The value shown to the left is for 16 Mbit memory.
FLASH_WP_WHOLE_MEM	(uint8_t)0x07	Write protection for the chip as a whole
FLASH_CMD_SIZE	(uint8_t)1	Command size (in bytes)
FLASH_STSREG_SIZE	(uint16_t)1	Status register size (in bytes)
FLASH_IDDATA_SIZE	(uint16_t)3	ID Data size (in bytes)
FLASH_ESIG_SIZE	(uint16_t)1	Electronic Signature size (in bytes)
FLASH_SHORT_SIZE	(uint32_t)0x0000fff0	Sets the maximum transfer size for lower-level functions (set to 0xFFFF or lower.)
FLASH_HI	(uint8_t)0x01	Port "H"
FLASH_LOW	(uint8_t)0x00	Port "L"
FLASH_OUT	(uint8_t)0x01	Port Output Setting
FLASH_IN	(uint8_t)0x00	Port Input Setting
FLASH_WBUSY_WAIT	(uint16_t)18000	Write Busy Waiting Time $18000 \times 1\ \mu s = 18$ ms
FLASH_T_WBUSY_WAIT	(uint16_t)MTL_T_1US	Write Busy Completion Polling Time
FLASH_T_EBUSY_WAIT	(uint16_t)MTL_T_1MS	Erase Busy Completion Polling Time
FLASH_T_DP_WAIT	(uint16_t)MTL_T_4US	Deep Standby Completion Waiting Time
FLASH_T_RES_WAIT	(uint16_t)MTL_T_30US	Release Deep Standby Completion Waiting Time
FLASH_T_CS_HOLD	(uint16_t)MTL_T_1US	CS Stability Waiting Time
FLASH_T_R_ACCESS	(uint16_t)MTL_T_1US	Reading Start Waiting Time
FLASH_REG_SRWD	(uint8_t)0x80	Status Register Write Disable
FLASH_REG_BP2	(uint8_t)0x10	Block Protection Bit2 (unused)
FLASH_REG_BP1	(uint8_t)0x08	Block Protection Bit1 (unused)
FLASH_REG_BP0	(uint8_t)0x04	Block Protection Bit0 (unused)
FLASH_REG_WEL	(uint8_t)0x02	Write Enable Latch Bit
FLASH_REG_WIP	(uint8_t)0x01	Write In Progress Bit
FLASH_BYTE_READ	2	Number of bytes used to identify the execution of a 1-byte read
FLASH_BYTE_WRITE	2	Number of bytes used to identify the execution of a 1-byte write

5.6 Structures and Unions

Shown below are the structures that are used in the sample code.

```
typedef union {
  uint32_t  ul;
  uint8_t uc[4];
} FLASH_EXCHG_LONG;                                  /* total 4byte                */
```

5.7 List of Variables

The following table shows a list of variables that are used in the sample code.

Table 5-28 List of Variables

Type	Variable Name	Description	Used in
STATIC uint8_t	gFlash_CmdBuf[5]	Command buffer	R_SPI_FLASH_Send_Cmd R_SPI_FLASH_Cmd_set

5.8 List of Functions

The following table lists the functions that are used in the sample code.

Table 5-29 List of Functions

Function Name	Outline
R_SPI_FLASH_Init_Driver	Initialize driver.
R_SPI_FLASH_Read_Status	Read status.
R_SPI_FLASH_Write_Protect	Set write protection.
R_SPI_FLASH_Read_Data	Read data.
R_SPI_FLASH_Write_Data	Write data.
R_SPI_FLASH_BulkErase	Erases all data in chip.
R_SPI_FLASH_SectorErase	Erases the sector.
R_SPI_FLASH_DeepPDown	Deep Power-down Processing
R_SPI_FLASH_ReleaseDeepPDown	Deep Power-down Release Processing
R_SPI_FLASH_ReadID	Read ID.
R_SPI_FLASH_Init_Port	Initialize port.
R_SPI_FLASH_Send_Cmd	Send command.
R_SPI_FLASH_Write_En	Enable writes.
R_SPI_FLASH_Write_Di	Disable writes.
R_SPI_FLASH_Read_StsReg	Read status register.
R_SPI_FLASH_Write_StsReg	Write status register.
R_SPI_FLASH_Wait_Busy	Busy wait.
R_SPI_FLASH_Write_Page	Write.
R_SPI_FLASH_Read_Memory	Read memory.
R_SPI_FLASH_Erase	Erase.
R_SPI_FLASH_DPD	Deep Power-down Command Processing
R_SPI_FLASH_ReleaseDPD	Deep Power-down Release Command Processing
R_SPI_FLASH_ID	Read ID.
R_SPI_FLASH_Cmd_set	Set command.

When using an MCU that incorporates cache memory, allocate the read/write buffer to a non-cache area.

The address of the buffer for storing read or write data is dependent on the individual MCU-specific clock synchronous single-master control software in the lower-level layer; there are cases in which it is necessary to specify a 4-byte boundary. Refer to the individual MCU-specific clock synchronous single-master control software application note.

5.9 Function Details

5.9.1 Driver Initialization

R_SPI_FLASH_Init_Driver

Synopsis	Initializes driver.
Headers	R_SPI_FLASH_m25p.h
Declaration	error_t R_SPI_FLASH_Init_Driver(void)
Explanation	• Calls the R_SPI_FLASH_Init_Port() function to initialize the CS# pin. • Calls the serial I/O driver's R_SIO_Init_Driver() function to initialize the I/O port. • This function must be called only once at system start time.
Arguments	void
Return value	• A description of the results of initialization is returned. FLASH_OK ; Successful operation FLASH_ERR_OTHER ; Other error Returns the return value of R_SIO_Init_Driver().
Remarks	None

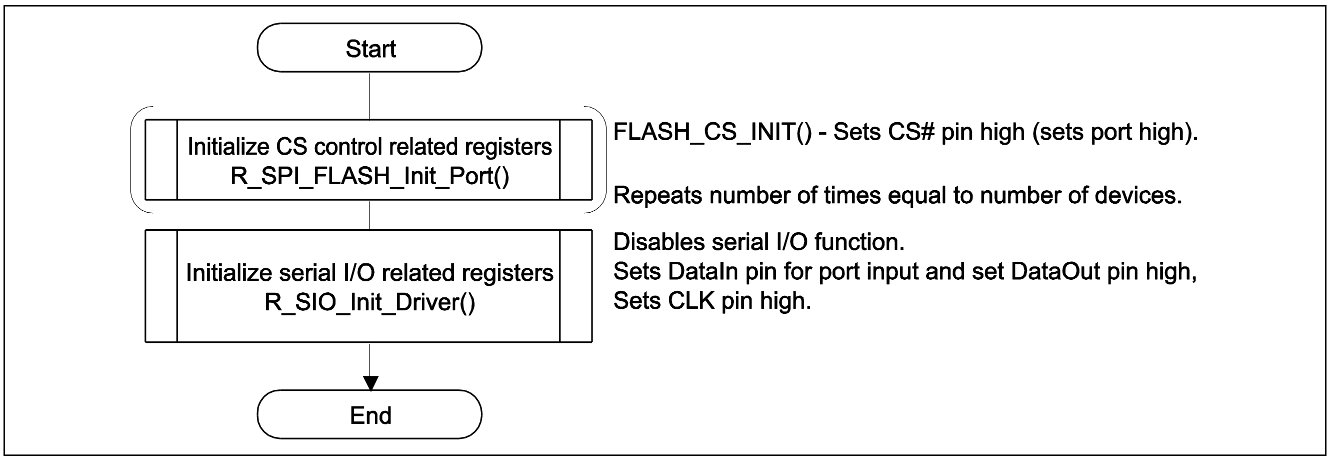

Start
Initialize CS control related registers
R_SPI_FLASH_Init_Port()
FLASH_CS_INIT() - Sets CS# pin high (sets port high).
Repeats number of times equal to number of devices.
Initialize serial I/O related registers
R_SIO_Init_Driver()
Disables serial I/O function.
Sets DataIn pin for port input and set DataOut pin high,
Sets CLK pin high.
End

Figure 8 Driver Initialization Processing Outline

5.9.2 Read Status Processing

R_SPI_FLASH_Read_Status

Synopsis	Reads status register.
Headers	R_SPI_FLASH_m25p.h
Declaration	error_t R_SPI_FLASH_Read_Status(uint8_t DevNo, uint8_t FAR* pStatus)

Explanation

- Reads the contents of the status register into pStatus.
  Set up a 1-byte read buffer.
- The read status buffer (pStatus) is loaded with the following information:

Bit 7: SRWD	1: BP2, BP1, BP0 are read-only bits
	0: BP2, BP1, BP0 are read/writable
Bit 6 to 5: Reserved (All "0")	
Bits 4 to 2: BP2, BP1, BP0	
Bit 1: WEL	1: Internal Write Enable Latch is set
	0: Internal Write Enable Latch is reset
Bit 0: WIP	1: Program or Erase cycle is in progress
	0: No Program or Erase cycle is in progress

Arguments

uint8_t	DevNo	;	Device number
uint8_t FAR*	pStatus	;	Buffer for storing read status

Return value

- Returns the result of reading the status register data.

FLASH_OK	;	Successful operation
FLASH_ERR_PARAM	;	Parameter error
FLASH_ERR_HARD	;	Hardware error
FLASH_ERR_OTHER	;	Other error

Remarks

- See the specification for the flash memory in use for the relation between protected areas and protection bits. There is a possibility of the BP bit not being allocated.

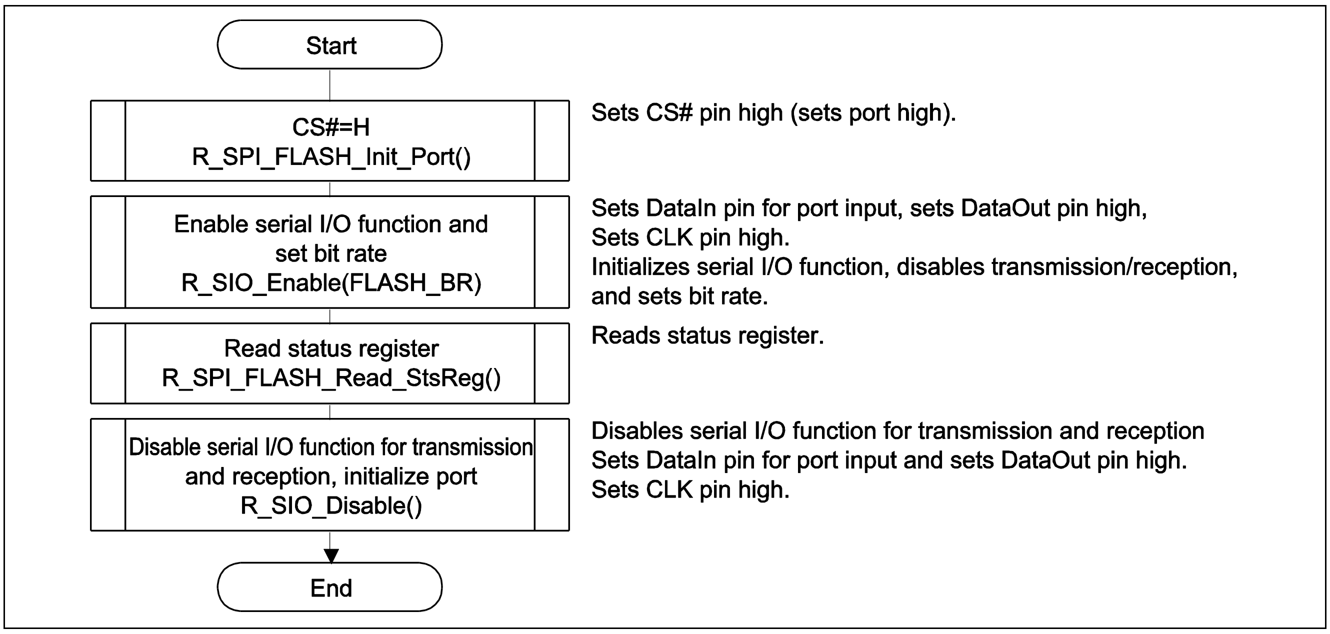

Start
CS#=H
R_SPI_FLASH_Init_Port()
Sets CS# pin high (sets port high).
Enable serial I/O function and set bit rate
R_SIO_Enable(FLASH_BR)
Sets DataIn pin for port input, sets DataOut pin high,
Sets CLK pin high.
Initializes serial I/O function, disables transmission/reception, and sets bit rate.
Read status register
R_SPI_FLASH_Read_StsReg()
Reads status register.
Disable serial I/O function for transmission and reception, initialize port
R_SIO_Disable()
Disables serial I/O function for transmission and reception
Sets DataIn pin for port input and sets DataOut pin high.
Sets CLK pin high.
End

Figure 9 Status Register Read Processing Outline

5.9.3 Write Protect Setup Processing

R_SPI_FLASH_Write_Protect

Synopsis	Performs write protect setup processing.
Headers	R_SPI_FLASH_m25p.h
Declaration	error_t R_SPI_FLASH_Write_Protect(uint8_t DevNo, uint8_t WpSts)

Explanation

- Sets write protection.
- The write protection setting data (WpSts) must be set up as follows:

WpSts=0:	BP0=0	BP1=0,	BP2=0
WpSts=1:	BP0=1	BP1=0,	BP2=0
WpSts=2:	BP0=0	BP1=1,	BP2=0
WpSts=3:	BP0=1	BP1=1,	BP2=0
WpSts=4:	BP0=0	BP1=0,	BP2=1
WpSts=5:	BP0=1	BP1=0,	BP2=1
WpSts=6:	BP0=0	BP1=1,	BP2=1
WpSts=7:	BP0=1	BP1=1,	BP2=1

Arguments

uint8_t	DevNo	;	Device number
uint8_t	WpSts	;	Write protect setting data

Return value

- Returns the result of write protect setup processing.

FLASH_OK	;	Successful operation
FLASH_ERR_PARAM	;	Parameter error
FLASH_ERR_OTHER	;	Other error

Remarks

- SRWD is set to 0.
- See the specification for the flash memory in use for the relation between protected areas and protection bits. There is a possibility of the BP bit not being allocated.

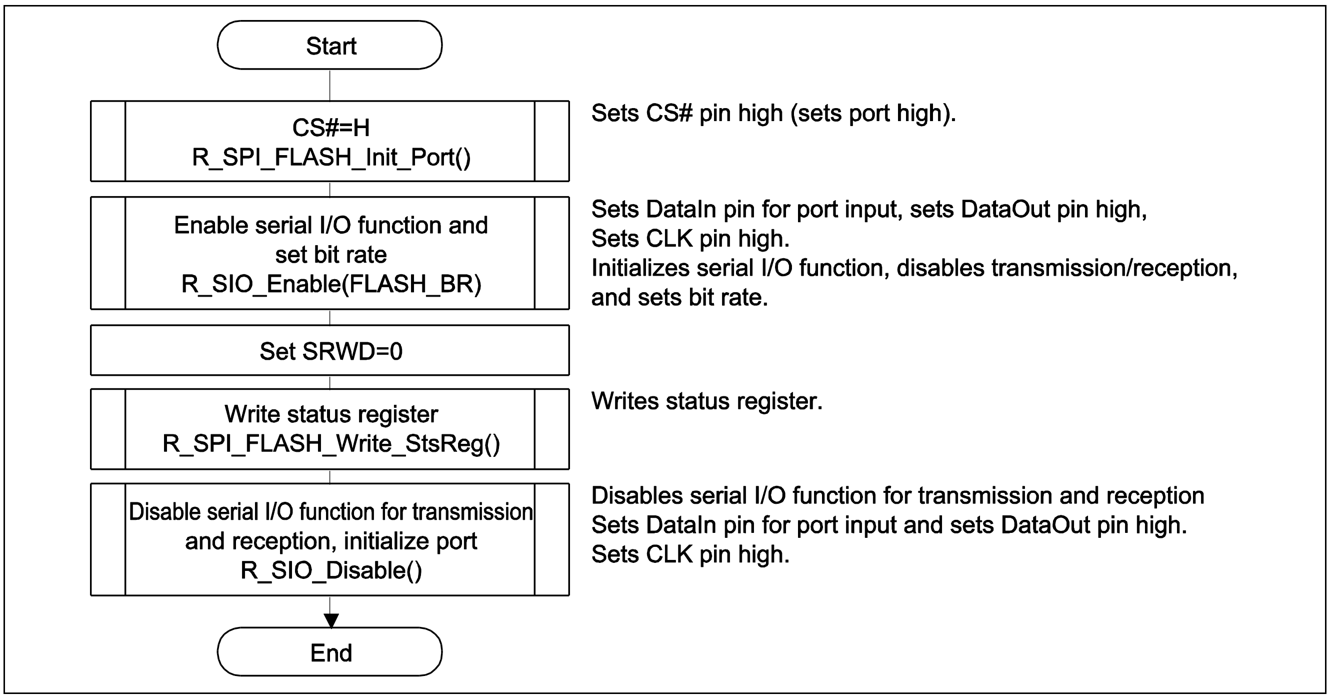

Start
CS#=H
R_SPI_FLASH_Init_Port()
Sets CS# pin high (sets port high).
Enable serial I/O function and set bit rate
R_SIO_Enable(FLASH_BR)
Sets DataIn pin for port input, sets DataOut pin high, Sets CLK pin high. Initializes serial I/O function, disables transmission/reception, and sets bit rate.
Set SRWD=0
Write status register
R_SPI_FLASH_Write_StsReg()
Writes status register.
Disable serial I/O function for transmission and reception, initialize port
R_SIO_Disable()
Disables serial I/O function for transmission and reception Sets DataIn pin for port input and sets DataOut pin high. Sets CLK pin high.
End

Figure 10 Write Protect Setup Processing Outline

5.9.4 Data Read Processing

R_SPI_FLASH_Read_Data

Synopsis	Reads data.
Headers	R_SPI_FLASH_m25p.h
Declaration	error_t R_SPI_FLASH_Read_Data(uint8_t DevNo, uint32_t RAddr, uint32_t RCnt, uint8_t FAR* pData)
Explanation	• Reads the specified number of bytes from flash memory, 1 byte at a time, starting at the specified address and stores the bytes in pData.

Arguments

uint8_t	DevNo	;	Device number
uint32_t	RAddr	;	Read start address
uint32_t	RCnt	;	Number of read bytes
uint8_t FAR*	pData	;	Pointer to read data buffer

Return value
• Returns the result of the data read.

FLASH_OK	;	Successful operation
FLASH_ERR_PARAM	;	Parameter error
FLASH_ERR_HARD	;	Hardware error
FLASH_ERR_OTHER	;	Other error

Remarks
• The highest read address is flash memory size -1.

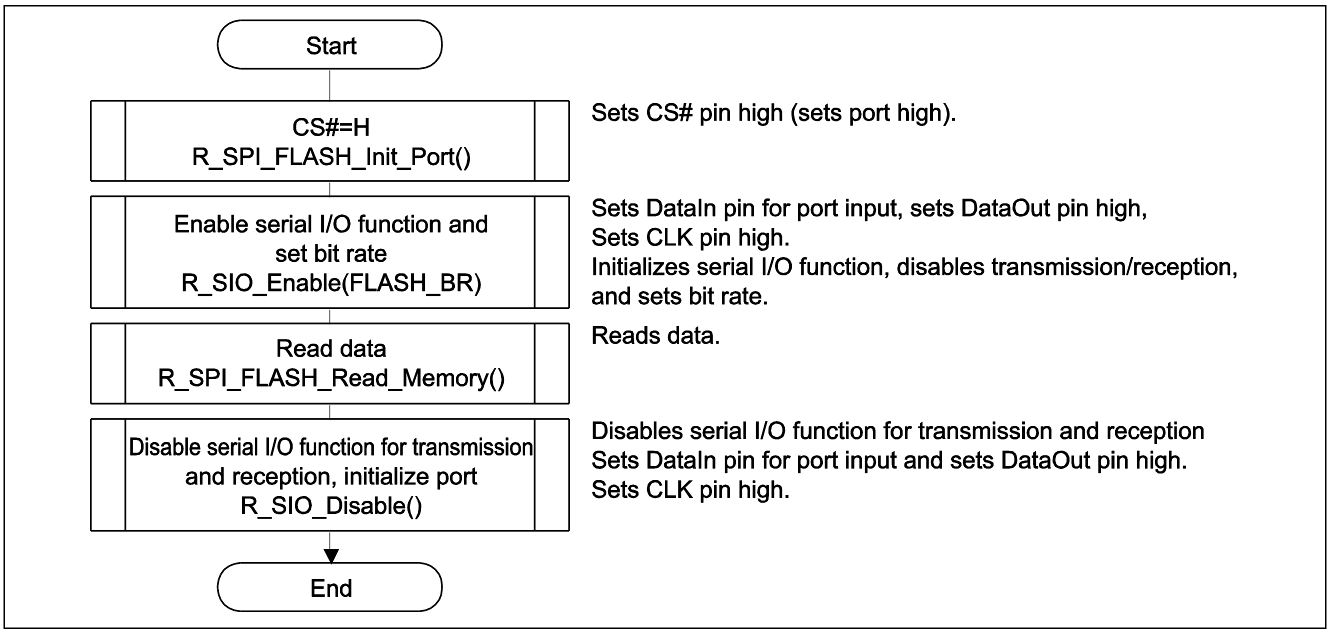

Start
CS#=H
R_SPI_FLASH_Init_Port()
Sets CS# pin high (sets port high).
Enable serial I/O function and set bit rate
R_SIO_Enable(FLASH_BR)
Sets DataIn pin for port input, sets DataOut pin high,
Sets CLK pin high.
Initializes serial I/O function, disables transmission/reception, and sets bit rate.
Read data
R_SPI_FLASH_Read_Memory()
Reads data.
Disable serial I/O function for transmission and reception, initialize port
R_SIO_Disable()
Disables serial I/O function for transmission and reception
Sets DataIn pin for port input and sets DataOut pin high.
Sets CLK pin high.
End

Figure 11 Data Read Processing Outline

5.9.5 Data Write Processing

R_SPI_FLASH_Write_Data

Synopsis	Writes data.
Headers	R_SPI_FLASH_m25p.h
Declaration	error_t R_SPI_FLASH_Write_Data(uint8_t DevNo, uint32_t WAddr, uint32_t WCnt, uint8_t FAR* pData)
Explanation	• Writes the data from pData into flash memory the specified number of bytes starting at the specified address.

Arguments

uint8_t	DevNo	;	Device number
uint32_t	WAddr	;	Write start address
uint32_t	WCnt	;	Number of bytes to write
uint8_t FAR*	pData	;	Pointer to write data buffer

Return value

- Returns the result of the data write.

FLASH_OK	;	Successful operation
FLASH_ERR_PARAM	;	Parameter error
FLASH_ERR_HARD	;	Hardware error
FLASH_ERR_OTHER	;	Other error

Remarks

- Writing into EEPROM is enabled only when write protection is off.
- Writing to protected areas is not possible. Furthermore, an error response is not returned even if this is attempted.
- The highest read address is flash memory size -1.

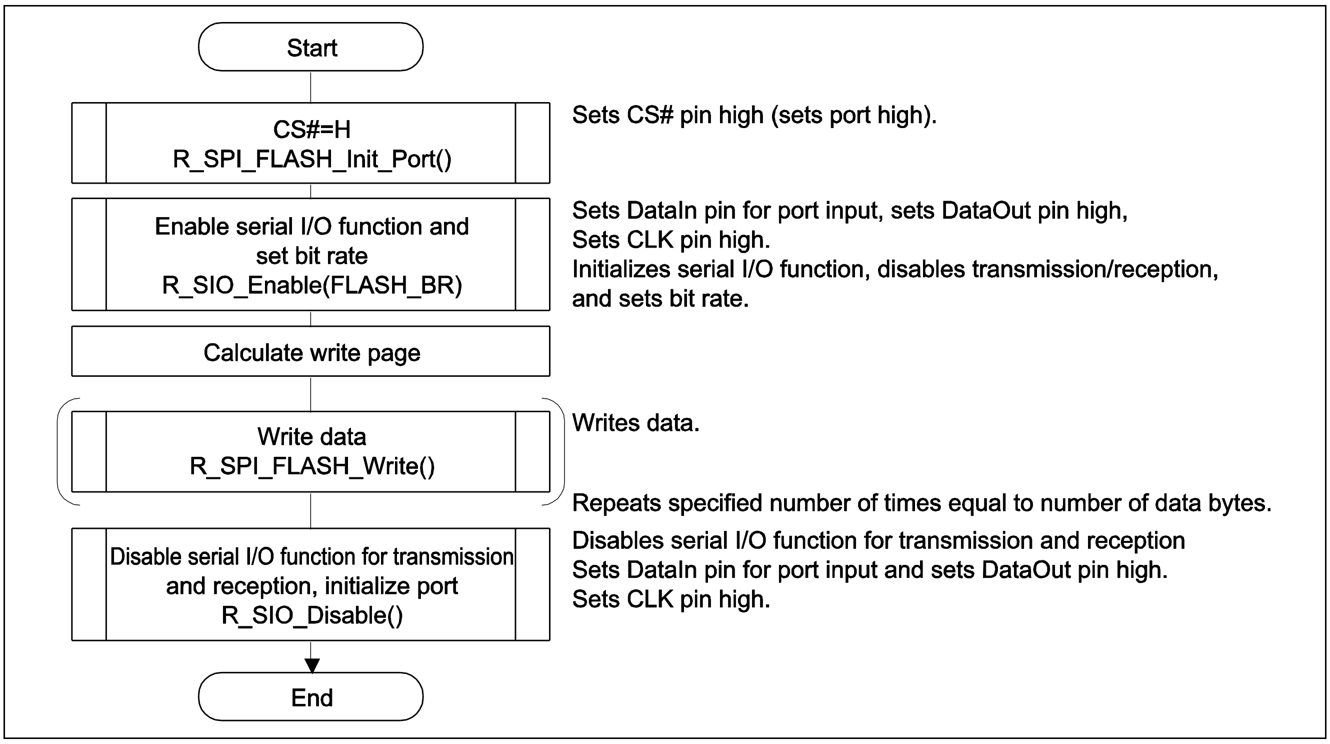

Start
CS#=H
R_SPI_FLASH_Init_Port()
Sets CS# pin high (sets port high).
Enable serial I/O function and set bit rate
R_SIO_Enable(FLASH_BR)
Sets DataIn pin for port input, sets DataOut pin high,
Sets CLK pin high.
Initializes serial I/O function, disables transmission/reception, and sets bit rate.
Calculate write page
Write data
R_SPI_FLASH_Write()
Writes data.
Repeats specified number of times equal to number of data bytes.
Disable serial I/O function for transmission and reception, initialize port
R_SIO_Disable()
Disables serial I/O function for transmission and reception
Sets DataIn pin for port input and sets DataOut pin high.
Sets CLK pin high.
End

Figure 12 Data Write Processing Outline

5.9.6 Chip Erasure Processing

R_SPI_FLASH_BulkErase

Synopsis	Performs chip erasure.
Headers	R_SPI_FLASH_m25p.h
Declaration	error_t R_SPI_FLASH_BulkErase(uint8_t DevNo)
Explanation	• Performs chip erasure.
Arguments	uint8_t DevNo ; Device number
Return value	• A description of the results of erasure is returned.
	FLASH_OK ; Successful operation
	FLASH_ERR_PARAM ; Parameter error
	FLASH_ERR_HARD ; Hardware error
	FLASH_ERR_OTHER ; Other error
Remarks	• Erasure of flash memory is only possible when memory protection is released.
	• Erasure is not possible if protection is in effect. If this is attempted, an error response is not returned.

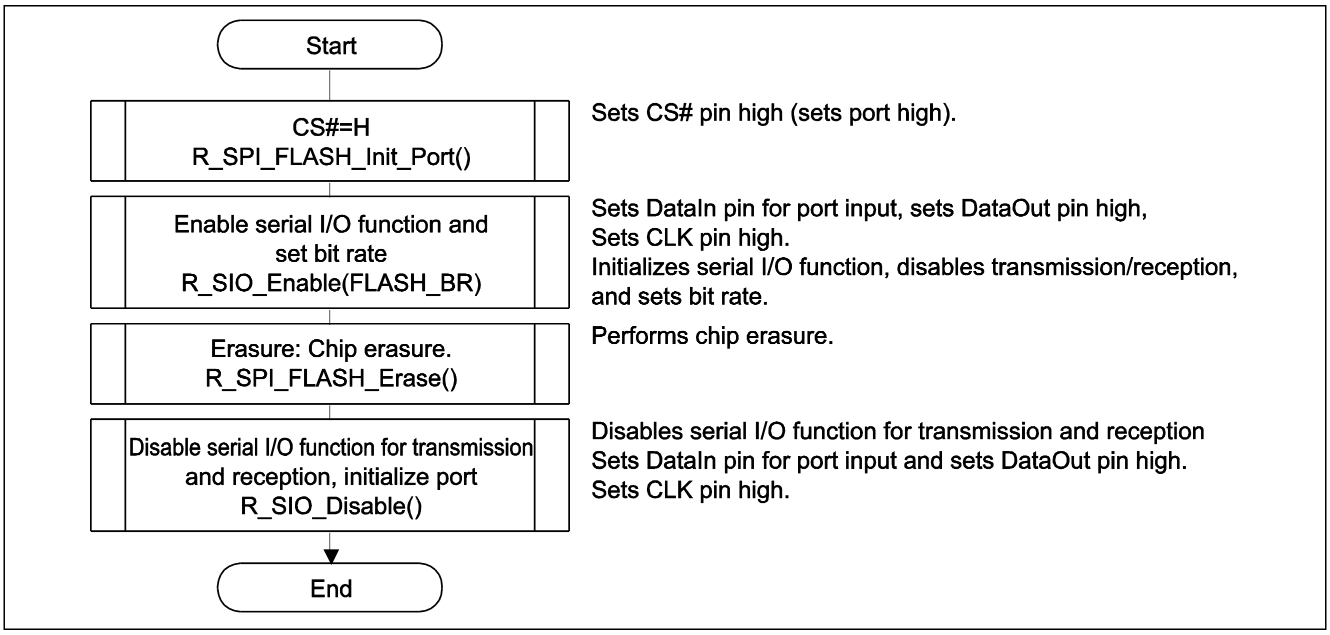

Start
CS#=H
R_SPI_FLASH_Init_Port()
Sets CS# pin high (sets port high).
Enable serial I/O function and set bit rate
R_SIO_Enable(FLASH_BR)
Sets DataIn pin for port input, sets DataOut pin high, Sets CLK pin high. Initializes serial I/O function, disables transmission/reception, and sets bit rate.
Erasure: Chip erasure.
R_SPI_FLASH_Erase()
Performs chip erasure.
Disable serial I/O function for transmission and reception, initialize port
R_SIO_Disable()
Disables serial I/O function for transmission and reception Sets DataIn pin for port input and sets DataOut pin high. Sets CLK pin high.
End

Figure 13 Chip Erasure Processing Outline

5.9.7 Sector Erasure Processing

R_SPI_FLASH_SectorErase

Synopsis	Performs sector erasure.			
Headers	R_SPI_FLASH_m25p.h			
Declaration	error_t R_SPI_FLASH_SectorErase(uint8_t DevNo, uint32_t EAddr)			
Explanation	• Erases all data in the specified sector.			
Arguments	uint8_t	DevNo	;	Device number
	uint32_t	EAddr	;	Erasure address
Return value	• A description of the results of erasure is returned.			
	FLASH_OK		;	Successful operation
	FLASH_ERR_PARAM		;	Parameter error
	FLASH_ERR_HARD		;	Hardware error
	FLASH_ERR_OTHER		;	Other error
Remarks	• Erasure of flash memory is only possible when memory protection is released. • Protected areas are not erasable. If this is attempted, an error response is not returned.			

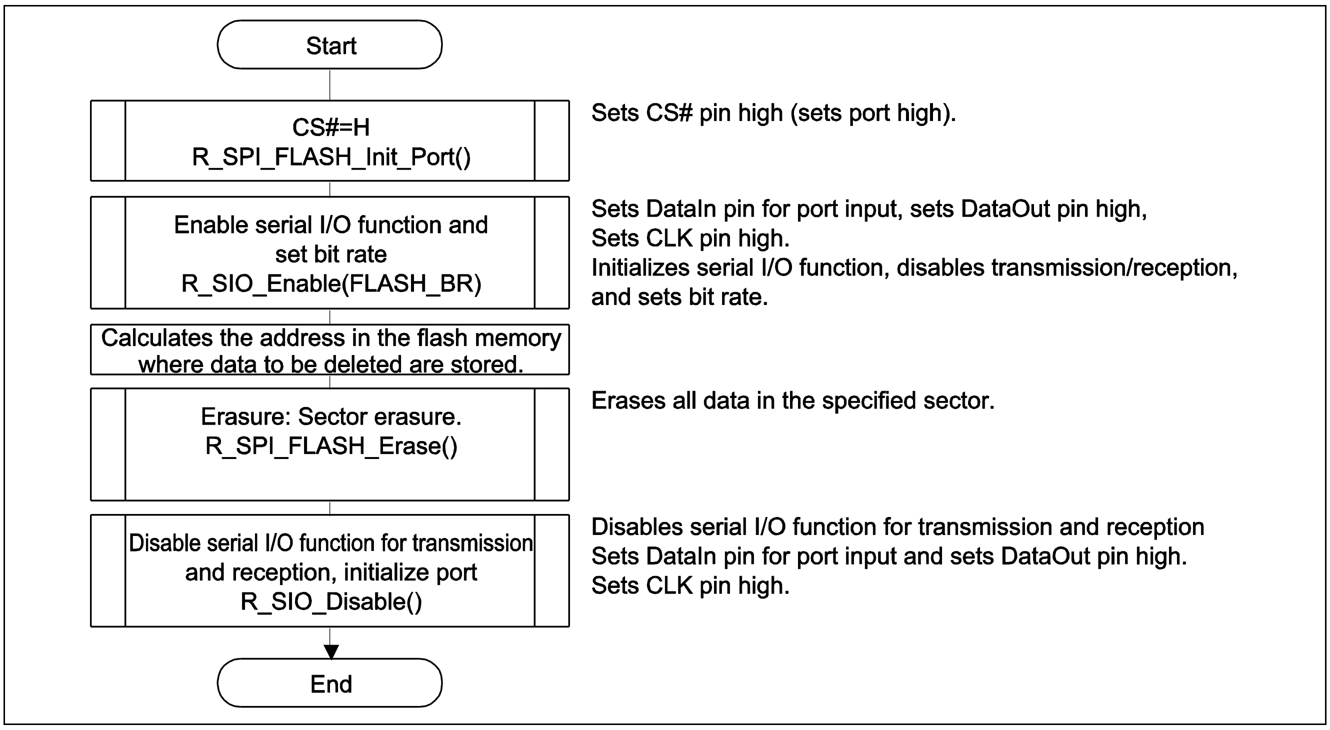

Start
CS#=H
R_SPI_FLASH_Init_Port()
Sets CS# pin high (sets port high).
Enable serial I/O function and set bit rate
R_SIO_Enable(FLASH_BR)
Sets DataIn pin for port input, sets DataOut pin high,
Sets CLK pin high.
Initializes serial I/O function, disables transmission/reception, and sets bit rate.
Calculates the address in the flash memory where data to be deleted are stored.
Erasure: Sector erasure.
R_SPI_FLASH_Erase()
Erases all data in the specified sector.
Disable serial I/O function for transmission and reception, initialize port
R_SIO_Disable()
Disables serial I/O function for transmission and reception
Sets DataIn pin for port input and sets DataOut pin high.
Sets CLK pin high.
End

Figure 14 Sector Erasure Processing Outline

5.9.8 Deep Power-down Processing

R_SPI_FLASH_DeepPDown

Synopsis	Sets deep power-down mode.		
Headers	R_SPI_FLASH_m25p.h		
Declaration	error_t R_SPI_FLASH_DeepPDown (uint8_t DevNo)		
Explanation	• Sets deep power-down mode.		
Arguments	uint8_t DevNo	;	Device number
Return value	• Returns the result of the processing.		
	FLASH_OK	;	Successful operation
	FLASH_ERR_PARAM	;	Parameter error
	FLASH_ERR_HARD	;	Hardware error
	FLASH_ERR_OTHER	;	Other error
Remarks	• This function is not supported (dependent on the device in use). Refer to the specifications of the flash memory in use.		

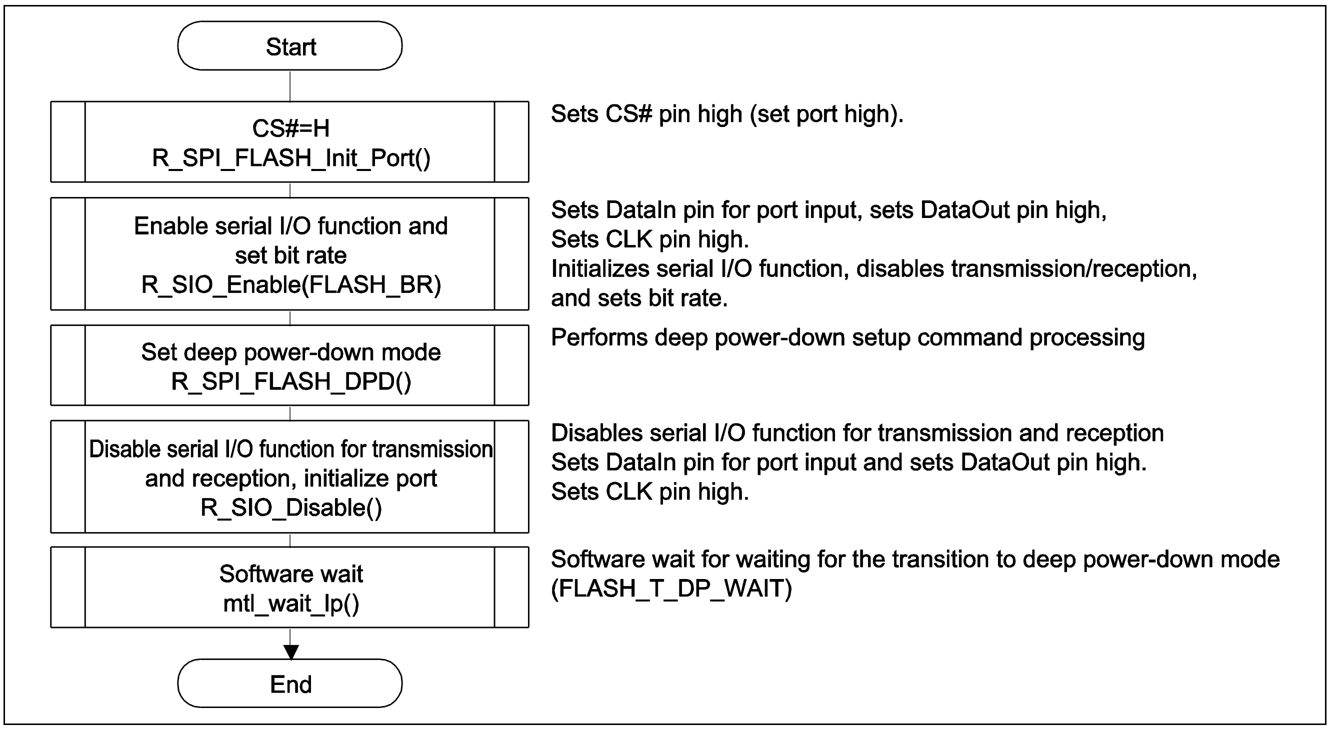

Start
CS#=H
R_SPI_FLASH_Init_Port()
Sets CS# pin high (set port high).
Enable serial I/O function and set bit rate
R_SIO_Enable(FLASH_BR)
Sets DataIn pin for port input, sets DataOut pin high,
Sets CLK pin high.
Initializes serial I/O function, disables transmission/reception, and sets bit rate.
Set deep power-down mode
R_SPI_FLASH_DPD()
Performs deep power-down setup command processing
Disable serial I/O function for transmission and reception, initialize port
R_SIO_Disable()
Disables serial I/O function for transmission and reception
Sets DataIn pin for port input and sets DataOut pin high.
Sets CLK pin high.
Software wait
mtl_wait_lp()
Software wait for waiting for the transition to deep power-down mode (FLASH_T_DP_WAIT)
End

Figure 15 Deep Power-down Processing Outline

5.9.9 Deep Power-down Release Processing

R_SPI_FLASH_ReleaseDeepPDown

Synopsis	Releases deep power-down mode
Headers	R_SPI_FLASH_m25p.h
Declaration	error_t R_SPI_FLASH_ReleaseDeepPDown (uint8_t DevNo, uint8_t FAR* pData)
Explanation	• Releases deep power-down mode and stores the signature in pData. Set up a 1-byte read buffer.
Arguments	uint8_t DevNo ; Device number uint8_t FAR* pData ; Pointer to read data buffer
Return value	• Returns the result of the processing. FLASH_OK ; Successful operation FLASH_ERR_PARAM ; Parameter error FLASH_ERR_HARD ; Hardware error FLASH_ERR_OTHER ; Other error
Remarks	• This function is not supported (dependent on the device in use). Refer to the specifications of the flash memory in use.

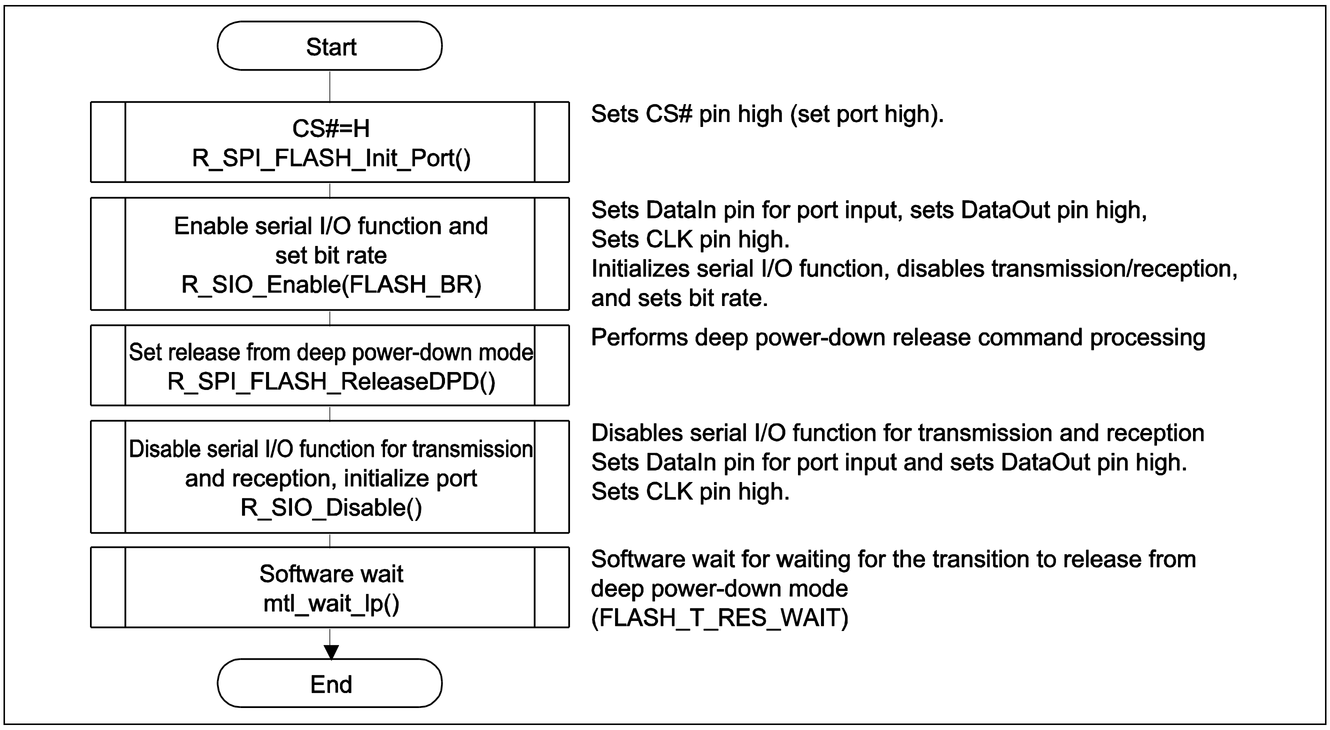

Start
CS#=H
R_SPI_FLASH_Init_Port()
Sets CS# pin high (set port high).
Enable serial I/O function and set bit rate
R_SIO_Enable(FLASH_BR)
Sets DataIn pin for port input, sets DataOut pin high, Sets CLK pin high. Initializes serial I/O function, disables transmission/reception, and sets bit rate.
Set release from deep power-down mode
R_SPI_FLASH_ReleaseDPD()
Performs deep power-down release command processing
Disable serial I/O function for transmission and reception, initialize port
R_SIO_Disable()
Disables serial I/O function for transmission and reception Sets DataIn pin for port input and sets DataOut pin high. Sets CLK pin high.
Software wait
mtl_wait_lp()
Software wait for waiting for the transition to release from deep power-down mode (FLASH_T_RES_WAIT)
End

Figure 16 Deep Power-down Release Processing Outline

5.9.10 ID Read Processing

R_SPI_FLASH_ReadID

Synopsis	Reads ID.
Headers	R_SPI_FLASH_m25p.h
Declaration	error_t R_SPI_FLASH_ReadID(uint8_t DevNo, uint8_t FAR* pData)
Explanation	• The manufacturer ID and device ID are read out and stored in pData. Set up three bytes as a buffer for use in reading.
Arguments	uint8_t DevNo ; Device number uint8_t FAR* pData ; Pointer to read data buffer
Return value	• Returns the result of the ID read. FLASH_OK ; Successful operation FLASH_ERR_PARAM ; Parameter error FLASH_ERR_HARD ; Hardware error FLASH_ERR_OTHER ; Other error
Remarks	• This function is not supported (dependent on the device in use). Refer to the specifications of the flash memory in use.

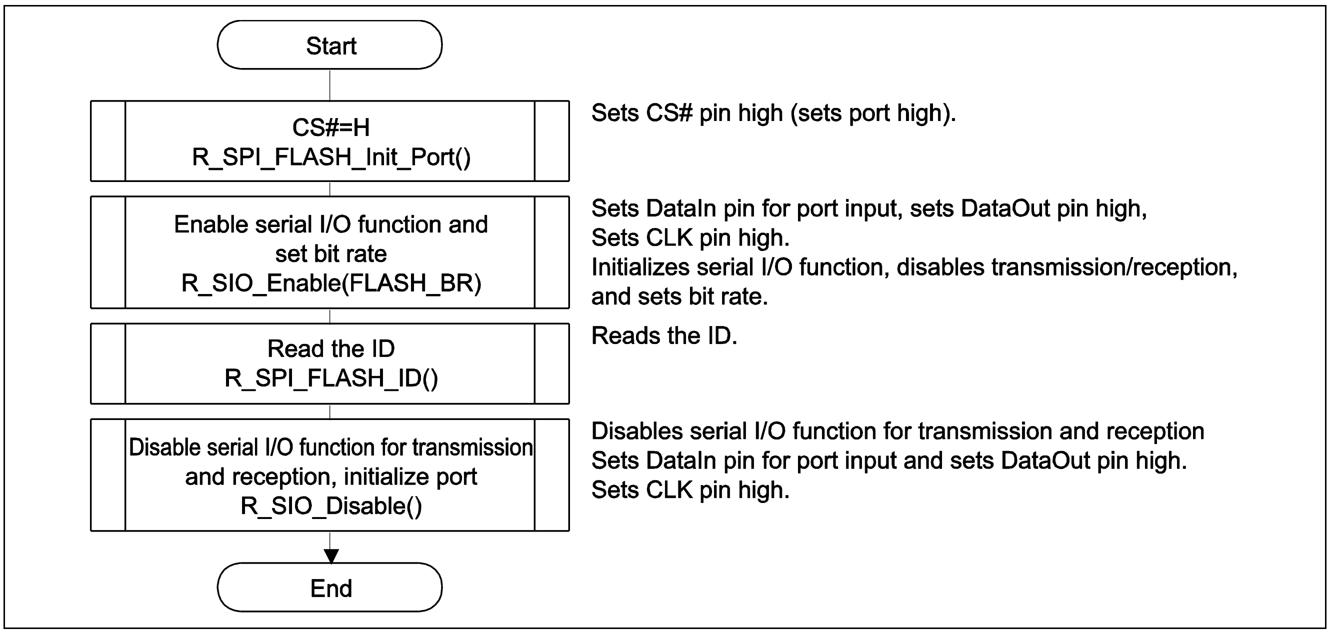

Start
CS#=H
R_SPI_FLASH_Init_Port()
Sets CS# pin high (sets port high).
Enable serial I/O function and set bit rate
R_SIO_Enable(FLASH_BR)
Sets DataIn pin for port input, sets DataOut pin high,
Sets CLK pin high.
Initializes serial I/O function, disables transmission/reception, and sets bit rate.
Read the ID
R_SPI_FLASH_ID()
Reads the ID.
Disable serial I/O function for transmission and reception, initialize port
R_SIO_Disable()
Disables serial I/O function for transmission and reception
Sets DataIn pin for port input and sets DataOut pin high.
Sets CLK pin high.
End

Figure 17 ID Read Processing Outline

5.9.11 Port Initialization Processing (Internal Function)

R_SPI_FLASH_Init_Port

Synopsis	Initializes ports (internal function).
Headers	R_SPI_FLASH_m25p_io.h
Declaration	void R_SPI_FLASH_Init_Port(uint8_t DevNo)
Explanation	• Initializes the port (CS#) of the specified device and sets it high.
Arguments	uint8_t DevNo ; Device number
Return value	void
Remarks	None

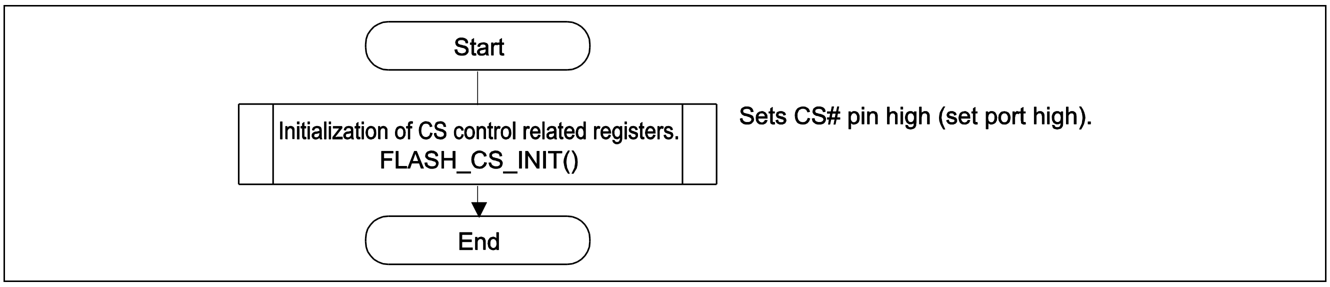

Start
Initialization of CS control related registers.
FLASH_CS_INIT()
End
Sets CS# pin high (set port high).

Figure 18 Port Initialization Processing Outline

5.9.12 Command Send Processing (Internal Function)

R_SPI_FLASH_Send_Cmd

Synopsis	Sends command (internal function).
Headers	R_SPI_FLASH_m25p_io.h
Declaration	STATIC error_t R_SPI_FLASH_Send_Cmd(uint8_t Cmd, uint32_t Addr, uint8_t CmdSize)
Explanation	• Sends the specified command.

Arguments

uint8_t	Cmd	;	Command code
uint32_t	Addr	;	Address
uint8_t	CmdSize	;	Command size

Return value

• Returns the result of the command send.

FLASH_OK	;	Successful operation
FLASH_ERR_HARD	;	Hardware error
FLASH_ERR_OTHER	;	Other error

Remarks None

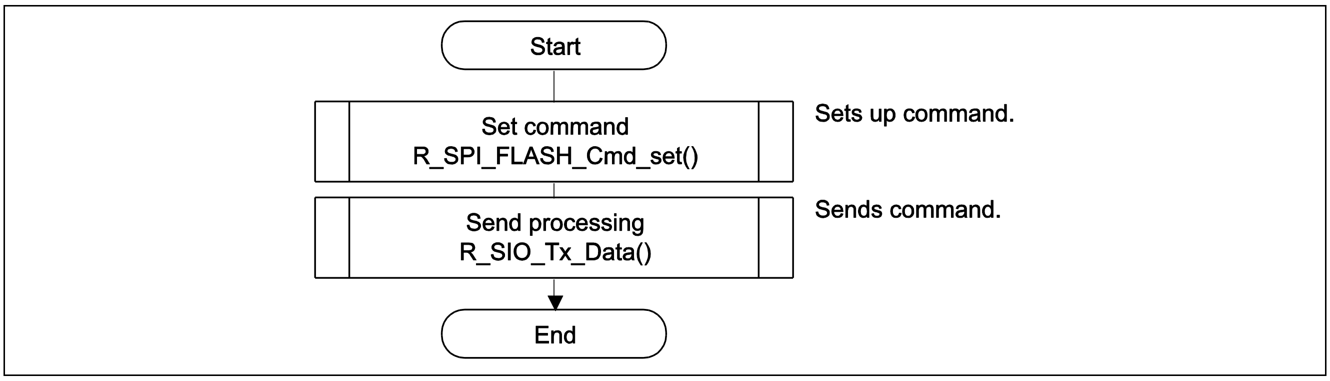


Figure 19 Command Send Processing Outline

5.9.13 Write Enable Command Processing (Internal Function)

R_SPI_FLASH_Write_En

Synopsis	Performs Write Enable command processing (internal function).
Headers	R_SPI_FLASH_m25p_io.h
Declaration	STATIC error_t R_SPI_FLASH_Write_En(uint8_t DevNo)
Explanation	• Sends a WREN command to enable the specified device for writes (setting the WEL bit).
Arguments	uint8_t DevNo ; Device number
Return value	• Returns the result of command processing.
	FLASH_OK ; Successful operation
	FLASH_ERR_HARD ; Hardware error
	FLASH_ERR_OTHER ; Other error
Remarks	None

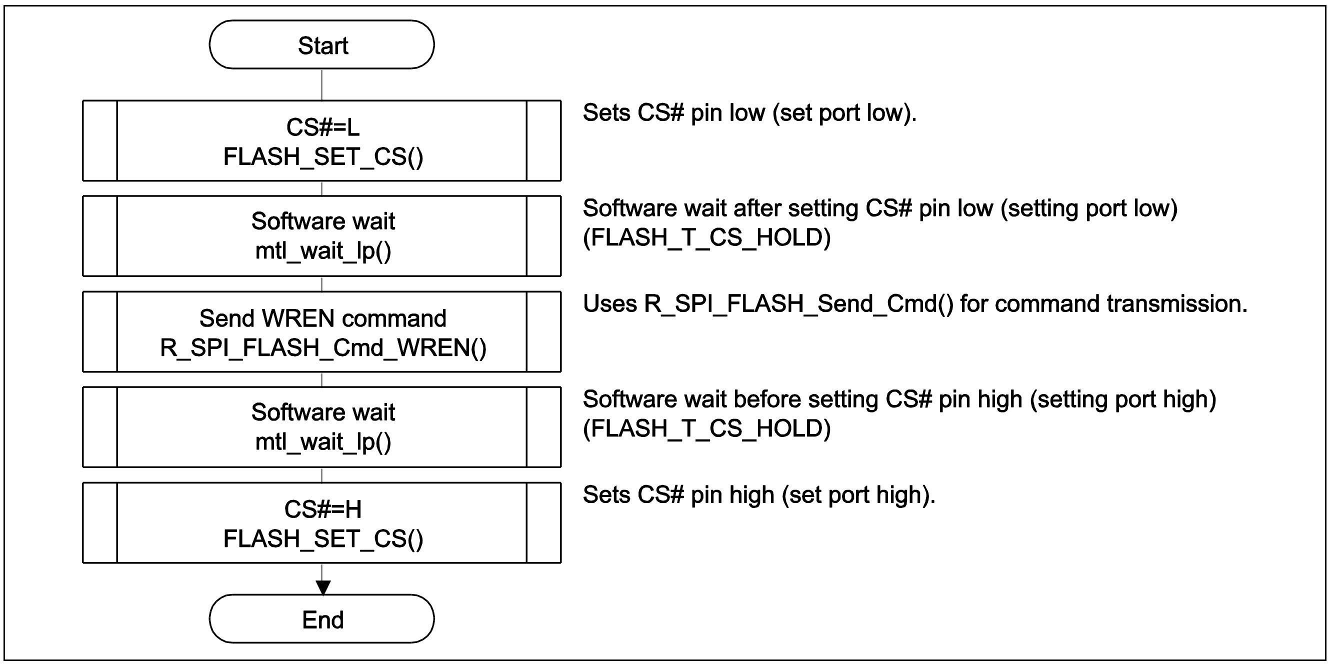

Start
CS#=L
FLASH_SET_CS()
Sets CS# pin low (set port low).
Software wait
mtl_wait_lp()
Software wait after setting CS# pin low (setting port low)
(FLASH_T_CS_HOLD)
Send WREN command
R_SPI_FLASH_Cmd_WREN()
Uses R_SPI_FLASH_Send_Cmd() for command transmission.
Software wait
mtl_wait_lp()
Software wait before setting CS# pin high (setting port high)
(FLASH_T_CS_HOLD)
CS#=H
FLASH_SET_CS()
Sets CS# pin high (set port high).
End

Figure 20 Write Enable Command Processing Outline

5.9.14 Write Disable Command Processing (Internal Function)

R_SPI_FLASH_Write_Di

Synopsis	Performs Write Disable command processing (internal function).
Headers	R_SPI_FLASH_m25p_io.h
Declaration	STATIC error_t R_SPI_FLASH_Write_Di(uint8_t DevNo)
Explanation	• Sends a WRDI command to disable the specified device for writes (clearing the WEL bit).
Arguments	uint8_t DevNo ; Device number
Return value	• Returns the result of command processing. FLASH_OK ; Successful operation FLASH_ERR_HARD ; Hardware error FLASH_ERR_OTHER ; Other error
Remarks	None

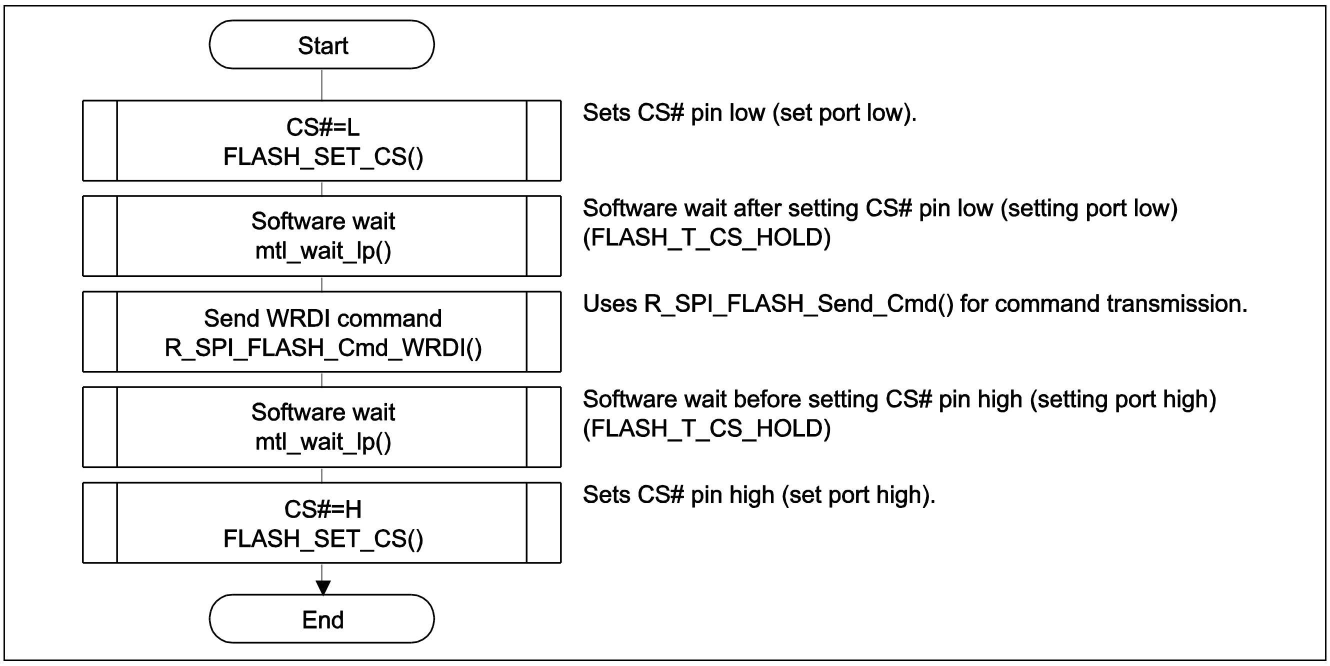

Start
CS#=L
FLASH_SET_CS()
Sets CS# pin low (set port low).
Software wait
mtl_wait_lp()
Software wait after setting CS# pin low (setting port low) (FLASH_T_CS_HOLD)
Send WRDI command
R_SPI_FLASH_Cmd_WRDI()
Uses R_SPI_FLASH_Send_Cmd() for command transmission.
Software wait
mtl_wait_lp()
Software wait before setting CS# pin high (setting port high) (FLASH_T_CS_HOLD)
CS#=H
FLASH_SET_CS()
Sets CS# pin high (set port high).
End

Figure 21 Write Disable Command Processing Outline

5.9.15 Read Status Register Command Processing (Internal Function)

R_SPI_FLASH_Read_StsReg

Synopsis	Performs Read Status Register command processing (internal function).
Headers	R_SPI_FLASH_m25p_io.h
Declaration	error_t R_SPI_FLASH_Read_StsReg(uint8_t DevNo, uint8_t FAR* pStsReg)
Explanation	• Sends an RDSR command to read the data from the status register into pStsReg. Set up a 1-byte read buffer. • The read status buffer (pStatus) is loaded with the following information: Bit 7:SRWD 1: BP2, BP1, BP0 are read-only bits 0: BP2, BP1, BP0 are read/writable Bit 6 to 5:Reserved (All "0") Bits 4 to 2:BP2, BP1, BP0 Bit 1:WEL 1: Internal Write Enable Latch is set 0: Internal Write Enable Latch is reset Bit 0:WIP 1: Program or Erase cycle is in progress 0: No Program or Erase cycle is in progress
Arguments	uint8_t DevNo ; Device number uint8_t FAR* pStsReg ; Pointer to buffer for storing read status
Return value	• Returns the result of command processing. FLASH_OK ; Successful operation FLASH_ERR_HARD ; Hardware error FLASH_ERR_OTHER ; Other error
Remarks	• See the specification for the flash memory in use for the relation between protected areas and protection bits. There is a possibility of the BP bit not being allocated.

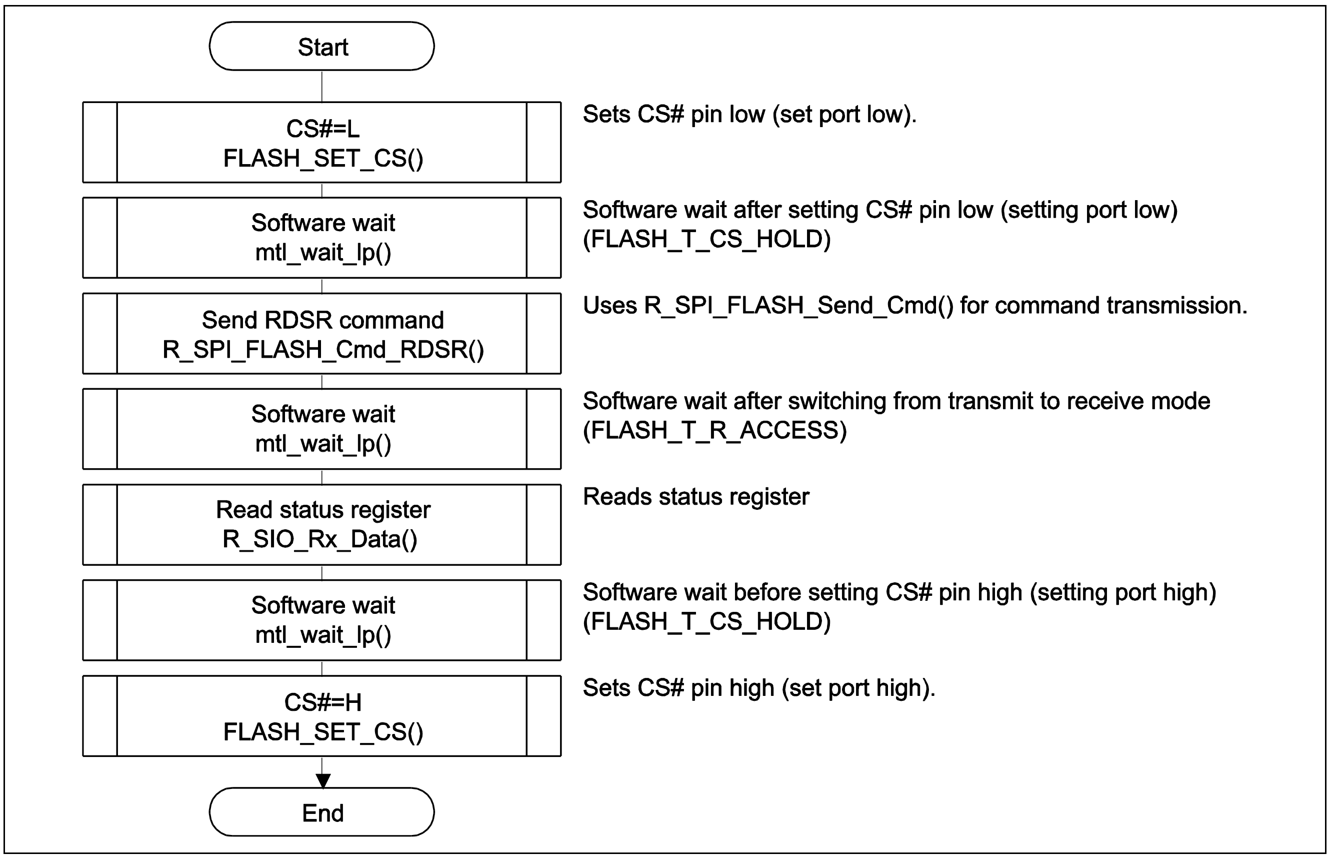

Start
CS#=L
FLASH_SET_CS()
Sets CS# pin low (set port low).
Software wait
mtl_wait_lp()
Software wait after setting CS# pin low (setting port low) (FLASH_T_CS_HOLD)
Send RDSR command
R_SPI_FLASH_Cmd_RDSR()
Uses R_SPI_FLASH_Send_Cmd() for command transmission.
Software wait
mtl_wait_lp()
Software wait after switching from transmit to receive mode (FLASH_T_R_ACCESS)
Read status register
R_SIO_Rx_Data()
Reads status register
Software wait
mtl_wait_lp()
Software wait before setting CS# pin high (setting port high) (FLASH_T_CS_HOLD)
CS#=H
FLASH_SET_CS()
Sets CS# pin high (set port high).
End

Figure 22 Read Status Register Command Processing Outline

5.9.16 Write Status Register Command Processing (Internal Function)

R_SPI_FLASH_Write_StsReg

Synopsis Performs Write Status Register command processing (internal function).

Headers R_SPI_FLASH_m25p_io.h

Declaration error_t R_SPI_FLASH_Write_StsReg(uint8_t DevNo, uint8_t FAR* pStsReg)

Explanation

- Performs EWSR command processing and sends a WRSR command to writes data from pStsReg into the status register. Set up a 1-byte write buffer.
- The write status buffer (pStatus) must be loaded with the following information:

Bit 7: SRWD	1: BP2, BP1, BP0 are read-only bits
	0: BP2, BP1, BP0 are read/writable
Bits 6 to 5: Reserved (All "0")	
Bits 4 to 2: BP2, BP1, BP0	
Bits 1 to 0: Read-only (All "0")	

Arguments

uint8_t	DevNo	;	Device number
uint8_t FAR*	pStsReg	;	Pointer to buffer for storing status data

Return value

- Returns the result of command processing.

FLASH_OK	;	Successful operation
FLASH_ERR_HARD	;	Hardware error
FLASH_ERR_OTHER	;	Other error

Remarks

- Writing to bits other than SRWD, BP2, BP1, or BP0 is ignored.
- See the specification for the flash memory in use for the relation between protected areas and protection bits. There is a possibility of the BP bit not being allocated.

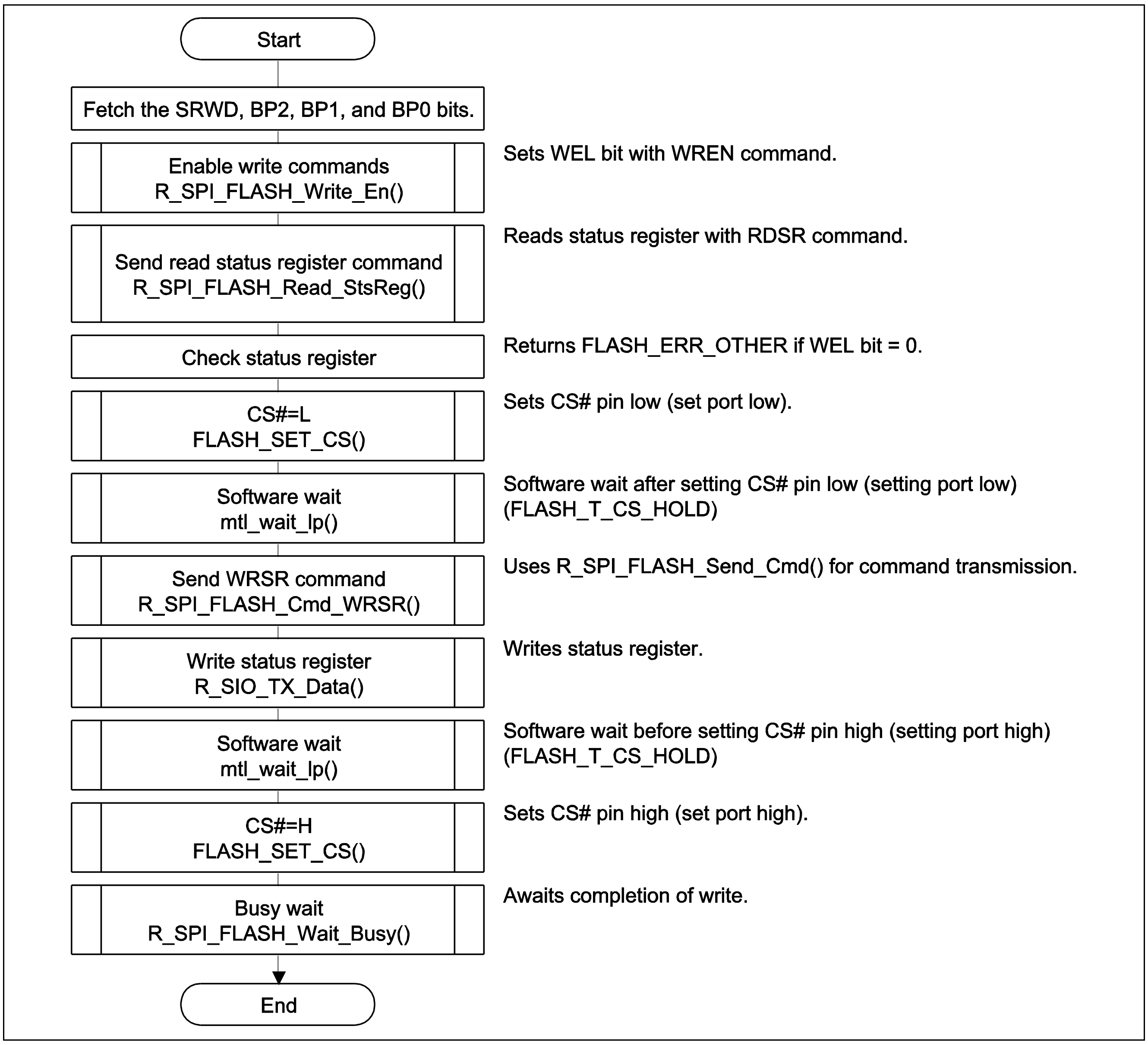

Start
Fetch the SRWD, BP2, BP1, and BP0 bits.
Enable write commands
R_SPI_FLASH_Write_En()
Sets WEL bit with WREN command.
Send read status register command
R_SPI_FLASH_Read_StsReg()
Reads status register with RDSR command.
Check status register
Returns FLASH_ERR_OTHER if WEL bit = 0.
CS#=L
FLASH_SET_CS()
Sets CS# pin low (set port low).
Software wait
mtl_wait_lp()
Software wait after setting CS# pin low (setting port low) (FLASH_T_CS_HOLD)
Send WRSR command
R_SPI_FLASH_Cmd_WRSR()
Uses R_SPI_FLASH_Send_Cmd() for command transmission.
Write status register
R_SIO_TX_Data()
Writes status register.
Software wait
mtl_wait_lp()
Software wait before setting CS# pin high (setting port high) (FLASH_T_CS_HOLD)
CS#=H
FLASH_SET_CS()
Sets CS# pin high (set port high).
Busy wait
R_SPI_FLASH_Wait_Busy()
Awaits completion of write.
End

Figure 23 Write Status Register Command Processing Outline

5.9.17 Busy Wait Processing (Internal Function)

R_SPI_FLASH_Wait_Busy

Synopsis	Performs busy wait processing (internal function).
Headers	R_SPI_FLASH_m25p_io.h
Declaration	STATIC error_t R_SPI_FLASH_Wait_Busy(uint8_t DevNo, uint16_t BusyTime, uint16_t BusyCnt)
Explanation	• Waits for the busy period at the BusyTime intervals if BusyCnt is found to be 0 with the Read Status Register command. • Waits for the busy period at the BusyTime intervals the number of times equal to BusyCnt if BusyCnt is found to be nonzero with the Read Status Register command.

Arguments

uint8_t	DevNo	;	Device number
uint16_t	BusyTime	;	Wait time for status check
uint16_t	BusyCnt	;	Counter

Return value

• Returns the result of wait processing.

FLASH_OK	;	Successful operation
FLASH_ERR_HARD	;	Hardware error
FLASH_ERR_OTHER	;	Other error

Remarks None

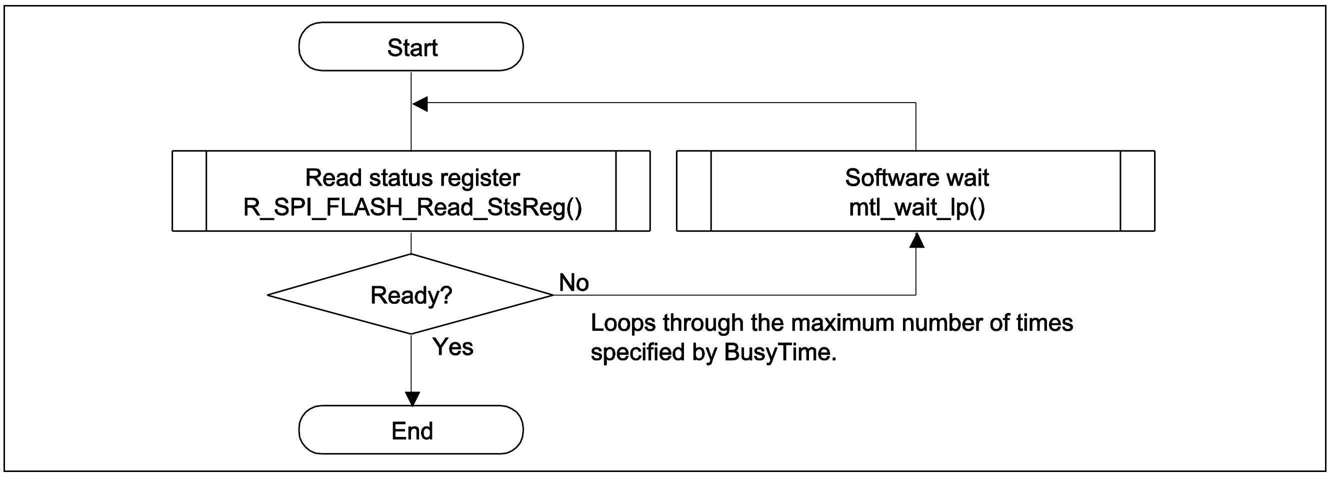

Start
Read status register
R_SPI_FLASH_Read_StsReg()
Software wait
mtl_wait_lp()
Ready?
No
Yes
Loops through the maximum number of times specified by BusyTime.
End

Figure 24 Busy Wait Processing Outline

5.9.18 Page Program Command Processing (Internal Function)

R_SPI_FLASH_Write_Page

Synopsis Performs Page Program Command processing (internal function).

Headers R_SPI_FLASH_m25p_io.h

Declaration STATIC error_t R_SPI_FLASH_Wait_Busy(uint8_t DevNo, uint16_t BusyTime, uint16_t BusyCnt)

Explanation
- Writes the specified number of bytes from pData into flash memory using the PP command starting at the specified address.

Arguments

uint8_t	DevNo	;	Device number
uint32_t	WAddr	;	Write start address
uint32_t	WCnt	;	Number of bytes to write
uint8_t FAR*	pData	;	Pointer to write data buffer

Return value
- Returns the result of the write processing.

FLASH_OK	;	Successful operation
FLASH_ERR_HARD	;	Hardware error
FLASH_ERR_OTHER	;	Other error

Remarks
- Writes beyond a page are not permitted.
- The function cannot write any protected page. It would then return no error.

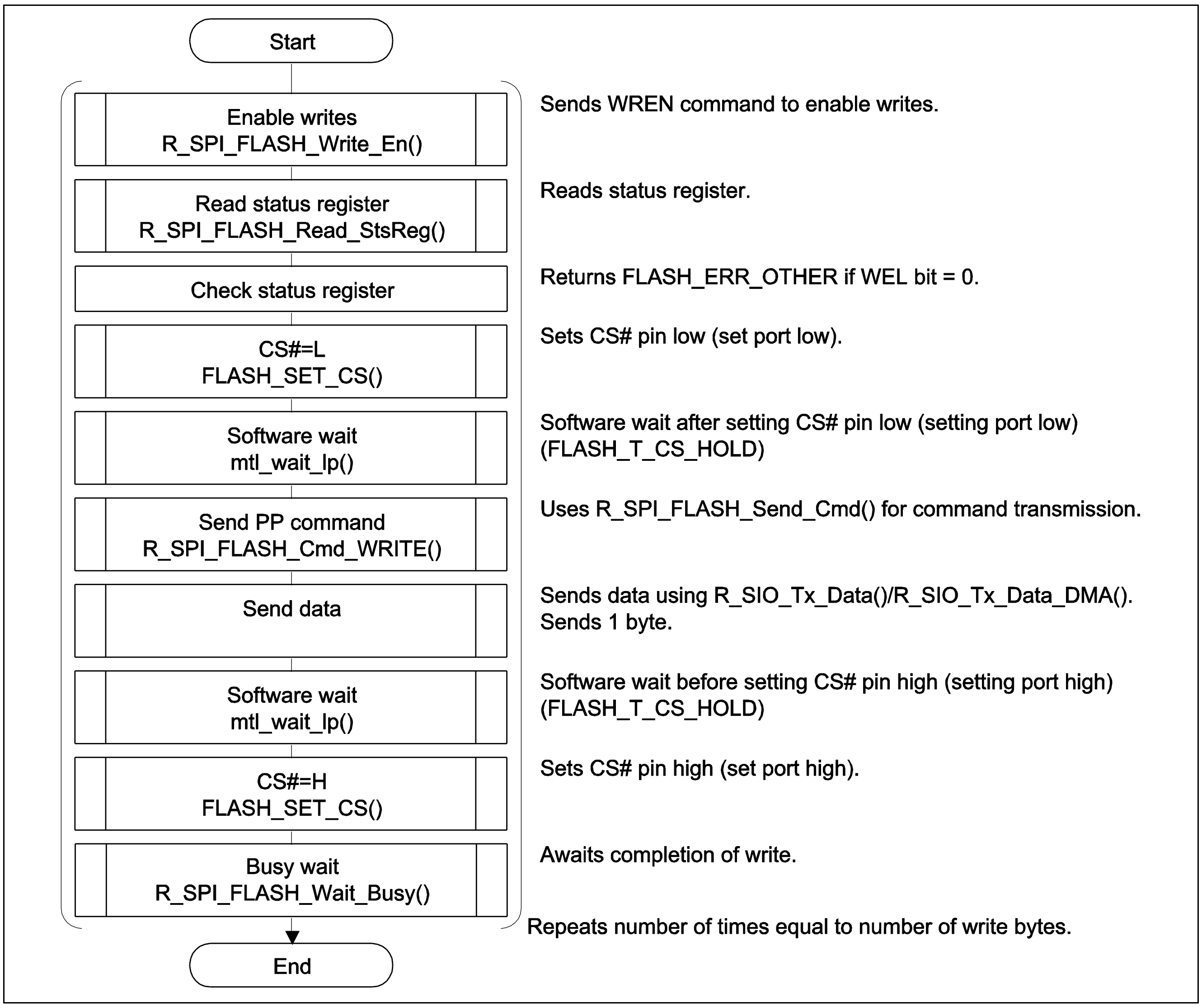

Start
Enable writes
R_SPI_FLASH_Write_En()
Sends WREN command to enable writes.
Read status register
R_SPI_FLASH_Read_StsReg()
Reads status register.
Check status register
Returns FLASH_ERR_OTHER if WEL bit = 0.
CS#=L
FLASH_SET_CS()
Sets CS# pin low (set port low).
Software wait
mtl_wait_lp()
Software wait after setting CS# pin low (setting port low) (FLASH_T_CS_HOLD)
Send PP command
R_SPI_FLASH_Cmd_WRITE()
Uses R_SPI_FLASH_Send_Cmd() for command transmission.
Send data
Sends data using R_SIO_Tx_Data()/R_SIO_Tx_Data_DMA(). Sends 1 byte.
Software wait
mtl_wait_lp()
Software wait before setting CS# pin high (setting port high) (FLASH_T_CS_HOLD)
CS#=H
FLASH_SET_CS()
Sets CS# pin high (set port high).
Busy wait
R_SPI_FLASH_Wait_Busy()
Awaits completion of write.
Repeats number of times equal to number of write bytes.
End

Figure 25   Page Program Command Processing Outline

5.9.19 Read Command Processing (Internal Function)

R_SPI_FLASH_Read_Memory

Synopsis	Performs Read command processing (internal function).
Headers	R_SPI_FLASH_m25p_io.h
Declaration	error_t R_SPI_FLASH_Read_Memory(uint8_t DevNo, uint32_t RAddr, uint32_t RCnt, uint8_t FAR* pData)
Explanation	• Reads the specified number of bytes from flash memory starting at the specified address, 1 byte at a time, using the READ command and stores the read data in pData.

Arguments

uint8_t	DevNo	;	Device number
uint32_t	RAddr	;	Read start address
uint32_t	RCnt	;	Number of read bytes
uint8_t FAR*	pData	;	Pointer to read data buffer

Return value

• Returns the result of command processing.

FLASH_OK	;	Successful operation
FLASH_ERR_HARD	;	Hardware error
FLASH_ERR_OTHER	;	Other error

Remarks

• The highest read address is flash memory size -1.

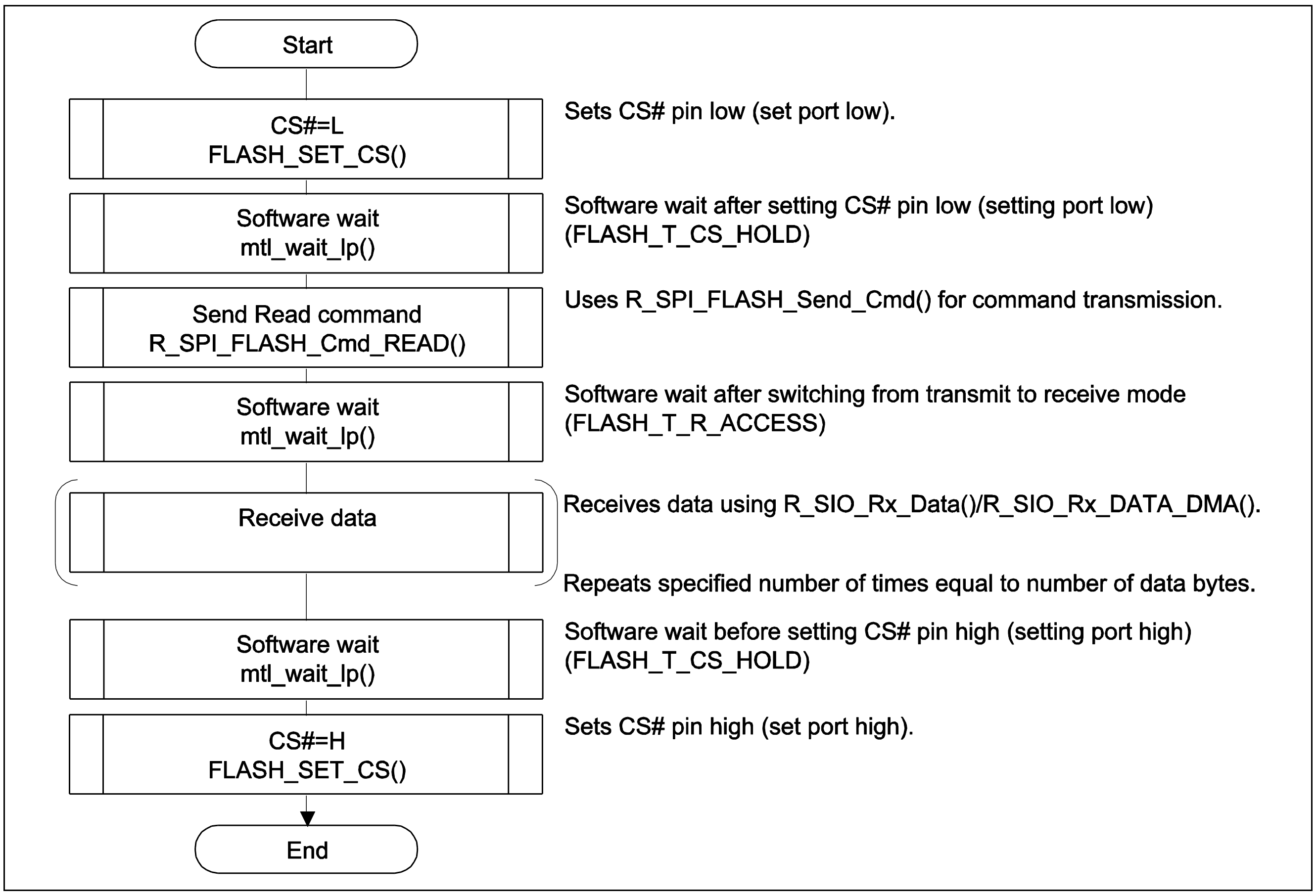

Start
CS#=L
FLASH_SET_CS()
Sets CS# pin low (set port low).
Software wait
mtl_wait_lp()
Software wait after setting CS# pin low (setting port low) (FLASH_T_CS_HOLD)
Send Read command
R_SPI_FLASH_Cmd_READ()
Uses R_SPI_FLASH_Send_Cmd() for command transmission.
Software wait
mtl_wait_lp()
Software wait after switching from transmit to receive mode (FLASH_T_R_ACCESS)
Receive data
Receives data using R_SIO_Rx_Data()/R_SIO_Rx_DATA_DMA().
Repeats specified number of times equal to number of data bytes.
Software wait
mtl_wait_lp()
Software wait before setting CS# pin high (setting port high) (FLASH_T_CS_HOLD)
CS#=H
FLASH_SET_CS()
Sets CS# pin high (set port high).
End

Figure 26 Read Command Processing Outline

5.9.20 Erase Command Processing (Internal Function)

R_SPI_FLASH_Erase

Synopsis	Performs Erase command processing (internal function)
Headers	R_SPI_FLASH_m25p_io.h
Declaration	error_t R_SPI_FLASH_Erase(uint8_t DevNo, uint32_t EAddr, uint8_t Etype)
Explanation	• Erases the memory array by a SE, or BE command. • Specify the following information for an erasure tyte (Etype). FLASH_B_ERASE: Bulk erasure FLASH_S_ERASE: Sector erasure

Arguments

uint8_t	DevNo	;	Device number
uint32_t	EAddr	;	Erasure address
uint32_t	Etype	;	Erasure type

Return value

• Returns the result of erase processing.

FLASH_OK	;	Successful operation
FLASH_ERR_HARD	;	Hardware error
FLASH_ERR_OTHER	;	Other error

Remarks

• Erasure of flash memory is only possible when write protection is released. Operation is as described below.
Bulk erasure: Erasure does not proceed and an error response is not returned.
Sector erasure: Write-protected areas are not erased. An error response is not returned even if such areas are encountered.

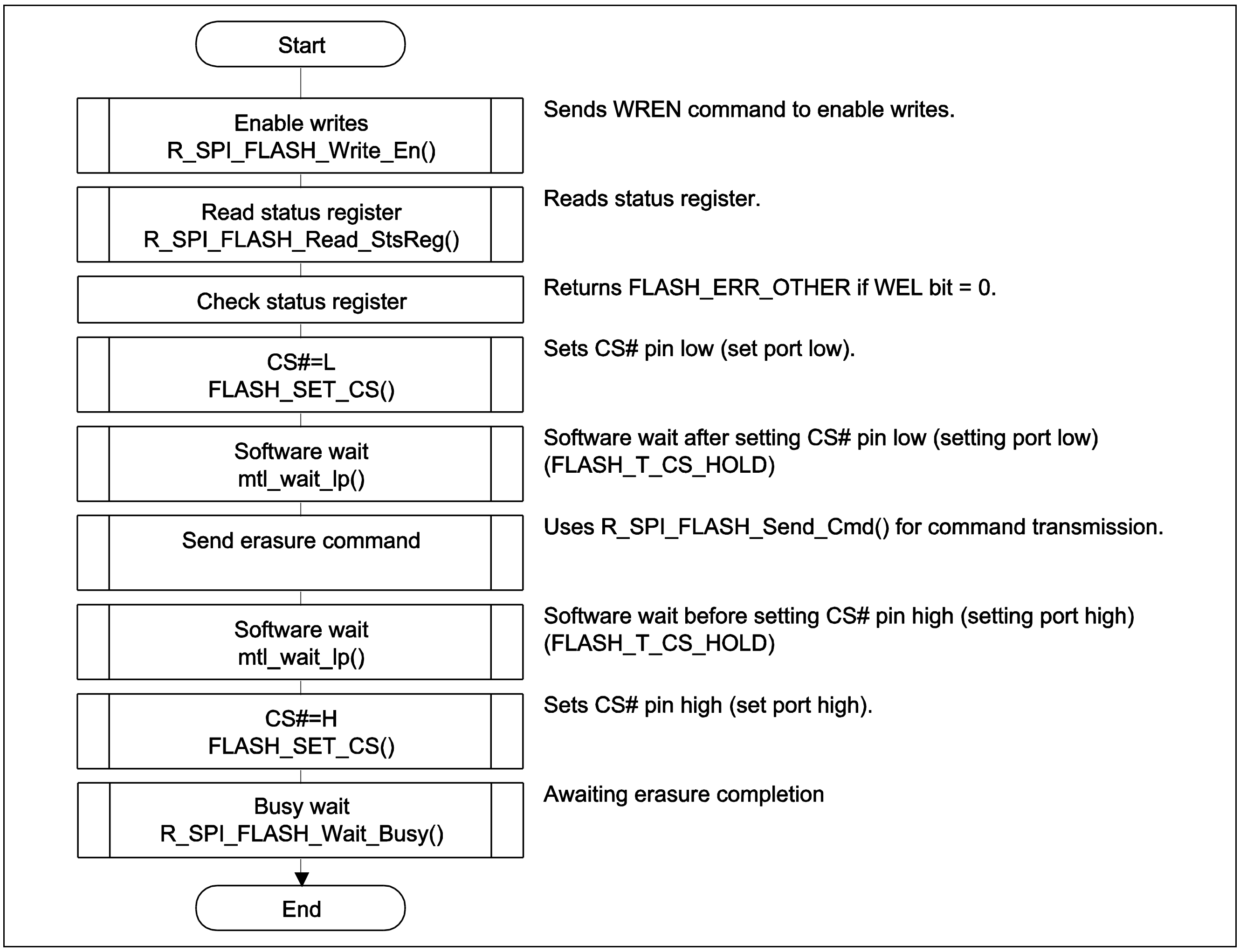

Start
Enable writes
R_SPI_FLASH_Write_En()
Sends WREN command to enable writes.
Read status register
R_SPI_FLASH_Read_StsReg()
Reads status register.
Check status register
Returns FLASH_ERR_OTHER if WEL bit = 0.
CS#=L
FLASH_SET_CS()
Sets CS# pin low (set port low).
Software wait
mtl_wait_lp()
Software wait after setting CS# pin low (setting port low) (FLASH_T_CS_HOLD)
Send erasure command
Uses R_SPI_FLASH_Send_Cmd() for command transmission.
Software wait
mtl_wait_lp()
Software wait before setting CS# pin high (setting port high) (FLASH_T_CS_HOLD)
CS#=H
FLASH_SET_CS()
Sets CS# pin high (set port high).
Busy wait
R_SPI_FLASH_Wait_Busy()
Awaiting erasure completion
End

Figure 27 Erase Command Processing Outline

5.9.21 Deep Power-down Command Processing (Internal Function)

R_SPI_FLASH_DPD

Synopsis	Performs Deep Power-down command processing (internal function).			
Headers	R_SPI_FLASH_m25p_io.h			
Declaration	error_t R_SPI_FLASH_DPD(uint8_t DevNo)			
Explanation	• Sets deep power-down by using a DP command.			
Arguments	uint8_t	DevNo	;	Device number
	uint8_t FAR*	pData	;	Pointer to read data buffer
Return value	• Returns the result of command processing.			
	FLASH_OK		;	Successful operation
	FLASH_ERR_HARD		;	Hardware error
	FLASH_ERR_OTHER		;	Other error
Remarks	None			

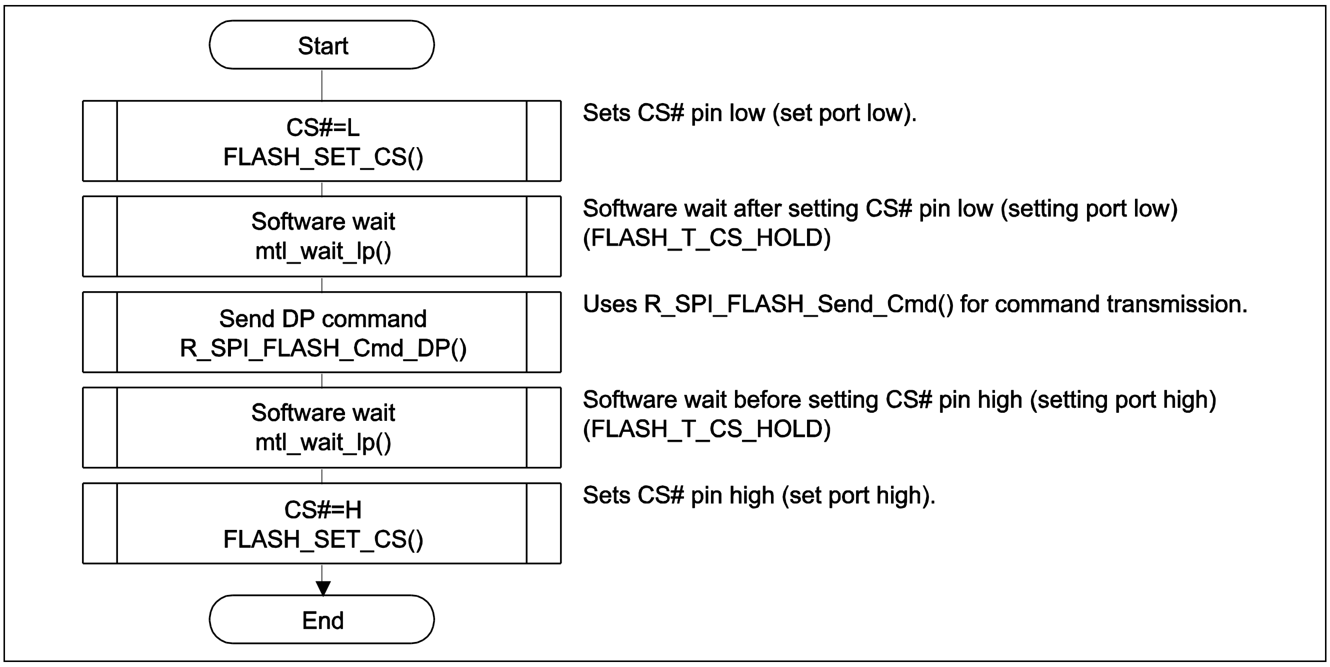

Start
CS#=L
FLASH_SET_CS()
Sets CS# pin low (set port low).
Software wait
mtl_wait_lp()
Software wait after setting CS# pin low (setting port low)
(FLASH_T_CS_HOLD)
Send DP command
R_SPI_FLASH_Cmd_DP()
Uses R_SPI_FLASH_Send_Cmd() for command transmission.
Software wait
mtl_wait_lp()
Software wait before setting CS# pin high (setting port high)
(FLASH_T_CS_HOLD)
CS#=H
FLASH_SET_CS()
Sets CS# pin high (set port high).
End

Figure 28 Deep Power-down Command Processing Outline

5.9.22 Deep Power-down Release Command Processing (Internal Function)

R_SPI_FLASH_ReleaseDPD

Synopsis	Performs Deep Power-down Release command processing (internal function).
Headers	R_SPI_FLASH_m25p_io.h
Declaration	error_t R_SPI_FLASH_ReleaseDPD(uint8_t DevNo, uint8_t FAR* pData)
Explanation	• Releases deep power-down by using a RES command, and then read out the signature and stores the result in pData. Set up a 1-byte read buffer.
Arguments	uint8_t DevNo ; Device number
	uint8_t FAR* pData ; Pointer to read data buffer
Return value	• Returns the result of command processing.
	FLASH_OK ; Successful operation
	FLASH_ERR_HARD ; Hardware error
	FLASH_ERR_OTHER ; Other error
Remarks	None

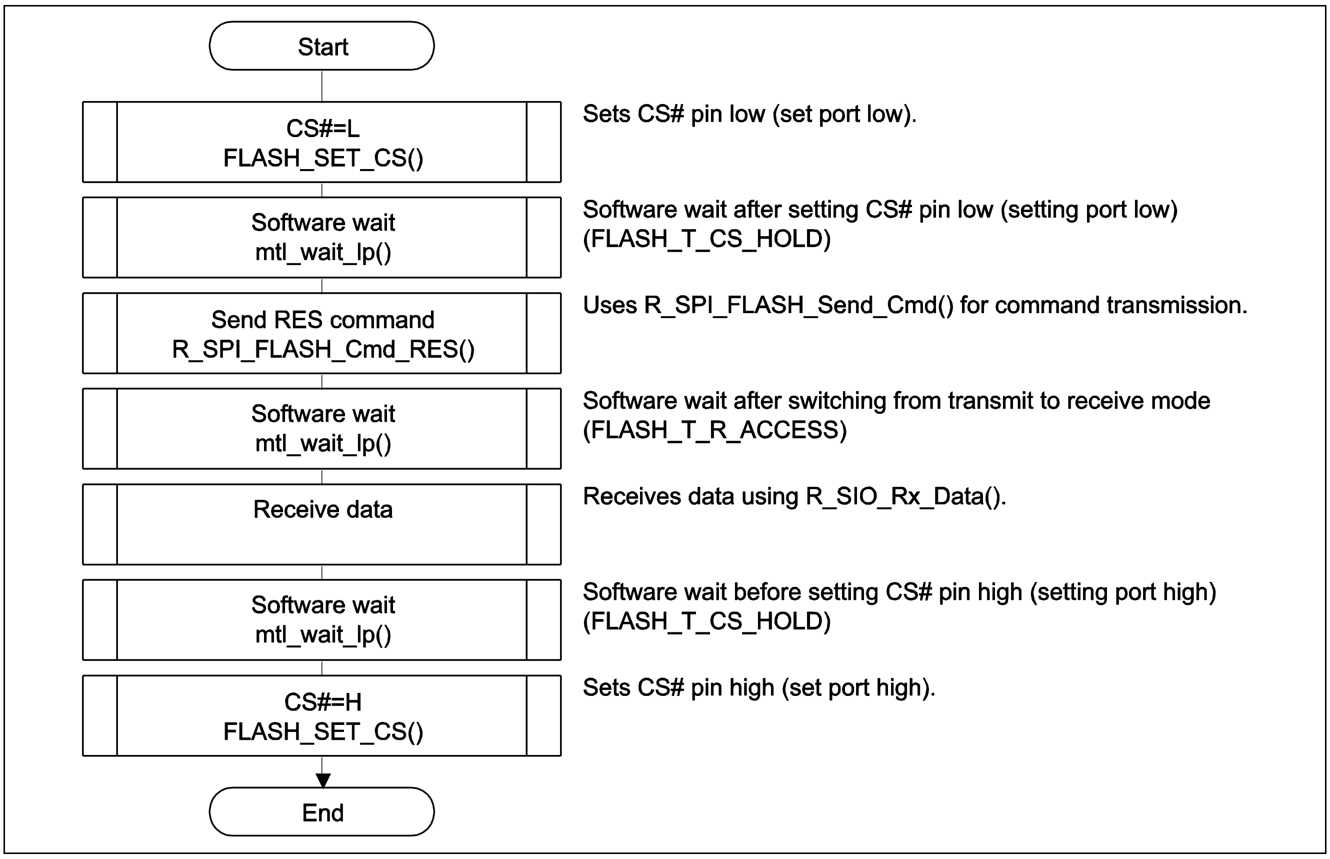

Start
CS#=L
FLASH_SET_CS()
Sets CS# pin low (set port low).
Software wait
mtl_wait_lp()
Software wait after setting CS# pin low (setting port low) (FLASH_T_CS_HOLD)
Send RES command
R_SPI_FLASH_Cmd_RES()
Uses R_SPI_FLASH_Send_Cmd() for command transmission.
Software wait
mtl_wait_lp()
Software wait after switching from transmit to receive mode (FLASH_T_R_ACCESS)
Receive data
Receives data using R_SIO_Rx_Data().
Software wait
mtl_wait_lp()
Software wait before setting CS# pin high (setting port high) (FLASH_T_CS_HOLD)
CS#=H
FLASH_SET_CS()
Sets CS# pin high (set port high).
End

Figure 29 Deep Power-down Release Command Processing Outline

5.9.23 Read ID Command Processing (Internal Function)

R_SPI_FLASH_ID

Synopsis	Performs Read ID command processing (internal function).
Headers	R_SPI_FLASH_m25p_io.h
Declaration	error_t R_SPI_FLASH_ID(uint8_t DevNo, uint8_t FAR* pData)
Explanation	• This function uses the RDID command to read out the manufacturer ID and device ID, and stores the result in pData. Set up three bytes as a buffer for use in reading.

Arguments

uint8_t	DevNo	;	Device number
uint8_t FAR*	pData	;	Pointer to read data buffer

Return value

• Returns the result of command processing.

FLASH_OK	;	Successful operation
FLASH_ERR_HARD	;	Hardware error
FLASH_ERR_OTHER	;	Other error

Remarks None

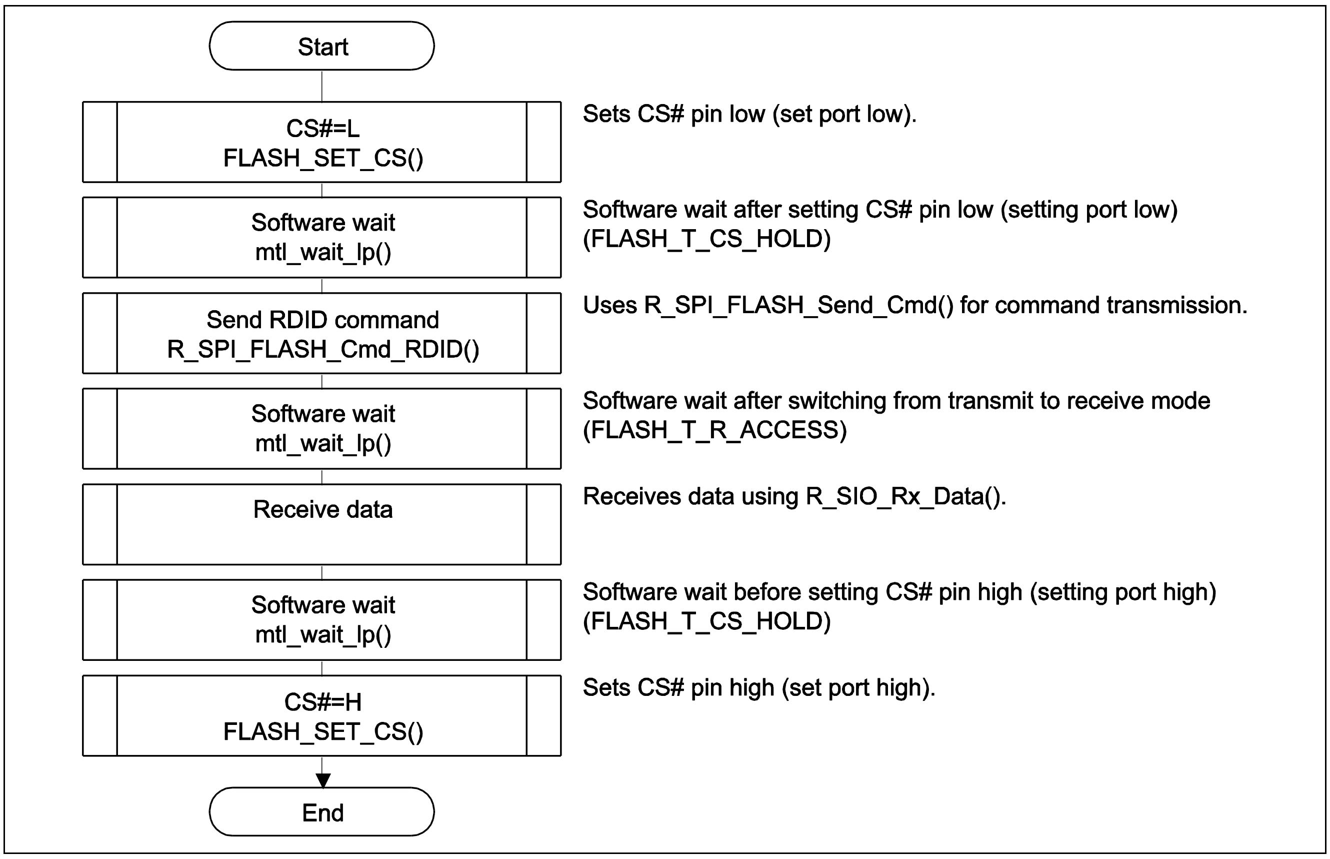

Start
CS#=L
FLASH_SET_CS()
Sets CS# pin low (set port low).
Software wait
mtl_wait_lp()
Software wait after setting CS# pin low (setting port low) (FLASH_T_CS_HOLD)
Send RDID command
R_SPI_FLASH_Cmd_RDID()
Uses R_SPI_FLASH_Send_Cmd() for command transmission.
Software wait
mtl_wait_lp()
Software wait after switching from transmit to receive mode (FLASH_T_R_ACCESS)
Receive data
Receives data using R_SIO_Rx_Data().
Software wait
mtl_wait_lp()
Software wait before setting CS# pin high (setting port high) (FLASH_T_CS_HOLD)
CS#=H
FLASH_SET_CS()
Sets CS# pin high (set port high).
End

Figure 30 Read ID Command Processing Outline

5.9.24 Command Setup Processing (Internal Function)

R_SPI_FLASH_Cmd_set

Synopsis	Sets up command (internal function).			
Headers	R_SPI_FLASH_m25p_io.h			
Declaration	STATIC void R_SPI_FLASH_Cmd_set(uint8_t Cmd, uint32_t Addr, uint8_t CmdSize)			
Explanation	• Sets a command and an address. Conversion is carried out according to the endian mode specified.			
Arguments	uint8_t	Cmd	;	Command (instruction code)
	uint32_t	Addr	;	Address information
	uint8_t	CmdSize	;	Command size
Return value	None			
Remarks	• The endian mode must be specified through MTL_MCU_LITTLE (mtl_com.h).			

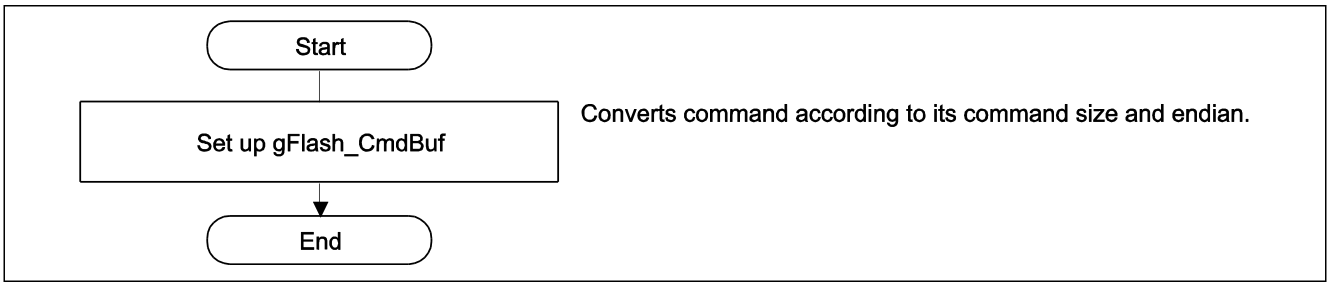


Figure 31 Command Setup Processing Outline

6. Application Example

This chapter gives an example of setting up the Serial Flash memory control section (the serial I/O control section is not covered).

For the serial I/O control section, refer to the application note for the individual MCU-specific clock synchronous single-master control software.

The baud rate is specified in this sample code because it is necessary to set it up according to the individual slave device.

The locations where settings are made are identified by the comments header "/** SET **/" in the defining file.

The common functions (e.g., mtl_wait_lp()) must be borrowed from the individual MCU-specific clock synchronous single-master control software.

6.1 Setting Up the Serial Flash memory Control Software

The settings to be made are identified by the comments header "/** SET **/" in the individual file.

6.1.1 R_SPI_FLASH_m25p.h

This is a definition file for this Serial Flash memory.

The settings to be made are identified by the comments header "/** SET **/" in the file.

1. Defining the number of devices to be used and their device number
   Specify the number of devices to be used and assign a device number to each of them.
   Given below is an example of using one device and assigning a device number of 0.
   A maximum of 2 devices can be controlled.

```
/*------------------------------------------------------------------------*/
/*  Define number of required serial FLASH devices.(1~N devices)           */
/*  Define the device number in accordance with the number of serial FLASH
devices           */
/*  to be connected.                                                      */
/*------------------------------------------------------------------------*/
/* Define no. of devices */
#define FLASH_DEV_NUM      1           /* 1devices             */

/* Define no. of slots */
#define FLASH_DEV0         0           /* Device 0             */
#define FLASH_DEV1         1           /* Device 1             */
```

2. Defining the size of the devices to be used
   Specify the size of the devices to be used
   Given below is an example of using 32 Mbit devices.

```
/*------------------------------------------------------------------------*/
/*  Define the serial FLASH device.                                        */
/*------------------------------------------------------------------------*/

//#define M25P05A                      /* 512kbit  ( 64kByte)  */
//#define M25P10A                      /* 1Mbit    ( 128kByte) */
//#define M25P20                       /* 2Mbit    ( 256kByte) */
//#define M25P40                       /* 4Mbit    ( 512kByte) */
//#define M25P16                       /* 16Mbit   ( 2MByte)   */
#define M25P32                         /* 32Mbit   ( 4MByte)   */
//#define M25P64                       /* 64Mbit   ( 8MByte)   */
```

6.1.2 R_SPI_FLASH_sfr.h

R_SPI_FLASH_sfr.h.XXX are made available for the purpose of evaluating the individual MCUs. Rename one of them to R_SPI_FLASH_sfr.h. If no header file is available that applies to the desired MCU, create your own R_SPI_FLASH_sfr.h while referring to these files.

The settings to be made are identified by the comments header "/** SET **/" in the file.

1. Read Command Type Setting
   Define the high-speed reading command (0x0B) if you intend to use it. Use the read command (0x03) if the high-speed command is not defined.
   The high-speed reading command (0x0B) is used in the example below.

```
/*----------------------------------------------------------------------------*/
/*   Select the read comand type.                                             */
/*   If you want to set the high speed read command(0x0B),
  define FLASH_0B_READ_SELECT        */
/*   Otherwise read command(0x03)is select.                                   */
/*----------------------------------------------------------------------------*/

#define FLASH_0B_READ_SELECT
```

2. Setting up the Chip Select signal
   Define the port to be used for the Chip Select signal.
   When connecting a second device, define the second port.
   Given below is a sample code for using port66.

```
/*----------------------------------------------------------------------------*/
/*  Define the CS port.                                                    */
/*----------------------------------------------------------------------------*/
#define FLASH_DR_CS0     PORT6.DR.BIT.B6/* FLASH CS0(Negative-true logic)    */
#define FLASH_DDR_CS0    PORT6.DDR.BIT.B6  /* FLASH CS0(Negative-true logic)  */

#if (FLASH_DEV_NUM > 1)
#define FLASH_DR_CS1                       /* FLASH CS1(Negative-true logic)  */
#define FLASH_DDR_CS1                      /* FLASH CS1(Negative-true logic)  */
#endif   /* #if (FLASH_DEV_NUM > 1) */
```

3. Setting the baud rate
   Set the baud rate in bits per second (bps).
   The value to be set is dependent on the type of MCU and serial I/O to be used.
   Given below is a sample code for the RX610's SCI, 3.125 Mbps (bits/second).

```
/*----------------------------------------------------------------------------*/
/*  Define the value of the bit rate register according to a communication
baud rate.        */
/*  The possible maximum transfer frequency of CLK is depends on hardware
circuit              */
/*  and MCU conditions.                                                       */
/*   Refer to MCU hardware manual/memory card specifications and specify the
buad rate.        */

  /* PCLK = 50MHz, n=0 for RX610 SCI */
#define FLASH_BR     (uint8_t)0x03           /* BRR initial setting        */
                         /*        ++----------------- 3.125MHz              */
```

Refer to the individual MCU hardware manual for the legitimate values.

6.1.3 R_SPI_FLASH_m25p_io.h

The settings to be made are identified by the comments header "/** SET **/" in the file.

1. Setting the value for time-out of erasure
   Since times taken by erasure differ according to the device, reconsider the values used below as required.
   The example below is written with 40 s as the period for time-out of erasure.

```
/*------------------------------------------------------------------------*/
/*   Define the software timer value of erase busy waiting.               */
/*   If you want to wait till the flash comes to ready status without time out,
  */
/*   comment the definition FLASH_EBUSY_WAIT_TIME.                        */
/*------------------------------------------------------------------------*/

#define FLASH_EBUSY_WAIT   (uint16_t)40000/* Erase busy waiting time  40000*
1ms =   40s */
```

2. Setting the value for time-out of writing
   Since times taken by writing differ according to the device, reconsider the values used below as required.
   The example below is written with 18 ms as the period for time-out of writing.

```
/* Transmit&receive waiting time */
#define FLASH_WBUSY_WAIT(uint16_t)18000/* Write busy waiting time    18000*
1us =  18ms */
```

6.1.4 R_SPI_FLASH_m25p_io.c

This is an I/O module file for this Serial Flash memory.

The settings to be made are identified by the comments header "/** SET **/" in the file.

1. Setting the definition of SFR
When an RL78 family or 78K0R family microcontroller is used, there will be predefined preprocessor symbols in the C compiler used. The program is coded using these predefined preprocessor symbols.

Also, when the microcontroller used is an RL78 family or 78K0R family product and furthermore, the IAR Systems integrated development environment is used, it will be necessary to set the header file in which the SFRs for the microcontroller used are defined.

See the clock synchronous single master control software for the individual microcontroller.

These settings are used for the SPI slave device select control signals.

Table 6.1 Microcontroller and SFR Area Define Settings

Integrated development environment	Microcontroller	SFR setting required?	Method
CubeSuite+ CS+	RL78	Not required	Not required
	78K0R	Not required	Not required
	RX	Not required	Not required
IAR Embedded Workbench	RL78	Required	#ifdef __ICCRL78__ #include <ior5f104pj.h> ← Change to match the microcontroller used. #include <ior5f104pj_ext.h> ← Change to match the microcontroller used. #endif
	78K0R	Required	#ifdef __ICC78K__ #include <io78f1009_64.h> ← Change to match the microcontroller used. #include <io78f1009_64_ext.h> ← Change to match the microcontroller used. #endif
	RX	(Not supported by this software)	(Not supported by this software)

The example below is for the 100-pin RL78/G14 microcontroller.

```
#ifdef __ICCRL78__                      /* IAR RL78 Compiler                */
   #include <ior5f104pj.h>              /* for RL78/G14 100pin (R5F104PJ)   */
   #include <ior5f104pj_ext.h>          /* for RL78/G14 100pin (R5F104PJ)   */
#endif   /* __ICCRL78__ */
```

7. Usage Notes

7.1 Usage Notes to be Observed when Building the Sample Code

Include R_SPI_FLASH_m25p.h when building this sample code into your application.

7.2 When Using a Cache-incorporated MCU

Specify a non-cache area for the buffer that is to be used for storing read/write data.

7.3 When Working with a Different Capacity within the Same Series

Reconsider the following definitions if you need to work with a different capacity but within the same series.

FLASH_MEM_SIZE
FLASH_SECT_ADDR
FLASH_WPAG_SIZE
FLASH_ADDR_SIZE
FLASH_WP_WHOLE_MEM

Since there is a chance that definitions other than the above will also require reconsideration, obtain the datasheet for the memory you are using and adjust definitions as required.

7.4 When Using Other Types of Slave Devices

The sample code can control other slave devices on the same SPI bus.

Refer to this sample code when preparing slave device control programs for such devices.

Note that it is possible to set different baud rates for different slave device control programs.

Website and Support

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

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

Revision Record

Rev.	Date	Description: Page	Description: Summary
1.00	June 30, 2011	—	First edition issued
1.01	Aug 31, 2011	All	Header changed “RX Family” to “MCU”.
		1	Added “78K0R/KE3-L” in Target Device.
		3	Added “78K0R/Kx3-L” in Conditions for Checking the Operation.
		4	Added an application note for 78K0R/Kx3-L in Related Application Notes.
		12	Added the mention of “Maximum user stack size” in Table 6 Note.
		12	Added “78K0R/Kx3-L” in Sizes of Required Memory.
		13	Undated Application Note file name.
		13	Added R_SPI_ FLASH _sfr.78k0r file.
		52	Added “R_SPI_ FLASH_m25p_io.c”.
1.05	Apr 30, 2014	1	Changed Product Name from MCU to RX Family, RL78 Family, 78K0R/Kx3-L.
		1	Modified Introduction to add short address.
		1	Added RX63N, RX63T, RX210, RX21A, RX220, RX111 RL78/G1C, RL78/L12, RL78/L13, RL78/L1C and RL78/G14 as supported devices.
		4, 12	Added 2.1 RX Family and 2.2 RL78 Family, 78K0R/Kx3-L.
		6 to 11	Added the following conditions to section 2.1. (4) RX63N RSPI (5) RX63N SCI (6) RX63T RSPI (7) RX63T SCI (8) RX210 RSPI (9) RX210 SCI (10) RX21A RSPI (11) RX21A SCI (12) RX220 RSPI (13) RX220 SCI (14) RX111 RSPI (15) RX111 SCI
		13 to 17	Added the following conditions to section 2.2. (2) RL78/G14 SAU Integrated Development Environment CubeSuite+ (3) RL78/G14 SAU Integrated Development Environment IAR Embedded Workbench (4) RL78/G1C SAU Integrated Development Environment CubeSuite+ (5) RL78/G1C SAU Integrated Development Environment IAR Embedded Workbench (6) RL78/L12 SAU Integrated Development Environment CubeSuite+ (7) RL78/L12 SAU Integrated Development Environment IAR Embedded Workbench (8) RL78/L13 SAU Integrated Development Environment CubeSuite+ (9) RL78/L13 SAU Integrated Development Environment IAR Embedded Workbench (10) RL78/L1C SAU Integrated Development Environment CubeSuite+

			(11) RL78/L1C SAU Integrated Development Environment IAR Embedded Workbench
		18	Updated application note title in section 3, Related Application Notes. -RX210, RX21A, RX220, RX63N, RX63T, RX111 Group Clock Synchronous Single Master Control Software Using the RSPI (R01AN1196EJ) -RX210, RX21A, RX220, RX63N, RX63T, RX111 Group Clock Synchronous Single Master Control Software Using the SCI (R01AN1229EJ) -RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C Group Clock Synchronous Single Master Control Software Using CSI Mode of Serial Array Unit (R01AN1195EJ)
		26, 34	Added 5.3.1 RX Family and 5.3.2 RL78 Family, 78K0R/Kx3-L.
		28 to 33	Added the following to section 5.3.1, Sizes of Required Memory. (4) RX63N RSPI (5) RX63N SCI (6) RX63T RSPI (7) RX63T SCI (8) RX210 RSPI (9) RX210 SCI (10) RX21A RSPI (11) RX21A SCI (12) RX220 RSPI (13) RX220 SCI (14) RX111 RSPI (15) RX111 SCI
		34 to 36	Added the following to section 5.3.2, Sizes of Required Memory. (2) RL78/G14 SAU Integrated Development Environment CubeSuite+ (3) RL78/G14 SAU Integrated Development Environment IAR Embedded Workbench (4) RL78/L13 SAU Integrated Development Environment CubeSuite+ (5) RL78/L13 SAU Integrated Development Environment IAR Embedded Workbench
		37	5.4 File Configuration Changed name for folder for the sample code. Changed application note number. Changed name of folder for storing the programs. Added new device register common definitions.
		76	Changed 6.1.4 R_SPI_FLASH_m25p_io.c for IAR Embedded Workbench.
		-	Changed “Table No” format.
1.06	Mar 31, 2016	-	Support CC-RL compiler.
		13 to 15	Added the following conditions to section 2.2. (3) RL78/G14 SAU Integrated Development Environment CS+ for CC (Compiler:CC-RL) (6) RL78/G1C SAU Integrated Development Environment CS+ for CC (Compiler:CC-RL)
		36 to 38	Added the following to section 5.3.2, Required Memory Size. (3) RL78/G14 SAU Integrated Development Environment CS+ for CC (Compiler:CC-RL) (6) RL78/L13 SAU Integrated Development Environment CS+ for CC (Compiler:CC-RL)

	39	Updated the software version in Table 5.22 File Configuration.
	78	Updated Table 6.1 Microcontrollers and SFR Area Define Settings.

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. 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 LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the 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.

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

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

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

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

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HALII Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 5.0