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

Renesas R1EX25xxx Series Serial EEPROM Control Software

R01AN0565EJ0106
Rev.1.06
Mar 31, 2016

Introduction

This application note explains how to control Renesas Electronics R1EX25xxx/HN58X25xx Series SPI Serial EEPROMs 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 EEPROM 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 Serial EEPROM Control Software
  http://www.renesas.com/driver/spi_serial_eeprom

Target Device

Serial EEPROM: Renesas Electronics R1EX25xxx Series SPI Serial EEPROM

MCU used for checking the operation of the sample code:

RX600 Series	: RX610, RX62N, RX63N, RX63T (using the SCI)
	: RX62N, RX63N, RX63T (using the RSPI)
RX200 Series	: RX210, RX21A, RX220 (using the SCI)
	: RX210, RX21A, RX220 (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 EEPROM control program controls the Renesas Electronics R1EX25xxx/HN58X25xxx series SPI Serial EEPROM 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 1.1 illustrates a sample configuration.

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

- The SPI Serial EEPROM control program is a block-type device driver that assumes a Renesas Electronics MCU as the master device and a Renesas Electronics R1EX25xxx/HN58X25xxx series SPI Serial EEPROM 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 EEPROM control program can control a maximum of two SPI Serial EEPROM 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).

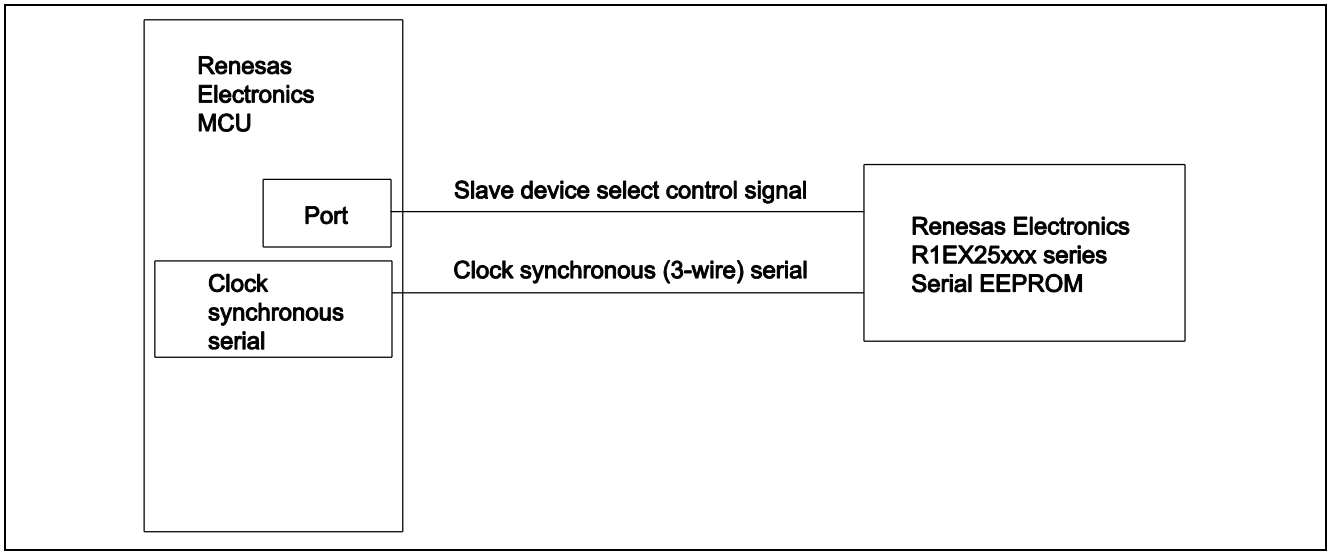


Figure 1.1 Sample Configuration

2. Conditions for Checking the Operation of the SPI Serial EEPROM 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	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
Microcomputer used for evaluation	RX610 Group (Program ROM: 2 MB, RAM: 128 KB)
Operating frequency	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	RX610 Group Clock Synchronous Single Master Control Software Using the SCI (R01AN0534EJ), 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	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
Microcomputer used for evaluation	RX62N Group (Program ROM: 512 KB, RAM: 96 KB)
Operating frequency	ICLK: 96 MHz, PCLK: 48 MHz
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	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
Microcomputer used for evaluation	RX62N Group (Program ROM: 512 KB, RAM: 96 KB)
Operating frequency	ICLK: 96 MHz, PCLK: 48 MHz
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	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
Microcomputer used for evaluation	RX63N Group (Program ROM: 1 MB, RAM: 128 KB)
Operating frequency (microcomputer)	ICLK: 96 MHz, PCLK: 48 MHz
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 (R01AN1196EJ), Ver. 2.04.R01
Evaluation board used	Renesas Starter Kit for the RX63N

(5) RX63N SCI

Table 2.5 Operating Conditions

Item	Description
Memory used for evaluation	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
Microcomputer used for evaluation	RX63N Group (Program ROM: 1 MB, RAM: 128 KB)
Operating frequency (microcomputer)	ICLK: 96 MHz, PCLK: 48 MHz
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 (R01AN1229EJ), 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	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
Microcomputer used for evaluation	RX63T Group (Program ROM: 512 KB, RAM: 48 KB)
Operating frequency (microcomputer)	ICLK: 96 MHz, PCLK: 48 MHz
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 (R01AN1196EJ), Ver. 2.04.R01
Evaluation board used	Renesas Starter Kit for the RX63T

(7) RX63T SCI

Table 2.7 Operating Conditions

Item	Description
Memory used for evaluation	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
Microcomputer used for evaluation	RX63T Group (Program ROM: 512 KB, RAM: 48 KB)
Operating frequency (microcomputer)	ICLK: 96 MHz, PCLK: 48 MHz
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 (R01AN1229EJ), Ver. 2.01.R01
Evaluation board used	Renesas Starter Kit for the RX63T

(8) RX210 RSPI

Table 2.8 Operating Conditions

Item	Description
Memory used for evaluation	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
Microcomputer used for evaluation	RX210 Group (Program ROM: 512 KB, RAM: 64 KB)
Operating frequency (microcomputer)	ICLK: 50 MHz, PCLK: 25 MHz
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 (R01AN1196EJ), Ver. 2.04.R01
Evaluation board used	Renesas Starter Kit for the RX210

(9) RX210 SCI

Table 2.9 Operating Conditions

Item	Description
Memory used for evaluation	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
Microcomputer used for evaluation	RX210 Group (Program ROM: 512 KB, RAM: 64 KB)
Operating frequency (microcomputer)	ICLK: 50 MHz, PCLK: 25 MHz
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 (R01AN1229EJ), Ver. 2.01.R01
Evaluation board used	Renesas Starter Kit for the RX210

(10) RX21A RSPI

Table 2.10 Operating Conditions

Item	Description
Memory used for evaluation	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
Microcomputer used for evaluation	RX21A Group (Program ROM: 512 KB, RAM: 64 KB)
Operating frequency (microcomputer)	ICLK: 50 MHz, PCLK: 25 MHz
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 (R01AN1196EJ), Ver. 2.04.R01
Evaluation board used	HSBRX21AP-B (Hokuto Denshi Co., Ltd.)

(11) RX21A SCI

Table 2.11 Operating Conditions

Item	Description
Memory used for evaluation	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
Microcomputer used for evaluation	RX21A Group (Program ROM: 512 KB, RAM: 64 KB)
Operating frequency (microcomputer)	ICLK: 50 MHz, PCLK: 25 MHz
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 (R01AN1229EJ), Ver. 2.01.R01
Evaluation board used	HSBRX21AP-B (Hokuto Denshi Co., Ltd.)

(12) RX220 RSPI

Table 2.12 Operating Conditions

Item	Description
Memory used for evaluation	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
Microcomputer used for evaluation	RX220 Group (Program ROM: 256 KB, RAM: 16 KB)
Operating frequency (microcomputer)	ICLK: 32 MHz, PCLK: 32 MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics High-performance embedded Workshop Version 4.09.01.007
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 (R01AN1196EJ), Ver. 2.04.R01
Evaluation board used	Renesas Starter Kit for the RX220

(13) RX220 SCI

Table 2.13 Operating Conditions

Item	Description
Memory used for evaluation	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
Microcomputer used for evaluation	RX220 Group (Program ROM: 256 KB, RAM: 16 KB)
Operating frequency (microcomputer)	ICLK: 32 MHz, PCLK: 32 MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics High-performance embedded Workshop Version 4.09.01.007
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 (R01AN1229EJ), Ver. 2.01.R01
Evaluation board used	Renesas Starter Kit for the RX220

(14) RX111 RSPI

Table 2.14 Operating Conditions

Item	Description
Memory used for evaluation	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
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. 2.03 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	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
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. 2.03 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	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
Microcomputer used for evaluation	78K0R/KE3-L (Program ROM: 64 KB, RAM: 3 KB)
Operating frequency	Main System Clock: 20 MHz CPU/peripheral Hardware Clock: 20 MHz Serial Clock: 2.5 MHz
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.
Integrated Debugger	Renesas Electronics Integrated Debugger ID78K0R-QB Ver.3.61
Version of the sample code	Ver. 2.01
Software used for evaluation	78K0R/Kx3-L Clock synchronous single-master control software using the SAU CSI mode (R01AN0708EJ), Ver. 2.00
Evaluation board used	78K0R/KE3-L Target Board (QB-78K0RKE3L-TB)

(2) RL78/G14 SAU Integrated Development Environment CS+ for CA,CX (Compiler: CA78K0R)

Table 2.17 Operating Conditions

Item	Description
Memory used for evaluation	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
Microcomputer used for evaluation	RL78/G14 Group (Program ROM: 256 KB, RAM: 24 KB)
Operating frequency (microcomputer)	Main system clock: 32 MHz CPU/peripheral hardware clock: 32 MHz Serial clock: 4 MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics CS+ for CA, CX V3.01.00
C compiler	Renesas Electronics RL78,78K0R compiler CA78K0R V1.71
	Compiler options: The default settings (-qx2) for the integrated development environment are used.
Version of the sample code	Ver. 2.04
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 (R01AN1195EJ), Ver. 2.05
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	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
Microcomputer used for evaluation	RL78/G14 Group (Program ROM: 256 KB, RAM: 24 KB)
Operating frequency (microcomputer)	Main system clock: 32 MHz CPU/peripheral hardware clock: 32 MHz Serial clock: 4 MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics CS+ for CC V3.03.00
C compiler	Renesas Electronics 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. 2.04
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 (R01AN1195EJ), Ver. 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	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
Microcomputer used for evaluation	RL78/G14 Group (Program ROM: 256 KB, RAM: 24 KB)
Operating frequency (microcomputer)	Main system clock: 32 MHz CPU/peripheral hardware clock: 32 MHz Serial clock: 4 MHz
Operating voltage	3.3 V
Integrated development environment	IAR Systems IAR Embedded Workbench for Renesas RL78 (Ver.1.30.2)
C compiler	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 ("level: low") for the integrated development environment are used.
Version of the sample code	Ver. 2.03
Software used for evaluation	RL78/G14 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/G14

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

Table 2.20 Operating Conditions

Item	Description
Memory used for evaluation	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
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: 4 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. 2.03 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 IAR Embedded Workbench

Table 2.21 Operating Conditions

Item	Description
Memory used for evaluation	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
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: 4 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. 2.03 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

(7) RL78/L12 SAU Integrated Development Environment CubeSuite+

Table 2.22 Operating Conditions

Item	Description
Memory used for evaluation	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
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: 4 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. 2.03 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

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

Table 2.23 Operating Conditions

Item	Description
Memory used for evaluation	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
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: 4 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. 2.03 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/L13 SAU Integrated Development Environment CubeSuite+

Table 2.24 Operating Conditions

Item	Description
Memory used for evaluation	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
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: 4 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. 2.03 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

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

Table 2.25 Operating Conditions

Item	Description
Memory used for evaluation	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
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: 4 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. 2.03 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/L1C SAU Integrated Development Environment CubeSuite+

Table 2.26 Operating Conditions

Item	Description
Memory used for evaluation	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
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: 4 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. 2.03 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

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

Table 2.27 Operating Conditions

Item	Description
Memory used for evaluation	Renesas Electronics R1EX25xxx/HN58X25xxx Series SPI Serial EEPROM
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: 4 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. 2.03 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 4.1 shows a sample wiring configuration.

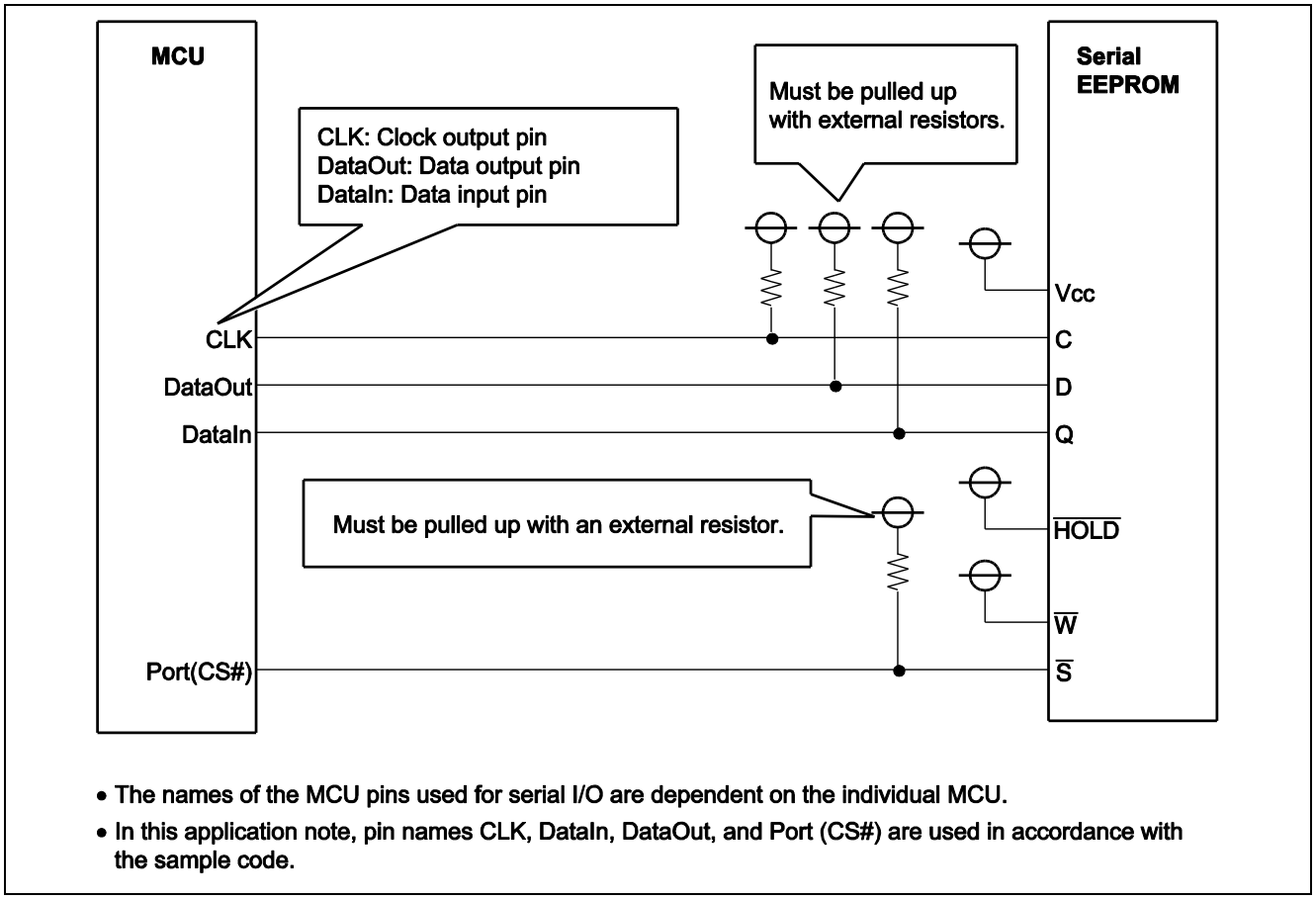


Figure 4.1 Sample Wiring Diagram for an MCU and an SPI Slave Device

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 EEPROM 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.

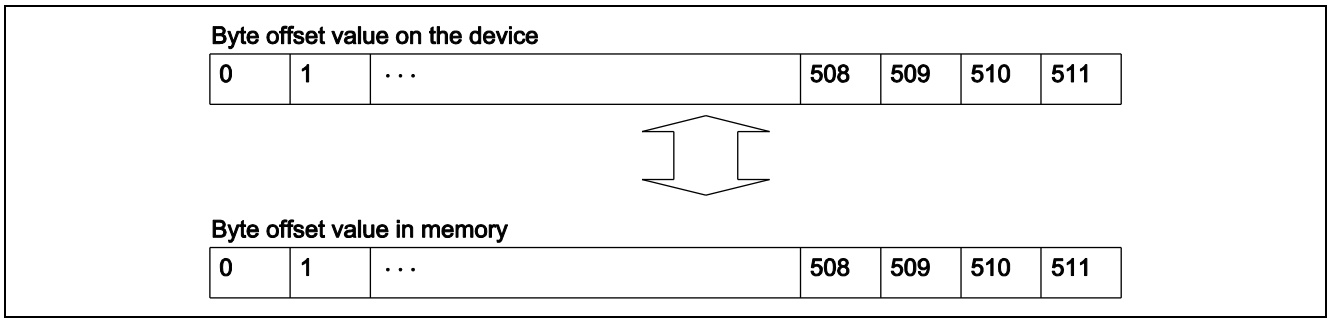
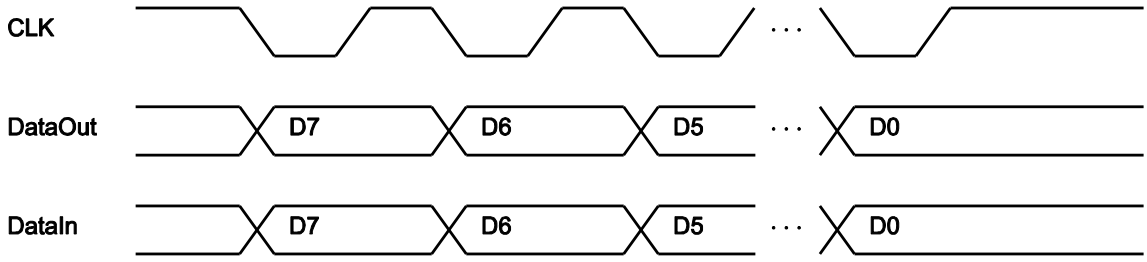


Figure 5.1 Storage Format of the Transferred Data

5.1.1 Clock Synchronous Mode Timing

The SPI mode 3 (CPOL=1, CPHA=1) timing shown in Figure 5.2 is used to control the Serial EEPROM.



- MCU->Serial EEPROM transmission: Transmission of transmit data is started on the falling edge of the transfer clock.
- Serial EEPROM->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 5.2 Clock Synchronous Mode Timing Setup

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

5.1.2 Serial EEPROM S# Pin Control

The S# pin of the Serial EEPROM 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 EEPROM and the falling edge of the C (MCU's CLK) signal of the Serial EEPROM is controlled with software wait processing to account for the Serial EEPROM S# setup time.

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

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

5.1.3 Serial EEPROM Instruction Code

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

Table 5.1 Instruction Set

Instruction	Description	Instruction Format
WREN	Write Enable	0000 0110
WRDI	Write Disable	0000 0100
RDSR	Read Status Register	0000 0101
WRSR	Write Status Register	0000 0001
READ	Read from Memory Array	0000 0011
WRITE	Write to Memory Array	0000 0010

5.1.4 Serial EEPROM Status Register

The status register of the Serial EEPROM can be read and written using dedicated instructions. The configuration of the status register is dependent on the size of the memory to be used.

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

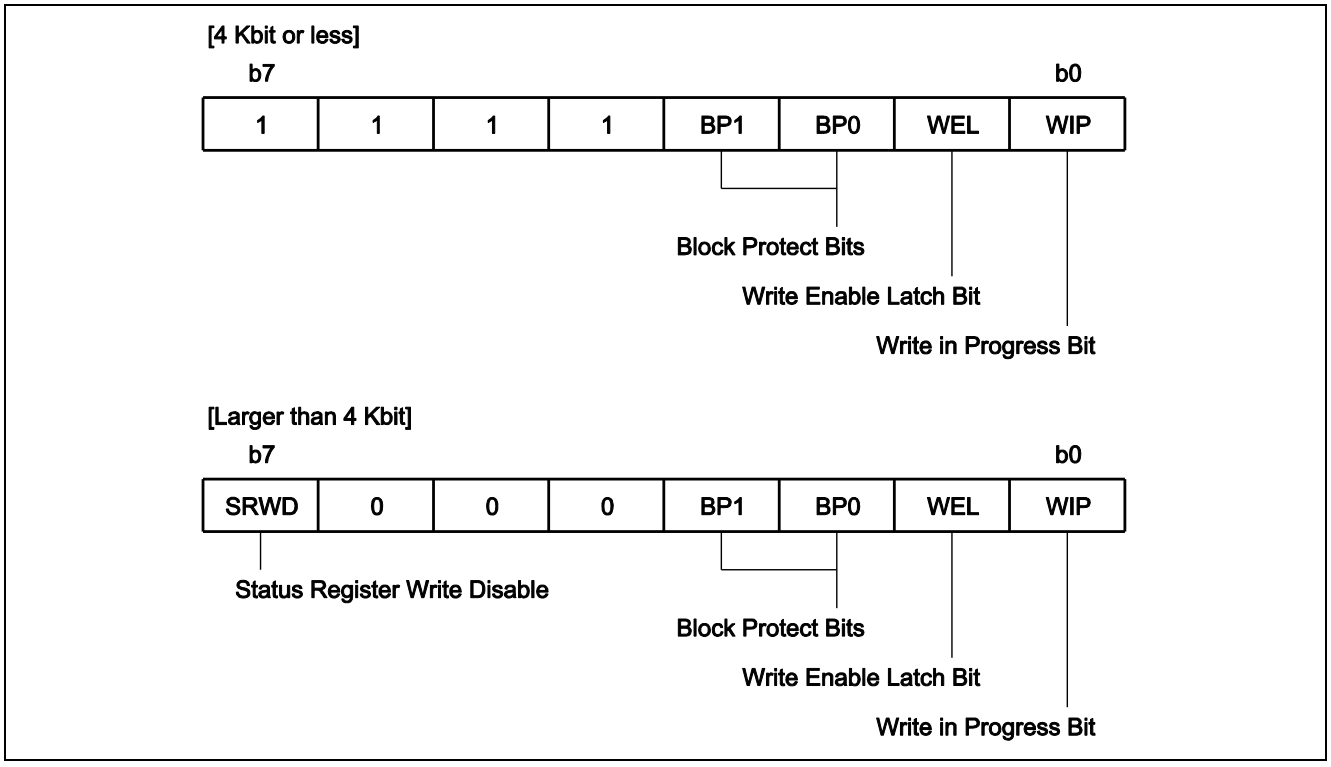


Figure 5.3 Status Register Configuration

5.2 Software Control Outline

5.2.1 Software Configuration

The sample code ranks in the higher-level layer of the SPI Serial EEPROM control software system (EEPROM control software shown in Figure 5.4).

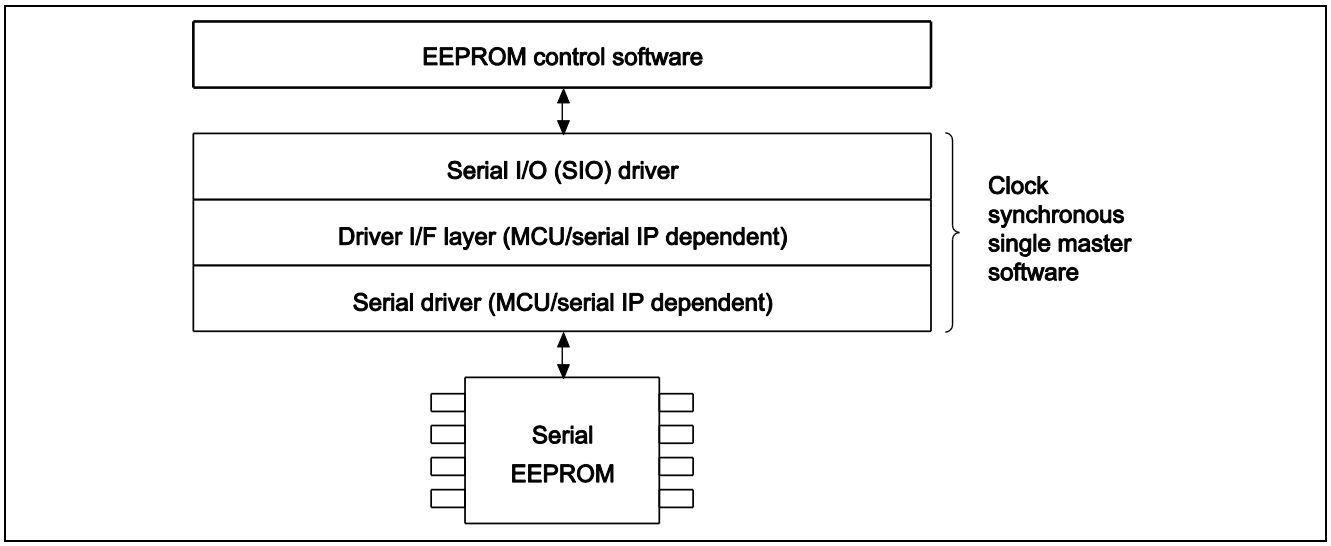


Figure 5.4 Software Configuration

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 EEPROM.
- Data transmission
  Send data to the Serial EEPROM.
- Data reception
  Receive response/data from the Serial EEPROM.
- 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_EEP_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 (EEP_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_EEP_Send_Cmd())

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

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

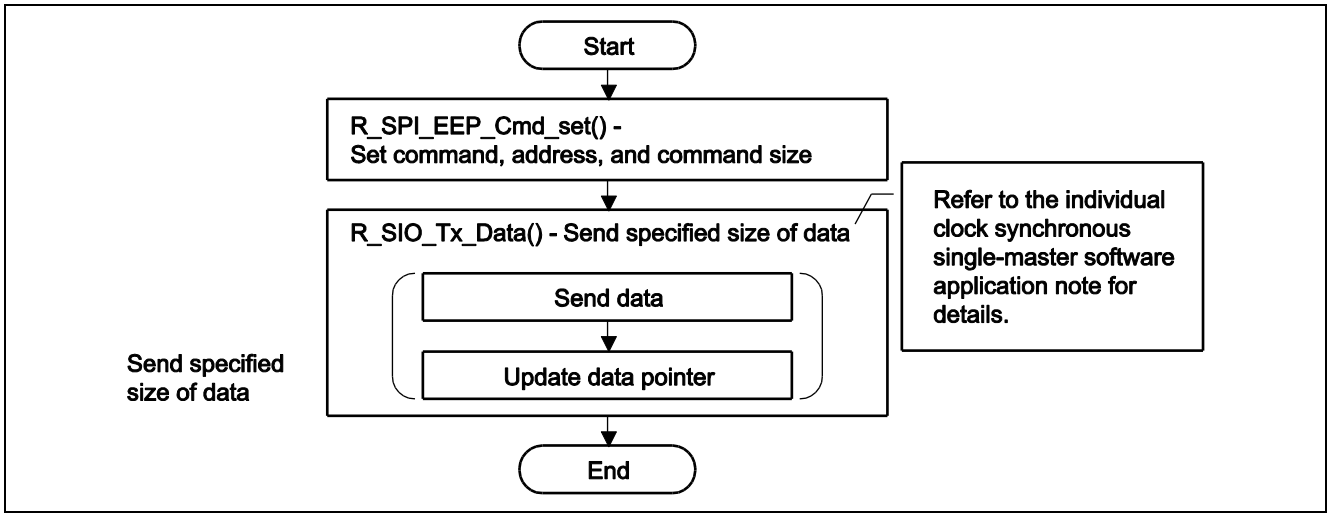


Figure 5.5 Outline of Command Transmission Processing (R_SPI_EEP_Send_Cmd())

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 EEPROM Control Software, for the environment.

Table 5.2 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	838 bytes (little endian) 705 bytes (little endian)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 bytes (little endian) 4+n bytes (little endian) (n denotes the number of devices to be used.)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
Maximum user stack size	152 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.
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 EEPROM Control Software, for the environment.

Table 5.3 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	705 bytes (little endian) 897 bytes (little endian)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 bytes (little endian) 4+n bytes (little endian) (n denotes the number of devices to be used.)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
Maximum user stack size	168 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 EEPROM Control Software, for the environment.

Table 5.4 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	705 bytes (little endian) 898 bytes (little endian)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 bytes (little endian) 4+n bytes (little endian) (n denotes the number of devices to be used.)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
Maximum user stack size	148 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 EEPROM Control Software, for the environment.

Table 5.5 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	705 bytes (little endian) 897 bytes (little endian)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 bytes (little endian) 4+n bytes (little endian) (n denotes the number of devices to be used.)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
Maximum user stack size	168 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 EEPROM Control Software, for the environment.

Table 5.6 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	705 bytes (little endian) 898 bytes (little endian)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 bytes (little endian) 4+n bytes (little endian) (n denotes the number of devices to be used.)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
Maximum user stack size	148 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 EEPROM Control Software, for the environment.

Table 5.7 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	705 bytes (little endian) 897 bytes (little endian)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 bytes (little endian) 4+n bytes (little endian) (n denotes the number of devices to be used.)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
Maximum user stack size	168 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 EEPROM Control Software, for the environment.

Table 5.8 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	705 bytes (little endian) 898 bytes (little endian)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 bytes (little endian) 4+n bytes (little endian) (n denotes the number of devices to be used.)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
Maximum user stack size	148 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 EEPROM Control Software, for the environment.

Table 5.9 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	705 bytes (little endian) 897 bytes (little endian)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 bytes (little endian) 4+n bytes (little endian) (n denotes the number of devices to be used.)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
Maximum user stack size	168 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 EEPROM Control Software, for the environment.

Table 5.10 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	705 bytes (little endian) 898 bytes (little endian)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 bytes (little endian) 4+n bytes (little endian) (n denotes the number of devices to be used.)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
Maximum user stack size	148 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 EEPROM Control Software, for the environment.

Table 5.11 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	705 bytes (little endian) 897 bytes (little endian)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 bytes (little endian) 4+n bytes (little endian) (n denotes the number of devices to be used.)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
Maximum user stack size	168 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 EEPROM Control Software, for the environment.

Table 5.12 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	705 bytes (little endian) 898 bytes (little endian)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 bytes (little endian) 4+n bytes (little endian) (n denotes the number of devices to be used.)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
Maximum user stack size	148 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 EEPROM Control Software, for the environment.

Table 5.13 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	705 bytes (little endian) 897 bytes (little endian)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 bytes (little endian) 4+n bytes (little endian) (n denotes the number of devices to be used.)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
Maximum user stack size	168 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 EEPROM Control Software, for the environment.

Table 5.14 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	705 bytes (little endian) 898 bytes (little endian)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 bytes (little endian) 4+n bytes (little endian) (n denotes the number of devices to be used.)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
Maximum user stack size	148 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 EEPROM Control Software, for the environment.

Table 5.15 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1,613 bytes (little endian)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	4+n bytes (little endian) (n denotes the number of devices to be used.)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
Maximum user stack size	184 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.
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 EEPROM Control Software, for the environment.

Table 5.16 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1,613 bytes (little endian)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	4+n bytes (little endian) (n denotes the number of devices to be used.)	R_SPI_EEP_usr.c R_SPI_EEP_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.
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 EEPROM Control Software, for the environment.

Table 5.17 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1,211 bytes 1,264 bytes	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 bytes 6 bytes (The number of devices to be used is 1 or 2.)	R_SPI_EEP_usr.c R_SPI_EEP_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 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 CS+ for CA,CX (Compiler: CA78K0R)

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

Table 5.18 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1,254 bytes 1,232 bytes	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 bytes 6 bytes (The number of devices to be used is 1 or 2.)	R_SPI_EEP_usr.c R_SPI_EEP_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 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 EEPROM Control Software, for the environment.

Table 5.19 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	879 bytes 864 bytes	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 bytes 6 bytes (The number of devices to be used is 1 or 2.)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
Maximum user stack size	98 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 SAU Integrated Development Environment IAR Embedded Workbench

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

Table 5.20 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	1,353 bytes 1,091 bytes	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	0 bytes 6 bytes (The number of devices to be used is 1 or 2.)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
Maximum user stack size	132 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+

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

Table 5.21 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	2,206 bytes	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	6 bytes (The number of devices to be used is 1 or 2.)	R_SPI_EEP_usr.c R_SPI_EEP_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 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 IAR Embedded Workbench

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

Table 5.22 Sizes of Required Memory

Memory Used	Size	Remarks
ROM	2,160 bytes	R_SPI_EEP_usr.c R_SPI_EEP_io.c
RAM	6 bytes (The number of devices to be used is 1 or 2.)	R_SPI_EEP_usr.c R_SPI_EEP_io.c
Maximum user stack size	84 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 5.23 File Configuration

\an_r01an0565ej0106_mcu_serial <DIR>			Folder for the sample code
r01an0565ej0106_mcu.pdf			Application note
\source <DIR>			Folder for storing the programs
	\r_spi_eep <DIR>		Folder for the Serial EEPROM control software
		R_SPI_EEP.h	Header file
		R_SPI_EEP_io.c	I/O module
		R_SPI_EEP_io.h	I/O module common definitions
		R_SPI_EEP_sfr.h.78k0r	Common register definitions (78K0R/Kx3-L)
		R_SPI_EEP_sfr.h.rl78g14	Common register definitions (RXRL78/G14)
		R_SPI_EEP_sfr.h.rl78g1c	Common register definitions (RXRL78/G1C)
		R_SPI_EEP_sfr.h.rl78l12	Common register definitions (RXRL78/L12)
		R_SPI_EEP_sfr.h.rl78l13	Common register definitions (RXRL78/L13)
		R_SPI_EEP_sfr.h.rl78l1C	Common register definitions (RXRL78/L1C)
		R_SPI_EEP_sfr.h.rx21a	Common register definitions (RX21A)
		R_SPI_EEP_sfr.h.rx62n	Common register definitions (RX62N)
		R_SPI_EEP_sfr.h.rx63n	Common register definitions (RX63N)
		R_SPI_EEP_sfr.h.rx63t	Common register definitions (RX63T)
		R_SPI_EEP_sfr.h.rx111	Common register definitions (RX111)
		R_SPI_EEP_sfr.h.rx210	Common register definitions (RX210)
		R_SPI_EEP_sfr.h.rx220	Common register definitions (RX220)
		R_SPI_EEP_sfr.h.rx610	Common register definitions (RX610)
		R_SPI_EEP_usr.c	User I/F module
	\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 5.24 Return Values

Constant Name	Value	Description
EEP_OK	(error_t)(0)	Successful Operation
EEP_ERR_PARAM	(error_t)(–1)	Parameter Error
EEP_ERR_HARD	(error_t)(–2)	Hardware Error
EEP_ERR_WP	(error_t)(–4)	Write-Protection Error
EEP_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 5.25 Command Definitions

Constant Name	Value	Description
EEP_CMD_WREN	(uint8_t)0x06	Write Enable
EEP_CMD_WRDI	(uint8_t)0x04	Write Disable
EEP_CMD_RDSR	(uint8_t)0x05	Read Status Register
EEP_CMD_WRSR	(uint8_t)0x01	Write Status Register
EEP_CMD_READ	(uint8_t)0x03	Read for Memory Array
EEP_CMD_WRITE	(uint8_t)0x02	Write for Memory Array

5.5.3 Miscellaneous Definitions

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

Table 5.26 Miscellaneous Definitions

Constant Name	Value	Description
EEP_LOG_ERR	1	Log type: Error
EEP_TRUE	(uint8_t)0x01	Flag "ON"
EEP_FALSE	(uint8_t)0x00	Flag "OFF"
EEP_WP_NONE	(uint8_t)0x00	Write-Protection Status: None setting
EEP_WP_UPPER_QUART	(uint8_t)0x01	Write-Protection Status: Upper quarter setting
EEP_WP_UPPER_HALF	(uint8_t)0x02	Write-Protection Status: Upper half setting
EEP_WP_WHOLE_MEM	(uint8_t)0x03	Write-Protection Status: Whole memory setting
EEP_MEM_SIZE	(uint32_t)0x100	Memory size (in bytes) The value shown to the left is for 2 Kbit memory.
EEP_WPAG_SIZE	(uint32_t)16	Page size (in bytes) The value shown to the left is for 2 Kbit memory.
EEP_ADDR_SIZE	(uint8_t)1	Address size (in bytes) The value shown to the left is for 2 Kbit memory.
EEP_UPPER_QUART	(EEP_MEM_SIZE - (EEP_MEM_SIZE / 4))	Start address used for upper 1/4 setting
EEP_UPPER_HALF	(EEP_MEM_SIZE / 2)	Start address used for upper 1/2 setting
EEP_CMD_SIZE	(uint8_t)1	Command size (in bytes)
EEP_STSREG_SIZE	(uint16_t)1	Status register size (in bytes)
EEP_SHORT_SIZE	(uint32_t)0x0000fff0	Sets the maximum transfer size for lower-level functions (set to 0xFFFF or lower.)
EEP_HI	(uint8_t)0x01	Port "H"
EEP_LOW	(uint8_t)0x00	Port "L"
EEP_OUT	(uint8_t)0x01	Port Output Setting
EEP_IN	(uint8_t)0x00	Port Input Setting
EEP_READY_WAIT	(uint16_t)8000	Write Busy Waiting Time 8000 × 10 μs = 80 ms
EEP_T_READY_WAIT	(uint16_t)MTL_T_10US	Write Busy Completion Polling Time
EEP_T_CS_HOLD	(uint16_t)MTL_T_1US	CS Stability Waiting Time
EEP_T_R_ACCESS	(uint16_t)MTL_T_1US	Reading Start Waiting Time
EEP_REG_SRWD	(uint8_t)0x80	Status Register Write Disable
EEP_REG_BP1	(uint8_t)0x08	Block Protect Bit1
EEP_REG_BP0	(uint8_t)0x04	Block Protect Bit0
EEP_REG_WEL	(uint8_t)0x02	Write Enable Latch Bit
EEP_REG_WIP	(uint8_t)0x01	Write In Progress Bit
EEP_BYTE_READ	2	Number of bytes used to identify the execution of a 1-byte read
EEP_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.

```
/* uint16_t <-> uint8_t conversion */
typedef union {
  uint32_t  ul;
  uint8_t uc[4];
} EEP_EXCHG_LONG;                          /* total 4byte                */

typedef struct {
  uint8_t EEPCmd;                          /* Command                    */
  uint8_t EEPAddr[3];                      /* Operand                    */
} EEP_CMD;                                 /* total 4byte                */
```

5.7 List of Variables

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

Table 5.27 List of Variables

Type	Variable Name	Description	Used in
STATIC EEP_CMD	gEep_CmdBuf	Command buffer	R_SPI_EEP_Send_Cmd R_SPI_EEP_Cmd_set
uint8_t	gEep_WP[EEP_DEV_NUM]	Command buffer	R_SPI_EEP_Init_Ram R_SPI_EEP_Write_Protect R_SPI_EEP_Write_Data

5.8 List of Functions

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

Table 5.28 List of Functions

Function Name	Outline
R_SPI_EEP_Init_Driver	Initialize driver.
R_SPI_EEP_Read_Status	Read status.
R_SPI_EEP_Write_Protect	Set write protection.
R_SPI_EEP_Read_Data	Read data.
R_SPI_EEP_Write_Data	Write data.
R_SPI_EEP_Init_Port	Initialize port.
R_SPI_EEP_Init_Ram	Initialize RAM.
R_SPI_EEP_Send_Cmd	Send command.
R_SPI_EEP_Write_En	Enable writes.
R_SPI_EEP_Write_Di	Disable writes.
R_SPI_EEP_Read_StsReg	Read status register.
R_SPI_EEP_Write_StsReg	Write status register.
R_SPI_EEP_Wait_WBusy	Busy wait.
R_SPI_EEP_Write_Page	Write.
R_SPI_EEP_Read_Memory	Read memory.
R_SPI_EEP_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_EEP_Init_Driver	
Synopsis	Initialize driver.
Headers	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h
Declaration	error_t R_SPI_EEP_Init_Driver(void)
Explanation	• This function initializes the driver. It performs the following operations on each device: — Calls the R_SPI_EEP_Init_Ram() function to initialize RAM. — Calls the R_SPI_EEP_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. EEP_OK ; Successful operation EEP_ERR_OTHER ; Other error Returns the return value of R_SIO_Init_Driver().
Remarks	None

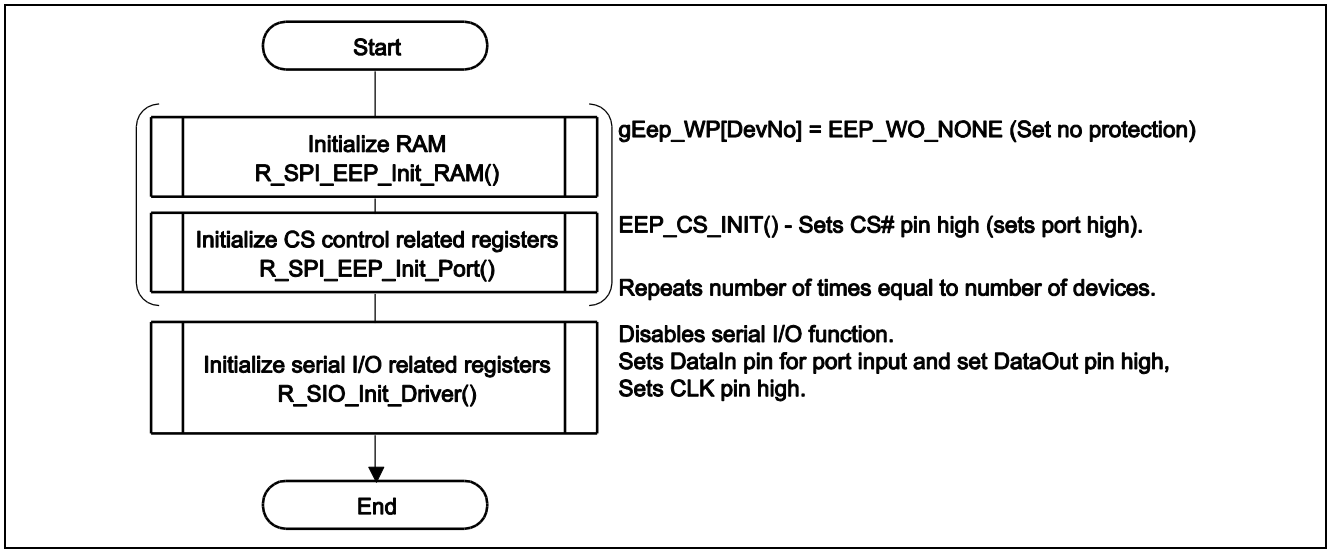


Figure 5.6 Driver Initialization Processing Outline

5.9.2 Read Status Processing

R_SPI_EEP_Read_Status

Synopsis Read status register.

Headers R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h

Declaration error_t R_SPI_EEP_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:
  - 4 Kbit or less
    - Bit 7 to 4: Reserved (All "1")
    - Bit 3 to 2: BP1, BP0 — 00: No protection; 01: Upper-quarter protection; 10: Upper-half protection; 11: Whole memory protection
    - Bit 1: WEL — 0: Write disabled; 1: Write enabled
    - Bit 0: WIP — 1: During write operation
  - More than 4 Kbit
    - Bit 7: SRWD — 0: Status register can be changed.; 1: Status register cannot be changed.
    - Bit 6 to 4: Reserved (All "0")
    - Bit3 to 2: BP1, BP0 — 00: No protection; 01: Upper-quarter protection; 10: Upper-half protection; 11: Whole memory protection
    - Bit 1: WEL — 0: Write disabled; 1: Write enabled
    - Bit 0: WIP — 1: During write operation

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.

EEP_OK	;	Successful operation
EEP_ERR_PARAM	;	Parameter error
EEP_ERR_HARD	;	Hardware error
EEP_ERR_OTHER	;	Other error

Remarks None

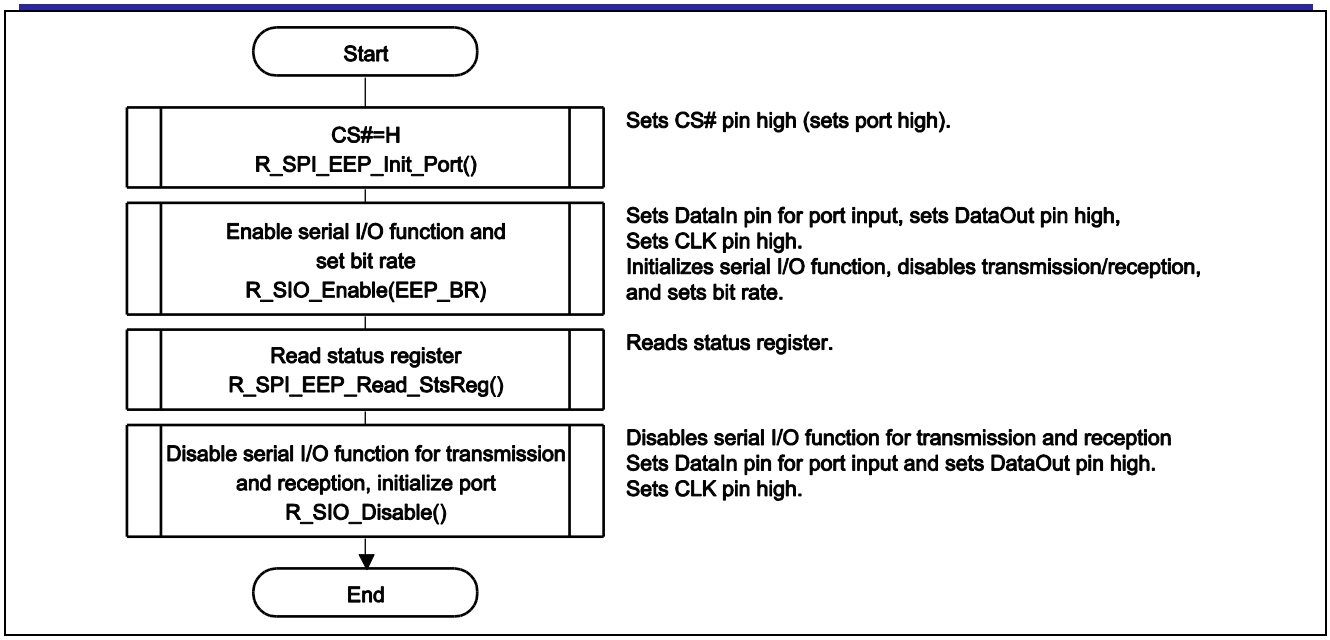


Figure 5.7 Status Register Read Processing Outline

5.9.3 Write Protect Setup Processing

R_SPI_EEP_Write_Protect	
Synopsis	Performs write protect setup processing.
Headers	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h
Declaration	error_t R_SPI_EEP_Write_Protect(uint8_t DevNo, uint8_t WpSts)
Explanation	• Sets write protection. • The write protection setting data (WpSts) must be set up as follows: EEP_WP_NONE: No protection EEP_WP_UPPER_QUART: Upper-quarter protection setting EEP_WP_UPPER_HALF: Upper-half protection setting EEP_WP_WHOLE_MEM: Whole memory protection setting
Arguments	uint8_t DevNo ; Device number uint8_t WpSts ; Write protect setting data
Return value	• Returns the result of write protect setup processing. EEP_OK ; Successful operation EEP_ERR_PARAM ; Parameter error EEP_ERR_OTHER ; Other error
Remarks	None

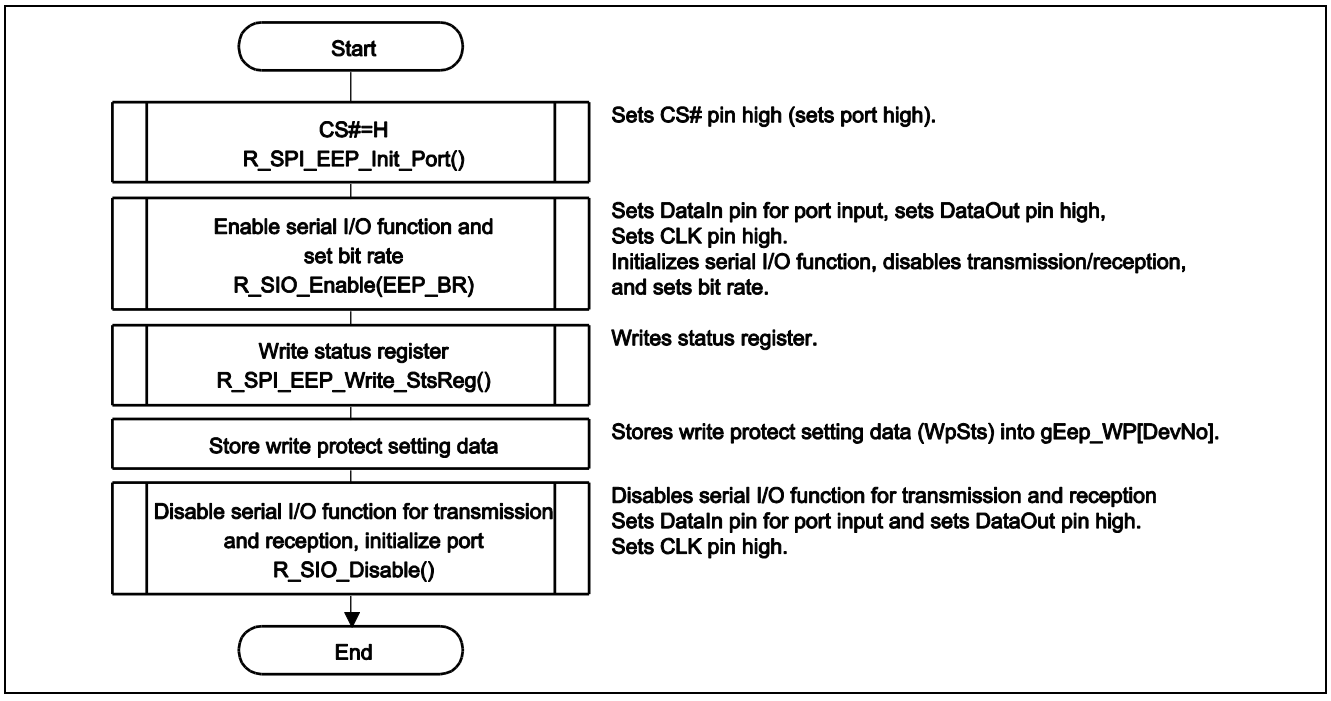


Figure 5.8 Write Protect Setup Processing Outline

5.9.4 Data Read Processing

R_SPI_EEP_Read_Data				
Synopsis	Reads data.			
Headers	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h			
Declaration	error_t R_SPI_EEP_Read_Data(uint8_t DevNo, uint32_t RAddr, uint32_t RCnt, uint8_t FAR* pData)			
Explanation	• Reads the specified number of bytes from EEPROM, 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.			
	EEP_OK		;	Successful operation
	EEP_ERR_PARAM		;	Parameter error
	EEP_ERR_HARD		;	Hardware error
	EEP_ERR_OTHER		;	Other error
Remarks	• The highest read address is EEPROM size -1.			

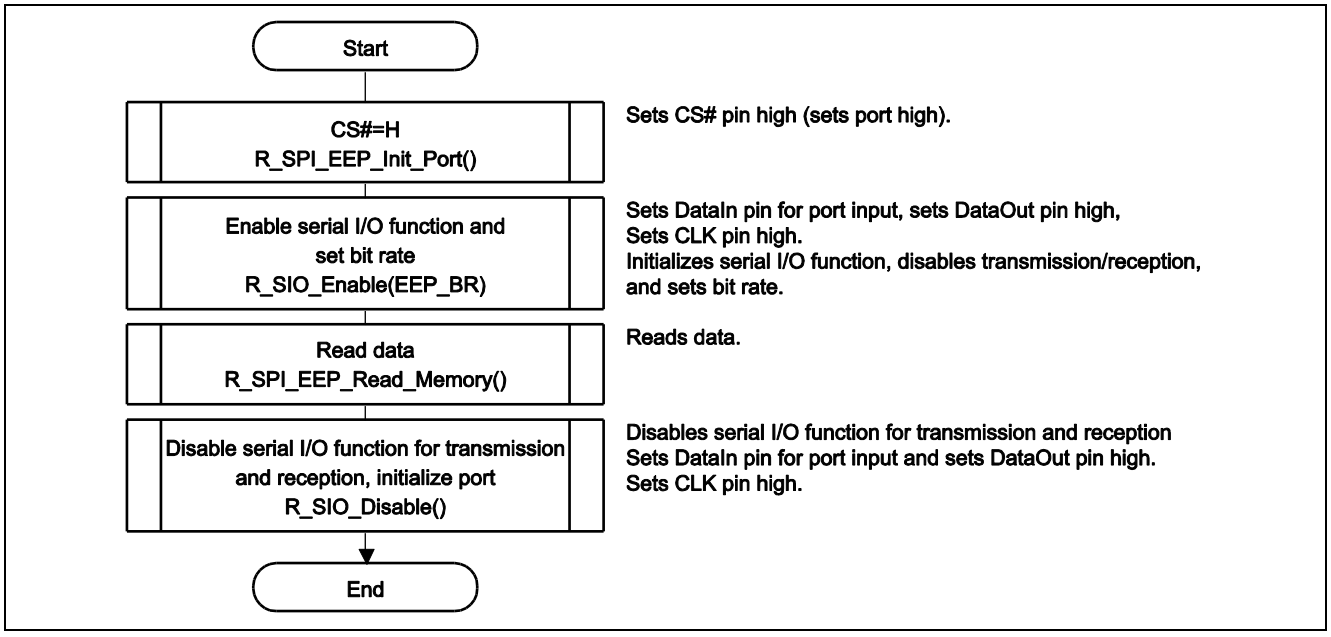


Figure 5.9 Data Read Processing Outline

5.9.5 Data Write Processing

R_SPI_EEP_Write_Data			
Synopsis	Writes data.		
Headers	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h		
Declaration	error_t R_SPI_EEP_Write_Data(uint8_t DevNo, uint32_t WAddr, uint32_t WCnt, uint8_t FAR* pData)		
Explanation	• Writes the data from pData into EEPROM 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.		
	EEP_OK		; Successful operation
	EEP_ERR_PARAM		; Parameter error
	EEP_ERR_HARD		; Hardware error
	EEP_ERR_WP		; Write protect error
	EEP_ERR_OTHER		; Other error
Remarks	• Writing into EEPROM is enabled only when write protection is off.		
	• Writing to protected page is not possible. Furthermore, an error response is returned if this is attempted.		
	• The highest write address is EEPROM size -1.		

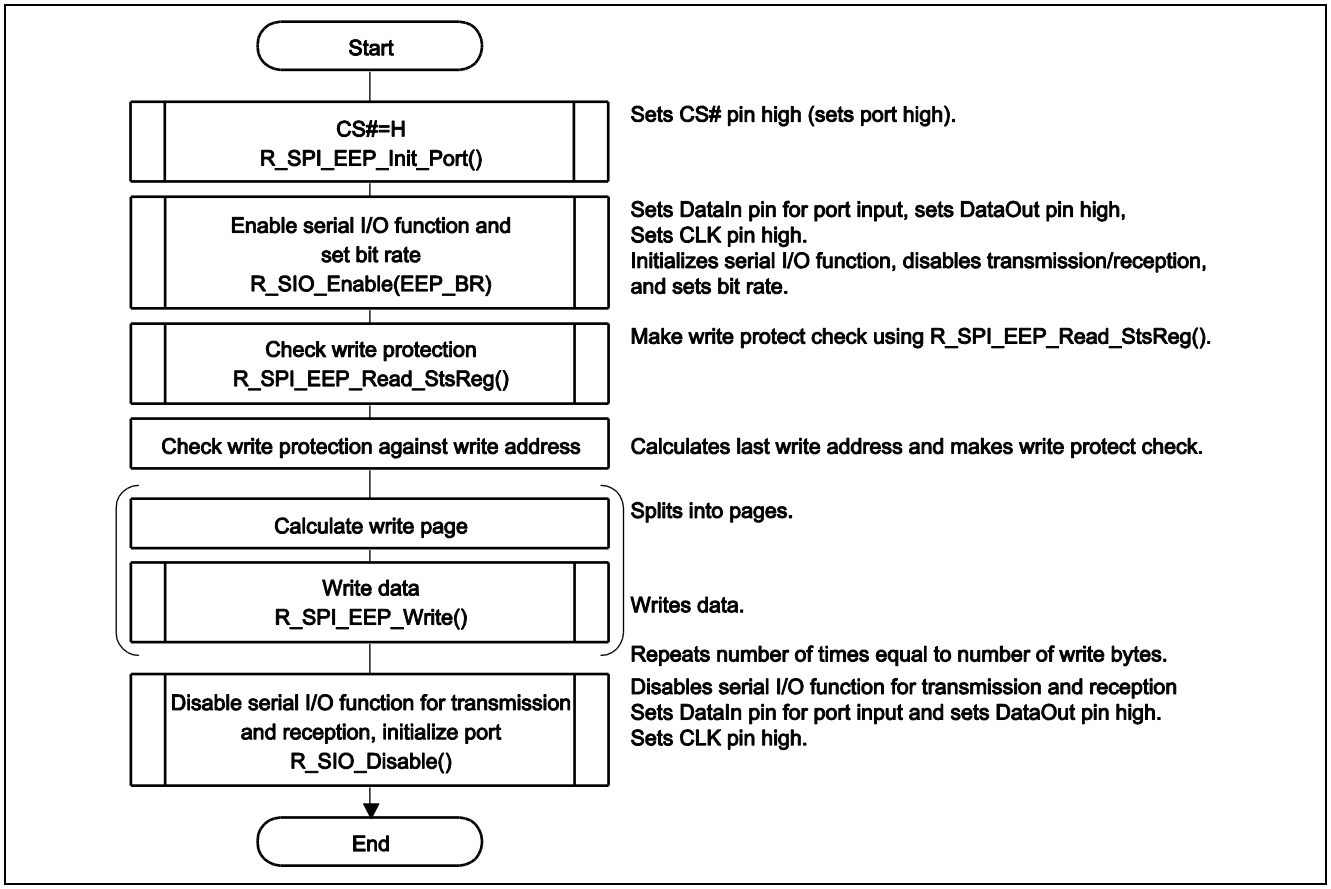


Figure 5.10 Data Write Processing Outline

5.9.6 Port Initialization Processing (Internal Function)

R_SPI_EEP_Init_Port	
Synopsis	Initializes ports (internal function).
Headers	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h
Declaration	void R_SPI_EEP_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

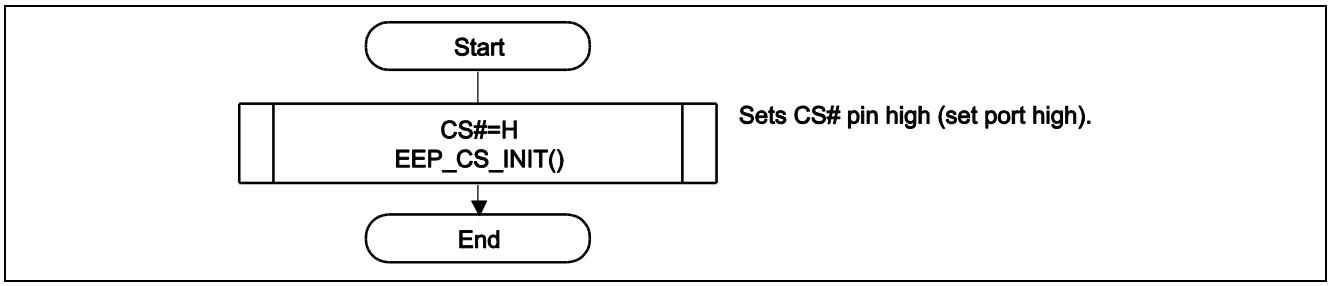


Figure 5.11 Port Initialization Processing Outline

5.9.7 RAM Initialization Processing (Internal Function)

R_SPI_EEP_Init_Ram	
Synopsis	Initializes port (internal function).
Headers	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h
Declaration	void R_SPI_EEP_Init_Ram(uint8_t DevNo)
Explanation	• Initialize RAM associated with the specified device.
Arguments	uint8_t DevNo ; Device number
Return value	void
Remarks	None

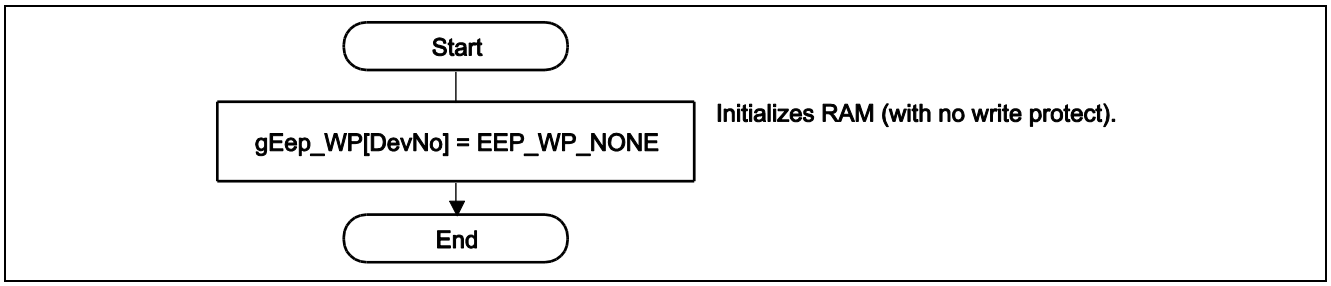


Figure 5.12 RAM Initialization Processing Outline

5.9.8 Command Send Processing (Internal Function)

R_SPI_EEP_Send_Cmd	
Synopsis	Sends command (internal function).
Headers	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h
Declaration	STATIC error_t R_SPI_EEP_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. EEP_OK ; Successful operation EEP_ERR_HARD ; Hardware error EEP_ERR_OTHER ; Other error
Remarks	None

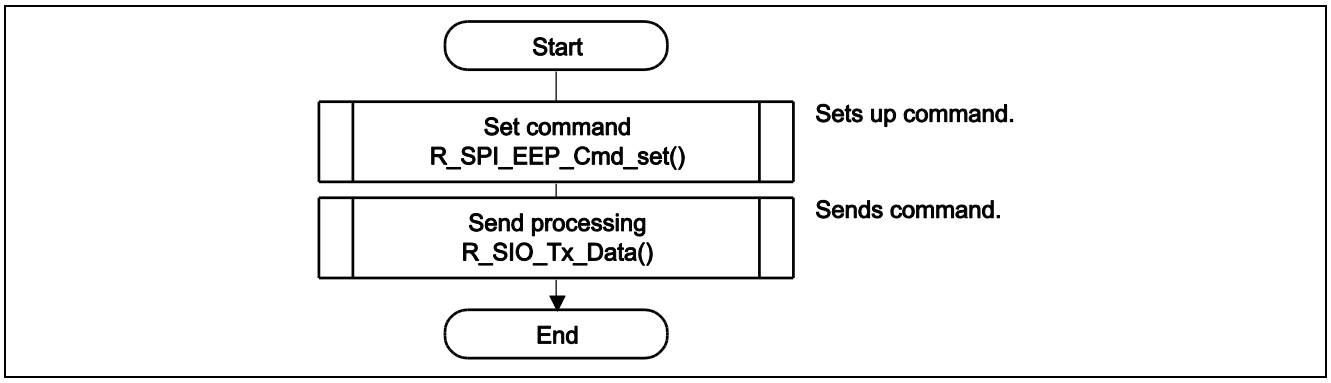


Figure 5.13 Command Send Processing Outline

5.9.9 Read Command Processing (Internal Function)

R_SPI_EEP_Read_Memory	
Synopsis	Performs Read command processing (internal function).
Headers	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h
Declaration	error_t R_SPI_EEP_Read_Memory(uint8_t DevNo, uint32_t RAddr, uint32_t RCnt, uint8_t FAR* pData)
Explanation	• Reads the specified number of bytes from EEPROM 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. EEP_OK ; Successful operation EEP_ERR_HARD ; Hardware error EEP_ERR_OTHER ; Other error
Remarks	• The highest read address is EEPROM size -1.

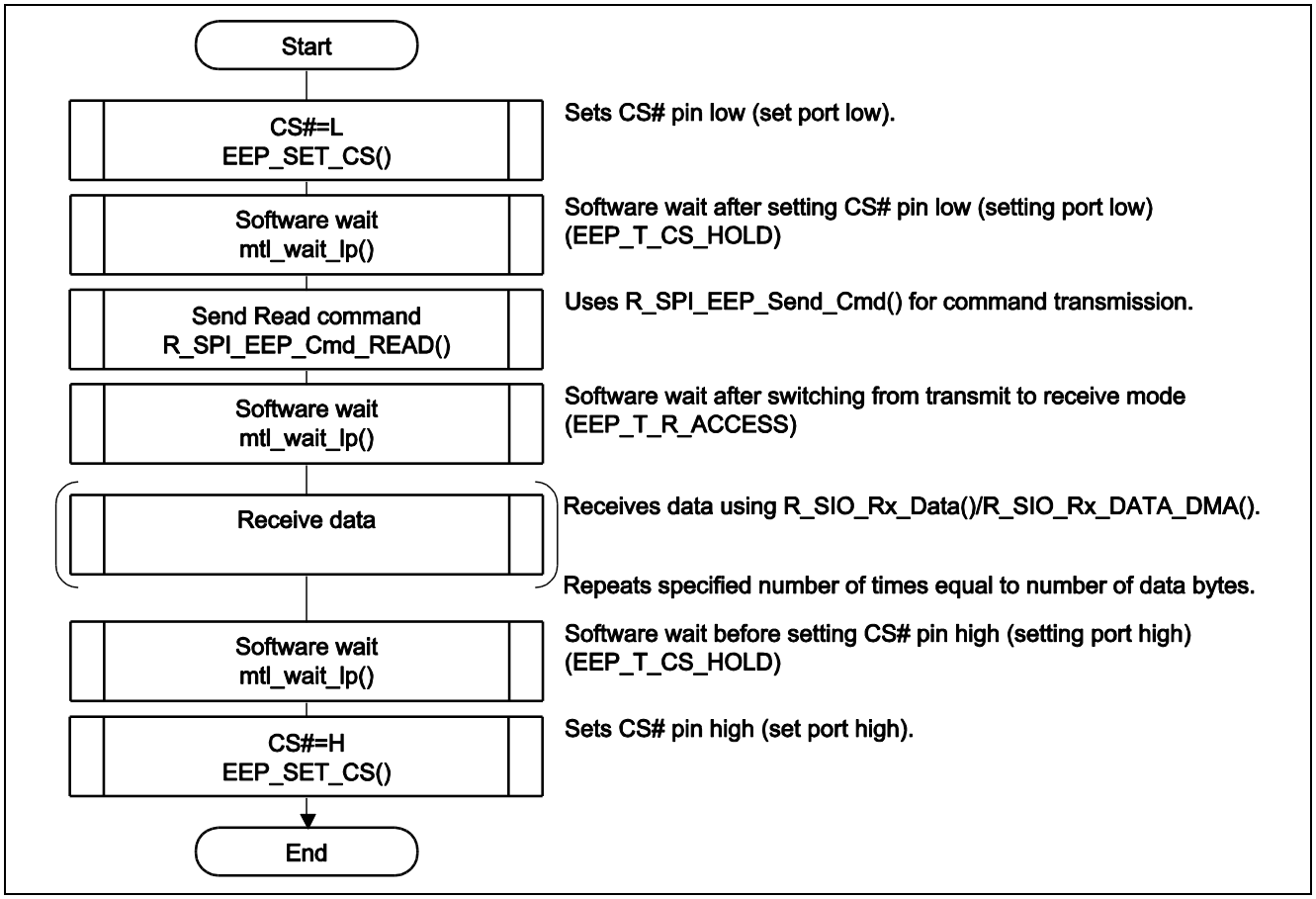


Figure 5.14 Read Command Processing Outline

5.9.10 Write Enable Command Processing

R_SPI_EEP_Write_En	
Synopsis	Performs Write Enable command processing.
Headers	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h
Declaration	STATIC error_t R_SPI_EEP_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. EEP_OK ; Successful operation EEP_ERR_HARD ; Hardware error EEP_ERR_OTHER ; Other error
Remarks	None

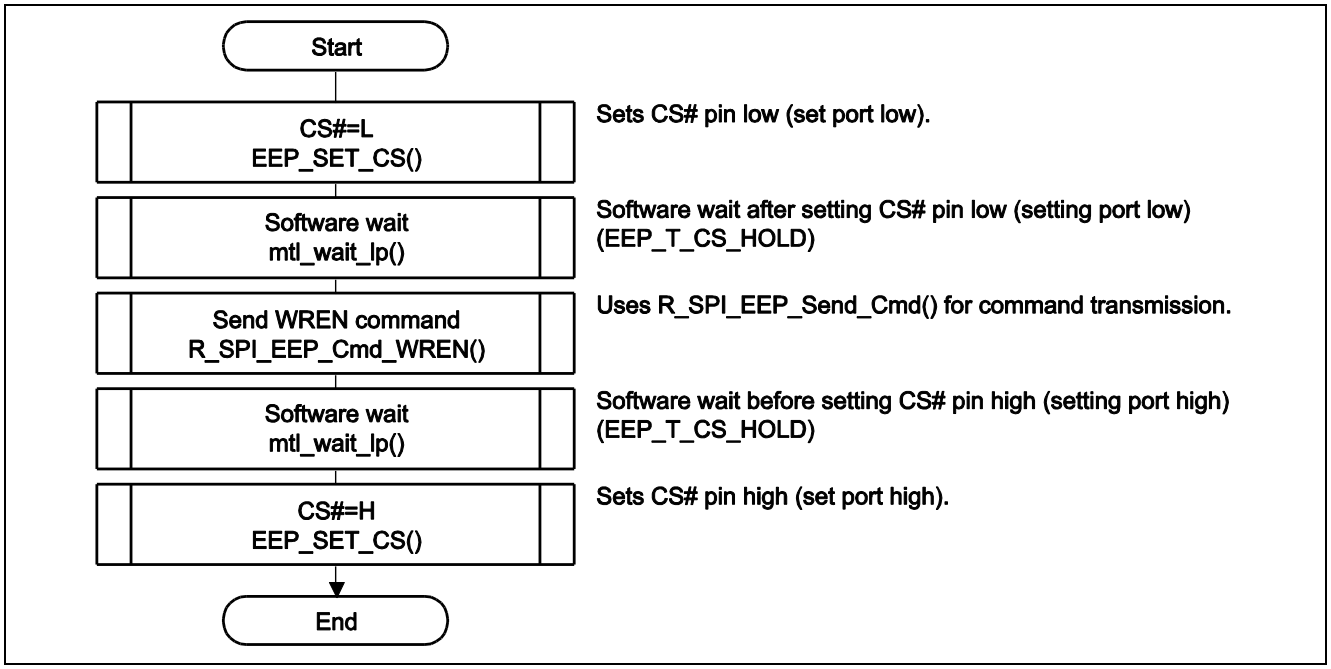


Figure 5.15 Write Enable Command Processing Outline

5.9.11 Write Disable Command Processing

R_SPI_EEP_Write_Di	
Synopsis	Performs Write Disable command processing.
Headers	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h
Declaration	STATIC error_t R_SPI_EEP_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. EEP_OK ; Successful operation EEP_ERR_HARD ; Hardware error EEP_ERR_OTHER ; Other error
Remarks	None

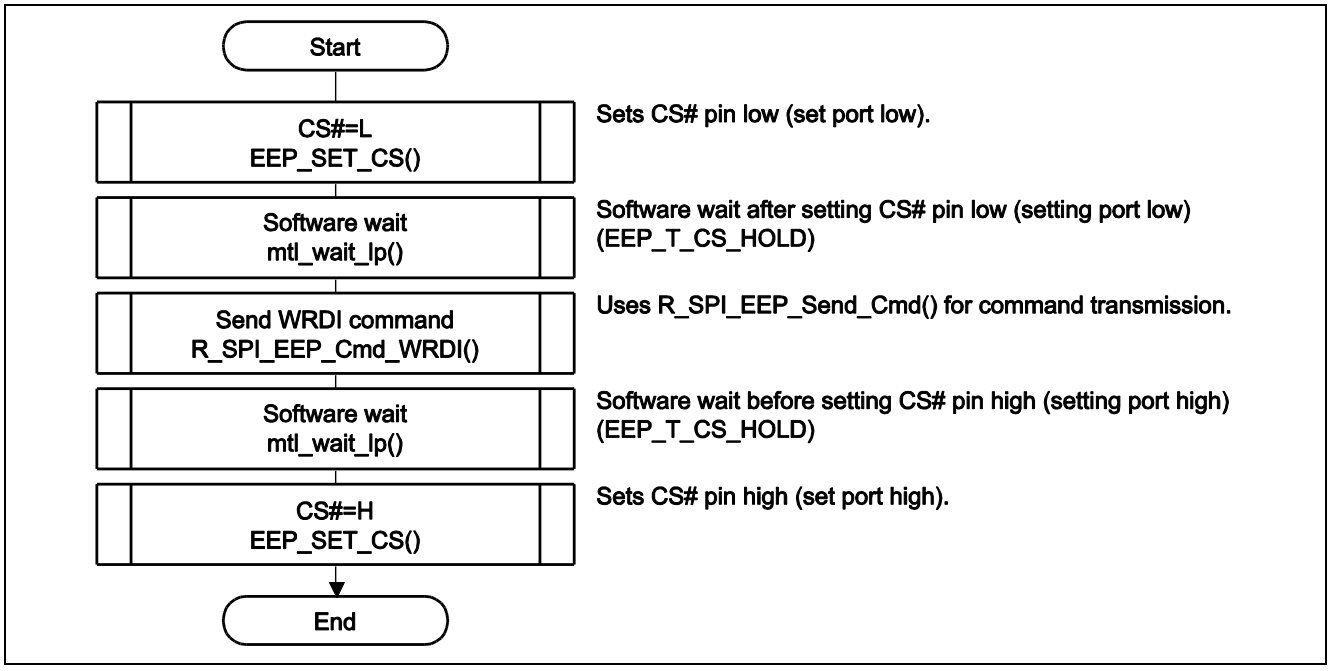


Figure 5.16 Write Disable Command Processing Outline

5.9.12 Read Status Register Command Processing (Internal Function)

R_SPI_EEP_Read_StsReg

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

Headers R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h

Declaration error_t R_SPI_EEP_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:
  - 4 Kbit or less
    - Bit 7 to 4: Reserved (All "1")
    - Bit 3 to 2: BP1, BP0 — 00: No protection; 01: Upper-quarter protection; 10: Upper-half protection; 11: Whole memory protection
    - Bit 1: WEL — 0: Write disabled; 1: Write enabled
    - Bit 0: WIP — 1: During write operation
  - More than 4 Kbit
    - Bit 7: SRWD — 0: Status register can be changed.; 1: Status register cannot be changed.
    - Bit 6 to 4: Reserved (All "0")
    - Bit3 to 2: BP1, BP0 — 00: No protection; 01: Upper-quarter protection; 10: Upper-half protection; 11: Whole memory protection
    - Bit 1: WEL — 0: Write disabled; 1: Write enabled
    - Bit 0: WIP — 1: During write operation

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.

EEP_OK	;	Successful operation
EEP_ERR_HARD	;	Hardware error
EEP_ERR_OTHER	;	Other error

Remarks None

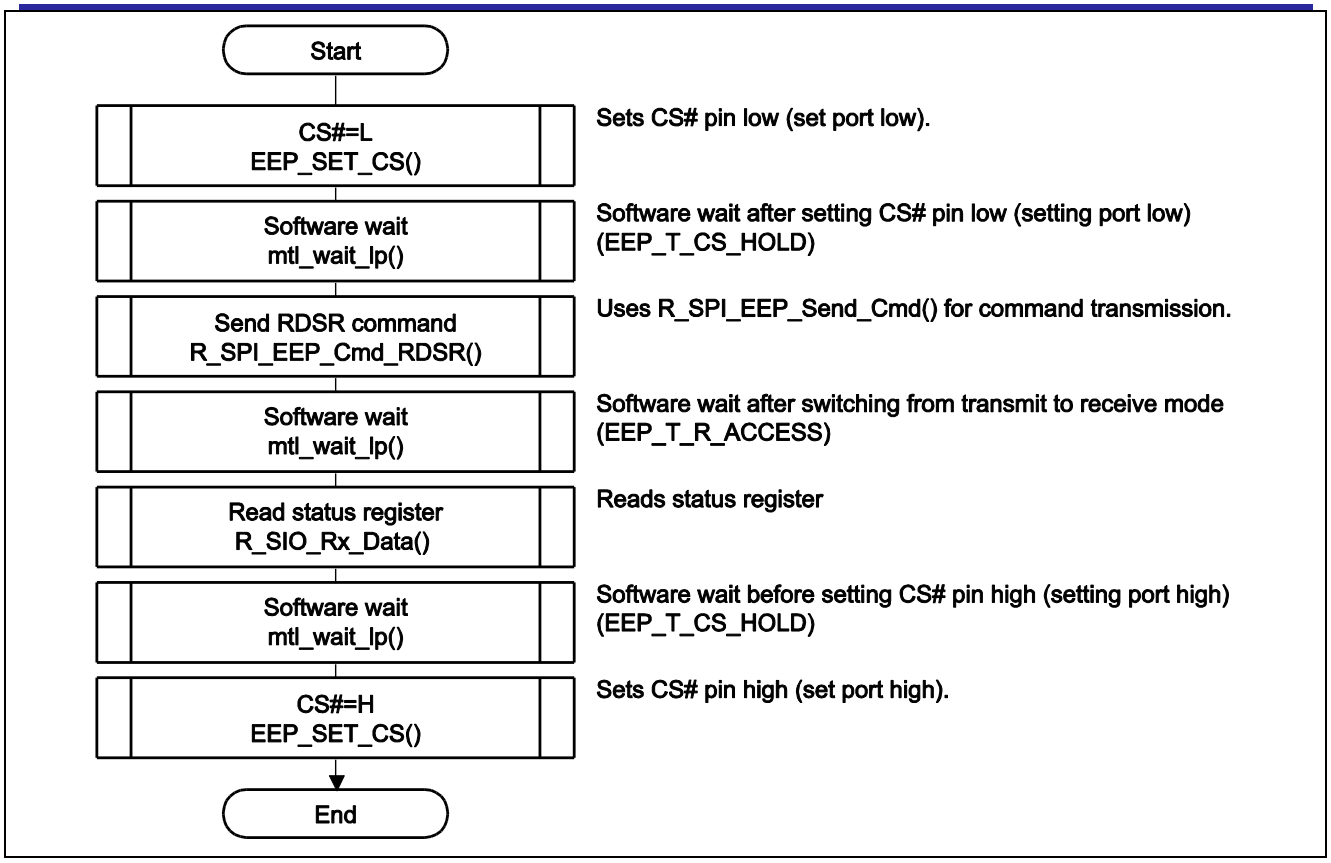


Figure 5.17 Read Status Register Command Processing Outline

5.9.13 Write Status Register Command Processing (Internal Function)

R_SPI_EEP_Write_StsReg

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

Headers R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h

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

Explanation

- Performs enable write command processing (issuing a WREN command) 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:
  - 4 Kbit or less
    Bit 7 to 4: Reserved (All "1")
    Bit 3 to 2: BP1, BP0 00: No protection
    01: Upper-quarter protection
    10: Upper-half protection
    11: Whole memory protection
    Bit 1to 0:Read-only (All "0")
  - More than 4 Kbit
    Bit 7: SRWD 0: Status register can be changed.
    1: Status register cannot be changed.
    Bit 6 to 4: Reserved (All "0")
    Bit3 to 2: BP1, BP0 00: No protection
    01: Upper-quarter protection
    10: Upper-half protection
    11: Whole memory protection
    Bit 1to 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.

EEP_OK	;	Successful operation
EEP_ERR_HARD	;	Hardware error
EEP_ERR_OTHER	;	Other error

Remarks

- The reserved bits of the status register are set to the same value as the read value.

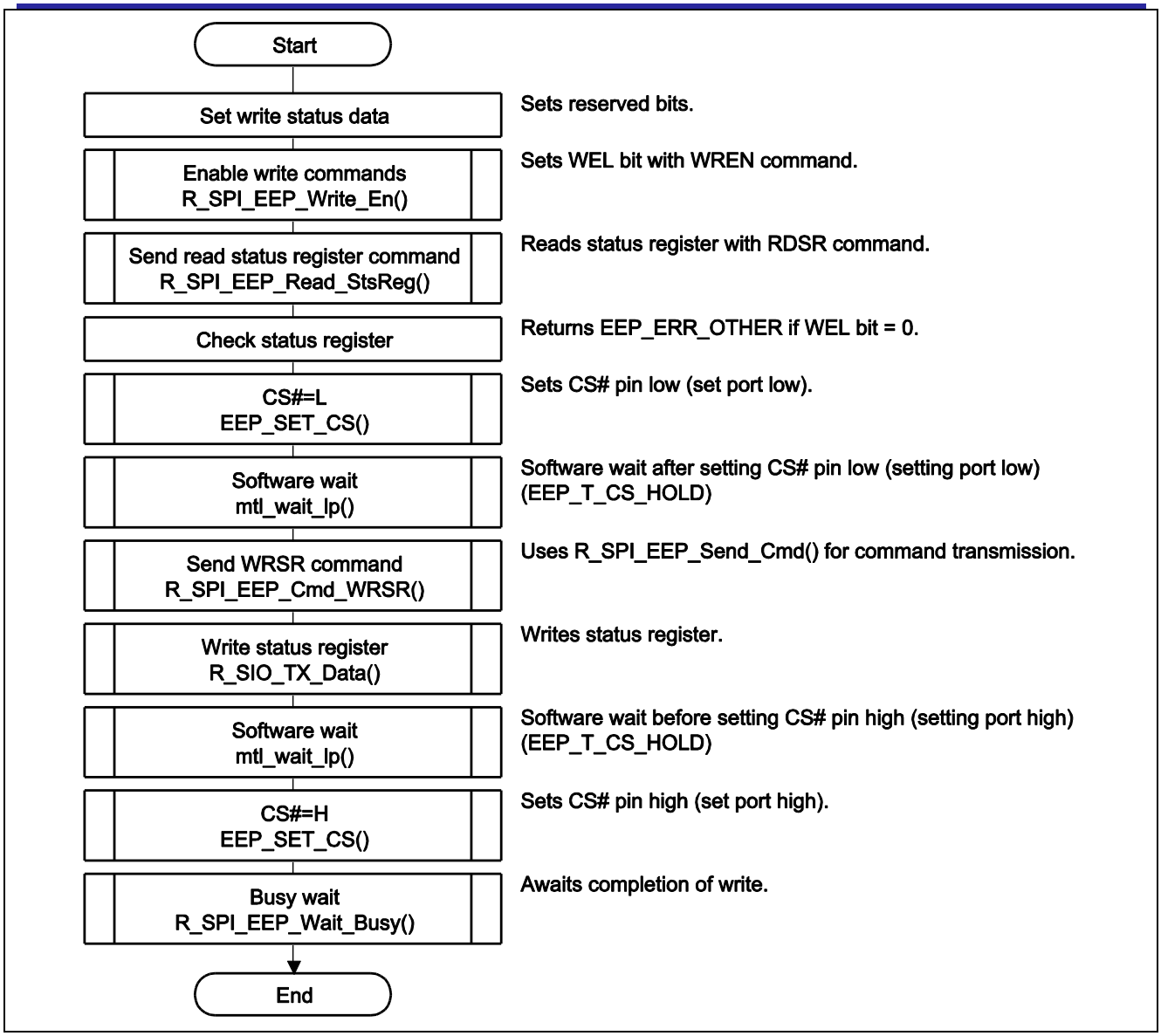


Figure 5.18 Write Status Register Command Processing Outline

5.9.14 Busy Wait Processing (Internal Function)

R_SPI_EEP_Wait_WBusy				
Synopsis	Performs busy wait processing (internal function).			
Headers	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h			
Declaration	STATIC error_t R_SPI_EEP_Wait_WBusy(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. EEP_OK ; Successful operation EEP_ERR_HARD ; Hardware error EEP_ERR_OTHER ; Other error			
Remarks	None			

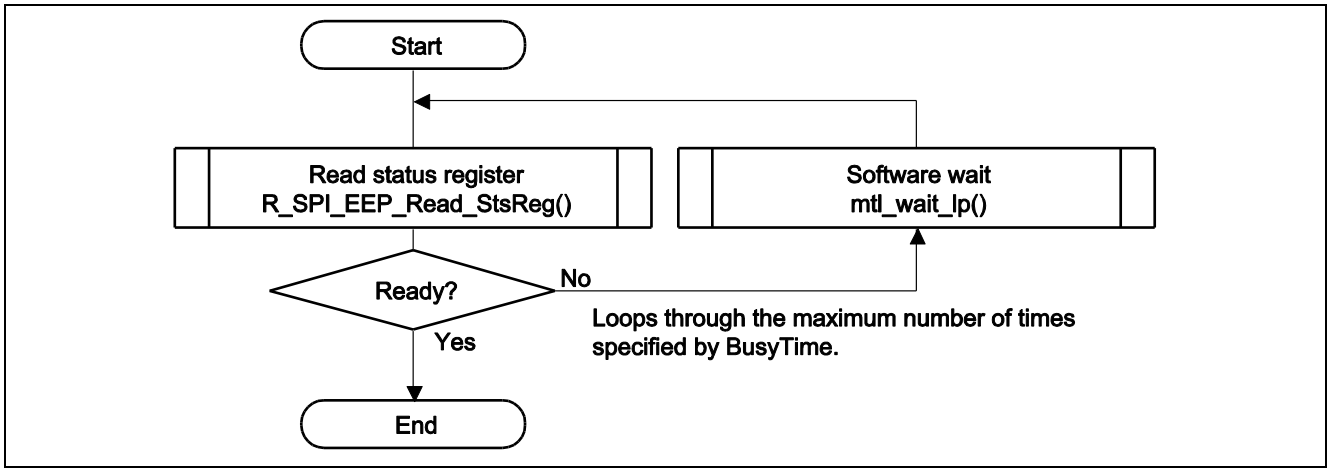


Figure 5.19 Busy Wait Processing Outline

5.9.15 Page Write Command Processing (Internal Function)

R_SPI_EEP_Write_Page	
Synopsis	Performs Page Write command processing (internal function).
Headers	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h
Declaration	error_t R_SPI_EEP_Write_Page(uint8_t DevNo, uint32_t WAddr, uint32_t WCnt, uint8_t FAR* pData)
Explanation	• Writes the specified number of bytes from pData into EEPROM using the WRITE 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. EEP_OK ; Successful operation EEP_ERR_HARD ; Hardware error EEP_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.

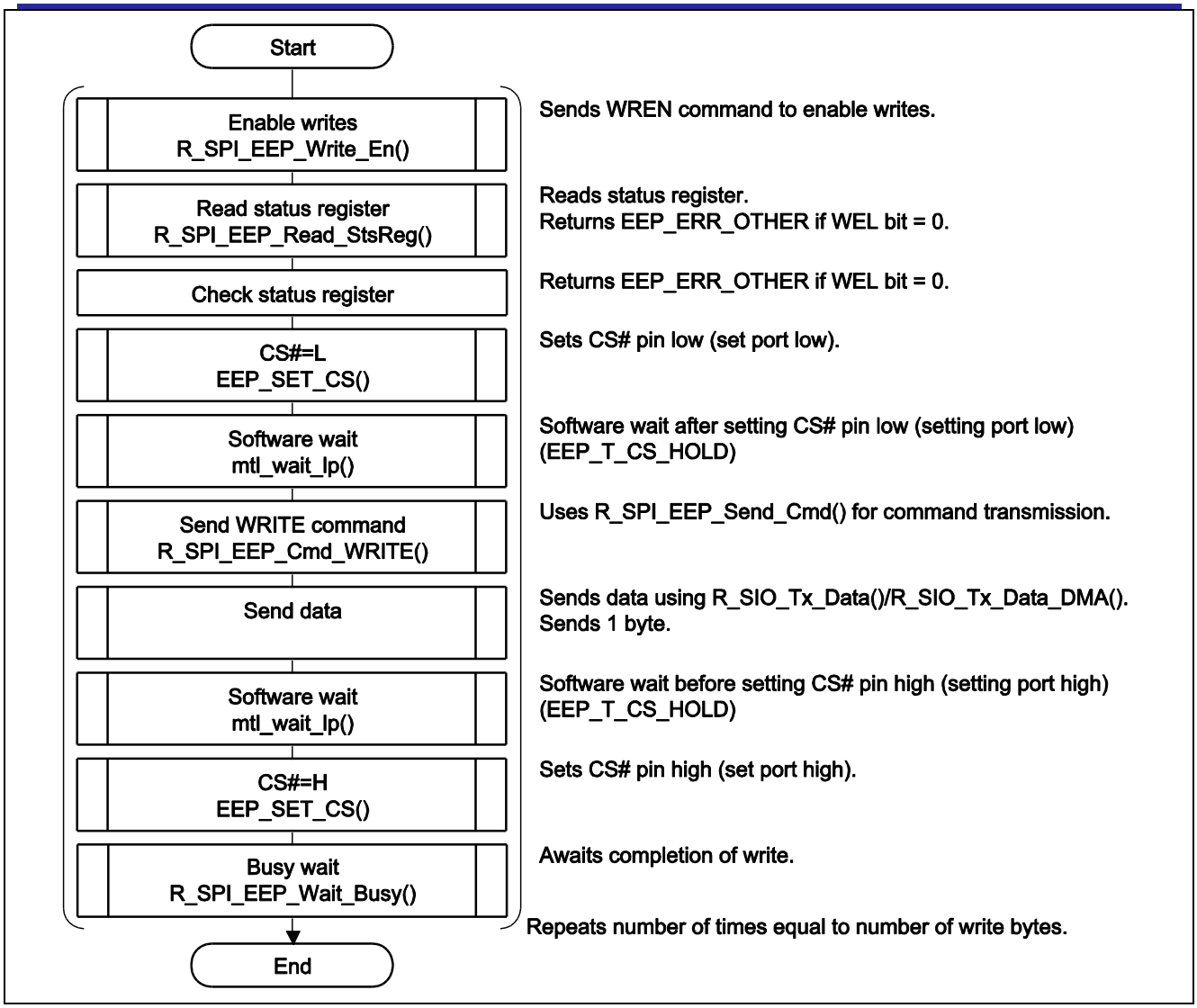


Figure 5.20 Page Write Command Processing Outline

5.9.16 Command Setup Processing (Internal Function)

R_SPI_EEP_Cmd_set				
Synopsis	Sets up command (internal function).			
Headers	R_SPI_EEP.h, R_SPI_EEP_io.h, R_SPI_EEP_sfr.h, R_SIO.h, mtl_com.h			
Declaration	STATIC void R_SPI_EEP_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).			

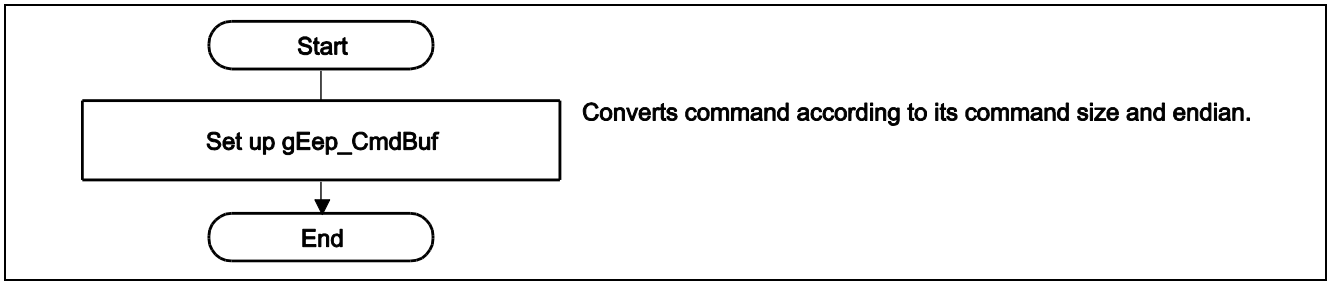


Figure 5.21 Command Setup Processing Outline

6. Application Example

This chapter gives an example of setting up the Serial EEPROM 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 EEPROM Control Software

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

6.1.1 R_SPI_EEP.h

This is a definition file for this Serial EEPROM.

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

/* Define no. of slots */
#define EEP_DEV0           0           /* Device 0             */
#define EEP_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 4 Kbit devices.

```
/*------------------------------------------------------------------------- */
/*  Define the serial EEPROM device.                                        */
/*------------------------------------------------------------------------- */
/*#define EEP_SIZE_002K       */ /*   2kbit  (256 Byte)        */
#define EEP_SIZE_004K            /*   4kbit  (512 Byte)        */
/*#define EEP_SIZE_008K       */ /*   8kbit  (  1kByte)        */
/*#define EEP_SIZE_016K       */ /*  16kbit  (  2kByte)        */
/*#define EEP_SIZE_032K       */ /*  32kbit  (  4kByte)        */
/*#define EEP_SIZE_064K       */ /*  64kbit  (  8kByte)        */
/*#define EEP_SIZE_128K       */ /* 128kbit  ( 16kByte)        */
/*#define EEP_SIZE_256K       */ /* 256kbit  ( 32kByte)        */
/*#define EEP_SIZE_512K       */ /* 512kbit  ( 64kByte)        */
```

6.1.2 R_SPI_EEP_sfr.h

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

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

(1) 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 EEP_DR_CS0    PORT6.DR.BIT.B6    /* EEPROM CS0   (Negative-true logic) */
#define EEP_DDR_CS0   PORT6.DDR.BIT.B6   /* EEPROM CS1   (Negative-true logic) */

#if (EEP_DEV_NUM > 1)
#define EEP_DR_CS1                       /* EEPROM CS1   (Negative-true logic) */
#define EEP_DDR_CS1                      /* EEPROM CS1   (Negative-true logic) */
#endif   /* #if (EEP_DEV_NUM > 1) */
```

(2) 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 EEP_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_EEP_io.c

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

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+	RL78	Not required	Not required
CS+	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_EEP.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 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 History

		Description	
Rev.	**Date**	**Page**	**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_EEP_sfr.78k0r file.
		41	Added “R_SPI_EEP_io.c”.
1.03	May 15, 2012	1	Added “RX62N group (using the RSPI)” and “RX62N group (using the SCI)” in Target Device.
		3	Changed “RX610” to “RX610 SCI” in Conditions for Checking the Operation. Changed description in Software used for evaluation.
		3	Changed “78K0R/Kx3-L” to “78K0R/Kx3-L SAU” in Conditions for Checking the Operation. Changed description in Software used for evaluation.
		4	Added “RX62N RSPI” in Conditions for Checking the Operation.
		4	Added “RX62N SCI” in Conditions for Checking the Operation.
		4	Changed Group name R01AN0534EJ, R01AN0323EJ and R01AN0708EJ in Related Application Notes.
		4	Added an application note for RX62N SCI in Related Application Notes.
		12	Changed “RX610” to “RX610 SCI” in Size of Required Memory.
		12	Changed “78K0R/Kx3-L” to “78K0R/Kx3-L SAU” in Size of Required Memory.
		13	Added “RX62N RSPI” in Size of Required Memory.
		13	Added “RX62N SCI” in Size of Required Memory.
		14	Undated Application Note file name.
		14	Added R_SPI_EEP_sfr.rx62x file.
1.05	Apr 30, 2014	1	Modified Introduction to add short address.
		1	RX63N, RX63T, RX210, RX21A, RX220, RX111 RL78/G1C, RL78/L12, RL78/L13,and RL78/L1C and RL78/G14 added as supported devices.
		4, 12	Added 2.1 RX Family and 2.2 RL78 Family, 78K0R/Kx3-L.
		4 to 11	The following added to section 2.1, Conditions for Checking the Operation of the SPI Serial EEPROM Control Software. (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
12 to 17	The following added to section 2.2, Conditions for Checking the Operation of the SPI Serial EEPROM Control Software. (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	The following added to 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 Group 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	The following added 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	The following added 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(2) RL78/G14 SAU Integrated Development Environment CubeSuite+
		37	5.4 File Configuration Application note numbers changed. Source file folder name changed. New device register common definitions added.
		64	6.1.2 R_SPI_EEP_sfr.h: Content corrected
		65	6.1.3 R_SPI_EEP_io.c: Content corrected
1.06	Mar 31. 2016	14	2.2 RL78 Family, 78K0R/Kx3-L Changed the following conditions. (2) RL78/G14 SAU Integrated Development Environment CS+ for CA,CX (Compiler: CA78K0R) Added the following conditions. (3) RL78/G14 SAU Integrated Development Environment CS+ for CC (Compiler: CC-RL)
		37	5.3.2 RL78 Family, 78K0R/Kx3-L Changed the following sizes. (2) RL78/G14 SAU Integrated Development Environment CS+ for CA,CX (Compiler: CA78K0R) Added the following sizes. (3) RL78/G14 SAU Integrated Development Environment CS+ for CC (Compiler: CC-RL)
		40	5.4 File Configuration Application note numbers changed.

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.

Notice

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

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

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

RENESAS

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

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

Renesas Electronics America Inc.
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