

# **S1C17801**

## **USB Rewriter**

### **Application Note**

When using the commands, follow the instructions of NOTICE\_Application Notes Sample Programs.pdf included in the downloaded compressed file.

## NOTICE

---

No part of this material may be reproduced or duplicated in any form or by any means without the written permission of Seiko Epson. Seiko Epson reserves the right to make changes to this material without notice. Seiko Epson does not assume any liability of any kind arising out of any inaccuracies contained in this material or due to its application or use in any product or circuit and, further, there is no representation that this material is applicable to products requiring high level reliability, such as, medical products. Moreover, no license to any intellectual property rights is granted by implication or otherwise, and there is no representation or warranty that anything made in accordance with this material will be free from any patent or copyright infringement of a third party. This material or portions thereof may contain technology or the subject relating to strategic products under the control of the Foreign Exchange and Foreign Trade Law of Japan and may require an export license from the Ministry of Economy, Trade and Industry or other approval from another government agency.

All other product names mentioned herein are trademarks and/or registered trademarks of their respective companies.

## OVERVIEW

This document is a reference material to download programs in the internal Flash-ROM of the S1C17801 and the external Flash-ROM of the SVT17801, using the USB storage of the S1C17801.

## OPERATING ENVIRONMENT

- S5U1C17801T1100 (Hereafter SVT17801:Software eValuation Tool for S1C17801)  
SVT17801 CPU board and SVT17 ICD board sets
- USB miniB cable
- PC

The GNU17 development tools must be installed.

The USB driver for the SVT17 ICD board must be installed.

The program download tools must be installed.

The USB drivers for program download tools must be installed.

## Table of Contents

<b>1. SPECIFICATIONS .....</b>	<b>1</b>
<b>2. FUNCTIONAL DESCRIPTION .....</b>	<b>2</b>
<b>3. SOFTWARE DESCRIPTION .....</b>	<b>3</b>
<b>3.1 File Configuration .....</b>	<b>3</b>
<b>3.2 Firmware Configuration .....</b>	<b>4</b>
<b>3.3 Memory Map .....</b>	<b>5</b>
<b>3.4 Module Description .....</b>	<b>8</b>
<b>3.5 Global Variables .....</b>	<b>8</b>
<b>3.6 Structures .....</b>	<b>8</b>
<b>3.7 Descriptor .....</b>	<b>9</b>
3.7.1 Device Descriptor .....	9
3.7.2 Configuration Descriptor .....	9
3.7.3 Interface Descriptor .....	9
3.7.4 String Descriptor .....	10
<b>3.8 Vendor Request .....</b>	<b>10</b>
3.8.1 GetRewriterInformation Request .....	10
3.8.2 Flash-ROM Erase Request .....	10
3.8.3 WriteROM Request .....	11
3.8.4 ReadROM Request .....	11
3.8.5 EndDownload Request .....	11
<b>3.9 Operation Procedure .....</b>	<b>11</b>
<b>3.10 Outline of the sample program operations .....</b>	<b>12</b>
<b>3.11 Operation Flowcharts .....</b>	<b>13</b>
3.11.1 Read command operation flow .....	13
3.11.2 Write command operation flow .....	13
<b>3.12 Mode Transition .....</b>	<b>14</b>
3.12.1 During system startup (by power-ON or reset) .....	14
3.12.2 Transition to another mode .....	14
<b>3.13 Flowcharts .....</b>	<b>15</b>
<b>3.14 Header Definitions .....</b>	<b>18</b>
<b>3.15 Definition of a compile option .....</b>	<b>18</b>
<b>REVISION HISTORY .....</b>	<b>19</b>

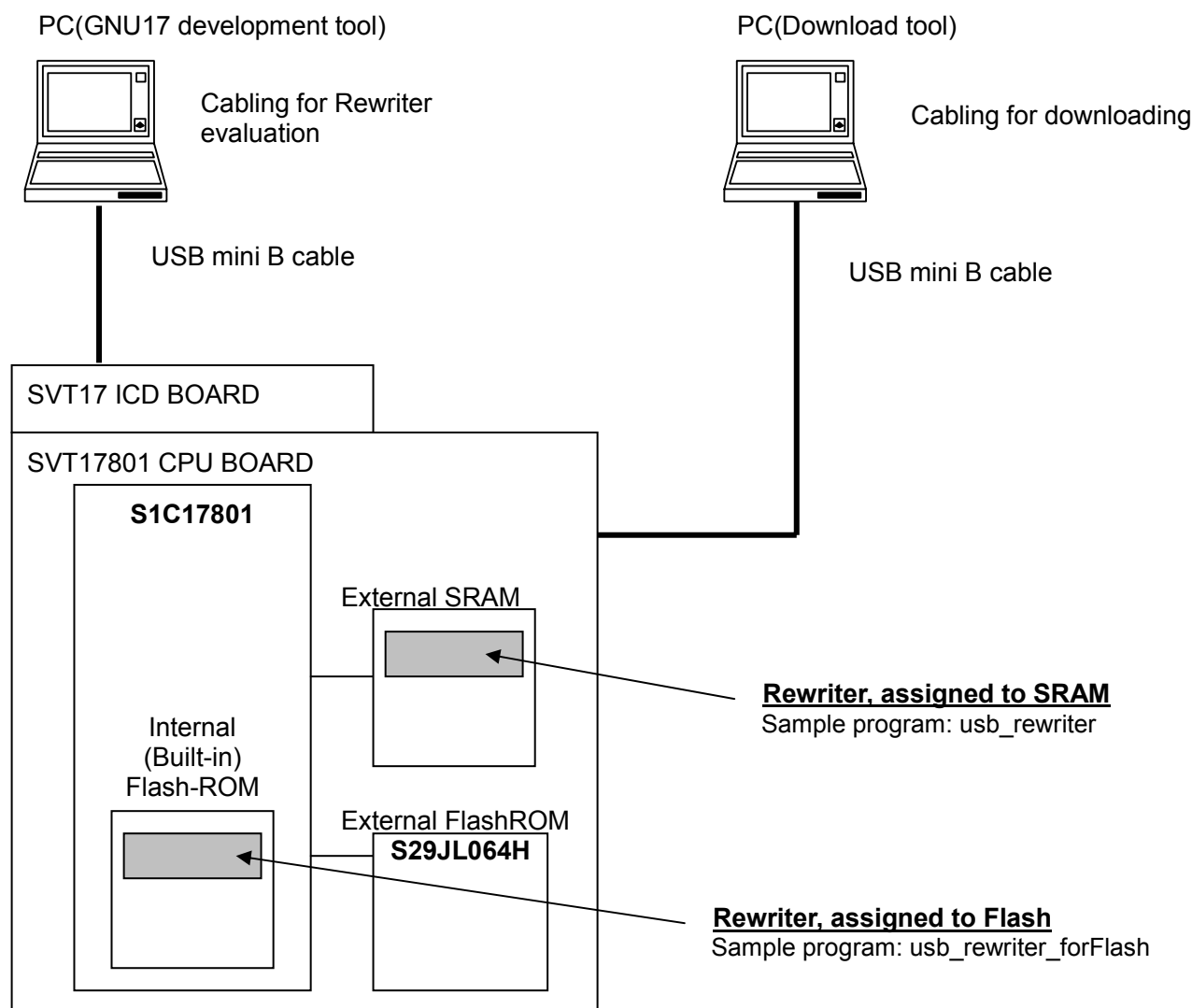
## 1. SPECIFICATIONS

Data can be written or erase into/from the flash memory built in S1C17801 and the Flash ROM (S29JL064H) of SVT17801, via the USB of S1C17801.

When the PC is connected to the S1C17801, the PC recognizes it as the USB Rewriter Device. The download software of the PC is used to write or erase data in/from the Flash memory with the Vendor Request Class commands. Also, data can be read from any area.

The Rewriter jumps to the downloaded ProgramCode when the “End Download” request is received. The external Flash memory (0x100000) is assumed to have the VectorTable. The destination address is obtained from ResetVector of the VectorTable, and control jumps.

The following shows the system connection.



Two sample programs are provided. They are: the program assigned to external SRAM and the program assigned to internal Flash-ROM. For program debugging, use the “program for external SRAM” that can be written any number of times.

## 2. FUNCTIONAL DESCRIPTION

---

## 2. FUNCTIONAL DESCRIPTION

Address            The 128K-byte internal Flash-ROM has address : address 0x20000 to 0x227FF are used to store the Rewriter, and address 0x22800 to 0x3FFFF can be rewritten. (The 96K-byte Flash-ROM has the address up to 0x37FFF.)  
Note that the Rewriter can be rewritten on the ICD only.

### 3. SOFTWARE DESCRIPTION

#### 3.1 File Configuration

File name	Function
boot.c	Startup module
main.c	Main function
inthdlr.c	Interrupt function
vector.c	Vector table settings
classrequest.c	Class-specific command process function
classrequest.h	Class-specific command process definitions
descriptor.c	Descriptor setup function
descriptor.h	Descriptor definitions
Vendorclass.c	Vendor class command process function
Vendorclass.h	Vendor class command process definitions
header\reg_801.h	Register definitions
header\vector.h	Vector table definitions
usb_drv/	USB driver API group folder
flashc_drv/	FLASHC driver API group folder
extflsc_drv/	External Flash memory-driver API group folder
usb_rewriter_gnu17IDE.lids	Linker script file
usb_rewriter_gnu17IDE.cmd	GDB command file
usb_rewriter_gnu17IDE.par	Parameter configuration file
usb_rewriter_gnu17IDE.mak	Make file
.cdtproject	Project file
.gnu17project	Project file
.project	Project file
GDB17 Launch for svt17801_uart.launch	Project startup file

For the USB driver functions, refer to the “Detailed description of USB driver” chapter.

For the FLASHC driver functions, refer to the “Detailed description of FLASHC Driver” chapter.

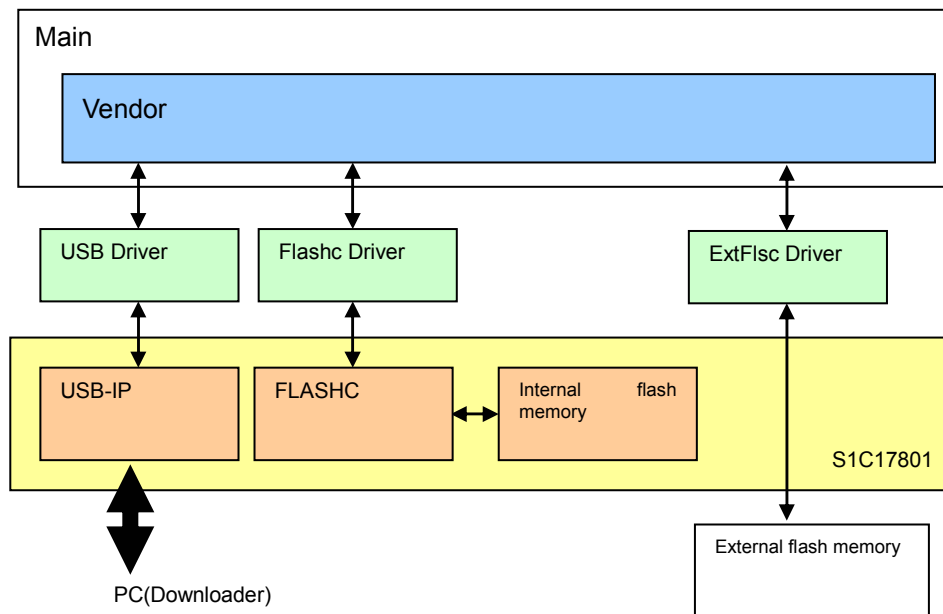
For the external Flash memory driver functions, refer to the “Detailed description of external Flash memory driver” chapter.

### 3. SOFTWARE DESCRIPTION

---

#### 3.2 Firmware Configuration

The following shows the simple and standard configuration of USB Rewriter.



This Rewriter is implemented with the Vendor Class.



### 3.3 Memory Map

The following shows the memory map of the SVT17801.

The rewriteable area of this Rewriter is both internal and external Flash memory.

However, if the Rewriter is already installed in the internal Flash-ROM, no data can be written or erased into/from the Rewriter storage area in order to protect the Rewriter program.

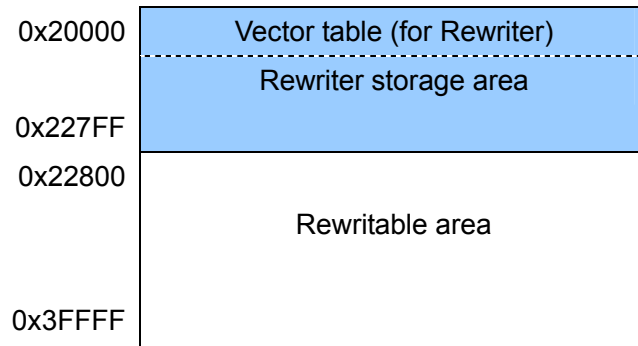
0x000000 to 0x000FFF	IRAM Area(4KByte)
0x001000 to 0x003FFF	Reserved
0x004000 to 0x0043FF	Peripheral Area
0x004400 to 0x005FFF	SAPB Peripheral Area
0x006000 to 0x007FFF	IVRAM Area (2Kbyte)
0x008000 to 0x01FFFF	Reserved
0x020000 to 0x03FFFF	Internal Flash Memory Area (Trap table)
0x040000 to 0x0FFFFFFF	Reserved
0x100000 to 0x8FFFFFFF	SRAM#CE0 (8MByte) External NOR Flash Area
0x900000 to 0xCFFFFFFF	SRAM#CE1 (4MByte) External SRAM Area
0xD00000 to 0xEFFFFFFF	SRAM#CE2 (2MByte) External I/F Area
0xF00000 to 0xFEFFFF	SRAM#CE3 (1MByte) Reserved
0xFF0000 to 0xFFFBFF	C17 IROM Area (Reserved)
0xFFFC00 to 0xFFFFFFF	C17 CPU core reserved I/O

### 3. SOFTWARE DESCRIPTION

---

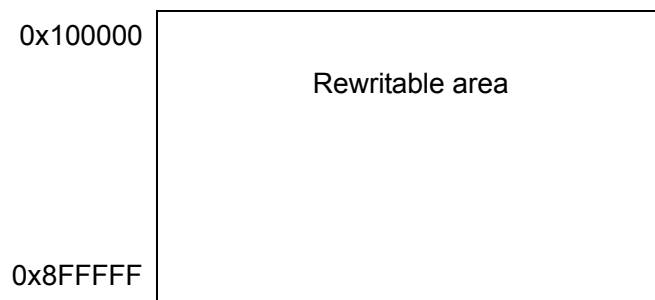
#### • Internal Flash memory

The internal Flash memory contains the vector table and main part of the Rewriter.  
Any data can be written or erased into/from the other areas.



#### • External Flash memory

In the entire area of external Flash memory, data can be written and erased by the Rewriter (Flash version).

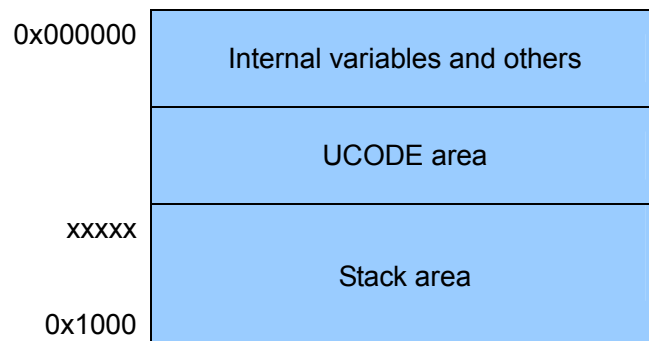


#### • IRAM

Used as a storage for variables of Rewriter and as the stack area.

Flash memory (erase/write) control codes are expanded in the UCODE area.

This prevents a malfunction of Flash Controller that fails if the operation codes are contained in the Flash memory when data is written in or erased from internal Flash memory.

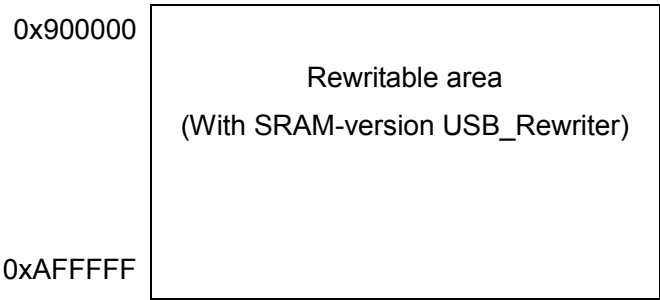


#### • IVRAM

Not used for the Rewriter

• External SRAM

The entire area of memory can be erased or overwritten for the rewriter in SRAM version.



### 3. SOFTWARE DESCRIPTION

#### 3.4 Module Description

File name : main.c

Function name	Function
Main	Initializes the USB memory and starts the vendor class process.
SnoozeUsb	Enables the snooze control.
ActiveUsb	Starts the USB controller operation.
USBIntEnable	Enables the USB interrupt.
USBIntDisable	Disables the USB interrupt.

File name : classrequest.c

Function name	Function
ResetFunc	Reset Request process
RemoteFunc	Remote Wakeup process
ClassFunc	Class-specific command process

File name : vendorclass.c

Function name	Function
VendorInit	Initializes the vendor class process.
VendorMain	The Main process in response to the vendor-class command request
VendorFunc	Command check of USB packet
UCODECopy	Expansion process of Flash memory control codes in RAM
UCODEClear	Clear process of Flash memory control codes expanded in RAM

#### 3.5 Global Variables

The following shows the global variables used in the sample program.

Variable name	Type	Function
ExitDownloader		

#### 3.6 Structures

The following describes the structure used in the sample program.

Definition name		
T_USB_CFG		
<b>Members</b>		
descriptorData	Unsigned char *	Descriptor data pointer
descriptorSize	unsigned short	Descriptor size
NegotiationMode	unsigned char	Auto negotiation mode
AddressMode	unsigned char	Auto address reply mode
ManagementMode	unsigned char	BUS processing mode of USB
ActiveFunc	Void (*)(void)	Active clock function
SnoozeFunc	Void (*)(void)	Snooze clock function
ResetFunc	Void (*)(void)	Reset process function
remoteFunc	int (*)(void)	RemoteWakeup Switch function
classFunc	int (*)(T_USB_REQUEST_PACKET*)	Class request function
vendorFunc	int (*)(T_USB_REQUEST_PACKET*)	Vendor request function
intEnableFunc	Void (*)(void)	USB Interrupt enable function
intDisableFunc	Void (*)(void)	USB Interrupt disable function
<b>Remarks</b>		
USB driver initialization structures		

For the other structures, refer to the “Detailed description of USB driver” Chapter.

### 3.7 Descriptor

The following lists the Rewriter descriptors.

#### 3.7.1 Device Descriptor

Offset	Field	Size	Value	Description
0	bLength	1	12h	Descriptor size (18 bytes)
1	bDescriptorType	1	01h	Descriptor type (Device)
2	bcdUSB	2	0110h	Release number of USB standard (USB1.1)
4	bDeviceClass	1	00h	Device class code
5	bDeviceSubClass	1	00h	Device subclass code
6	bDeviceProtocol	1	00h	Device protocol code
7	bMaxPacketSize0	1	40h	Maximum packet size (64 bytes) at endpoint 0
8	idVendor	2	04B8h	Vendor ID (SEIKO EPSON) * 1
10	idProduct	2	0509h	Product ID (General firmware downloader)
12	bcdDevice	2	0100h	Device release number
14	iManufacturer	1	01h	String descriptor index that identifies the manufacturer
15	iProduct	1	02h	String descriptor index that identifies the product name
16	iSerialNumber	1	03h	String descriptor index that identifies the serial number
17	bNumConfiguration	1	01h	Number of devices that can be configured

#### 3.7.2 Configuration Descriptor

Offset	Field	Size	Value	Description
0	bLength	1	09h	Descriptor size (9 bytes)
1	bDescriptorType	1	02h	Descriptor type (Configuration)
2	wTotalLength	2	0012h	Total length of return data (18 bytes)
4	bNumInterfaces	1	01h	Number of supported interfaces
5	bConfigurationValue	1	01h	Selection value of this configuration
6	iConfiguration	1	00h	String descriptor index
7	bmAttributes	1	C0h	Configuration attributes
8	MaxPower	1	20h	Maximum power consumption (40 mA)

#### 3.7.3 Interface Descriptor

Offset	Field	Size	Value	Description
0	bLength	1	09h	Descriptor size (9 bytes)
1	bDescriptorType	1	04h	Descriptor type (Interface)
2	bInterfaceNumber	1	00h	Interface number
3	bAlternateSetting	1	00h	Selection value of alternate settings
4	bNumEndpoints	1	00h	Number of endpoints used for this interface
5	bInterfaceClass	1	FFh	Interface class code
6	bInterfaceSubClass	1	FFh	Interface subclass code
7	bInterfaceProtocol	1	FFh	Interface protocol code
8	iInterface	1	00h	String descriptor index that identifies the interface

### 3. SOFTWARE DESCRIPTION

#### 3.7.4 String Descriptor

Offset	Field	Size	Value	Description
0	bLength	1	??h	Descriptor size (?? bytes)
1	bDescriptorType	1	03h	Descriptor type (String)
2	bString	?	?	String coded in UNICODE system

\*String1 : "EPSON" \*1

\*String2 : "EPSON USB REWRITER" \*1

\*String3 : "Rev.0.01" \*1

\*1 : EPSON is a name used for tentative registration. Be sure to change this name into the pertinent maker's name on your side.

### 3.8 Vendor Request

#### 3.8.1 GetRewriterInformation Request

This request is used to obtain the Rewriter information and to determine whether the current Rewriter is supported by the Downloader tool or not.

bmRequest Type	bRequest	wValue	wIndex	wLength	Data stage
11000000b	FFh	not significant	not significant	36	IN: Rewriter ID information

The following shows the information in the Data Stage.

Byte position	Length	Contents	Description
0	36	"EPSON USB REWRITER" in Unicode	Codes of the left characters

#### 3.8.2 Flash-ROM Erase Request

This request passes the data erase address in the Data Stage. Data between the passed start address and the end address is erased.

bmRequest Type	bRequest	wValue	wIndex	wLength	Data stage
010000000b	FDh	Not significant	Not significant	8	OUT: Data erase address

The following shows the information in the Data Stage.

Byte position	Length	Description
0	4	Erase start address
4	4	Erase end address

When addresses are specified, data is erased in every data erase unit (in units of blocks) according to the properties of the connected Flash-ROM.

For the block definitions, refer to the Data Sheets of the connected Flash-ROM.

Data is erased from every blocks as follows:

- (1) The block of data that matches the erase start address and erase end address is erased.
- (2) The erase start address and the erase end address must be the outside of Rewriter storage area.

Data is erased from the Flash-ROM only when the data transaction of this request has completed. Because the next request is waited until the data erasure completes, it may take several seconds to succeed data erasure. If data erasure from Flash-ROM has failed, a Request error is returned to the next request.

#### 3.8.3 WriteROM Request

This request is used to write the “wLength” length of data in the specified addressed.

bmRequest Type	bRequest	wValue	wIndex	wLength	Data stage
01000000b	FEh	Address (Low)	Address (High)	Data length (in bytes)	OUT: Write data

This request is valid for internal and external Flash-ROMs only.

Note that the request should be issued for an internal Flash-ROM area outside of the Rewriter storage area.

Data is written for each data packet one by one. When data is received with a data packet, the data contained in this packet is written at a time.

If data writing has failed, the Rewriter returns a Request Error. (As data is transferred in units of 64 bytes for Data Fase, every 64 bytes of data is written.)

It may take a time to write data in the Flash-ROM. Therefore, the successful time of the next data packet writing after the currently succeeded data packet is undefined.

#### 3.8.4 ReadROM Request

This request is used to read the “wLength” length of data from the specified addressed.

bmRequest Type	bRequest	wValue	wIndex	wLength	Data stage
11000000b	FCh	Address (Low)	Address (High)	Data length	IN: Read data

#### 3.8.5 EndDownload Request

This is a request for termination of the Rewriter operation.

When this request is received, the USB cable disconnection is waited.

When the USB cable is disconnected, control jumps to the beginning of the external Flash-ROM (address 0x100000) based on the VectorTable information.

bmRequest Type	bRequest	wValue	wIndex	wLength	Data stage
01000000b	FBh	Reserved (=00h)	Reserved (=00h)	No (0 bytes)	No

### 3.9 Operation Procedure

#### Import the project

(1) Start the IDE, and import the “usb\_rewriter” project.

\* For the import procedure, refer to S5U1C17001C Manual “3. Software Development Procedure.”

\* Copy required drivers from the driver folder.

#### Build

(1) Build the “usb\_rewriter” project using the IDE.

#### Cabling and power-on

(1) Connect SVT17801, USB miniB cable and PC.

(2) Reset SVT17 ICD board.

#### Execution

(1) Execute the “usb\_rewriter” project using the IDE.

(2) Connect the Flash-ROM to the PC and make sure that the memory is recognized as a USB device.

When the Flash-ROM is connected to the PC in the first time, its drivers are requested to install. Specify

### 3. SOFTWARE DESCRIPTION

---

the “DLUsb.inf” and “DLUsb.sys” driver files.

(3) Execute the “Downloader.exe” file on the PC, and execute the following commands.

- (1) Download the USB firmware.
  - The contents of the file specified by FlashROM are downloaded.
  - The supported file format is Motorola-S3.
- (2) Read data from the Flash-ROM.
  - The FlashROM data is read from the USB device, and it is written in the specified file.
  - The file format should be as specified. (The data write file name must be specified as “\*.mot”.)
  - The data read address range should be as specified.
- (3) Read data from RAM area.
  - Data is read from the RAM, and it is written in the specified file.
  - The file format should be as specified. (The data write file name must be specified as “\*.mot”.)
- (4) Transition to the Normal mode.
  - The USB device is transitioned to the Normal mode.

\* Check for data writing

After data has been written in the Flash-ROM, the written data is checked and, if it is OK, the data writing operation terminates. If data was not written correctly, the data must be written in this area again.

The “Downloader.exe” checks whether the data written in the Flash-ROM by the ReadROM request is correct or not. If an error is found, a Request error is returned immediately.

For details about the “Downloader.exe” file, refer to the “USB Downloader Specifications”.

#### 3.10 Outline of the sample program operations

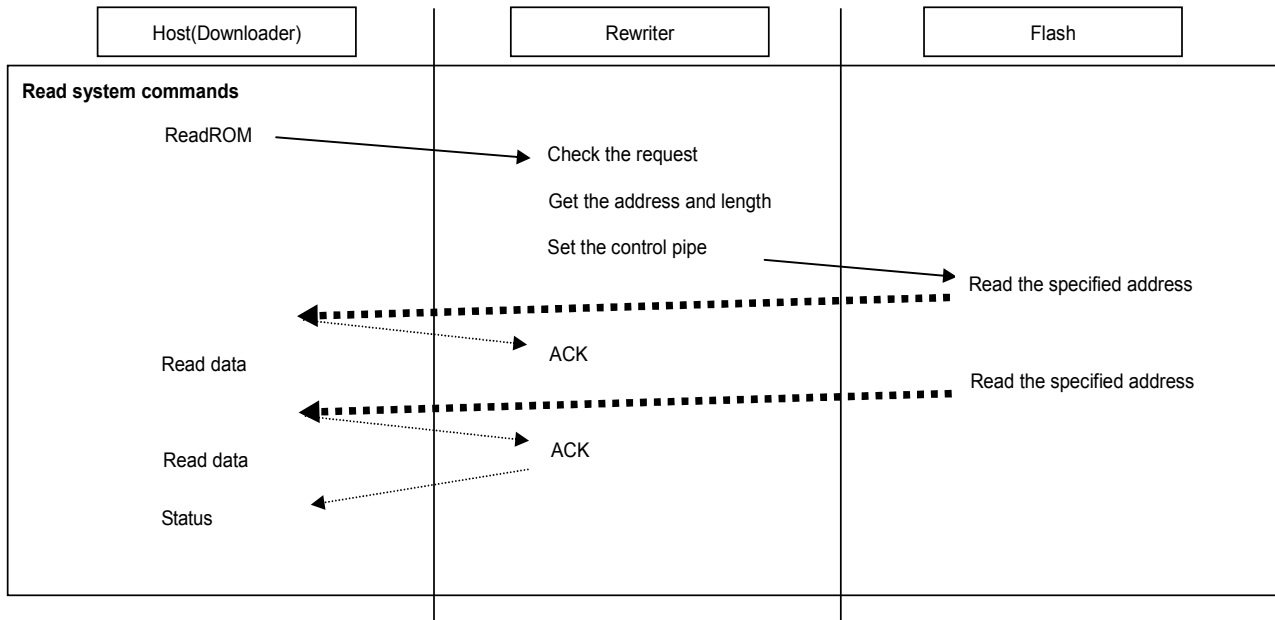
- (1) Initialize the GPIO (ADD, DATA), SRAMC and CMU.
- (2) Check if the VBUS state has changed or not.
- (3) Check the VBUS state change again, and if it has changed and if the program is written, exit the Rewriter mode and jump to the “boot”. Otherwise, go to Step (4).
- (4) Set the ITC for USB memory.
- (5) Enable a CPU interrupt.
- (6) Initialize the USB drivers.
- (7) Initialize the VendorClass.
- (8) Wait for the following processes:
  - a. Check for a USB interrupt.
  - b. Process according to the USB bus state
  - c. Execute the Main process of the VendorClass. (USB Rewriter process)



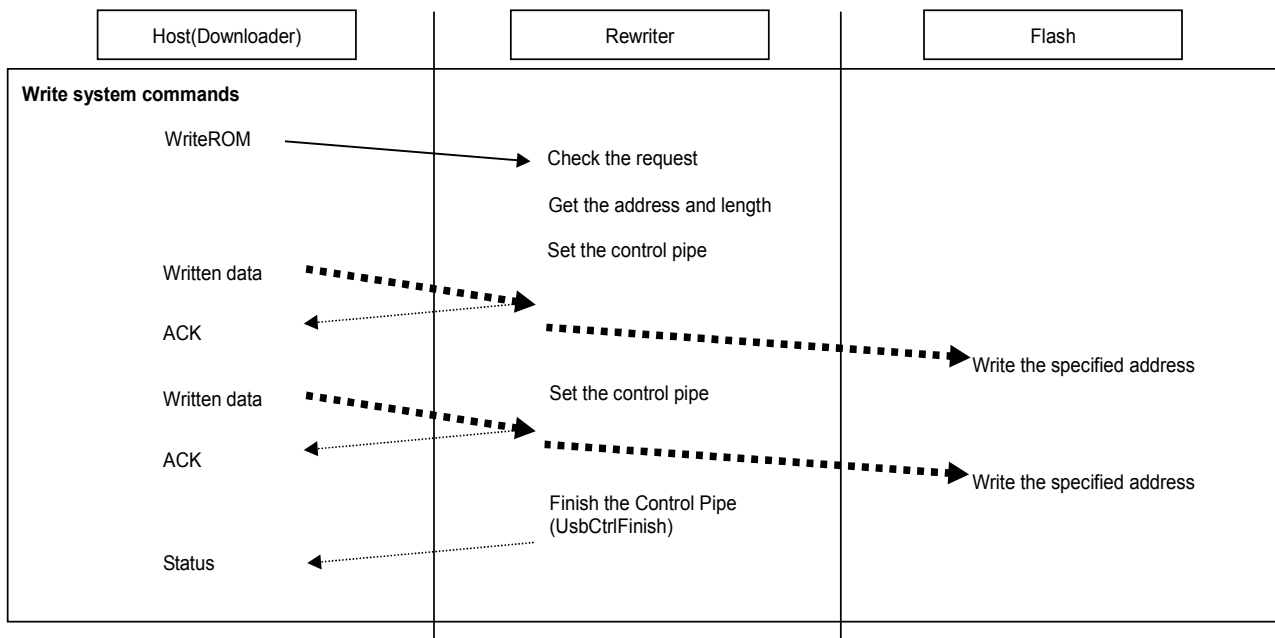
## 3.11 Operation Flowcharts

The following outlines the system operation image.

### 3.11.1 Read command operation flow



### 3.11.2 Write command operation flow



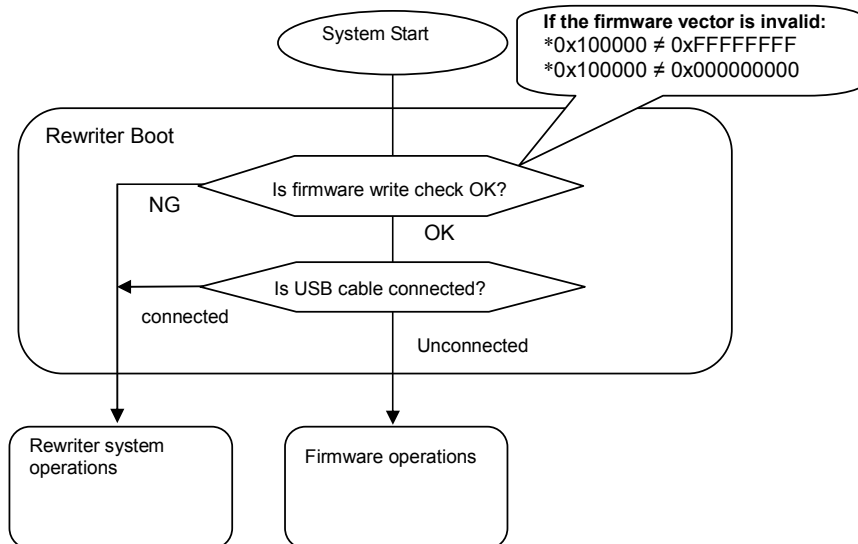
### 3. SOFTWARE DESCRIPTION

---

#### 3.12 Mode Transition

##### 3.12.1 During system startup (by power-ON or reset)

When the Rewriter is booted, it operates as follows.



As illustrated above, the Rewriter operates:

- (1) If the firmware is not written, or
- (2) If the firmware is written and the USB cable is connected.

##### 3.12.2 Transition to another mode

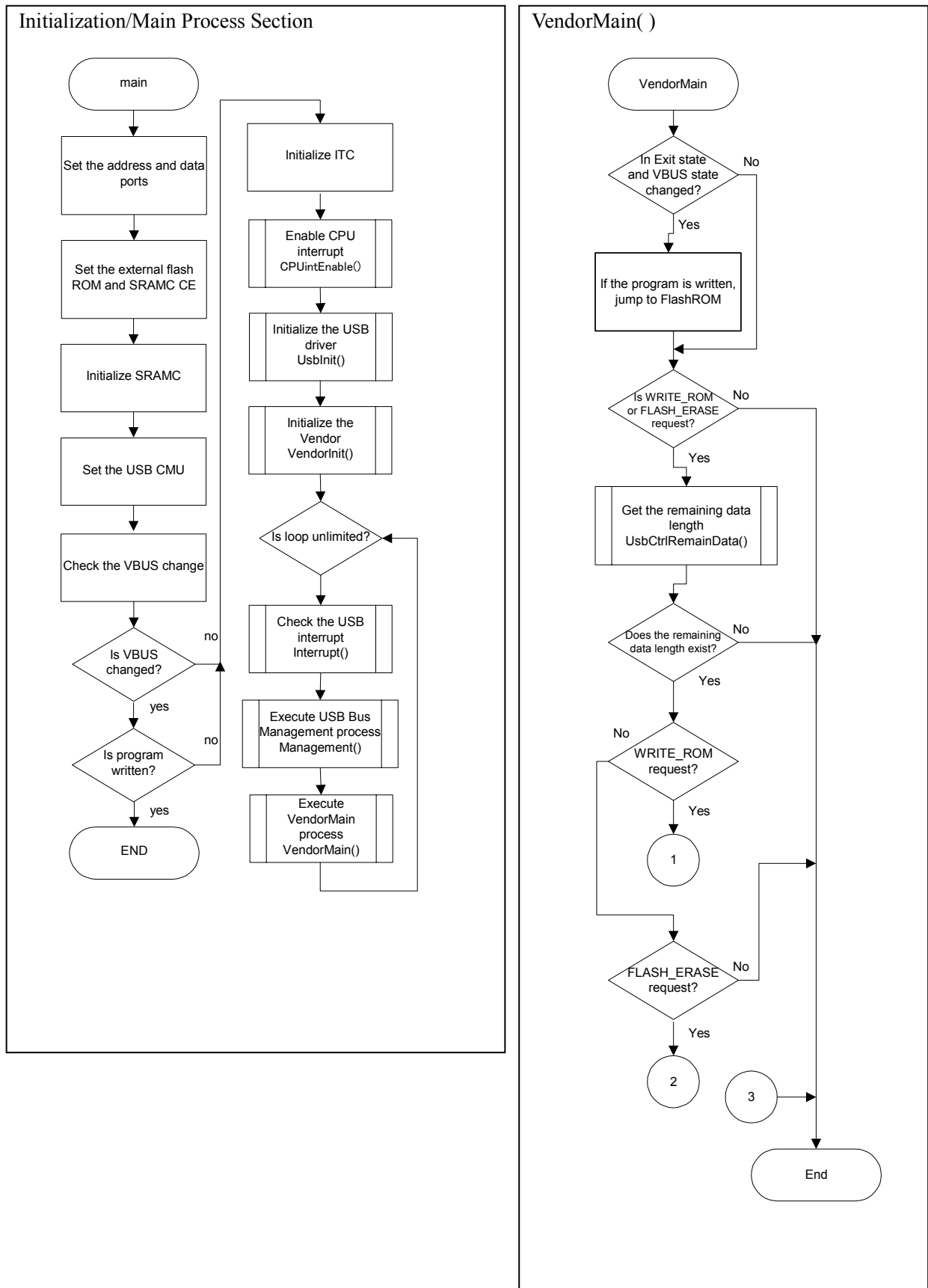
The Rewriter operation is transitioned to another mode when the “EndDownload” request is received and when the USB cable is disconnected.

When the power supply is turned off and on again, or when the system is reset, control is lost if the SystemStart conditions are not satisfied.

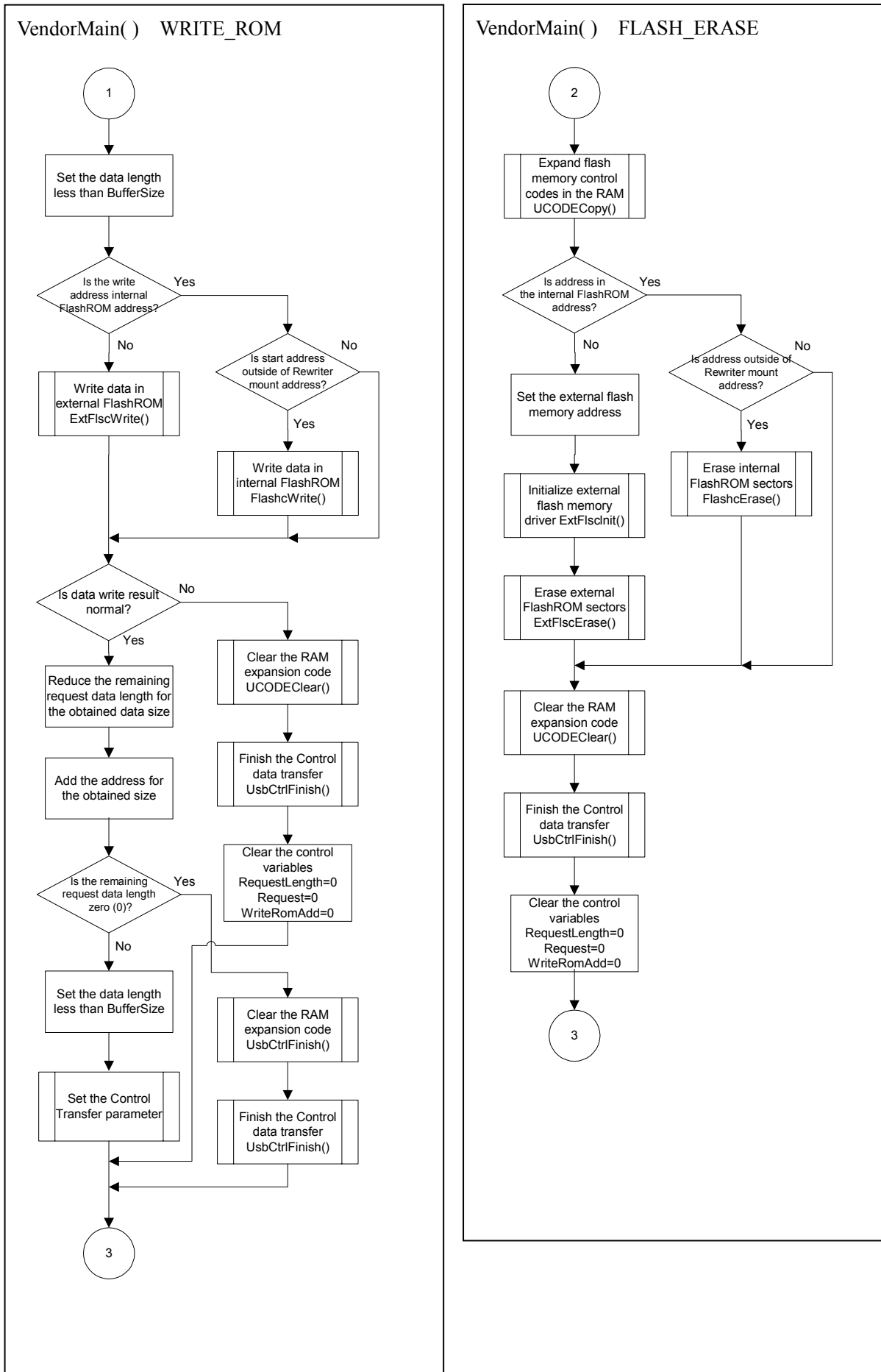
- \* If the Vector Table of the firmware is determined to be invalid, the Rewriter continues operation even when the “EndDownload” request is received.
- \* The Vector Table of the firmware means the external Flash memory (address 0x100000).

## 3.13 Flowcharts

The following shows the flowchart of the main routine and interrupt handler functions.



### 3. SOFTWARE DESCRIPTION





### 3. SOFTWARE DESCRIPTION

---

#### 3.14 Header Definitions

The tables below show the definitions used in the driver functions.

Definition name	Value	Description
USB_CONFIG_NUM	1	Number of configurations
USB_INTERFACE_NUM	1	Number of interfaces
GET_REWRITER_INFO	0xFF	Rewriter information acquisition command number
END_DOWNLOAD	0xFB	Download termination command number
WRITE_ROM	0xFE	WriteROM command number
READ_ROM	0xFC	ReadROM command number
FLASH_ERASE	0xFD	Flash-ROM erase command number

#### 3.15 Definition of a compile option

An unused function can be enabled or disabled by a compile option (Symbol definition).  
As the unused functions are disabled by default, it should be noted during compiling.

##### Symbol List

Definition name	Description
FOR_CLEAR_FEATURE	Enables ClearFeature of the Standard request.
FOR_SET_FEATURE	Enables SetFeature of the Standard request.
FOR_GET_INTERFACE	Enables GetInterface of the Standard request.
FOR_SET_INTERFACE	Enables SetInterface of the Standard request.
FOR_CLASS_REQUEST	Enables the Class request.
FOR_TEST_MODE	Enables TestMode.
FOR_BULK_TRANSFER	Enables the Bulk Transfer process.
FOR_NEGOTIATION	Enables the manual negotiation process.
FOR_ADDRESS	Enables the manual SetAddress process.
FOR_MANAGEMENT	Enables the manual management of USB processing.
FOR_REMOTEWAKEUP	Enables the RemoteWakeup function.
FOR_SNOOZE_OFF	Enables the Snooze OFF function.
FOR_USB_MINIMUM_API	Makes the minimum required API configuration.

## REVISION HISTORY

[illegible]

### AMERICA

---

**EPSON ELECTRONICS AMERICA, INC.****HEADQUARTERS**

2580 Orchard Parkway  
San Jose, CA 95131, USA  
Phone: +1-800-228-3964      FAX: +1-408-922-0238

**SALES OFFICES****Northeast**

301 Edgewater Place, Suite 210  
Wakefield, MA 01880, U.S.A.  
Phone: +1-800-922-7667      FAX: +1-781-246-5443

### EUROPE

---

**EPSON EUROPE ELECTRONICS GmbH****HEADQUARTERS**

Riesstrasse 15 Muenchen Bayern,  
80992 GERMANY  
Phone: +49-89-14005-0      FAX: +49-89-14005-110

### ASIA

---

**EPSON (CHINA) CO., LTD.**

7F, Jinbao Bldg., No. 89 Jinbao St.,  
Dongcheng District,  
Beijing 100005, China  
Phone: +86-10-6410-6655      FAX: +86-10-6410-7320

**SHANGHAI BRANCH**

7F, Block B, Hi-Tech Bldg., 900, Yishan Road,  
Shanghai 200233, CHINA  
Phone: +86-21-5423-5522      FAX: +86-21-5423-5512

**EPSON HONG KONG LTD.**

20/F., Harbour Centre, 25 Harbour Road  
Wanchai, Hong Kong  
Phone: +852-2585-4600      FAX: +852-2827-4346  
Telex: 65542 EPSCO HX

**EPSON (CHINA) CO., LTD.****SHENZHEN BRANCH**

12/F, Dawning Mansion, Keji South 12th Road,  
Hi-Tech Park, Shenzhen  
Phone: +86-755-2699-3828      FAX: +86-755-2699-3838

**EPSON TAIWAN TECHNOLOGY & TRADING LTD.**

14F, No. 7, Song Ren Road,  
Taipei 110  
Phone: +886-2-8786-6688      FAX: +886-2-8786-6660

**EPSON SINGAPORE PTE., LTD.**

1 HarbourFront Place,  
#03-02 HarbourFront Tower One, Singapore 098633  
Phone: +65-6586-5500      FAX: +65-6271-3182

**SEIKO EPSON CORPORATION****KOREA OFFICE**

50F, KLI 63 Bldg., 60 Yoido-dong  
Youngdeungpo-Ku, Seoul, 150-763, KOREA  
Phone: +82-2-784-6027      FAX: +82-2-767-3677

**GUMI OFFICE**

2F, Grand B/D, 457-4 Songjeong-dong,  
Gumi-City, KOREA  
Phone: +82-54-454-6027      FAX: +82-54-454-6093

---

**SEIKO EPSON CORPORATION****SEMICONDUCTOR OPERATIONS DIVISION****IC Sales Dept.****IC International Sales Group**

421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN  
Phone: +81-42-587-5814      FAX: +81-42-587-5117