EPSON
EXCEED YOUR VISION

S1D13U11

Software Technical Manual

Document Number: X96A-A-002-01.3

SEIKO EPSON CORPORATION

Rev.1.3

NOTICE

Table of Contents

1. Overview 1

2. USB Specification 2

3. Word Definition 3

4. System Diagram 4

5. Function Description 5

5.1 USB Device Function 5

5.2 Protocol Sequencer 5

5.3 LCD Interface 5

5.4 I/O Interface 5

5.5 SPI Sequential Control 6

5.6 Start-up Display 6

5.7 Wakeup Display 6

5.8 Power Management 7

6. USB Description 8

6.1 Endpoint 8

6.2 Standard Request 8

6.3 Vender Request 9

6.4 Descriptor 9

6.4.1 Device Descriptor 9
6.4.2 Device Qualifier Descriptor 10
6.4.3 Configuration Descriptor 10
6.4.4 Interface Descriptor 11
6.4.5 HS Endpoint Descriptor (Command Block) 11
6.4.6 HS Endpoint Descriptor (Status Block) 11
6.4.7 HS Endpoint Descriptor (Event Block) 12
6.4.8 HS Endpoint Descriptor (Display Data Transfer) 12
6.4.9 FS Endpoint Descriptor (Command Block) 12
6.4.10 FS Endpoint Descriptor (Status Block) 13
6.4.11 FS Endpoint Descriptor (Event Block) 13
6.4.12 FS Endpoint Descriptor (Display Data Transfer) 13
6.4.13 Other Speed Configuration Descriptor 14
6.4.14 String Language ID Descriptor 14
6.4.15 String Descriptor 14

7. Control Procedure 15

7.1 Reset 15

7.2 Initialization 16

7.2.1 Serial Flash ROM is not used 16
7.2.2 Serial Flash ROM is used 17

7.3 LCD Interface 18

7.3.1 Register Access 18
7.3.2 Initialization 18

7.3.3 Display Data Transfer 19
7.3.4 Interrupt Processing 20
7.4 I/O Interface 21
7.4.1 GPIO Initialization 21
7.4.2 GPIO Input Data Read 22
7.4.3 GPIO Output Data Write 23
7.4.4 Key-Scan 23
7.4.5 I2C 24
7.4.6 SPI 24
7.4.7 SPI Sequential Control 25
7.4.8 Buzzer 25
7.5 Power Management 26
7.5.1 Entering Sleep Mode 26
7.5.2 Returning Sleep Mode 27
8. Vender Protocol 28
8.1 Basic Specification 28
8.2 Command Block 29
8.3 Status Block 31
8.4 Event Block 32
8.5 Notes 32
9. Vender Request 33
9.1 SOFT_RESET 33
10. Command Description 34
10.1 CFG_GETINFO 34
10.2 CFG_DOWNLOAD 36
10.3 CFG_SWITCH 38
10.4 LCDC_READ 39
10.5 LCDC_WRITE 41
10.6 LCDC_VRAM_ACC_ENABLE 43
10.7 LCDC_VRAM_ACC_DISABLE 44
10.8 LCDC_WAKEUP_ON_CONFIG 45
10.9 I2C_CONFIG 47
10.10 I2C_ACCESS 48
10.11 SPI_CONFIG 50
10.12 SPI_ACCESS 53
10.13 SPI_SEQUENCE_START 55
10.14 SPI_SEQUENCE_STOP 57
10.15 GPIO_CONFIG 59
10.16 GPIO_INT_CONFIG 61
10.17 GPIO_INT_CONTROL 63
10.18 GPIO_READ 65
10.19 GPIO_WRITE 67
10.20 KEYSCAN_CONTROL 69
10.21 KEYSCAN_READ 71
10.22 BUZZER_CONTROL 73

10.23 EVENT_INT_CONTROL 75
11. Event Description 77
11.1 LCDC_EVENT 77
11.2 SPI_INT_EVENT 78
11.3 SPI_SEQ_EVENT 79
11.4 GPI_EVENT 80
11.5 WAKEUP_EVENT 81
11.6 KEYSCAN_EVENT 82
12. Error Processing 83
12.1 Error Recovery 83
Appendix-A Configuration Data Format 84
A.1 For USB download 85
A.2 For Serial Flash ROM 86
A.3 Special Command List 87
A.4 Start-up Display Setting Example 88
A.5 Start-up Display Data Example 90
Appendix-B Difference of bDataRegAcc 91
Appendix-C SPI Sequential Control Setting Example 93
Appendix-D Example of Command Sample 94
D.1 SPI_CONFIG 94
D.2 I2C_CONFIG 95
D.3 LCDC_WAKEUP_ON_CONFIG 96
Revision History 97
Sales and Technical Support 98

1. Overview

This is the software technical manual for the S1D13U11 color LCD controller with USB2.0 FS/HS device port. Included in this document are hardware control procedures and technical information for system implementation. Please refer to “S1D13U11 Hardware Technical Manual” for hardware information in the S1D13U11.

2. USB Specification

The S1D13U11 conforms to the following USB Standards.

- Universal Serial Bus Specification Revision 2.0

  Speed mode: HS, FS support
- Universal Serial Bus (USB) Language Identifiers (LANGIDs) Version1.0

3. Word Definition

USB..... USB product with control function confoms to USB standard
USB Host ..... USB product with host function conforms to USB standard
USB Device ..... USB product with device function conforms to USB standard
Enumeration..... USB recognition processing between USB host and USB device
USB Bus Reset..... USB bus reset function conforms to USB strandard
USB Resume..... USB resume function conforms to USB strandard
USB Suspend ..... USB suspend function conforms to USB strandard
Remoto Wakeup..... USB remote wakeup function conforms to USB standard
Hardware Reset..... Hardware reset by RESET# pin
VRAM ..... Frame buffer memory of LCD display data
LCD Interface ..... Interface to LCD panel control

4. System Diagram

The S1D13U11 supports the USB2.0 FS/HS device function as the host CPU interface.

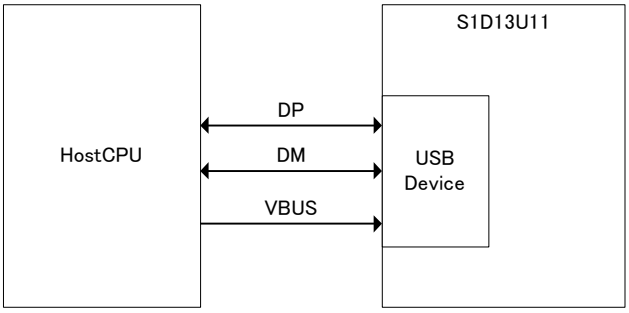

Figure 4.1 System Diagram

5. Function Description

The S1D13U11 has the following functions.

- USB device function
- Protocol sequencer
- LCD interface
- I/O interface
- SPI sequential control
- Start-up display
- Wakeup display
- Power management

5.1 USB Device Function

The S1D13U11 USB2.0 FS/HS device port conforms to USB standard.

5.2 Protocol Sequencer

The S1D13U11 has a protocol sequencer to simplify host CPU control. The host CPU controls the S1D13U11 with various types of protocol.

Configuration data is necessary to run the protocol sequencer. This data can be download into the S1D13U11 embedded SRAM from the USB port or theh SPI interface (external serial flash ROM).

5.3 LCD Interface

The S1D13U11 contains the following functions for the LCD interface.

- Write window
- Picture-in-Picture display
- Alpha-Blend
- Double buffer display
- Multi buffer display
- Virtual display
- PWM output
- GPO output
- FOUT output

Please refer to the “S1D13U11 Hardware Technical Manual” for LCD interface functions.

5.4 I/O Interface

The S1D13U11 has SPI, I2C, GPIO, Key-scan and Buzzer functions in the I/O interface.

Please refer to the “S1D13U11 Hardware Technical Manual” for I/O interface functions.

5.5 SPI Sequential Control

The S1D13U11 has an SPI sequential control function which registers the access procedures for the SPI device. This function executes the stored procedures automatically by trigger. The trigger factor can be selected as constant intervals or an INT0 interrupt.

The following are examples of the use of this function.

Case: Touch screen controller is selected as the SPI device.

- The host CPU receives the coordinate data by INT0 interrupt when the touch screen is touched.
- The host CPU receives the coordinate data from the touch screen controller at constant interval.

5.6 Start-up Display

S1D13U11 has a start-up display function which can initialize the LCD display from reset without host CPU control. For function it is necessary to store the initialization data in external serial flash ROM.

Please refer to "Appendix-A Configuration Data Format" for the serial flash data format.

5.7 Wakeup Display

S1D13U11 has a wakeup display function which will wake the LCD display from sleep mode without host CPU control. For function it is necessary to register the access procedures and store the display data into SDRAM. In sleep mode, the SDRAM operates on self-refresh.

The to/from sleep mode procedures are shown in Figure 5.1 and Figure 5.2.

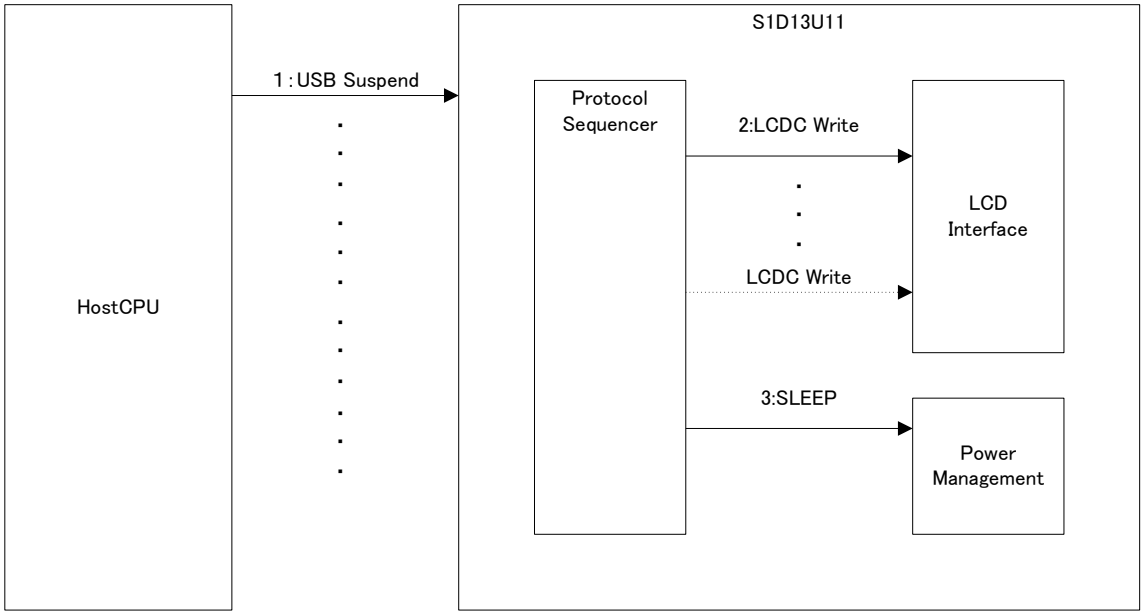

Figure 5.1 Enter to Sleep mode

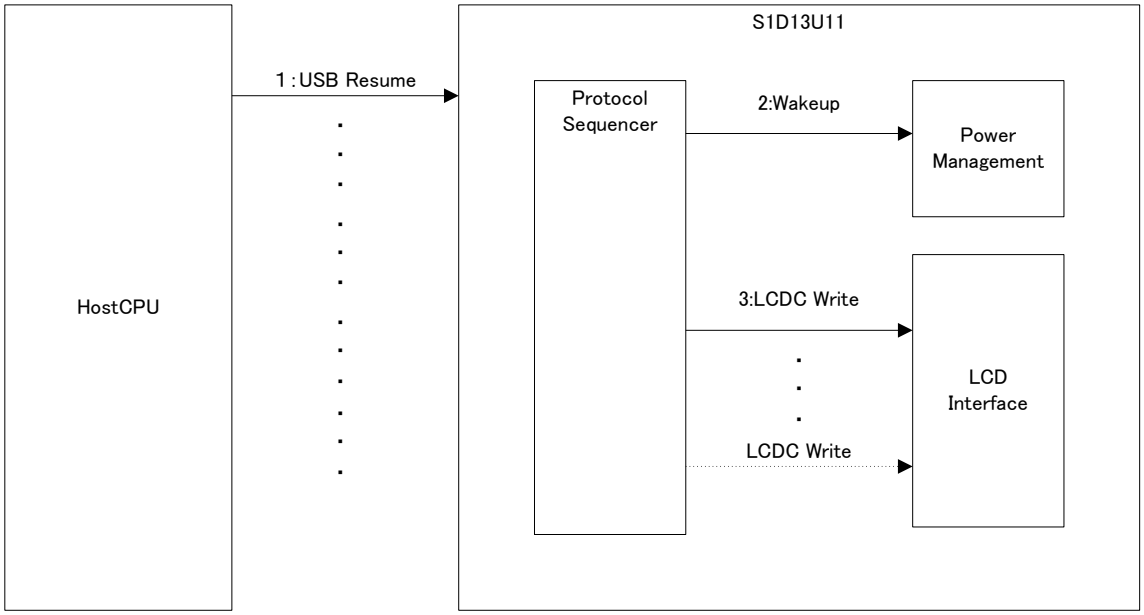

Figure 5.2 Return from Sleep mode

5.8 Power Management

The S1D13U11 enters into sleep mode through the following.

- USB suspend
- VBUS pin = Low

The S1D13U11 returns from sleep mode through the following.

- USB resume
- VBUS pin = High
- USB bus reset
- INT0 pin change
- INT1 pin change

6. USB Description

6.1 Endpoint

The S1D13U11 has one USB port and five endpoints. Each endpoint is described in Table 6.1.

Table 6.1 Endpoint Description

Endpoint				Description
Number	Direction	Type	Max Packet	
0	IN/OUT	CONTROL	HS:64 FS:64	For Standard request, Vender request
1	OUT	BULK	HS:512 FS:64	For Command transmission, Data transmission (except display data)
2	IN	BULK	HS:512 FS:64	Status reception, Data reception
3	IN	INT	HS:512 FS:64	Event reception
4	OUT	BULK	HS:512 FS:64	Display data transmission

6.2 Standard Request

S1D13U11 supports the following requests as the standard request. Please refer to the “USB Standard” for details on standard request.

Table 6.2 Standard Request

Request	Support
CLEAR_FEATURE	Yes
GET_CONFIGURATION	Yes
GET_DESCRIPTOR	Yes
GET_INTERFACE	Yes
GET_STATUS	Yes
SET_ADDRESS	Yes
SET_CONFIGURATION	Yes
SET_DESCRIPTOR	No
SET_FEATURE	Yes
SET_INTERFACE	Yes
SYNCH_FRAME	No

6.3 Vender Request

The S1D13U11 supports the following request as the vender request.

Table 6.3 Vender Request

Request
SOFT_RESET

6.4 Descriptor

6.4.1 Device Descriptor

The device descriptor is shown in Table 6.4. The value of idProduct is changed before / after the configuration data download.

Table 6.4 Device Descriptor

Offset	Content	Size (Byte)	Value	Comment
0	bLength	1	12h	
1	bDescriptorType	1	01h	DEVICE
2	bcdUSB	2	0200h	USB 2.00
4	bDeviceClass	1	00h	no
5	bDeviceSusbClass	1	00h	
6	bDeviceProtocol	1	00h	
7	bMaxPacketSize0	1	40h	
8	idVendor	2	04B8h	
10	idProduct	2	052Xh	Before download: 052Eh After download: 052Fh
12	bcdDevice	2	0100h	Program version 1.00
14	iManufacturer	1	01h	
15	iProduct	1	02h	
16	iSerialNumber	1	00h	
17	bNumConfigurations	1	01h	

6.4.2 Device Qualifier Descriptor

The device qualifier descriptor is shown in Table 6.5.

Table 6.5 Device Qualifier Descriptor

Offset	Content	Size (Byte)	Value	Description
0	bLength	1	0Ah	
1	bDescriptorType	1	06h	DEVICE_QUALIFER
2	bcdUSB	2	0200h	USB 2.00
4	bDeviceClass	1	00h	no
5	bDeviceSubClass	1	00h	
6	bDeviceProtocol	1	00h	
7	bMaxPacketSize0	1	40h	
8	bNumConfigurations	1	01h	
9	bReserved	1	00h	

6.4.3 Configuration Descriptor

The configuration descriptor is shown in Table 6.6. The value of bmAttributes is changed before / after the configuration data download.

Table 6.6 Configuration Descriptor

Offset	Content	Size (Byte)	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	02h	CONFIGURATION
2	wTotalLength	2	002Eh	46 bytes
4	bNumInterfaces	1	01h	Number of interface: 1
5	bConfigurationValue	1	01h	
6	iConfiguration	1	00h	
7	bmAttributes	1	XXh	Before download: C0h After download: E0h
8	bMaxPower	1	2Dh	90mA

6.4.4 Interface Descriptor

The interface descriptor is shown in Table 6.7.

Table 6.7 Interface Descriptor

Offset	Content	Size (Byte)	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	04h	
2	bInterfaceNumber	1	00h	
3	bAlternateSetting	1	00h	
4	bNumEndpoints	1	04h	
5	bInterfaceClass	1	FFh	
6	bInterfaceSubClass	1	00h	
7	bInterfaceProtocol	1	FFh	
8	iInterface	1	00h	

6.4.5 HS Endpoint Descriptor (Command Block)

The HS endpoint descriptor of command block is shown in Table 6.8.

Table 6.8 HS Endpoint Descriptor (Command Block)

Offset	Content	Size (Byte)	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	ENDPOINT
2	bEndpointAddress	1	01h	EP1-OUT
3	bmAttributes	1	02h	BULK
4	wMaxPacketSize	2	0200h	512 byte
6	bInterval	1	00h	

6.4.6 HS Endpoint Descriptor (Status Block)

The HS endpoint descriptor of status block is shown in Table 6.9.

Table 6.9 HS Endpoint Descriptor (Status Block)

Offset	Content	Size (Byte)	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	ENDPOINT
2	bEndpointAddress	1	82h	EP2-IN
3	bmAttributes	1	02h	BULK
4	wMaxPacketSize	2	0200h	512 byte
6	bInterval	1	00h	

6.4.7 HS Endpoint Descriptor (Event Block)

The HS endpoint descriptor of event block is shown in Table 6.10.

Table 6.10 HS Endpoint Descriptor (Event Block)

Offset	Content	Size (Byte)	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	ENDPOINT
2	bEndpointAddress	1	83h	EP3-IN
3	bmAttributes	1	03h	INTERRUPT
4	wMaxPacketSize	2	0040h	64 byte
6	bInterval	1	08h	1ms

6.4.8 HS Endpoint Descriptor (Display Data Transfer)

The HS endpoint descriptor of display data transfer is shown in Table 6.11.

Table 6.11 HS Endpoint Descriptor (Display Data Transfer)

Offset	Content	Size (Byte)	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	ENDPOINT
2	bEndpointAddress	1	04h	EP4-OUT
3	bmAttributes	1	02h	BULK
4	wMaxPacketSize	2	0200h	512 byte
6	bInterval	1	00h	

6.4.9 FS Endpoint Descriptor (Command Block)

The FS endpoint descriptor of command block is shown in Table 6.12.

Table 6.12 FS Endpoint Descriptor (Command Block)

Offset	Content	Size (Byte)	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	ENDPOINT
2	bEndpointAddress	1	01h	EP1-OUT
3	bmAttributes	1	02h	BULK
4	wMaxPacketSize	2	0040h	64 byte
6	bInterval	1	00h	

6.4.10 FS Endpoint Descriptor (Status Block)

The FS endpoint descriptor of status block is shown in Table 6.13.

Table 6.13 FS Endpoint Descriptor (Status Block)

Offset	Content	Size (Byte)	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	ENDPOINT
2	bEndpointAddress	1	82h	EP2-IN
3	bmAttributes	1	02h	BULK
4	wMaxPacketSize	2	0040h	64 byte
6	bInterval	1	00h	

6.4.11 FS Endpoint Descriptor (Event Block)

The FS endpoint descriptor of event block is shown in Table 6.14.

Table 6.14 FS Endpoint Descriptor (Event Block)

Offset	Content	Size (Byte)	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	ENDPOINT
2	bEndpointAddress	1	83h	EP3-IN
3	bmAttributes	1	03h	INTERRUPT
4	wMaxPacketSize	2	0040h	64 byte
6	bInterval	1	01h	1ms

6.4.12 FS Endpoint Descriptor (Display Data Transfer)

The FS endpoint descriptor of display data transfer is shown in Table 6.15.

Table 6.15 FS Endpoint Descriptor (Display Data Transfer)

Offset	Content	Size (Byte)	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	ENDPOINT
2	bEndpointAddress	1	04h	EP4-OUT
3	bmAttributes	1	02h	BULK
4	wMaxPacketSize	2	0040h	64 byte
6	bInterval	1	00h	

6.4.13 Other Speed Configuration Descriptor

The other speed configuration descriptor is shown in Table 6.16. The value of bmAttributes is changed before / after the configuration data download.

Table 6.16 Other Speed Configuration Descriptors

Offset	Content	Size (Byte)	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	07h	OTHER_SPEED_CONFIGURATION
2	wTotalLength	2	002Eh	46 byte
4	bNumInterfaces	1	01h	Number of interface: 1
5	bConfigurationValue	1	01h	
6	iConfiguration	1	00h	
7	bmAttributes	1	XXh	Before download: C0h After download: E0h
8	bMaxPower	1	2Dh	90mA

6.4.14 String Language ID Descriptor

The string language ID descriptor is shown in Table 6.17.

Table 6.17 String Language ID Descriptor

Offset	Content	Size (Byte)	Value	Description
0	bLength	1	04h	
1	bDescriptorType	1	03h	STRING
2	bString	2	0409	Language ID (US)

6.4.15 String Descriptor

The string descriptor is shown in Table 6.18 and Table 6.19.

Table 6.18 String Descriptor (Index1)

Offset	Content	Size (Byte)	Value	Description
0	bLength	1	0Ch	
1	bDescriptorType	1	03h	STRING
2	bString	10	-	“EPSON” (by UNICODE)

Table 6.19 String Descriptor (Index2)

Offset	Content	Size (Byte)	Value	Description
0	bLength	1	12h	
1	bDescriptorType	1	03h	STRING
2	bString	16	-	“S1D13U11” (by UNICODE)

7. Control Procedure

In this section, the control procedure for the S1D13U11 is described by flow chart.

For example, “CFG_DOWNLOAD” (FEh) means that command name is “CFG_DOWNLOAD” and “FEh” is command code. Please refer to “10. Command Description” for details on each command.

Similarly, “LCDC_EVENT" (00h) means that event name is “LCDC_EVENT” and “00h” is event code. Please refer to “11. Event Description” for details on each event.

7.1 Reset

The S1D13U11 has the following reset functions.

- Hardware reset
- Software reset
- USB bus reset

The hardware reset uses the RESET# pin of the S1D13U11. After hardware reset, the S1D13U11 is initialized.

The software reset is generated by software command from vender request. After software reset, the S1D13U11 is initialized the same as a hardware reset.

The S1D13U11 has a USB bus reset function. After USB bus reset, only USB interface is initialized to the status before USB enumeration. The LCD interface, I/O interface and the configuration data keep the same status as before the USB bus reset.

Table 7.1 Reset Function

Items	Hardware reset	Software reset	USB bus reset
Reset method	RESET# pin	Command	USB bus status
After reset	Hardware reset	Hardware reset	USB interface reset
USB	Disconnect	Disconnect	Disconnect
LCD interface	Initialization	Initialization	Keep status
I/O interface	Initialization	Initialization	Keep status
Configuration data	Require download	Require download	No download required

7.2 Initialization

There are two cases for initialization of the S1D13U11.

- Serial Flash ROM is not used.
- Serial Flash ROM is used.

After a hardware reset, the S1D13U11 reads the Serial Flash ROM. If the S1D13U11 reads the ASCII code 53, 31, 44, 31, 33, 55, 31, 31 from the ROM Address header, the Serial Flash ROM is used. In this case, the configuration data is downloaded from Serial Flash ROM automatically, and the host CPU does not need to download configuration data.

7.2.1 Serial Flash ROM is not used

When the serial flash ROM is not used, the S1D13U11 is initialized as shown in Figure 7.1.

"USB cable connection" is not necessary, when the USB cable is always connected.

After "CFG_SWITCH (FFh) transmission", the S1D13U11 connection is automatically disconnected.

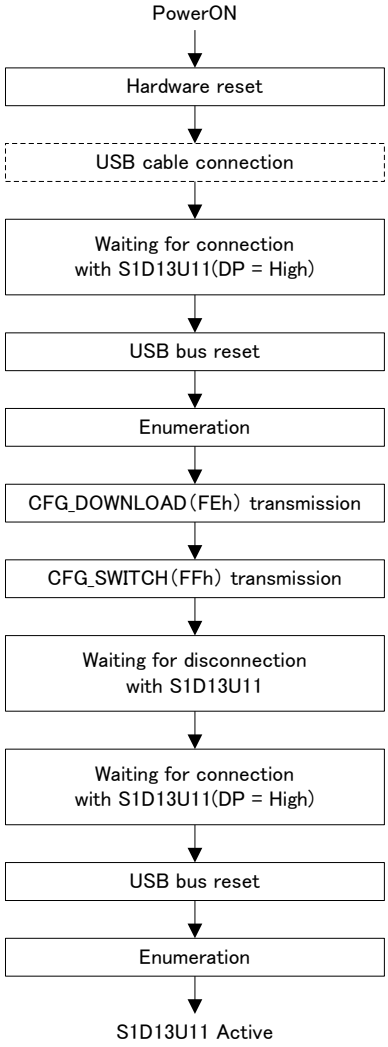

Figure 7.1 Initialization flow when Serial Flash ROM is not used

7.2.2 Serial Flash ROM is used

When the serial flash ROM is used, the S1D13U11 is initialized as shown in Figure 7.2.

“USB cable connection” is not necessary when the USB cable is always connected.

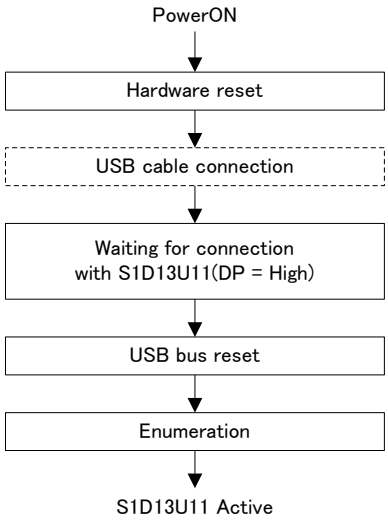

Figure 7.2 Initialization flow when Serial Flash ROM is used

7.3 LCD Interface

The control procedures for the LCD interface are described in this section.

- Register access
- Initialization
- Display data transfer
- Interrupt processing

7.3.1 Register Access

The LCD interface registers can be controlled as shown in Figure 7.3 and Figure 7.4.

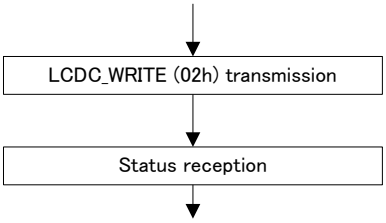

Figure 7.3 Register Write flow

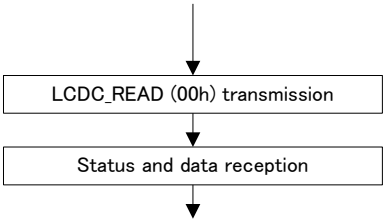

Figure 7.4 Register Read flow

7.3.2 Initialization

The initialization of the LCD interface is done by register setting according to system implementation. Please refer to “S1D13U11 Hardware Technical Manual” for register details and the control procedure.

7.3.3 Display Data Transfer

The display data transfer for the LCD interface can be controlled as shown in Figure 7.5.

”LCD_WRITE (02h) transmittion” and”Status reception” are necessary when the display settings are changed.

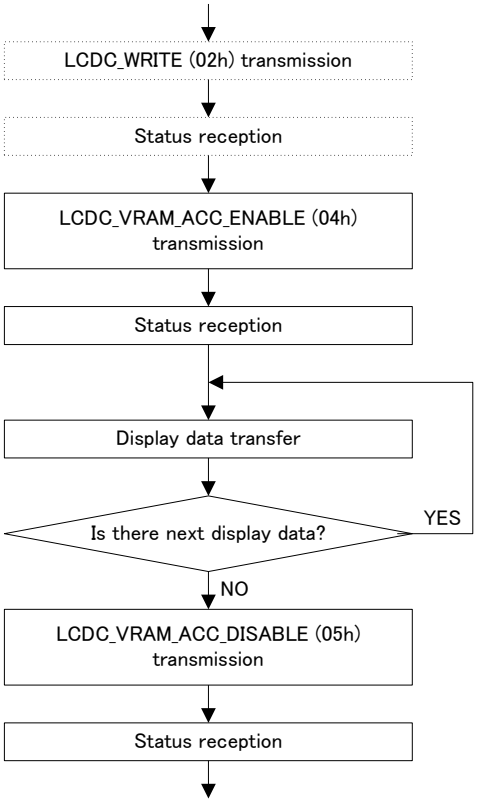

Figure 7.5 Display Data Transfer flow

7.3.4 Interrupt Processing

The interrupt processing of the LCD interface can be controlled as shown in Figure 7.6.

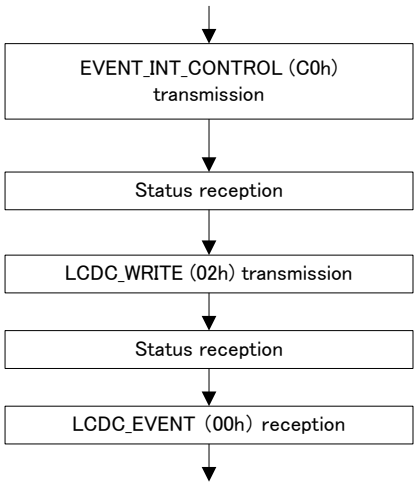

Figure 7.6 Interrupt Processing flow

7.4 I/O Interface

The control procedures for the I/O interface are described in this section.

- GPIO initialization
- GPIO input data read
- GPIO output data write
- Key-scan
- I2C
- SPI
- SPI sequential control
- Buzzer

7.4.1 GPIO Initialization

The GPIO configuration is initialized as shown in Figure 7.7. This procedure is not necessary when GPIO is used with default values.

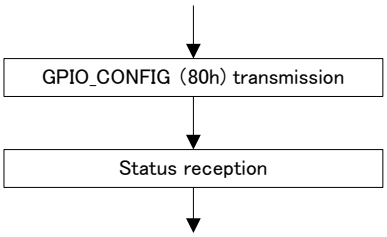

Figure 7.7 GPIO Initialization flow

7.4.2 GPIO Input Data Read

There are two methods of GPIO input data read. The GPIO input data read from the command is controlled as shown in Figure 7.8.

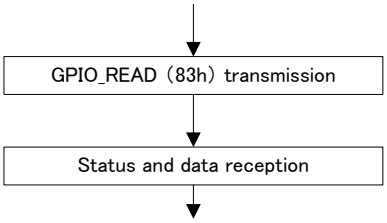

Figure 7.8 GPIO Input Data Read flow (Command)

"GPI_EVENT (80h) reception" occurs only when the status of the GPI input is changed. The GPIO input data read from event is controlled as shown in Figure 7.9.

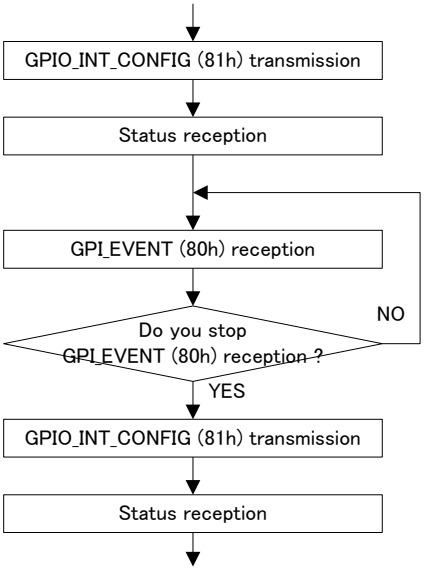

Figure 7.9 GPIO Input Data Read flow (Event)

7.4.3 GPIO Output Data Write

The GPIO output data write can be controlled as shown in Figure 7.10.

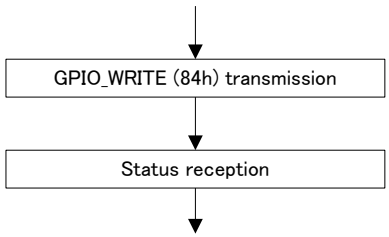

Figure 7.10 GPIO Output Data Write flow

7.4.4 Key-Scan

The key-scan can be controlled as shown in Figure 7.11.

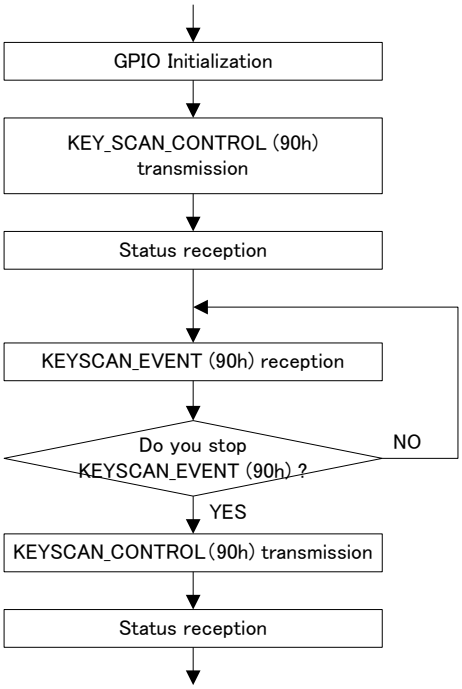

Figure 7.11 Key-scan Control flow

7.4.5 I2C

The I2C can be controlled as shown in Figure 7.12.

“I2C_CONFIG (20h) transmission” and “Status reception” are not necessary when default settings are used.

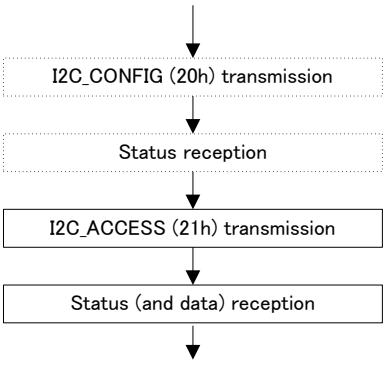

Figure 7.12 I2C Control flow

7.4.6 SPI

The SPI can be controlled as shown in Figure 7.13.

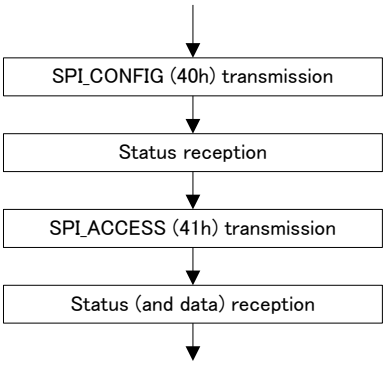

Figure 7.13 SPI Control flow

7.4.7 SPI Sequential Control

The SPI sequential control can be controlled as shown in Figure 7.14.

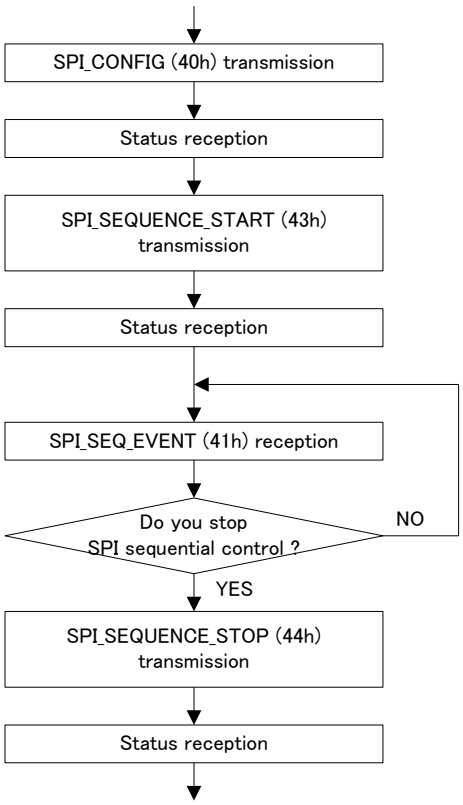

Figure 7.14 SPI Sequential Control flow

7.4.8 Buzzer

The buzzer can be controlled as shown in Figure 7.15.

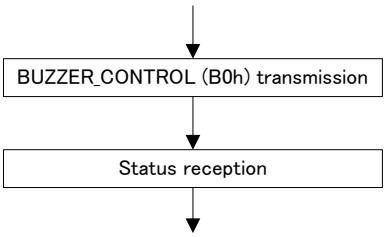

Figure 7.15 Buzzer Control flow

7.5 Power Management

The control procedures for power management are described in this section.

7.5.1 Entering Sleep Mode

Entering sleep mode is controlled as shown in Figure 7.16.

“LCDC_WAKEUP_ON_CONFIG (06h) transmission” is necessary when LCD wakeup display is used.

”EVENT_INT_CONTROL (C0h) transmission” is necessary when USB remote wakeup is used.

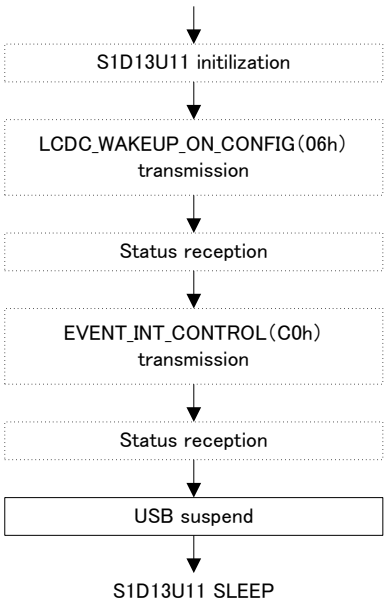

Figure 7.16 Entering Sleep Mode flow

7.5.2 Returning Sleep Mode

There are two methods for returning from sleep mode.

- USB resume
- INT0/INT1 input change

Returning from sleep mode (USB resume) is controlled as shown in Figure 7.17.

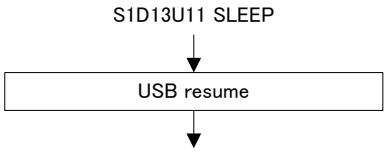

Figure 7.17 Returning from Sleep Mode flow (USB Resume)

Returning sleep mode (INT0/INT1 input changing) is controlled as shown in Figure 7.18.

“INT0/INT1 input change” is an external operation. This operation may be pressing the wake up key (INT1) or the touch screen (INT0).

When the wake up key (INT1) is pressed, “WAKEUP_EVENT" (81h) is received. When the touch screen (INT0) is pressed, “SPI_INT_EVENT" (40h) or “SPI_SEQ_EVENT" (41h) are received.

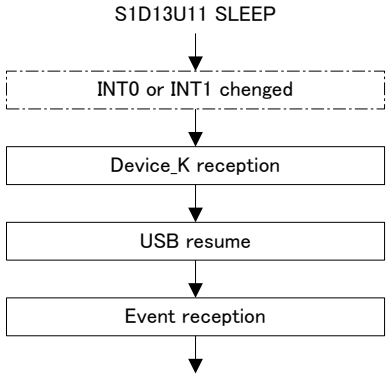

Figure 7.18 Returing from Sleep Mode flow (INT0/INT1 Input Changing)

8. Vender Protocol

In this section, the vender protocol for the S1D13U11 is described.

8.1 Basic Specification

The S1D13U11 protocol is described below.

- The host CPU transmits command block (command + additional data) to the S1D13U11.
- The S1D13U11 transmits status block (status + additional data) to the host CPU after command reception.
- The host CPU cannot transmit new command block to the S1D13U11 until the status block of the previous command is received.
- The S1D13U11 transmits event block (event + additional data) to the host CPU.
- The host CPU transmits LCD display data to the S1D13U11.

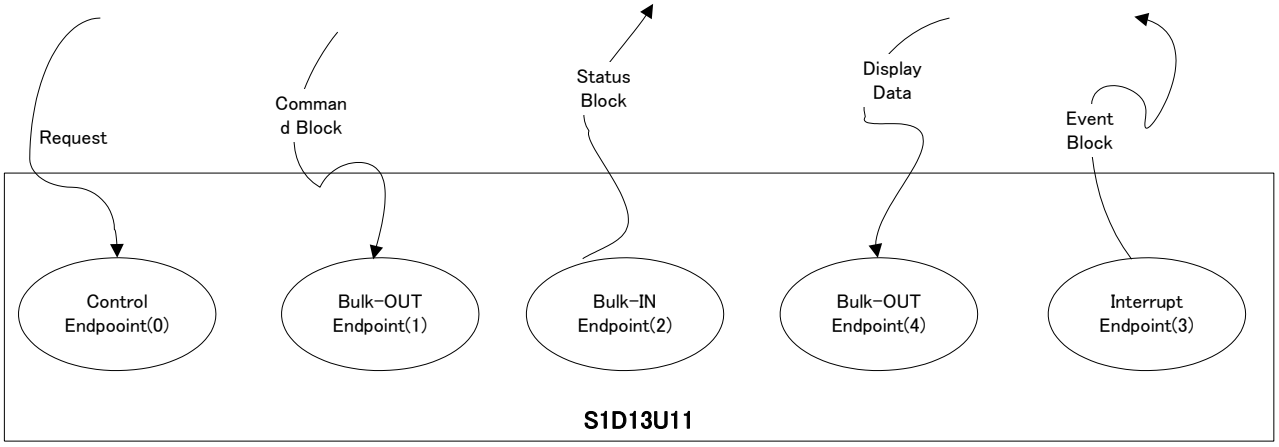

Figure 8.1 Vender Protocol Basic Specification

8.2 Command Block

The command block consists of command and additional data.

The size of command block should be 1040 bytes or less. However, only “CFG_DOWNLOAD (FEh)” can transmit over 1040 bytes.

Table 8.1 shows the command block.

Table 8.1 Command Block

	7	6	5	4	3	2	1	0
0	bCBWCB							
1	bCBWTag							
2	wCBWReserved							
3								
4-15	Parameter							
16-n	Data[0-x]							

- bCBWCB
  This byte is the command code. Please refer to “Table 8.2 Command Code List” for the details of each command code.

- bCBWTag
  This byte is the command block tag. It is used to relate between command block and status block. The host CPU specifies the value.

- wCBWReserved
  These bytes are reserved for future use. “0000h" is the defaut value.

- Parameter
  These bytes are the command parameter. Please refer to “10. Command Description” for the value of each command parameter.

- Data
  These bytes are the data. Some command parameters do not have data.

Table 8.2 Command Code List

Command Code	Value	Classification	Outline
CFG_GETINFO	FDh	Configuration	Version information is read.
CFG_DOWNLOAD	FEh	Configuration	The configuration data is downloaded.
CFG_SWITCH	FFh	Configuration	The configuration data downloaded is enabled.
LCDC_READ	00h	LCD	Data is read from LCDC registers.
LCDC_WRITE	02h	LCD	Data is written to LCDC registers.
LCDC_VRAM_ACC_ENABLE	04h	LCD	The display data transfer is enabled.
LCDC_VRAM_ACC_DISABLE	05h	LCD	The display data transfer is disabled.
LCDC_WAKEUP_ON_CONFIG	06h	LCD	The control procedure of wake-up is registered.
I2C_CONFIG	20h	I2C	The configuration of I2C is set.
I2C_ACCESS	21h	I2C	Data is read and written from/to I2C device.
SPI_CONFIG	40h	SPI	The configuration of SPI is set.
SPI_ACCESS	41h	SPI	Data is read and written from/to SPI device.
SPI_SEQUENCE_START	43h	SPI	The SPI sequential function is set and begun.
SPI_SEQUENCE_STOP	44h	SPI	The SPI sequential function is stopped.
GPIO_CONFIG	80h	GPIO	The configuration of GPIO is set.
GPIO_INT_CONFIG	81h	GPIO	The configuration of GPIO interrupt is set.
GPIO_INT_CONTROL	82h	GPIO	The GPIO_EVENT is controlled.
GPIO_READ	83h	GPIO	The input data of GPIO is read.
GPIO_WRITE	84h	GPIO	The output data of GPIO is written.
KEYSCAN_CONTROL	90h	Key-scan	The key-scan is controlled.
KEYSCAN_READ	91h	Key-scan	The key-scan data is read.
BUZZER_CONTROL	B0h	Buzzer	The buzzer is controlled.
EVENT_INT_CONTROL	C0h	Event	The event notification is controlled.

8.3 Status Block

The status block consists of status and additional data.

Table 8.3 shows the status block.

Table 8.3 Status Block

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	wCSWReserved							
3								
4-7	Parameter							
8-n	Data[0-x]							

- bCSWStatus
  This byte is the status code. Please refer to “Table 8.4 Status Code List” for the details of each status code.

- bCSWTag
  This byte is the status block tag. It is used to relate between command block and status block. The value is the same as bCBWTag (command block tag).
  However, when the status code of bCSWStatus is “PROTOCOL_ERROR”, the value becomes “FFh”.

- wCSWReserved
  These bytes are reserved for future use. “0000h" is the defaut value.

- Parameter
  These bytes are the status parameter. Please refer to “10. Command Description” for value of each status parameter.

- Data
  These bytes are the data. Some status parameters do not have data.

Table 8.4 Status Code List

Status Code	Value	Outline
SUCCESS	00h	Command was received correctly.
INVALID_PARAM	01h	There was an error in the parameter of command block.
CMD_ERROR	02h	There were the following errors. ・ Command was not received correctly. ・ Command that is not supported was received.
PROTOCOL_ERROR	FFh	There were the following errors. ・ Command block length was less than 16 bytes. ・ There was a difference in number of parameters between command block and actual data.

8.4 Event Block

The event block consists of the event and additional data.

Table 8.5 shows the event block.

Table 8.5 Event Block

	7	6	5	4	3	2	1	0
0	bEventType							
1	bEventReserved							
2	wLength							
3								
4-n	Data[0-x]							

- bEventType
  This byte is the event code. Please refer to Table 8.6 “Event Code List” for the details of each event code.

- bEventReserved
  This byte is reserved for the future use. “00h" is the defualt value.

- wLength
  This is the number of effective bytes in the data field.

- Data
  These bytes are the data. Some status parameters do not have data.

Table 8.6 Event Code List

Event Code	Value	Outline
LCDC_EVENT	00h	LCD interrupt was occurred.
SPI_INT_EVENT	40h	There was a change in SPI interrupt pin (INT0).
SPI_SEQ_EVENT	41h	A SPI sequential procedure was finished.
GPI_EVENT	80h	There was a change in GPIO input pin.
WAKEUP_EVENT	81h	There was a change on wake up key (INT1).
KEYSCAN_EVENT	90h	There was a change in key-scan data.

8.5 Notes

- Protocol data format is little endian unless otherwise stated.
- When the number of event blocks exceeds ten, new events are not stored in endpoint and are lost. Therefore, the host CPU must read the event block at constant interval.

9. Vender Request

In this section, vender request of the S1D13U11 is described.

9.1 SOFT_RESET

This command initializes the S1D13U11 by software reset.

Table 9.1 SOFT_RESET

bmRequestType	bRequest	wValue	wIndex	wLength	Data
40h	FFh	0000h	0000h	0000h	None

- Function description
  The S1D13U11 is initialized by the host CPU.
  The reset processing is executed after the status stage of control transfer ends. After this command executes, USB re-connecting is necessary because USB connection is initialized.
  Please refer to “7.1 Reset” for the details of soft reset.

10. Command Description

This section descripes the S1D13U11 commands.

10.1 CFG_GETINFO

This command gets the version information from the S1D13U11.

The CFG_GETINFO command is shown in Table 10.1 and the status is shown in Table 10.2.

Table 10.1 CFG_GETINFO Command

	7	6	5	4	3	2	1	0
0	FDh (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							
8	00h							
9	00h							
10	00h							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							

Table 10.2 CFG_GETINFO Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	0004h(wReadSize)							
5								
6	00h							
7	00h							
8	bMode							
9	00h							
10	wVersion							
11								

- bCSWStatus
  The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCSWReserved. ・ Data “0004h” was not set in wReadSize.

- bMode
  The status of the download configuration data is shown.

Value	Explanation
00h	Data disable
01h	Data enable

- wVersion
  The version number is shown in BCD. The content is different depending on the value of bMode.

bMode	Content
00h	Version number of the S1D13U11
01h	Version number of configuration data

10.2 CFG_DOWNLOAD

This command downloads the configuration data into the embedded SRAM of the S1D13U11. Please refer to "Appendix-A Configuration Data" for the content of the configuration data.

The size of the configuration data is fixed at 96K bytes. This command cannot be used after the "CNF_SWITCH" command is executed (the configuration data is enabled).

The CFG_DOWNLOAD command is shown in Table 10.3 and the status is shown in Table 10.4.

Table 10.3 CFG_DOWNLOAD Command

	7	6	5	4	3	2	1	0
0	FEh (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	00000000h(dwOffset)							
5								
6								
7								
8	00018000h(dwSize)							
9								
10								
11								
12	00h							
13	00h							
14	00h							
15	00h							
16	CFG_data[0]							
. . .	. . .							
n	CFG_data[x]							

- dwOffset
  These bytes specify offset value of configuration data.
  In the S1D13U11, "00000000h" is specified.

- dwSize
  These bytes specify data size of configuration data.
  In the S1D13U11, "00018000h" is specified.

- CFG_data
  These bytes are data field of configuration data.

Table 10.4 CFG_DOWNLOAD Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							

- bCSWStatus

The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved. ・ Data “exceeding 96K bytes” was set in dwOffset+dwSize. ・ Data “multiple in 256 bytes” was not set in dwSize. ・ Data “multiple in 256 bytes” was not set in dwOffset.
CMD_ERROR	There was the following errors. ・ The configuration data has already enabled before download.

10.3 CFG_SWITCH

This command switches to enable the configuration data.

The protocol sequencer of the S1D13U11 starts when this command is executed. After this command executes, USB re-connecting is necessary because USB is disconnected.

The CFG_SWITCH command is shown in Table 10.5 and the status is shown in Table 10.6.

Table 10.5 CFG_SWITCH Command

	7	6	5	4	3	2	1	0
0	FFh (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							
8	00h							
9	00h							
10	00h							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							

Table 10.6 CFG_SWITCH Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							

- bCSWStatus
  The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved.
CMD_ERROR	There was the following errors. ・ The configuration data was enabled before download. ・ The configuration data was not downloaded before enable. ・ The configuration data was not correct. (Data was corrupt).

10.4 LCDC_READ

This command reads from the LCDC register of the S1D13U11. Please refer to the “S1D13U11 Hardware Technical Manual’” for details on the LCDC registers.

This command can not be used when the "LCDC_VRAM_ACC_ENABLE" command is executed.

The LCDC_READ command is shown in Table 10.7 and the status is shown in Table 10.8.

Table 10.7 LCDC_READ Command

	7	6	5	4	3	2	1	0
0	00h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	00h							
5	00h							
6	bDataRegAcc	00h						
7	00h							
8	00h							
9	00h							
10	00h							
11	00h							
12	wReadSize							
13								
14	wRegAddress							
15								

- bDataRegAcc
  This bit is set to “1b” when the SDRAM or LUT (Look-up table) are read. For all other cases, this bit is set to ”0b”.
  Please refer to “Appendix-B bDataRegAcc Difference” for details of bDataRegAcc.

- wReadSize
  These bytes specify the amount of read data in the range of “0002h - 0400h” in even numbers.

- bRegAddress
  These bytes specify the LCDC register address. An even number address is set due to 16 bit access.

Table 10.8 LCDC_READ Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	wReadSize							
5								
6	00h							
7	00h							
8	wReadData[0]							
9								
. . .	. . .							
n-1	wReadData[x]							
n								

- bCSWStatus
  The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved. ・ Data “outside range” was set in wReadSize. ・ Data “odd number address” was set in wRegAddress.
CMD_ERROR	There was the following errors. ・ “LCDC_VRAM_ACC_ENABLE" command was enabled.

- wReadSize
  These bytes show the amount (size) of read data.

- wReadData
  These bytes are the read data field from the LCDC register.

10.5 LCDC_WRITE

This command writes to the LCDC register of the S1D13U11. Please refer to the “S1D13U11 Hardware Technical Manual” for details on the LCDC registers.

This command can not be used when th e"LCDC_VRAM_ACC_ENABLE" command is executed.

The LCDC_WRITE command is shown in Table 10.9 and the status is shown in Table 10.10.

Table 10.9 LCDC_WRITE Command

	7	6	5	4	3	2	1	0
0	02h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	00h							
5	00h							
6	bDataRegAcc	00h						
7	00h							
8	wWriteSize							
9								
10	00h							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							
16	wRegAddress[0]							
17								
18	wWriteData[0]							
19								
. . .	. . .							
n-3	wRegAddress[x]							
n-2								
n-1	wWriteData[x]							
n								

- bDataRegAcc
  This bit is set to “1b” when the LUT (Look-up table) is written. For all other cases, this bit is set to ”0b”.
  Please refer to “Appendix-B bDataRegAcc Difference” for details of bDataRegAcc.

- wWriteSize
  These bytes specify amount (size) of write data the range of “0004h - 0400h” in multiples of four.

- wRegAddress
  These bytes specify LCDC register address. An even address is set due to 16 bit access.
  When “FFFFh" is specified, the LCDC register is not written. Instead, the wait time (in msec) is set with the value of wRegData. In this case, these bytes specify a number in the range of “0000h - 0100h”. When a number outside the range is specified, “0100h” is used.

- wWriteData
  These bytes are the write data field of LCDC register. Data size is a multiple of 16 bits.

Table 10.10 LCDC_WRITE Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							

- bCSWStatus
  The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved. ・ Data “outside range” was set in wWriteSize. ・ Data “odd number address” was set in wRegAddress.
CMD_ERROR	There were the following errors. ・ “LCDC_VRAM_ACC_ENABLE" command was enabled.

10.6 LCDC_VRAM_ACC_ENABLE

This command enables VRAM access for display data transfer.

When this command is enabled, both the "LCDC_WRITE" and "LCDC_READ" commands cannot be used.

The LCDC_VRAM_ACC_ENABLE command is shown in Table 10.11 and the status is shown in Table 10.12.

Table 10.11 LCDC_VRAM_ACC_ENABLE Command

	7	6	5	4	3	2	1	0
0	04h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	dwOnePictureSize							
5								
6								
7								
8	00h							
9	00h							
10	00h							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							

- dwOnePictureSize
  These bytes specify the display data size of one transfer. The display size is specified in multiples of eight.

Table 10.12 LCDC_VRAM_ACC_ENABLE Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							

- bCSWStatus
  The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved. ・ Data “00000000h” was set in dwOnePictureSize. ・ Data “multiple of eight” was not set in dwOnePictureSize.
CMD_ERROR	There were the following errors. ・ “LCDC_VRAM_ACC_ENABLE" command was already enabled.

10.7 LCDC_VRAM_ACC_DISABLE

This command disables VRAM access for display data transfer.

When this command is received durring display data transfer, the VRAM access is stopped immediately.

The LCDC_VRAM_ACC_DISABLE command is shown in Table 10.13 and the status is shown in Table 10.14.

Table 10.13 LCDC_VRAM_ACC_DISABLE Command

	7	6	5	4	3	2	1	0
0	05h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							
8	00h							
9	00h							
10	00h							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							

Table 10.14 LCDC_VRAM_ACC_DISABLE Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							

- bCSWStatus

  The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved.

10.8 LCDC_WAKEUP_ON_CONFIG

This command registers the control procedure for the LCD wake-up display function.

When the S1D13U11 enters into sleep mode or returns from sleep mode, the control procedure for LCDC registers is executed automatically.

While the control procedure is executed, status changes of the USB bus can not be detected. Please refer to "Appendix-D LCDC_WAKEUP_ON_CONFIG Command Sample" for the setting of this control procedure.

The LCDC_WAKEUP_ON_CONFIG command is shown in Table 10.15 and the status is shown in Table 10.16.

Table 10.15 LCDC_WAKEUP_ON_CONFIG Command

	7	6	5	4	3	2	1	0
0	06h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	00h							
5	bType							
6	00h							
7	00h							
8	wWriteSize							
9								
10	00h							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							
16	wRegAddress[0]							
17								
18	wWriteData[0]							
19								
. . .	. . .							
n-3	wRegAddress[x]							
n-2								
n-1	wWriteData[x]							
n								

- bType
  This byte specifies type of the control procedure.

Value	Explanation
00h	Enter to sleep mode
01h	Return from sleep mode

- wWriteSize
  These bytes specify the amount (size) of data following the command block in the range of "0004h - 0100h" in multiples of four.

- wRegAddress
  These bytes specify the LCDC register address. An even address is set due to 16 bit access.
  When “FFFFh" is specified, LCDC register is not written. Instead, the wait time (in msec) is set with the value of wRegData. In this case, these bytes specify a number in the in the range of “0000h-0100h”. When a number outside the range is specified, “0100h” is used.

- wWriteData
  These bytes are the write data field of LCDC register. Thedata size is a multiple of 16 bits.

Table 10.16 LCDC_WAKEUP_ON_CONFIG Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							

- bCSWStatus
  The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved. ・ Data “regulated value” was not set in bType. ・ Data “outside range” was set in wWriteSize. ・ Data “odd number address” was set in wRegAddress.

10.9 I2C_CONFIG

This command sets the I2C configuration.

When this command is not set, the I2C transfer rate is 100kbps.

The I2C_CONFIG command is shown in Table 10.17 and the status is shown in Table 10.18.

Table 10.17 I2C_CONFIG Command

	7	6	5	4	3	2	1	0
0	20h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	bTransferRate							
5	00h							
6	00h							
7	00h							
8	00h							
9	00h							
10	00h							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							

- bTransferRate
  This byte specifies transfer rate of I2C.

Value	Explanation
01h	100kbps (Standard Mode)
02h	400kbps (Fast Mode)

Table 10.18 I2C_CONFIG Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							

- bCSWStatus
  The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved. ・ Data “regulated value” was not se in bTransferRate.

10.10 I2C_ACCESS

This command is used access the I2C device.

There are three types of the access methods.

The I2C_ACCESS command is shown in Table 10.19 and tThe status is shown in Table 10.20.

Table 10.19 I2C_ACCESS Command

	7	6	5	4	3	2	1	0
0	21h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	00h							
5	bSlaveAddress							
6	bEnRepeatedStartCondition							
7	00h							
8	wWriteSize							
9								
10	00h							
11	00h							
12	wReadSize							
13								
14	00h							
15	00h							
16	bWrData[0]							
. . .	. . .							
n	bWrData[x]							

- bSlaveAddress
  This byte specifies the slave address (7bit) of an I2C device in the range of “00h-7Fh”.

- bEnRepeatedStartCondition
  This byte specifies the condition code when read after write to I2C device.

Value	Explanation
01h	Not use
02h	Stop condition and start condition are used.

- wWriteSize
  These bytes specify amount (size) of write data to the I2C device in the range of “0000h-0400h”.
  When there are no write accesses, “0000h" is specified.

- wReadSize
  These bytes specify amount (size) of read data from the I2C device in the range of “0000h-0400h”.
  When there are no read accesses, “0000h" is specified.

- bWrData
  These bytes are the write data field to the I2C device.

Access	wWriteSize	wReadSize
Write	Except “0000h”	“0000h”
Read	“0000h”	Except “0000h”
Read after Write	Not use	Not use

Table 10.20 I2C_ACCESS Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	wReadSize							
5								
6	bI2C_Status							
7	00h							
8	bRdData[0]							
. . .	. . .							
n	bRdData[x]							

- bCSWStatus
  The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved. ・ Data “regulated value” was not set in bEnRepeatedStartCondition. ・ Data “outside range” was set in wWriteSize and wReadSize. ・ Data “00h” was set in both WWriteSize and wReadSize. ・ Data “outside range” was set in bSlaveAddres

- wReadSize
  These bytes show the amount (size) of data read from the I2C device.

- bI2C_Status
  This byte shows I2C transfer status.

Value	Explanation
00h	Command was received correctly.
01h	There was no response from I2C device.
02h	Data transfer was stopped while data write.

- bRdData
  These bytes are the read data field from the I2C device.

10.11 SPI_CONFIG

This command sets the SPI configuration.

When SPI is used, this command must be executed.

The SPI_CONFIG command is shown in Table 10.21 and the status is shown in Table 10.22.

Table 10.21 SPI_CONFIG Command

	7	6	5	4	3	2	1	0
0	40h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	bCh0_CPHA	bCh0_CPOL	bCh0_SS		bCh0_INT		bCh0_LSB	0b
5	bCh0_TransferRate							
6	bCh0_SS_Mode							
7	00h							
8	bCh1_CPHA	bCh1_CPOL	bCh1_SS		00b		bCh1_LSB	0b
9	bCh1_TransferRate							
10	bCh1_SS_Mode							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							

- bCh0_CPHA/bCh1_CPHA
  These bits specify phase of SPI clock pin.

Value	Explanation
0b	When the clock becomes active, data becomes effective.
1b	When the clock becomes non-active, data becomes effective.

- bCh0_CPOL/bCh1_CPOL
  These bits specify polarity of SPI clock pin.

Value	Explanation
0b	Active High
1b	Active Low

- bCh0_SS/bCh1_SS
  These bits specify polarity of Slave Select pin.

Value	Explanation
00b	Not use
01b	
10b	Active High
11b	Active Low

- bCh0_INT
  These bits specify polarity of INT0 pin.

Value	Explanation
00b	Not use
01b	
10b	Active Low
11b	Active High

- bCh0_LSB/bCh1_LSB
  This bit specifies data arrangement.

Value	Explanation
0b	MSB First
1b	LSB First

- bCh0_TransferRate/bCh1_TransferRate
  These bytes specify transfer rate.

Value	Explanation
01h	15Mbps
02h	7.5Mbps
03h	3.75Mbps
04h	1.87Mbps
05h	937Kbps
06h	468Kbps
07h	234Kbps
08h	117Kbps
09h	58.5Kbps
0Ah	29.2Kbps
0Bh	14.6Kbps
0Ch	7.3Kbps
0Dh	3.6Kbps
0Eh	1.8Kbps

- bCh0_SS_Mode/bCh1_SS_Mode
  These bytes specify the mode of the Slave Select pin.
  Please refer to “Figure 10.1 Difference of bChx_SS_Mode” for the differences between each mode.

Value	Explanation
00h	SS is asserted forward one byte.
01h	All SS is asserted until the data transfer is completed.

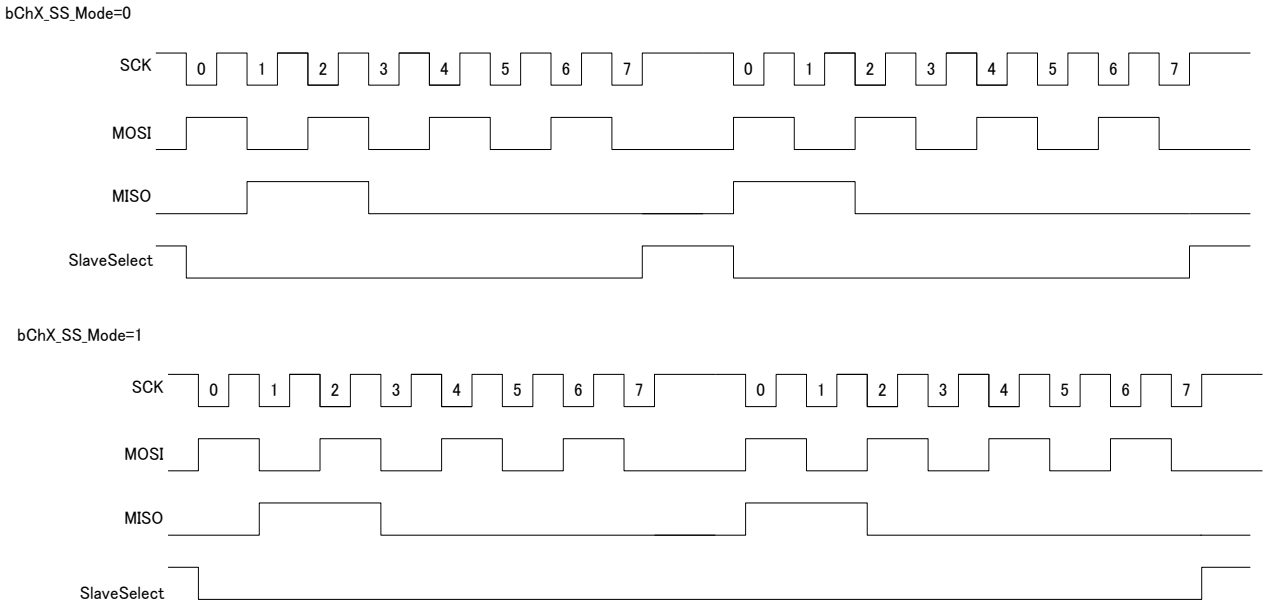

Figure 10.1 Difference of bChx_Mode

Table 10.22 SPI_CONFIG Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							

- bCSWStatus

The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved. ・ Data “regulated value” was not set in bCh0_TransferRate/bCh1_TransferRate. ・ Data “regulated value” was not set in bCh0_SS_Mode/bCh1_SS_Mode.
CMD_ERROR	There was an error in the following parameters. ・ When the SPI sequential command was executed.

10.12 SPI_ACCESS

This command is used access to The SPI device.

There are three types of accesses.

The SPI_ACCESS command is shown in Table 10.23 and the status is shown in Table 10.24.

Table 10.23 SPI_ACCESS Command

	7	6	5	4	3	2	1	0
0	41h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	00h						bChannel	
5	00h							
6	00h							
7	00h							
8	wWriteSize							
9								
10	00h							
11	00h							
12	wReadSize							
13								
14	00h							
15	00h							
16	bWrData[0]							
. . .	. . .							
n	bWrData[x]							

- bChannel
  These bits specify channel number.

Value	Explanation
00b	Ch0
01b	Ch1
10b	Ch1 (Serial flash ROM)

- wWriteSize
  These bytes specify the amount (size) of write data to the SPI device in the range of “0000h-0400h”.
  When there are no write accesses, 0000h" is specified.

- wReadSize
  These bytes specify the amount (size) of read data from the SPI device in the range of “0000h-0400h”.
  When there are no read accesses, “0000h" is specified.

- bWrData
  These bytes are the write data field to the SPI device.

Access	wWriteSize	wReadSize
Write	Except “0000h”	“0000h”
Read	“0000h”	Except “0000h”
Read after Write	Except “0000h”	Except “0000h”

Table 10.24 SPI_ACCESS Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	wReadSize							
5								
6	00h							
7	00h							
8	bRdData[0]							
. . .	. . .							
n	bRdData[x]							

- bCSWStatus
  The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved. ・ Data “regulated value” was not set in bChannel. ・ Data “outside range” was set in wWriteSize and wReadSize. ・ Data “00h” was set in both WWriteSize and wReadSize.
CMD_ERROR	There was the following errors. ・ "SPI_CONFIG “ command was not executed.

- wReadSize
  These bytes show the amount (size) of read data from the SPI device.

- bRdData
  These bytes are the read data field from SPI device.

10.13 SPI_SEQUENCE_START

This command starts the SPI sequential function. Please refer to "Appendix-C SPI Sequential Control" for information on setting up the SPI sequential procedure.

The following methods are used to stop the SPI sequential function.

- "SPI_SEQUENCE_STOP (44h)" command
- USB bus reset
- USB cable disconnection
- Hardware reset
- Software reset

The SPI_SEQUENCE_START command is shown in Table 10.25 and tThe status is shown in Table 10.26.

Table 10.25 SPI_SEQUENCE_START Command

	7	6	5	4	3	2	1	0
0	43h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	00h						bChannel	
5	bMode							
6	bCycleTime							
7	00h							
8	wSequenceSize							
9								
10	00h							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							
16	bSequence[0]							
17	bData[0]/bSequence[1]							
. . .	. . .							
n	bSequence[x]/bData[x]							

- bChannel
  These bits specify the channel number.

Value	Explanation
00b	Ch0
01b	Ch1

- bMode
  These bytes specify start trigger for the SPI sequence.

Value	Explanation
00h	Constant interval of bCycTime wait time.
01h	Detection of INT0 pin change. (Note)

  Note: INT0 interrupt should be enabled with the "SPI_CONFIG (20h)" command.

- bCycleTime
  This byte specifies constant interval (in msec) of the sequence execution in the range of “01h-FFh”. (Ex. 1Ah = 26ms)
  When bMode is specified as “01h", this byte is not used.

- wSequenceSize
  These bytes specify size of the SPI sequence data in the range of “0001h-0100h”.

- bSequence
  These bytes specify the operation of the SPI sequence.

Value	Name	Explanation
00h	WRITE	One byte is written.
01h	READ	One byte is read. (Note)
02h	SS_ASSERT	Slave Select is asserted.
03h	SS_NEGATE	Slave Select is negated.
04h	INT_DIS	The detection of INT0 change is disabled.
05h	INT_ENB	The detection of INT0 change is enabled.
06h	WAIT	Waiting time (bData) is installed.

  Note: Maximum number in a sequence is 58.

- bData
  These bytes are the write data field or the wait time field.
  These bytes are used when bSequence is specified as WRITE or WAIT.

Table 10.26 SPI_SEQUENCE_START Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							

- bCSWStatus
  The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved. ・ Data “regulated value” was not set in bChannel. ・ Data “regulated value” was not set in bMode. ・ Data “regulated value” was not set in wSequenceSize. ・ Data “over 59” was set in READ of bSequence. ・ Data “0000h” was not set in wCBWReserved. ・ Data “regulated value” was not set in bSequence. ・ Data “01h” was set in bMode when the interrupt is not used by “SPI_CONFIG”.
CMD_ERROR	There was the following errors. ・ “SPI_SEQUNCE_START" had already been started. ・ “SPI_CONFIG" was not set.

10.14 SPI_SEQUENCE_STOP

This command stops the SPI sequential function.

The SPI_SEQUENCE_STOP command is shown in Table 10.27 and the status is shown in Table 10.28.

Table 10.27 SPI_SEQUENCE_STOP Command

	7	6	5	4	3	2	1	0
0	44h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	00h						bChannel	
5	00h							
6	00h							
7	00h							
8	00h							
9	00h							
10	00h							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							

- bChannel
  These bits specify channel number.

Value	Explanation
00b	Ch0
01b	Ch1

Table 10.28 SPI_SEQUENCE_STOP Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							

- bCSWStatus

The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved. ・ Data “regulated value” was not set in bChannel.
CMD_ERROR	There was the following errors. ・ “SPI_CONFIG" was not set.

10.15 GPIO_CONFIG

This command sets the GPIO pins configuration.

The GPIO_CONFIG command is shown in Table 10.29 and the status is shown in Table 10.30.

Table 10.29 GPIO_CONFIG Command

	7	6	5	4	3	2	1	0
0	80h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	bmGPIOA_Dir[7:0]							
5	bmGPIOB_Dir[7:0]							
6	bmGPIOA_Pullup[7:0]							
7	bmGPIOB_Pullup[7:0]							
8	bKeyScanLine							
9	00h							
10	00h							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							

- bmGPIOA_Dir[7:0]/bmGPIOB_Dir[7:0]
  This bit specifies the direction of the GPIO pins.
  The default value is input.

Value	Explanation
0b	Input
1b	Output

- bmGPIOA_PullUp[7:0]/bmGPIOB_PullUp[7:0]
  This bit specifies ON/OFF control of the GPIO pull-up resistor.
  The default value is Pull-up ON.

Value	Explanation
0b	Pull-up OFF
1b	Pull-up ON

- bKeyScanLine
  This byte specifies the number of key-scan lines.
  The default value is unused.

Value	Explanation
00h	Unused
02h	8x2Line(GPIOA [7:0] and GPIOB [1:0])
04h	8x4Line(GPIOA [7:0] and GPIOB [3:0])
08h	8x8Line(GPIOA [7:0] and GPIOB [7:0])

Table 10.30 GPIO_CONFIG Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							

- bCSWStatus

  The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved. ・ Data “regulated value” was not set in bKeyScanLine.
CMD_ERROR	There was the following errors. ・ Key-scan had already been started by “KEYSCAN_CONTROL ". ・ GPIO interrupt was enabled by “GPIO_INT_CONTROL”.

10.16 GPIO_INT_CONFIG

This command configures the GPIO interrupt.

This command is used for the GPIO input pins.

The GPIO_INT_CONFIG command is shown in Table 10.31 and the status is shown in Table 10.32.

Table 10.31 GPIO_INT_CONFIG Command

	7	6	5	4	3	2	1	0
0	81h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	bmGPIOA_IntMode [7:0]							
5	bmGPIOB_IntMode [7:0]							
6	bmGPIOA_IntLevel [7:0]							
7	bmGPIOB_IntLevel [7:0]							
8	00h							
9	00h							
10	00h							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							

- bmGPIOA_IntMode[7:0]/bmGPIOB_IntMode[7:0]
  This byte specifies the interrupt mode of the GPIO pin.

Value	Explanation
0b	Level interrupt
1b	Edge interrupt

- bmGPIOA_IntLevel[7:0]/bmGPIOB_IntLevel[7:0]
  This byte specifies the interrupt logic of the GPIO pin.
  This byte is used only when level interrupt is selected.

Value	Explanation
0b	Active Low
1b	Active High

Table 10.32 GPIO_INT_CONFIG Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							

- bCSWStatus

The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved.
CMD_ERROR	There was the following errors. ・ GPIO direction was not set as input pin by “GPIO_CONFIG ". ・ GPIO interrupt was enabled by GPIO_INT_CONTROL".

10.17 GPIO_INT_CONTROL

This command configures the GPIO event notification.

This command is used for GPIO input pins.

The event notification is enabled when this command is executed.

The following methods can be used to stop the GPIO event notification.

- "GPIO_INT_CONTROL (82h)" command
- USB bus reset
- USB cable disconnection
- Hardware reset
- Software reset

The GPIO_INT_CONTROL command is shown in Table 10.33 and the status is shown in Table 10.34.

Table 10.33 GPIO_INT_CONTROL Command

	7	6	5	4	3	2	1	0
0	82h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	bmGPIOA_IntEnable[7:0]							
5	bmGPIOB_IntEnable[7:0]							
6	bmGPIOA_IntPosEdge[7:0]							
7	bmGPIOB_IntPosEdge[7:0]							
8	bmGPIOA_IntNegEdge[7:0]							
9	bmGPIOB_IntNegEdge[7:0]							
10	00h							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							

- bmGPIOA_IntEnable/bmGPIOB_IntEnable
  This bit enables/disables event notification.

Value	Explanation
0b	Disable
1b	Enable

- bmGPIOA_PosEdge/bmGPIOB_PosEdge
  This bit enables interrupt on the rising edge.
  This bit is used when GPIO edge interrupt is selected with the "GPIO_INT_CONFIG (81h) "command.

Value	Explanation
0b	Disable
1b	Enable

- bmGPIOA_NegEdge/bmGPIOB_NegEdge
  This bit enables interrupt on the falling edge.
  This bit is used when GPIO edge interrupt is selected with the "GPIO_INT_CONFIG (81h)" command.

Value	Explanation
0b	Disable
1b	Enable

Table 10.34 GPIO_INT_CONTROL Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							

- bCSWStatus
  The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data "00h" was not set in data field of "00h". ・ Data "0000h" was not set in wCBWReserved.
CMD_ERROR	There was the following errors. ・ GPIO direction was not set as input pin by "GPIO_CONFIG ". ・ GPIO interrupt was not set by "GPIO_INT_CONFIG ".

10.18 GPIO_READ

This command reads input status from the GPIO pin.

The GPIO_READ command is shown in Table 10.35 and the status is shown in Table 10.36.

Table 10.35 GPIO_READ Command

	7	6	5	4	3	2	1	0
0	83h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							
8	00h							
9	00h							
10	00h							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							

Table 10.36 GPIO_READ Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	0002h(wReadSize)							
5								
6	00h							
7	00h							
8	bmGPIOA_Status[7:0]							
9	bmGPIOB_Status[7:0]							

- bCSWStatus
  The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved.

- wReadSize
  These bytes show the amount (size) of read data. These bytes are fixed to “0002h”.

- bmGPIOA_Status/bmGPIOB_Status
  This bit shows the input status of the GPIO pin.
  When GPIO pin is used as key-scan, this bit becomes “0b".

Value	Explanation
0b	Low
1b	High

10.19 GPIO_WRITE

This command writes output data to the GPIO pin.

This command is used when the GPIO pin is set as output by the “GPIO_CONFIG (80h)” command.

The GPIO_WRITE command is shown in Table 10.37 and the status is shown in Table 10.38.

Table 10.37 GPIO_WRITE Command

	7	6	5	4	3	2	1	0
0	84h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							
8	0002h(wWriteSize)							
9								
10	00h							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							
16	bmGPIOA[7:0]							
17	bmGPIOB[7:0]							

- wWriteSize
  These bytes specify the amount (size) of write data. These bytes are fixed to “0002h".

- bmGPIOA[7:0]/bmGPIOB[7:0]
  This bit specifies the output status of the GPIO pin.

Value	Explanation
0b	Low
1b	High

Table 10.38 GPIO_WRITE Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							

- bCSWStatus

  The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved. ・ Data “0002h” was not set in wWriteSize.
CMD_ERROR	There was the following errors. ・ GPIO was not set as output pin by “GPIO_CONFIG (80h)”.

10.20 KEYSCAN_CONTROL

This command sets the key-scan configuration.

This command is used when key-scan is enabled with the "GPIO_CONFIG (80h)" command.

The following methods can be used to stop the key-scan operation.

- "KEYSCAN_CONTROL (90h)" command
- USB bus reset
- USB cable disconnection
- Hardware reset
- Software reset

The command is shown in Table 10.39. The status is shown in Table 10.40.

Table 10.39 KEYSCAN_CONTROL Command

	7	6	5	4	3	2	1	0
0	90h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	bKeyScanEnable							
5	bKeyScanDriveMode							
6	bKeyScanClock							
7	bKeyScanSamplingClock							
8	bKeyScanInterval							
9	00h							
10	00h							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							

- bKeyScanEnable
  This byte specifies the operation control of the key-scan.

Value	Explanation
00h	Stop
01h	Start

- bKeyScanDriveMode
  This byte specifies the operation control of the key-scan.

Value	Explanation
00h	Hi-Z drive mode
01h	High drive mode

- bKeyScanClock
  This byte specifies the sampling clock.

Value	Explanation
00h	12MHz
01h	6MHz
02h	3MHz
03h	1.5MHz

- bKeyScanSamplingClock
  This byte specifies number of clocks per 1line.

Value	Explanation
00h	2Clock
01h	4Clock
02h	6Clock
03h	8Clock

- bKeyScanInterval
  This byte specifies the scanning interval.

Value	Explanation
00h	1.365msec
01h	2.731msec
02h	5.461msec
03h	10.92msec

Table 10.40 KEYSCAN_CONTROL Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							

- bCSWStatus
  The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved. ・ Data “regulated value” was not set in bKeyScanEnable. ・ Data “regulated value” was not set in bKeyScanClock. ・ Data “regulated value” was not set in bKeyScanSamplingClock. ・ Data “regulated value” was not set in bKeyScanInterval.
CMD_ERROR	There was the following errors. ・ Key-scan was not set by “GPIO_CONFIG (80h) “.

10.21 KEYSCAN_READ

This command reads key-scan data.

The following are necessary to use this command.

- Key-scan is enabled by “GPIO_CONFIG “command.
- Key-scan is started by ”KEYSCAN_CONTROL” command.

The KEYSCAN_READ command is shown in Table 10.41 and the status is shown in Table 10.42.

Table 10.41 KEYSCAN_READ Command

	7	6	5	4	3	2	1	0
0	91h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							
8	00h							
9	00h							
10	00h							
11	00h							
12	wReadSize							
13								
14	00h							
15	00h							

- wReadSize
  These bytes specifis the amount (size) of read data.

Value	Explanation
02h	2 bytes (2Line)
04h	4 bytes (4Line)
08h	8 bytes (8Line)

Table 10.42 KEYSCAN_READ Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	wReadSize							
5								
6	00h							
7	00h							
8	bKeyScanData0[7:0]							
9	bKeyScanData1[7:0]							
10	bKeyScanData2[7:0]							
11	bKeyScanData3[7:0]							
12	bKeyScanData4[7:0]							
13	bKeyScanData5[7:0]							
14	bKeyScanData6[7:0]							
15	bKeyScanData7[7:0]							

- bCSWStatus

  The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved. ・ Data “number of key-scan line” was not set in wReadSize correctly.
CMD_ERROR	There was the following errors. ・ Key-scan was not set by “GPIO_CONFIG (80h) “. ・ Key-scan was not started by “KEYSCAN_CONTROL".

- wReadSize

  These bytes show the amount (size) of read data from the key-scan.

Value	Explanation
02h	2 bytes (2Line)
04h	4 bytes (4Line)
08h	8 bytes (8Line)

- bKeyScanData0-7

  This bit shows the status of the key-scan data.

Value	Explanation
0b	Key is not pushed.
1b	Key is pushed.

10.22 BUZZER_CONTROL

This command controls the buzzer.

The BUZZER_CONTROL command is shown in Table 10.43 and the status is shown in Table 10.44.

Table 10.43 BUZZER_CONTROL Command

	7	6	5	4	3	2	1	0
0	B0h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	bBuzzerControl							
5	00h							
6	bBuzzerCycle							
7	00h							
8	bBuzzerLen							
9	00h							
10	00h							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							

- bBuzzerControl
  This byte specifies operation control of the buzzer.

Value	Explanation
00h	Buzzer stop
01h	Buzzer start

- bBuzzerCycle
  These bytes specify the High/Low cycle of buzzer.
  The cycle time T can be calculated as following. (n = these bytes)
  $T = 10.67us \times 2 \times (n+1)$

Value	Explanation
00h	21.33us(46.88kHz)
01h	42.68us(23.43kHz)
.	.
.	.
.	.
FFh	5.46ms(183Hz)

- bBuzzerLen
  These bytes specify output length of the buzzer in the range of “00h-0Eh”.

Value	Explanation
00h	100msec
01h	200msec
02h	300msec
03h	400msec
04h	500msec
05h	600msec
06h	700msec
07h	800msec
08h	900msec
09h	1000msec
0Ah	1100msec
0Bh	1200msec
0Ch	1300msec
0Dh	1400msec
0Eh	1500msec

Table 10.44 BUZZER_CONTROL Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							

- bCSWStatus
  The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved. ・ Data “regulated value” was not set in bBuzzerControl. ・ Data “regulated value” was not set in bBuzzerLen.

10.23 EVENT_INT_CONTROL

This command controls event notification.

The following events cannot be detected in the default setting.

- “LCDC_EVENT (00h)”
- ”WAKEUP_EVENT (81h)”
- “ SPI_INT_EVENT (40h)”

The above events notify at once, and next event detection becomes disabled. Therefore if next event detect is necessary, this command should be executed again.

“SPI_INT_EVENT (40h)” cannot be executed in the following cases.

- The SPI sequential control is operating.
- SPI interrupt is not used by “SPI_CONFIG (40h) “command.

The EVENT_INT_CONTROL command is shown in Table 10.45 and the status is shown in Table 10.46.

Table 10.45 EVENT_INT_CONTROL Command

	7	6	5	4	3	2	1	0
0	C0h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	bType							
5	00h							
6	00h							
7	00h							
8	00h							
9	00h							
10	00h							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							

- bType
  This byte specifies the event to enable.

Value	Explanation
00h	“LCDC_EVENT "(00h)
01h	“SPI_INT_EVENT "(40h)
02h	“WAKEUP_EVENT" (81h)

Table 10.46 EVENT_INT_CONTROL Status

	7	6	5	4	3	2	1	0
0	bCSWStatus							
1	bCSWTag							
2	0000h (wCSWReserved)							
3								
4	00h							
5	00h							
6	00h							
7	00h							

- bCSWStatus

  The feedback status from the command is as follows.

SUCCESS	Command was received correctly.
INVALID_PARAM	There was an error in the following parameters. ・ Data “00h” was not set in data field of “00h”. ・ Data “0000h” was not set in wCBWReserved. ・ Data “regulated value” was not set in bType.
CMD_ERROR	There was the following errors. ・ SPI interrupt is not used by “SPI_CONFIG (40h)” command. ・ Ch0 SPI sequential control is operating.

11. Event Description

In this section, the S1D13U11 events are described.

11.1 LCDC_EVENT

This event shows interrupts on the LCD interface.
Please refer to “S1D13U11 Hardware Technical Manual” for the details of the interrupt factor.
This event notifies at once, and next event detection becomes disabled. Therefore, if next event detect is necessary, the “EVENT_INT_CONTROL (C0h)” command should be executed again.
The LCDC_EVENT event is shown in Table 11.1.

Table 11.1 LCDC_EVENT

	7	6	5	4	3	2	1	0
0	00h (bEventType)							
1	00h (bEventReserved)							
2	0000h(wLength)							
3								

11.2 SPI_INT_EVENT

This event indicates changes of the INT0 pin as an SPI interrupt.
This event notifies at once, and next event detection becomes disabled. Therefore, if next event detect is necessary, the "EVENT_INT_CONTROL (C0h)" command should be executed again.
The SPI_INT_EVENT event is shown in Table 11.2.

Table 11.2 SPI_INT_EVENT

	7	6	5	4	3	2	1	0
0	40h (bEventType)							
1	00h (bEventReserved)							
2	0002h(wLength)							
3								
4	00h						00b (bChannel)	
5	00h							

- bChannel
  These bits specify the channel number of the interrupt pin. This bit is fixed to "00b".

11.3 SPI_SEQ_EVENT

This event indicates the finish of SPI sequential control.
The SPI_SEQ_EVENT event is shown in Table 11.3.

Table 11.3 SPI_SEQ_EVENT

	7	6	5	4	3	2	1	0
0	41h (bEventType)							
1	00h (bEventReserved)							
2	wLength							
3								
4	00h						00b (bChannel)	
5	00h							
6	bSequenceData[0]							
. . .	. . .							
n	bSequenceData[x]							

- wLength
  These bytes show amount (size) of read data.

- bChannel
  These bits show the channel number of the interrupt pin. This bit is fixed to “00b".

- bSequenceData
  These bytes are the read data field.

11.4 GPI_EVENT

This event shows changes on the GPIO input.
The GPI_EVENT event is shown in Table 11.4.

Table 11.4 GPI_EVENT

	7	6	5	4	3	2	1	0
0	80h (bEventType)							
1	00h (bEventReserved)							
2	0004h(wLength)							
3								
4	bmGPIOA_IntStatus[7:0]							
5	bmGPIOB_IntStatus[7:0]							
6	bmGPIOA_Status[7:0]							
7	bmGPIOB_Status[7:0]							

- bmGPIOA_IntStatus/bmGPIOB_IntStatus
  This bit shows changes on the GPIO input.

Value	Explanation
0b	There is no change.
1b	There is a change.

- bmGPIOA_Status/bmGPIOB_Status
  This bit shows the status of the GPIO pin.
  When key-san is selected, this bit becomes “0b".

Value	Explanation
0b	Low
1b	High

11.5 WAKEUP_EVENT

This event shows a change of Wakeup key (INT1).
This event notifies at once, and next event detection becomes disabled. Therefore, if next event detect is necessary, the "EVENT_INT_CONTROL (C0h)" command should be executed again.
The WAKEUP_EVENT event is shown in Table 11.5.

Table 11.5 WAKEUP_EVENT

	7	6	5	4	3	2	1	0
0	81h (bEventType)							
1	00h (bEventReserved)							
2	0000h(wLength)							
3								

11.6 KEYSCAN_EVENT

This event shows changes in key-scan data.
The KEYSCAN_EVENT event is shown in Table 11.6.

Table 11.6 KEYSCAN_EVENT

	7	6	5	4	3	2	1	0
0	90h (bEventType)							
1	00h (bEventReserved)							
2	wLength							
3								
4	bKeyScanData0[7:0]							
5	bKeyScanData1[7:0]							
6	bKeyScanData2[7:0]							
7	bKeyScanData3[7:0]							
8	bKeyScanData4[7:0]							
9	bKeyScanData5[7:0]							
10	bKeyScanData6[7:0]							
11	bKeyScanData7[7:0]							

- wLength
  These bytes show amount (size) of read data.

Value	Explanation
02h	2 bytes (2Line)
04h	4 bytes (4Line)
08h	8 bytes (8Line)

- bKeyScanData0-7
  This bit shows the status of the key-scan.

Value	Explanation
0b	Key is not pushed.
1b	Key is pushed.

12. Error Processing

In this section, S1D13U11 error processing is described.

12.1 Error Recovery

When a command error is detected (bCSWStatus: except SUCCESS), the S1D13U11 is “STALL" for ENDPOINT1 (command transmission) and ENDPOINT2 (status reception). The host CPU must control the error recovery with the following procedures.

1． Clear STALL of ENDPOINT1 (command transmission) *Note
2． Clear STALL of ENDPOINT2 (status reception) *Note
3． Status block is received.

Note: Clear by USB standard request CLEAR_FEATURE (ENDPOINT_HALT).

The error recovery procedure is shown in Figure 12.1.

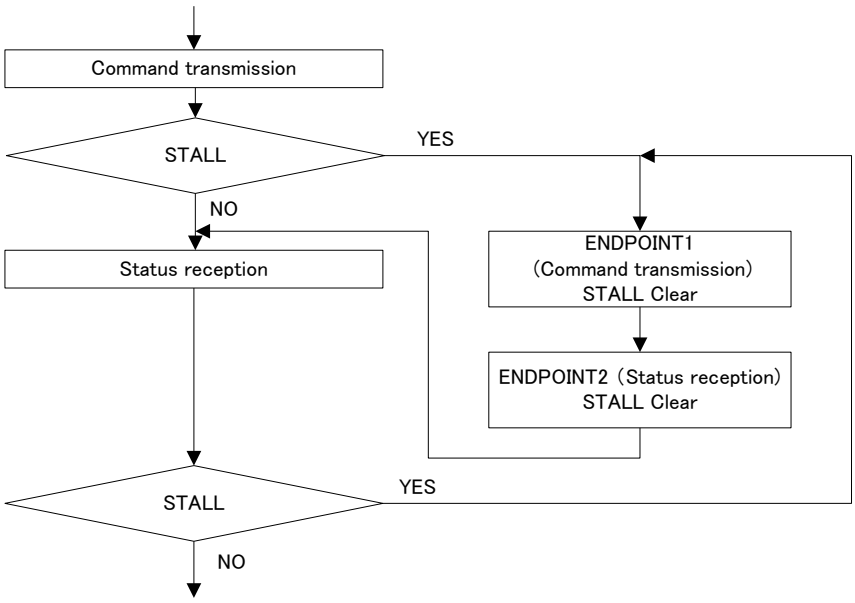

Figure 12.1 Error Recovery flow

Appendix-A Configuration Data Format

This section describes the S1D13U11 configuration data format.
There are two types of configuration data; for USB download and for serial flash ROM.
The USB download data is the format of the basic configuration data. The serial flash ROM data is a format in which additional information is added to the USB download data.

The structure of configuration data is shown in Figure A.1.

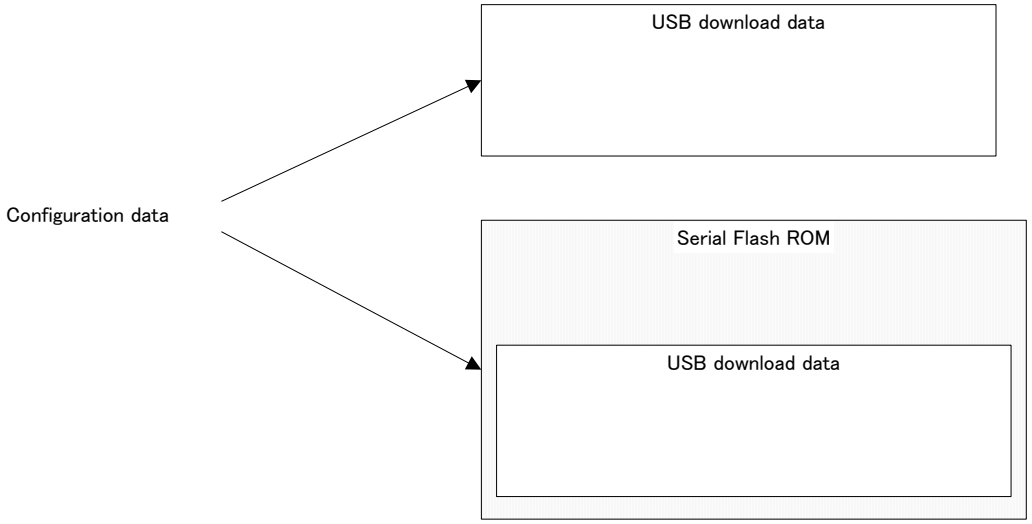

Figure A.1 Structure of Configuration Data

A.1 For USB download

The configuration data for the USB download is shown in Table A.1.

Table A.1 Data for USB Download

Type		Offset	Size (Byte)	Explanation	M/O note 1
Sequencer data					
	Sequencer Data	000000h	90112	Sequencer data from Epson.	M
USB data					
	USB ID	016000h	2	Whether idVendor, idProduct, and bcdDevice are changed is specified. Bit15-3: “0b” fix Bit2: bcdDevice is changed. Bit1: idProduct is changed. Bit0: idVendor is changed. For example: 03h When idVendor and idProduct are changed.	O
		016002h	2	IdVendor is specified.	O
		016004h	2	IdProduct is specified.	O
		016006h	2	BcdDevice is specified.	O
	LangID	016008h	4	WLangID of String Language ID Descriptor is specified.	O
		01600Ch	4	Please fill it by “00h".	-
	String1	016010h	128	Please specify the value in which manufacturer of String Descriptor is shown with UNICODE. Note 2	O
	String2	016090h	128	Please specify the value in which product of String Descriptor is shown with UNICODE. Note 2	O
	String3	016110h	128	Please specify the value in which serial number of String Descriptor is shown with UNICODE. Note 2	O
Reserved					
	Reserved	016190h	7790	Please fill it by “00h".	-
CRC					
	CRC	017FFE	2	CRC value from “000000h” to “017FFDh”. CRC is calculated as follows. CRC-16-CCIT $X^{16} + X^{12} + X^5 + 1$ Initial value: “FFFFh"	-

Note 1: M must set the value. O: Please fill with “00h" when not using.

Note 2: Please fill unused area with “00h".

A.2 For Serial Flash ROM

The configuration data for Serial Flash ROM is shown in Table A.2.

Table A.2 Data for Serial Flash ROM

Type		Offset	Size (Byte)	Explanation	M/O note 1
Header					
	Identifier	000000h	8	Please set "S1D13U11 "by ASCII code. S1D13U11: 53 31 44 31 33 55 31 31	M
	Reserved	000008h	8	Please fill it by “00h".	-
Data for USB download					
	Data for USB download	000010h	98304	Data for the Appendix-A.1 USB download.	M
GPIO setting					
	GPIO composition	018200h	16	The configuration of GPIO is specified. “GPIO_CONFIG (80h)" command block is specified.	O
	Reserved	018210h	3568	Please fill it by “00h".	-
Start-up display					
	LCDCReg Count	019000h	2	The value in which the number of data of LCDCInitialReg fields is divided by four is specified. $LCDC\,\mathrm{Re}\,gCount = LCDCInitial\,\mathrm{Re}\,g / 4$	O
	Picture size	019002h	2	The number of data of Picture Data fields is specified.	O
	LCDCInitialReg	019004h	1020	The register address and the write data in the LCD interface are specified. The format is the same as wRegAddress and wWriteData of “LCDC_WRITE (02h)" command. A special command can be specified in this field. Note 2	O
	Reserved	019400h	3072	Please bury it by “00h".	-
	PictureData	01A000h	24576	The display data of the start-up display is specified. Only RGB 5:6:5 is supported.	O

Note 1: M must set the value. O: Please fill with “00h" when not using.

Note 2: Please refer Table A.3.

A.3 Special Command List

The special command list is shown in Table A.3.

Table A.3 Special Command List

Command	wRegAddress	wRegData	Explanation
WAIT	FFFFh	XXXXh	LCD interface registers are not written, and the wait time (in msec) is installed with the value of wRegData, specified in the range of 0000h - 00FFh.
WAIT_LOW	FFF0h	0000h	Wait until the wRegData bit of the wRegAddress following this command becomes “LOW". Example) wRegAddress[0] = FFF0h、wRegData[0] = 0000h wRegAddress[1] = 0090h、wRegData[1] = 0001h Waitfor register 0090h bit 0 to become “0".
WAIT_HIGH	FFF1h	0000h	Waits until the wRegData bit of the wRegAddress following this command becomes “High". Example) wRegAddress[0] = FFF1h、wRegData[0] = 0000h wRegAddress[1] = 0090h、wRegData[1] = 0010h Waitfor register 0090h bit 4 to become “1".
PIC_WRITE	FFF2h	0000h	The PictureData is transmitted to VRAM. This command is used when the LCD interface is controlled in the following order. 1. Initialization of LCD interface 2. VRAM write 3. LCD LED backlight ON When this command is not used, VRAM write is executed after LCDCInitialReg.

A.4 Start-up Display Setting Example

The sample settings for the start-up display are shown in Table A.4.
Please refer to the "S5U13U11P00C100 Evaluation Board User Manual" for information on the S1D13U11 evaluation board.

Table A.4 LCDCInitialReg Setting Sample

No	wRegAddress	wRegData	Register Name
1	006Eh	0003h	REG[6Eh] General Purpose Output Register 1
2	0004h	0001h	REG[04h] PLL Control Register
3	0006h	002Ah	REG[06h] PLL Setting Register 0
4	000Ch	000Ah	REG[0Ch] PLL Setting Register 3
5	000Eh	0080h	REG[0Eh] SS Control Register 0
6	0010h	0055h	REG[10h] SS Control Register 1
7	0012h	00C2h	REG[12h] Clock Source Select Register
8	0014h	0007h	REG[14h] LCD Panel Type Register
9	0016h	0027h	REG[16h] Horizontal Display Width Register (HDISP)
10	0018h	004Fh	REG[18h] Horizontal Non-Display Period Register (HNDP)
11	001Ah	00EFh	REG[1Ah] Vertical Display Height Register 0 (VDISP)
12	001Ch	0000h	REG[1Ch] Vertical Display Height Register 1 (VDISP)
13	001Eh	0016h	REG[1Eh] Vertical Non-Display Period Register (VNDP)
14	0020h	000Ah	REG[20h] PHS Pulse Width Register (HSW)
15	0022h	0076h	REG[22h] PHS Pulse Start Position Register (HPS)
16	0024h	0002h	REG[24h] PHS Pulse Width Register (VSW)
17	0026h	0028h	REG[26h] PVS Pulse Start Position Register (VPS)
18	0028h	0000h	REG[28h] PCLK Polarity Register
19	0082h	0003h	REG[82h] SDRAM Control Register 0
20	008Ch	00FFh	REG[8Ch] SDRAM Refresh Counter Register 0
21	008Eh	0003h	REG[8Eh] SDRAM Refresh Counter Register 1
22	0090h	0050h	REG[90h] SDRAM Write Buffer Memory Size Register 0
23	0004h	0081h	REG[04h] PLL Control Register
24	FFFFh	0001h	WAIT 1msec
25	006Eh	0001h	REG[6Eh] General Purpose Output Register 1
26	FFFFh	0001h	WAIT 1msec
27	0012h	00C2h	REG[12h] Clock Source Select Register
28	0068h	00E8h	REG[68h] Power Save Register
29	0068h	0000h	REG[68h] Power Save Register
30	0068h	0001h	REG[68h] Power Save Register
31	FFFFh	0001h	WAIT 1msec
32	0084h	0000h	REG[84h] SDRAM Status Register 0
33	FFFFh	0001h	WAIT 1msec
34	0084h	0082h	REG[84h] SDRAM Status Register 0
35	FFFFh	0001h	WAIT 1msec
36	002Ah	0000h	REG[2Ah] Display Mode Register
37	0052h	0000h	REG[52h] Input Mode Register
38	005Ah	0010h	REG[5Ah] Write Window X Start Position Register
39	005Ch	0016h	REG[5Ch] Write window Y Start Position Register 0
40	005Eh	0002h	REG[5Eh] Write window Y Start Position Register 1
41	0060h	003Ch	REG[5Ah] Write Window X End Position Register
42	0062h	0026h	REG[5Ch] Write window Y End Position Register 0
43	0064h	0002h	REG[5Eh] Write window Y End Position Register 1
44	002Ah	0001h	REG[2Ah] Display Mode Register

No	wRegAddress	wRegData	Register Name
45	0050h	0080h	REG[50h] Display Control Register
46	006Eh	0000h	REG[6Eh] General Purpose Output Register 1
47	FFFFh	0001h	WAIT 1msec
48	00B6h	0001h	REG[B6h] Interrupt Clear Register
49	00B6h	0000h	REG[B6h] Interrupt Clear Register
50	00B2h	0001h	REG[B2h] Interrupt Control Register
51	0098h	004Fh	REG[98h] Alpha-blending Horizontal Size Register
52	009Ah	003Bh	REG[9Ah] Alpha-blending Vertical Size Register 0
53	009Ch	0003h	REG[9Ch] Alpha-blending Vertical Size Register 1
54	009Eh	0060h	REG[9Eh] Alpha-blending Value Register
55	00A0h	0000h	REG[A0h] Alpha-blending Input 1 Start Address Register 0
56	00A2h	0000h	REG[A2h] Alpha-blending Input 1 Start Address Register 1
57	00A4h	0000h	REG[A4h] Alpha-blending Input 1 Start Address Register 2
58	00ACh	0000h	REG[ACh] Alpha-blending Output Start Address Register 0
59	00AEh	0000h	REG[AEh] Alpha-blending Output Start Address Register 1
60	00B0h	0000h	REG[B0h] Alpha-blending Output Start Address Register 2
61	0054h	00FFh	REG[54h] Transparency Key Color Red Register
62	0056h	00FFh	REG[56h] Transparency Key Color Green Register
63	0058h	00FFh	REG[58h] Transparency Key Color Blue Register
64	0094h	0001h	REG[94h] Alpha-blending Control Register
65	0094h	0000h	REG[94h] Alpha-blending Control Register
66	FFF1h	0001h	WAIT_HIGH
67	00B4h	0001h	REG[B4h] Interrupt Status Register
68	00B6h	0001h	REG[B6h] Interrupt Clear Register
69	00B6h	0000h	REG[B6h] Interrupt Clear Register
70	FFF2h	0000h	PIC_WRITE
71	0072h	004Bh	PWM High Duty Register 0
72	0074h	0000h	PWM High Duty Register 1
73	0076h	0000h	PWM High Duty Register 2
74	0078h	0000h	PWM High Duty Register 3
75	007Ah	0096h	PWM Low Duty Register 0
76	007Ch	0000h	PWM Low Duty Register 1
77	007Eh	0000h	PWM Low Duty Register 2
78	0080h	0000h	PWM Low Duty Register 3
79	0070h	0086h	PWM Control Register

A.5 Start-up Display Data Example

The S5U13U11P00C100 data sample for the start-up display is shown in Figure A.2.

Figure A.2 Start-up Data Sample (184x65 pixels)

Appendix-B Difference of bDataRegAcc

bDataRegAcc settings are shown in the following figures.

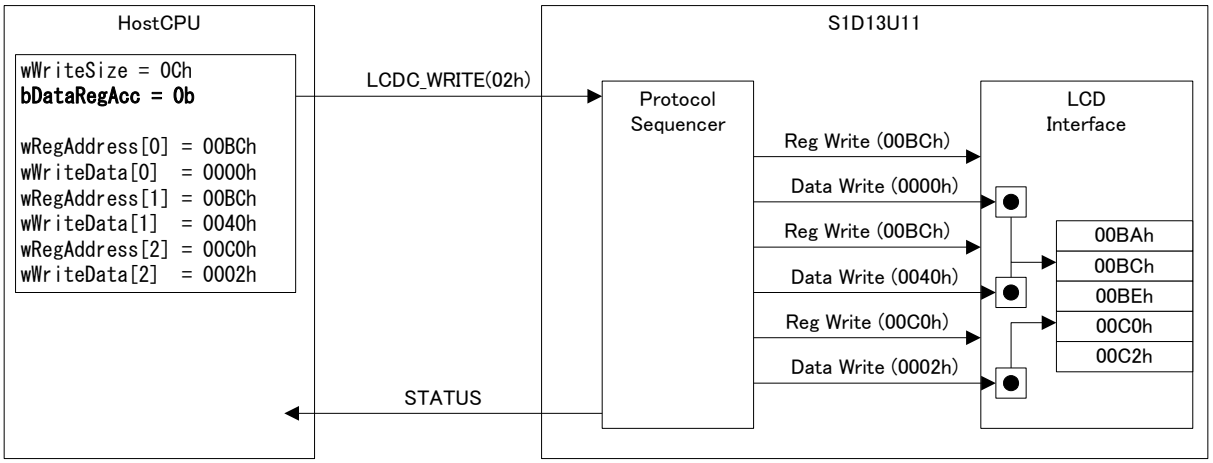

Figure B.1 Register Write Procedure (bDataRegAcc=0b)

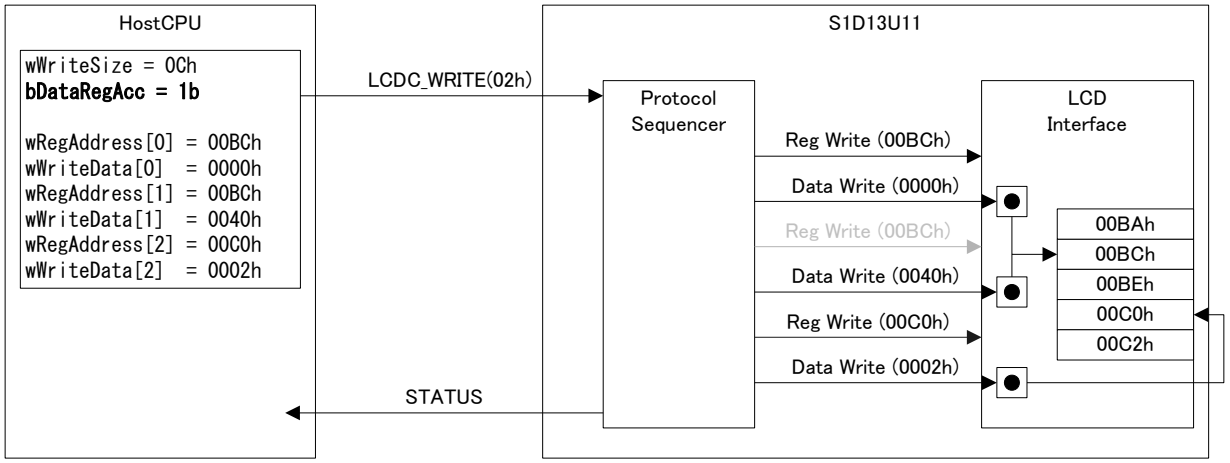

Figure B.2 Register Write Procedure (bDataRegAcc=1b)

When bDataRegAcc is set to “1b”, ”Reg Write(00BCh)” is not accessed two times. This function is used for the look-up table (register 00BCh). Please refer to “S1D13U11 Hardware Technical Manual” for the details on the look-up table. The register read procedure is also similar.

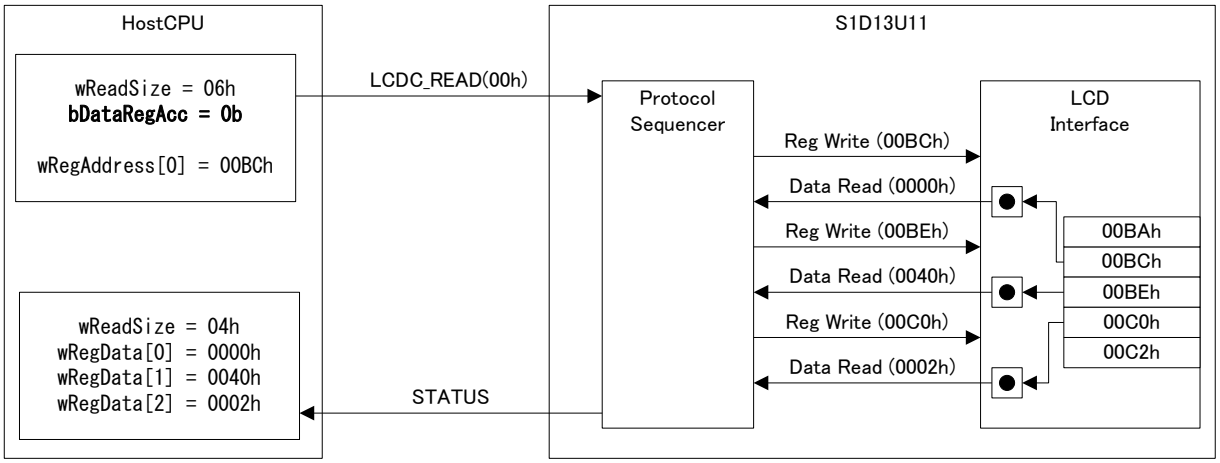

Figure B.3 Register Read Procedure (bDataRegAcc=0b)

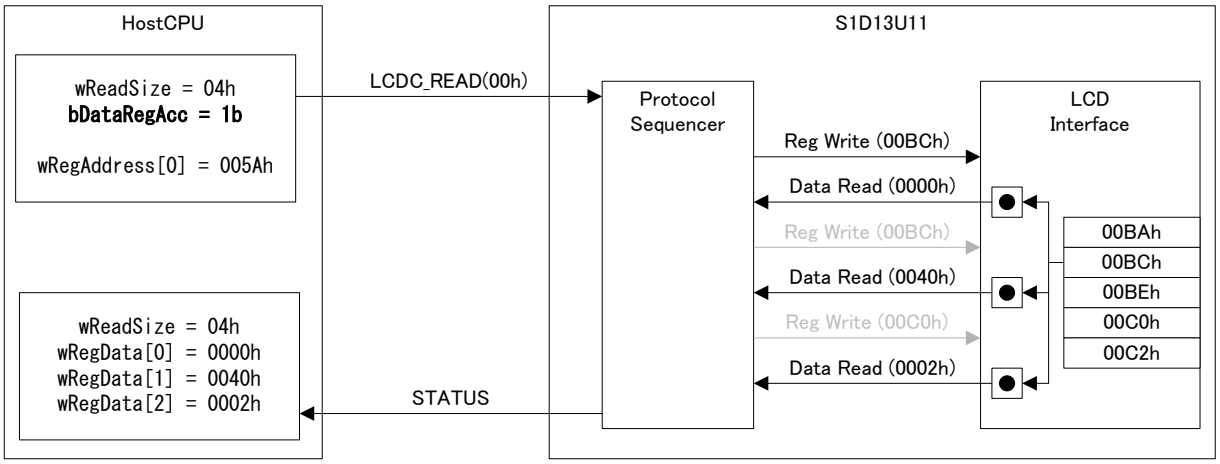

Figure B.4 Register Read Procedure (bDataRegAcc=1b)

Appendix-C SPI Sequential Control Setting Example

This section shows an example for setting the SPI sequential control when a TSC2046 (Texas Instruments) touch screen controller is connected to the S1D13U11.

Table C.1 Example of SPI Sequential Control

Offset	Field	Name	Value	Explanation
0	bSequence[0]	INT_DIS	04h	The interrupt of INT0 signal is disabled.
1	bSequence[1]	SS_ASSERT	02h	SS0 # signal is asserted.
2	bSequence[2]	WRITE	00h	Y position data is requested.
3	bData	-	90h	
4	bSequence[3]	READ	01h	Y position data is read.
5	bSequence[4]	READ	01h	
6	bSequence[5]	SS_NEGATE	03h	SS0 # signal is negated.
7	bSequence[6]	SS_ASSERT	02h	SS0 # signal is asserted.
8	bSequence[7]	WRITE	00h	Z1 position data is requested.
9	bData	-	B0h	
10	bSequence[8]	READ	01h	Z1 position data is read.
11	bSequence[9]	READ	01h	
12	bSequence[10]	SS_NEGATE	03h	SS0 # signal is negated.
13	bSequence[11]	SS_ASSERT	02h	SS0 # signal is asserted.
14	bSequence[12]	WRITE	00h	Z2 position data is requested.
15	bData	-	C0h	
16	bSequence[13]	READ	01h	Z2 position data is read.
17	bSequence[14]	READ	01h	
18	bSequence[15]	SS_NEGATE	03h	SS0 # signal is negated.
19	bSequence[16]	SS_ASSERT	02h	SS0 # signal is asserted.
20	bSequence[17]	WRITE	00h	X position data is requested.
21	bData	-	D0h	
22	bSequence[18]	READ	01h	X position data is read.
23	bSequence[19]	READ	01h	
24	bSequence[20]	SS_NEGATE	03h	SS0 # signal is negated.
25	bSequence[21]	INT_ENB	05h	The interrupt of INT0 signal is enabled.

The example data for “SPI_SEQ_EVENT” (41h) is shown in Table C.2.

Table C.2 Example of SPI_SEQ_EVENT Data

Offset	Field	Value	Explanation
0	bSequenceData[0]	XXh	Y position data
1	bSequenceData[1]	XXh	
2	bSequenceData[2]	XXh	Z1 position data
3	bSequenceData[3]	XXh	
4	bSequenceData[4]	XXh	Z2 position data
5	bSequenceData[5]	XXh	
6	bSequenceData[6]	XXh	X position data
7	bSequenceData[7]	XXh	

Appendix-D Example of Command Sample

This section gives and example of command sample as shown in the following tables.

D.1 SPI_CONFIG

An example of the “SPI_CONFIG (40h)” command is shown in Table D.1 and Table D.2.

Table D.1 SPI Setting Example

	Ch0	Ch1
Clock phase	Low Edge	High Edge
Clock polarity	Active Low	Active Low
Slave select pin	Active Low	Active High
INT0 pin	Active Low	-
Data arrangement	MSB	LSB
Transfer rate	1.87Mbps	937kbps
Mode of slave select	All data active	1 byte active

Table D.2 SPI_CONFIG

	7	6	5	4	3	2	1	0
0	40h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	0b (bCh0_ CPHA)	0b (bCh0_ CPOL)	11b (bCh0_SS)		11h (bCh0_INT)		0b (bCh0_ LSB)	0b
5	04h(bCh0_TransferRate)							
6	00h(bCh0_SS_Mode)							
7	00h							
8	1b (bCh1_ CPHA)	0b (bCh1_ CPOL)	10b (bCh1_SS)		00b (bCh1_INT)		1b (bCh1_ LSB)	0b
9	05h(bCh1_TransferRate)							
10	00h(bCh1_SS_Mode)							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							

D.2 I2C_CONFIG

An example of the “I2C_CONFIG (20h)” command is shown in Table D.3 and Table D.4.

Table D.3 I2C Setting Example (100kbps)

	7	6	5	4	3	2	1	0
0	20h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	01h(bTransferRate)							
5	00h							
6	00h							
7	00h							
8	00h							
9	00h							
10	00h							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							

Table D.4 I2C Setting Example (400kbps)

	7	6	5	4	3	2	1	0
0	20h (bCBWCB)							
1	bCBWTag							
2	0000h (wCBWReserved)							
3								
4	02h(bTransferRate)							
5	00h							
6	00h							
7	00h							
8	00h							
9	00h							
10	00h							
11	00h							
12	00h							
13	00h							
14	00h							
15	00h							

D.3 LCDC_WAKEUP_ON_CONFIG

An example of the “LCDC_WAKEUP_ON_CONFIG (06h)” command is shown in Table D.5 and Table D.6.

Table D.5 Enter to Sleep Mode Example

No	wRegAddress	wRegData	Register Name
0	0070h	0084h	REG[70h] PWM Control Register
1	0070h	0000h	REG[70h] PWM Control Register
2	006Eh	0001h	REG[6Eh] General Purpose Output Register
3	FFFFh	0064h	WAIT 100msec
4	002Ah	0000h	REG[2Ah] Display Mode Register
5	009Eh	0000h	REG[9Eh] Alpha-blending Value Register
6	FFFFh	0001h	WAIT 1msec
7	0084h	0008h	REG[84h] SDRAM Status Register
8	FFFFh	0001h	WAIT 1msec
9	0068h	0000h	REG[68h] Power Save Register
10	0012h	0001h	REG[12h] Clock Source Select Register
11	0004h	0001h	REG[04h] PLL Control Register
12	FFFFh	001Eh	WAIT 30msec
13	006Eh	0003h	REG[6Eh] General Purpose Output Register 1

Table D.6 Return from Sleep Mode Example

No	wRegAddress	wRegData	Register Name
0	006Eh	0001h	REG[6Eh] General Purpose Output Register 1
1	FFFFh	0064h	WAIT 100msec
2	0004h	0081h	REG[04h] PLL Control Register
3	FFFFh	0001h	WAIT 1msec
5	0012h	00C2h	REG[12h] Clock Source Select Register
6	0068h	00E8h	REG[68h] Power Save Register
7	0068h	0000h	REG[68h] Power Save Register
8	0068h	0001h	REG[68h] Power Save Register
9	FFFFh	0001h	WAIT 1msec
10	0084h	0000h	REG[84h] SDRAM Status Register
11	FFFFh	0001h	WAIT 1msec
12	002Ah	0001h	REG[2Ah] Display Mode Register
13	0050h	0080h	REG[50h] Display Control Register
14	006Eh	0000h	REG[6Eh] General Purpose Output Register 1
15	FFFFh	0001h	WAIT 1msec
16	0070h	0086h	REG[70h] PWM Control Register

Revision History

Attachment-1

Rev. No.	Date	Page	Category	Contents
Rev 1.0	2009/10/20	All		
Rev 1.1	2011/02/23	69		Add “bKeyScanDriveMode” to the command KEYSCAN_CONTROL.
		22		7.2 Initialization – add paragraph "After a hardware reset, the S1D13U11 reads the Serial Flash ROM. If the S1D13U11 reads the ASCII..."
Rev 1.2	2013/01/30	48		bEnRepeatedStartCondition - change Value 01h Explanation to "Not use"
		49		bWrData - change Access Read after Write wWriteSize to "Not use" and wReadSize to "Not use"
Rev 1.3	2018/03/29	All		Update page formatting
		98		Update Slales and Suport page

Sales and Technical Support

For more information on Epson Display Controllers, visit the Epson Global website.

https://global.epson.com/products_and_drivers/semicon/products/display_controllers/

For Sales and Technical Support, contact the Epson representative for your region.

https://global.epson.com/products_and_drivers/semicon/information/support.html