

安全にご使用いただくために

本製品は安全に充分配慮して設計を行っていますが、誤った使い方をすると火災や感電などの事故につながり大変危険です。ご使用の際は、警告/注意事項を必ず守ってください。

表示について

この取扱説明書は、次のような表示をしています。表示の内容をよく理解してから本文をお読みください。

警告 この表示を無視して誤った取扱いをすると、火災や感電などにより、人が死亡または重傷を負う可能性がある内容を示しています。

注意 この表示を無視して誤った取扱いをすると、感電やその他の事故により、人が負傷または物的損害が発生する可能性がある内容を示しています。

警告

- 製品の分解や改造などは、絶対に行わないでください。
- 無理に曲げる、落とす、傷つける、上に重い物を載せることは行わないでください。
- 製品が水・薬品・油などの液体によって濡れた場合、ショートによる火災や感電の恐れがあるため使用しないでください。

注意

- 本製品は電子機器ですので、静電気を与えないでください。
- 高温多湿の場所、温度差の激しい場所、チリやほこりの多い場所、振動や衝撃の加わる場所、スピーカなどの磁気を帯びた物の近くで保管しないでください。
- 煙が出たり異臭がする場合は、直ちにパソコンや周辺機器の電源を切り、USBケーブルをPCから抜いてください。
- 本製品は、医療機器、原子力機器、航空宇宙機器、輸送機器など人命に関わる設備や機器、及び高度な信頼性を必要とする設備や機器での使用は意図されておりません。これらの設備、機器制御システムに本製品を使用し、本製品の故障により人身事故/火災事故/その他の障害が発生した場合、いかなる責任も負いかねます。
- 取り付け時、鋭い部分で手を切らないように、十分注意して作業を行ってください。
- 配線を誤ったことによる損失、逸失利益などが発生した場合でも、いかなる責任も負いかねます。

その他のご注意

- 本書の内容に関して、将来予告なしに変更することがあります。
- 本書の内容につきましては万全を期して作成しておりますが、万一不審な点や誤りなどお気づきになりましたらご連絡お願い申し上げます。
- 本製品の運用を理由とする損失、逸失利益などの請求につきましては、いかなる責任も負いかねますので、予めご了承ください。
- 製品改良のため、将来予告なく外観または仕様の一部を変更する場合があります。
- 本製品は日本国内仕様となっており、海外での保守及びサポートは行っておりません。
- 本製品を廃棄するときは地方自治体の条例に従ってください。条例の内容については各地方自治体にお問い合わせください。
- 本製品の保証や修理に関しましては、添付の保証書に内容を明記しております。必ず内容をご確認の上、大切に保管してください。
- “REX”は株式会社リコーが商標権を所有しておりますが、弊社はその使用許諾契約により本商標の使用が認められています。
- Windowsは米国マイクロソフト社の米国およびその他の国における登録商標です。その他本書に記載されている商品名/社名などは、各社の商標または登録商標です。なお本書では、TM、[®]マークは明記しておりません。

第3章 SPI/I2C制御ユーティリティについて

/ /

USB61

<https://www.ratocsystems.com/>



Web

/ /

<https://www.ratocsystems.com> > usb61_download ▾

[REX-USB61ダウンロード\[RATOC\] - RATOC Systems](#)

(3-1)

Usb61Ut.y. exe SPI I2C

SPI/I2C

SPI
I2C
PORT

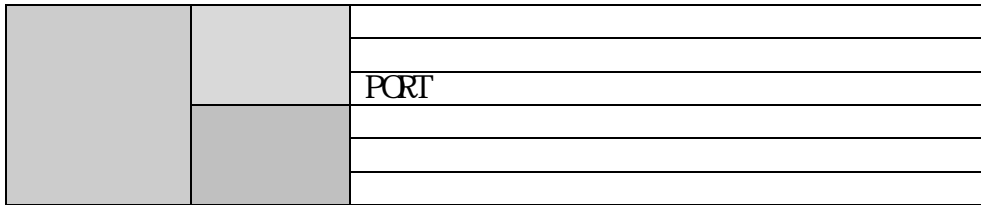
Read/Wite

(BIN)

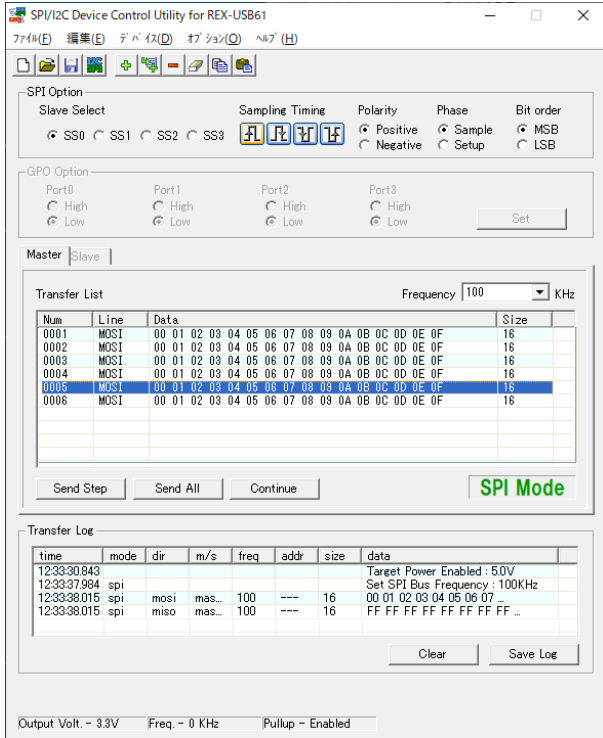
(CSV)

3-1

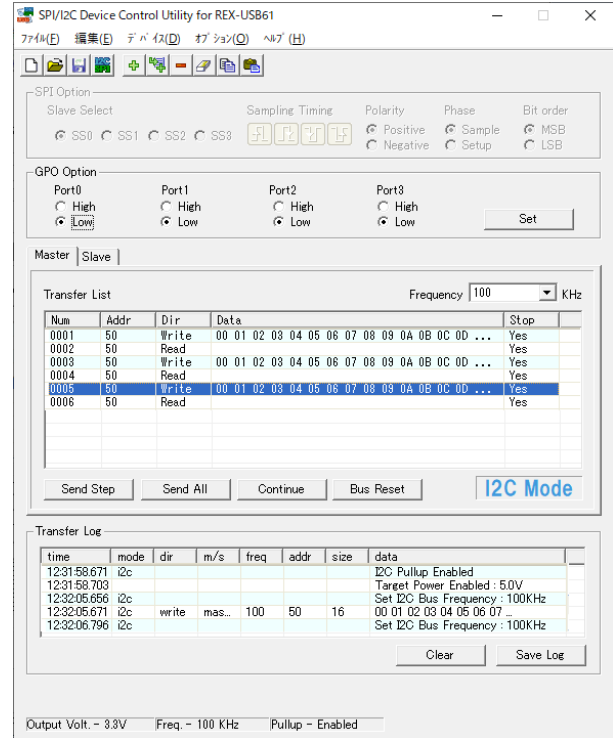
	SPI/I2C	
SPI		
		4
		()
I2C		
		()



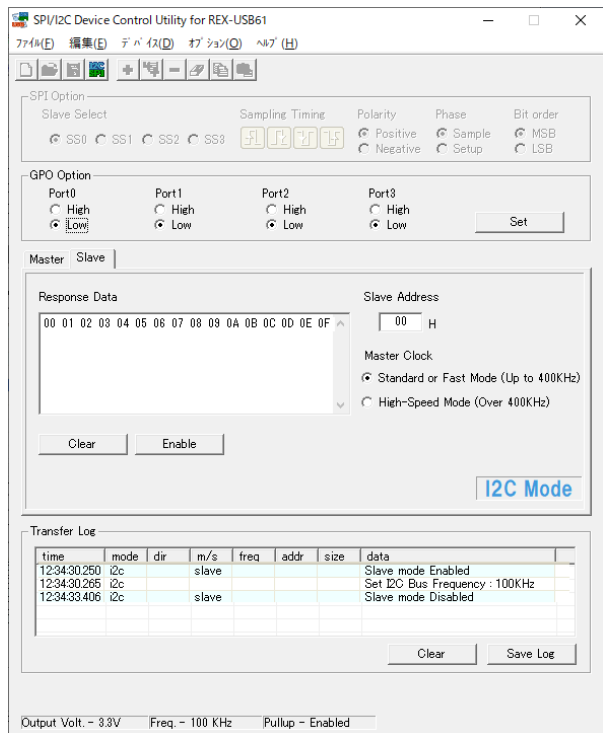
(3-2)



3-1. SPI



3-2. I2C



3-3. I2C

_____ ()

()

()

SPI/I2C

SPI/I2C






()

I2C

(3.3V 5.0V)

1
/

()

	[]
	[]
	[]
 SPI/I2C	[]
	[]

	[]
	[]
	[]
	[]
	[]

SPI Option

SPI Option SPI
 Slave Select
 Sampling Timing

 Polarity (Positive) (Negative)

 Phase (Sampling)
 (Set up)
 Bit order MSB LSB
 Sampling Timing Polarity, Phase

GPO Option

Port0 3 PCRT
 High Low
 Set Port

Master

Transfer List
 Num
 Addr
 Dir Read Write
 Line MCSI MISO

 Data
 Stop
 Size
 Send
 Send All
 Continue
 Bus Reset I2C

Slave

Slave Address) Page. 4 16

 Response Data
 Clear
 Enable
 Master Clock [Standard or Fast Mode (Up to 400kHz)]
 (400kHz)
 [High-Speed Mode (Over 400kHz)]
 (400kHz)

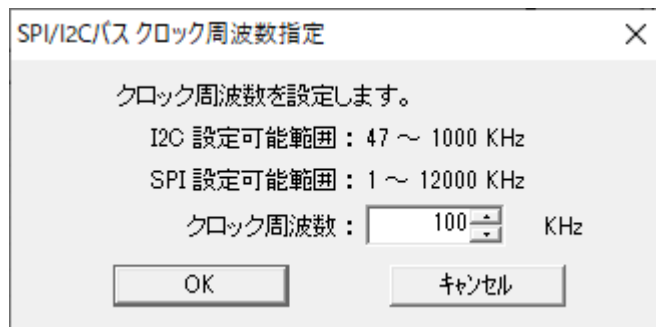
Master/Slave
Frequency

1kHz

SPI
(I2C: 47kHz 1MHz / SPI: 12MHz)

4 API

usb61_spi_set_freq()



[]

Device Mode

SPI Mode

I2C Mode

Output Volt

Freq

Pull up

I2C

Log

Transfer Log

time

(hh:mm:ss:ms)

mode

SPI/I2C

(spi/i2c)

dir

(read/write, mso/mosi)

ms

/

(master/slave)

freq

(kHz)

addr

I2C

(16)

size

(10)

data

(8)

)

Clear

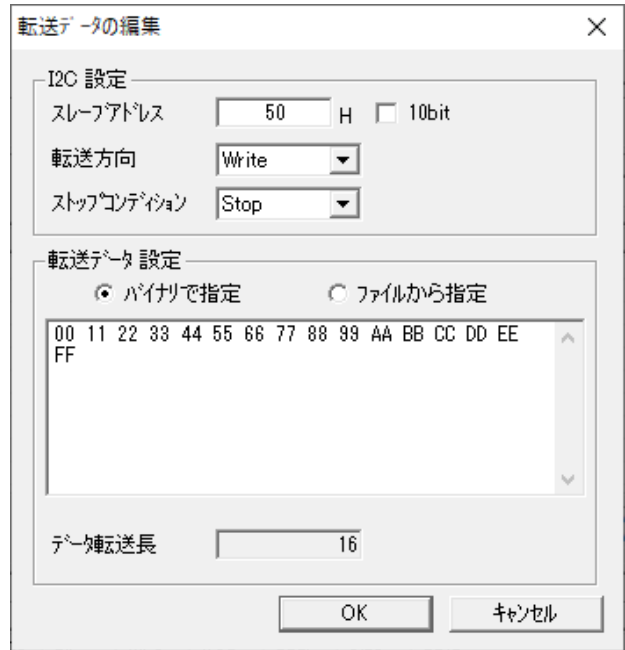
Save Log

CSV

(Transfer List)



3-4 SPI



3-5 I2C

I2C

)
10bit
)

I2C

Page. 4-16
10
Page. 4-16

Read

16

Write

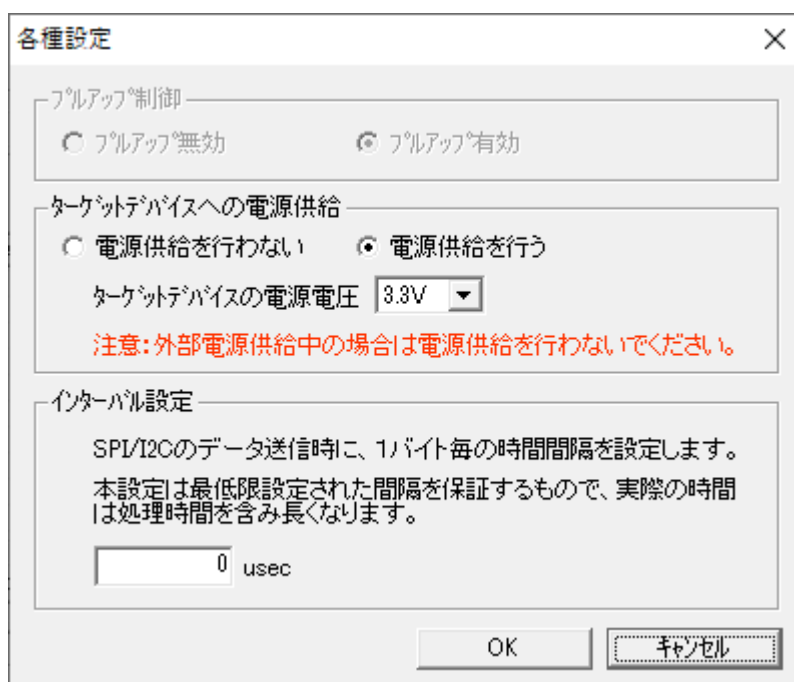
(16)

(10)

65535

(9)

I2C



3-6

I2C

(5V 1MHz [40kHz 100kHz]
I2C)

3.3V 5.0V

(3-3)

ATMEL AT24C02B AT25080A

SPI
[SPI/I2C]
SPI/I2C

SPI Mode

(SPI Mode)

[]
Frequency

1kHz

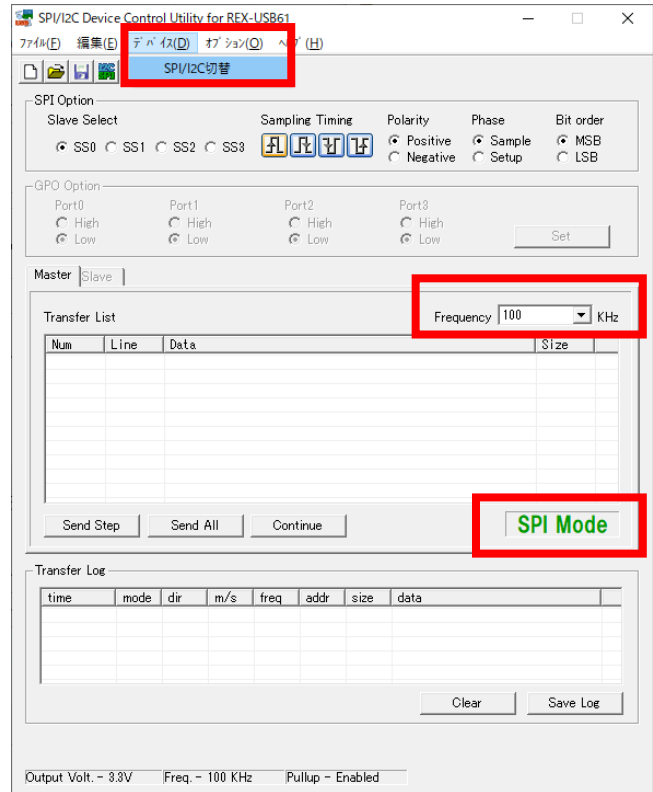
SPI

(I2C: 47kHz 1MHz / SPI: 12MHz)

API

usb61_spi_set_freq()

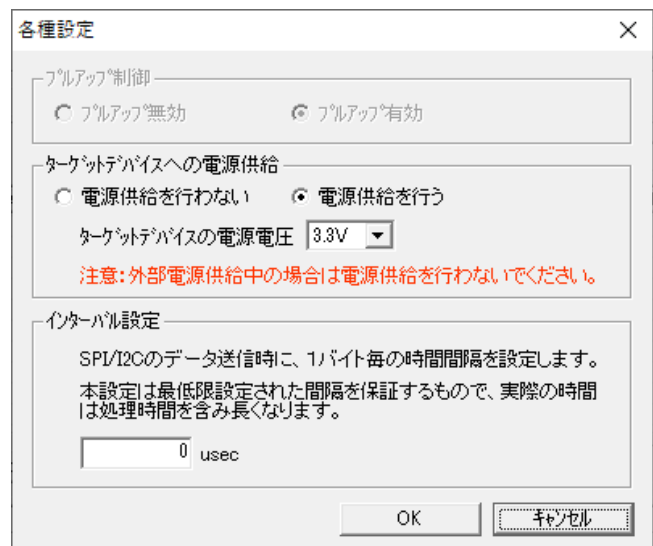
4



3-7.

[]

[]
1



3-8

50h	11	22	33	44	55	66	77	88	50h	8byte
-----	----	----	----	----	----	----	----	----	-----	-------

[(Write / Read)]
Transfer List

16

(1)

06h --- Write Enable

(2)

02h --- Write

00h 50h ---

11h 22h.. ---

(8byte)

(3)

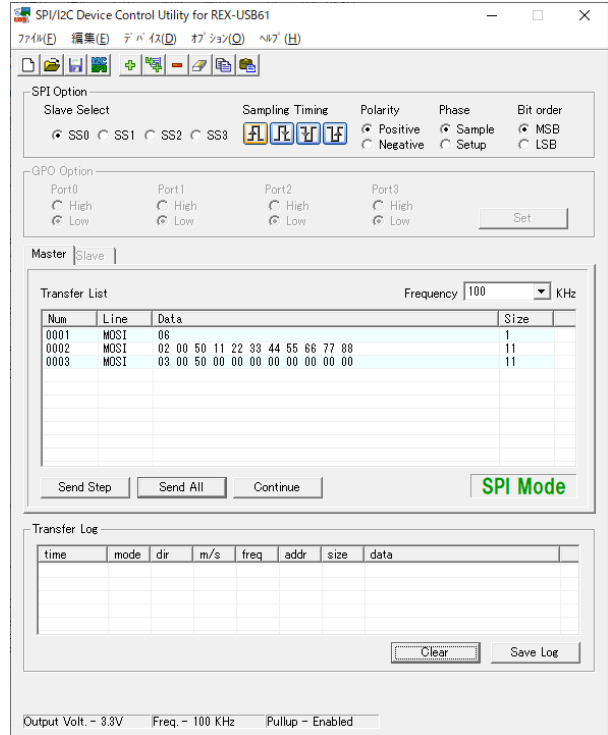
03h --- Read

00h 50h ---

00h 00h.. --- Read

(8byte)

(8byte)



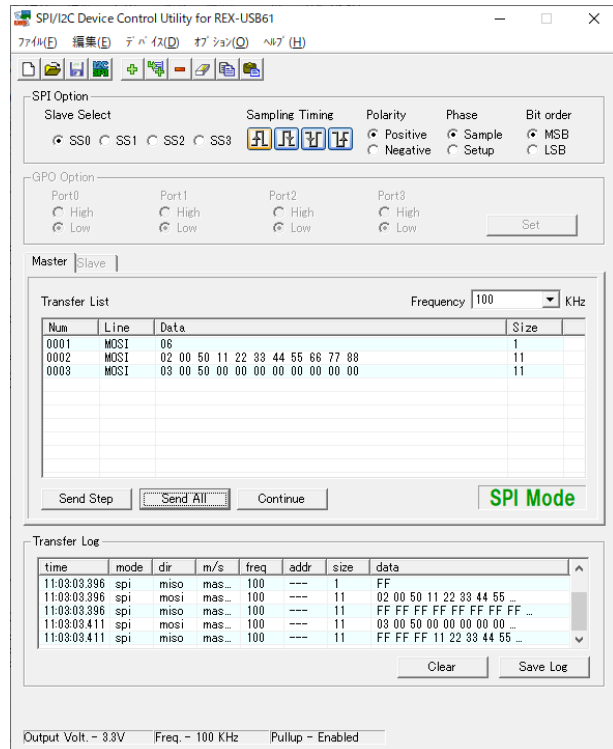
3-9

[(Write / Read)]

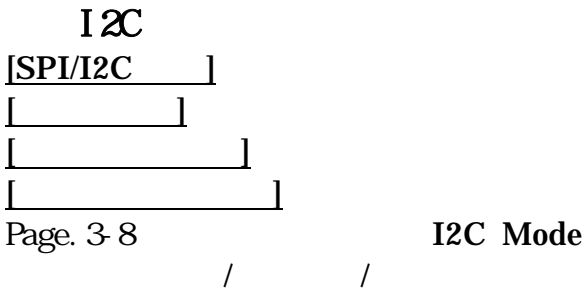
Send All

Transfer List

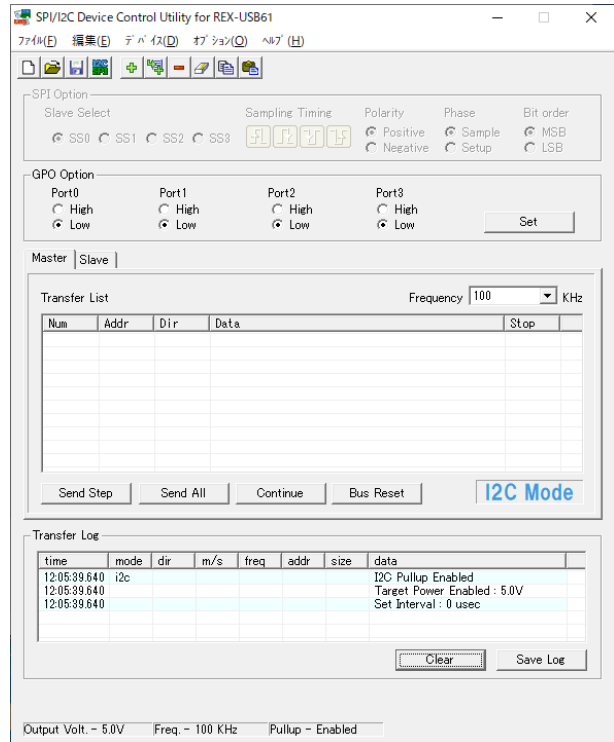
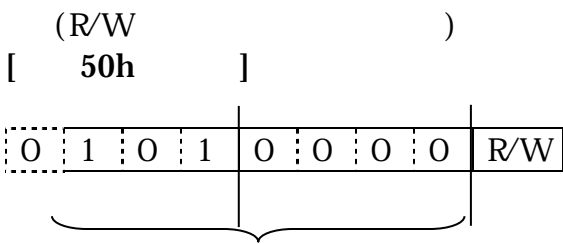
Transfer Log



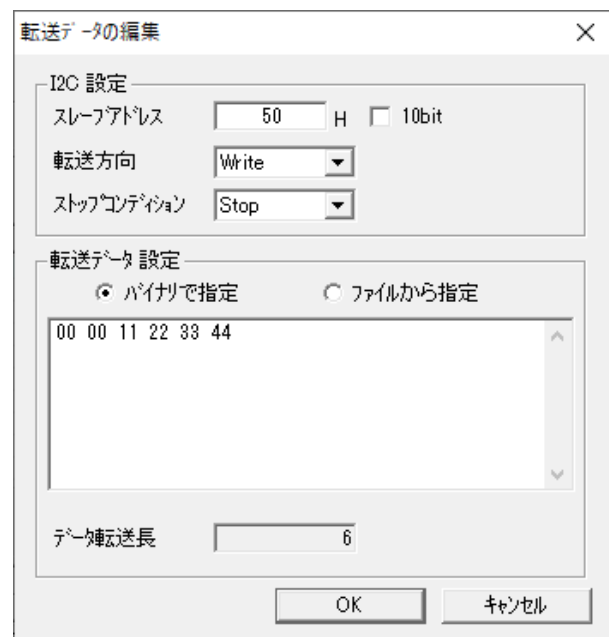
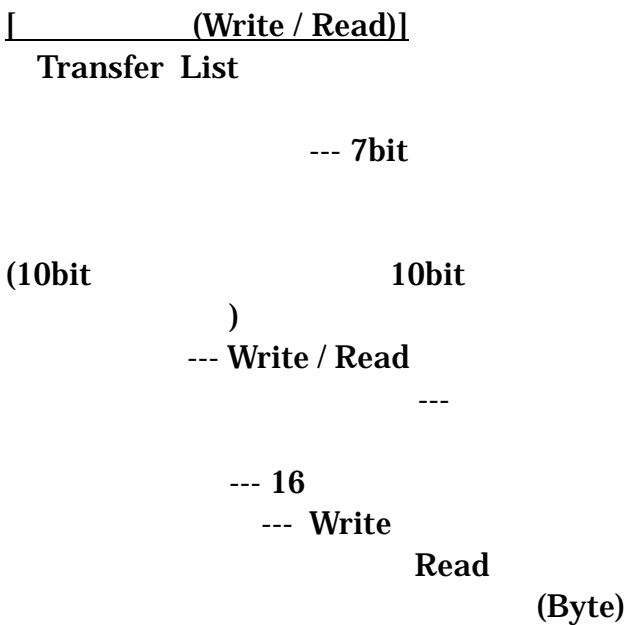
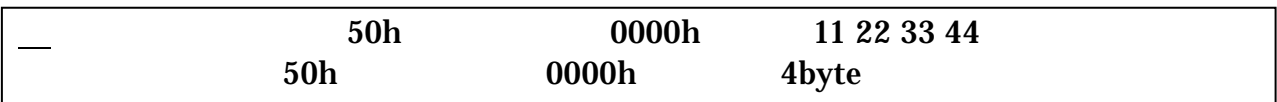
3-10



(Master)

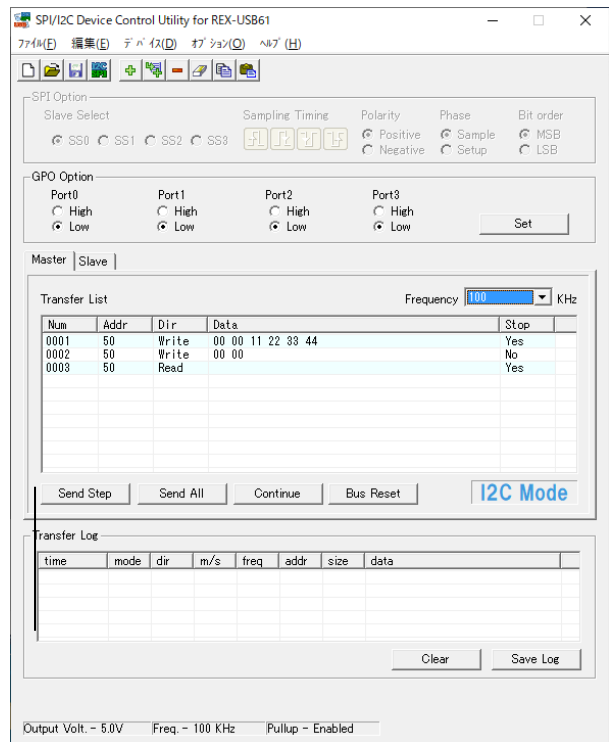


3-11.



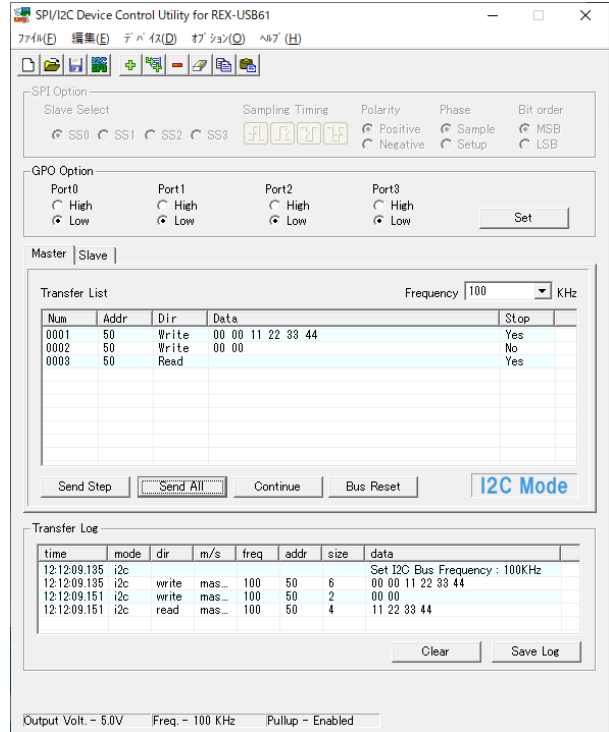
3-12

- (1) <Write>
 --- 50 (7bit)
 --- Write
 --- (Yes)
- 00h 00h ---
 11h 22h.. ---
- (2) <Read Write>
 --- 50 (7bit)
 --- Write
 --- (No)
- 00h 00h --- Read
- (3) <Read>
 --- 50 (7bit)
 --- Read
 --- (Yes)
- 4()



3-13

- [(Write / Read)]
 Send All Transfer List
- Transfer Log



3-14

I2C

_____ 50h

[SPI/I2C]

[_____]

[_____]

[_____]

Page. 3-8

I2C Mode

/ /

(Slave)

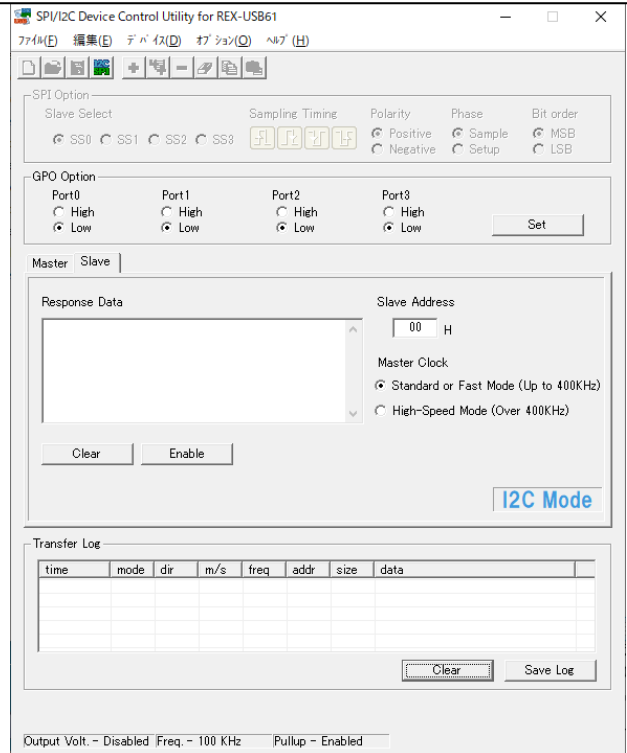
Slave Address

7bit

Master Clock

Response Data
Read

Transfer Log



3-15. I2C

(3-4)

SPI / I 2C
 SPI I 2C

10 16 0 65536 16

[h] [H]

[,]

/

() TAB

3-2

	#
	#
	MDE=
	SPI I 2C MDE
	SPI I 2C

FREQUENCY=							
	<p>1KHz SPI</p> <p>usb61_spi_set_freq()</p> <p>4 API</p> <table border="1"> <tr> <td></td> <td></td> </tr> <tr> <td>SPI</td> <td>100KHz</td> </tr> <tr> <td>I2C</td> <td>100KHz</td> </tr> </table>			SPI	100KHz	I2C	100KHz
SPI	100KHz						
I2C	100KHz						
	<p>SPI I2C</p> <table border="1"> <tr> <td></td> <td></td> </tr> <tr> <td>SPI</td> <td>1 12000</td> </tr> <tr> <td>I2C</td> <td>47 1000</td> </tr> </table>			SPI	1 12000	I2C	47 1000
SPI	1 12000						
I2C	47 1000						
INTERVAL=							
	(μ S)						
	0						
0 65535	(0μ S 65535μ S)						

POWER=							
	OFF						
	<table border="1"> <tr> <td>OFF</td> <td>OFF</td> </tr> <tr> <td>3.3V</td> <td>ON3</td> </tr> <tr> <td>5.0V</td> <td>ON5</td> </tr> </table>	OFF	OFF	3.3V	ON3	5.0V	ON5
OFF	OFF						
3.3V	ON3						
5.0V	ON5						
WAIT=							
	100ns (100ns 60)						
	1 600 (100ns 60)						
REPEAT=nn							
	REPEAT { } { } Page. 3-17 REPEAT						
	nn=1 65536{...}						
PULLUP=							
	SDA SCL Pull-up Pull-up (ON) 5V 1MHz OFF						
	ON OFF						
FILEn							
	n 5 " " ()						
	n=1 5 " "						

	END
	END END (
	END)

I2C

3-3 I2C

	ADDRESSMODE=
I2C	7 10 (
7)
7	10
	ADDRESS=
I2C	
READ	WRITE
0	1023
	READ
xxH	
1	65536
	READF
	FILEn
xxH FILEn	1 65536
	WRITE
xxH xxH ...	
	WRITEF
	FILEn
FILEn	
	STOP

RESET				
(STCP)				
GPO=				
DO0 DO3(13 16)				
1				
10 0 15				
16 0h Fh				
Bit3		Bit2		Bit1
DO3		DO2		DO1
Bit0				DO0

SPI

3-4 SPI

SS=n																
	0															
	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="background-color: #e0e0e0;"></td> <td style="background-color: #e0e0e0;">SSx</td> </tr> <tr> <td>0</td> <td>SS0</td> </tr> <tr> <td>1</td> <td>SS1</td> </tr> <tr> <td>2</td> <td>SS2</td> </tr> <tr> <td>3</td> <td>SS3</td> </tr> </table>		SSx	0	SS0	1	SS1	2	SS2	3	SS3					
	SSx															
0	SS0															
1	SS1															
2	SS2															
3	SS3															
	n=0 3															
SAMPLING=n																
	0															
	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="background-color: #e0e0e0;"></td> <td style="background-color: #e0e0e0;"></td> <td style="background-color: #e0e0e0;"></td> </tr> <tr> <td>0</td> <td></td> <td></td> </tr> <tr> <td>1</td> <td></td> <td></td> </tr> <tr> <td>2</td> <td></td> <td></td> </tr> <tr> <td>3</td> <td></td> <td></td> </tr> </table>				0			1			2			3		
0																
1																
2																
3																
	n=0 3															
FB=																
	M&B															
	M&B LSB															
SSSET																
	Low															
SSRESET																
	H gh															

SPI Read/Wite SPI	
	FILEn (FILEn)
	xxH xxH ... FILEn
	FILEm FILEn (FILEn)
	FILEm FILEn

REPEAT

REPEAT

STOP

REPEAT=10 READ 1 STOP	10 STOP
REPEAT=10 { READ 1 STOP }	1 STOP 10
REPEAT=10 READ 1 STOP	10 STOP
REPEAT=10 { READ 1 } STOP	10 STOP
REPEAT=10 { READ 1 STOP }	1 STOP 10

(3-5)

Load ---

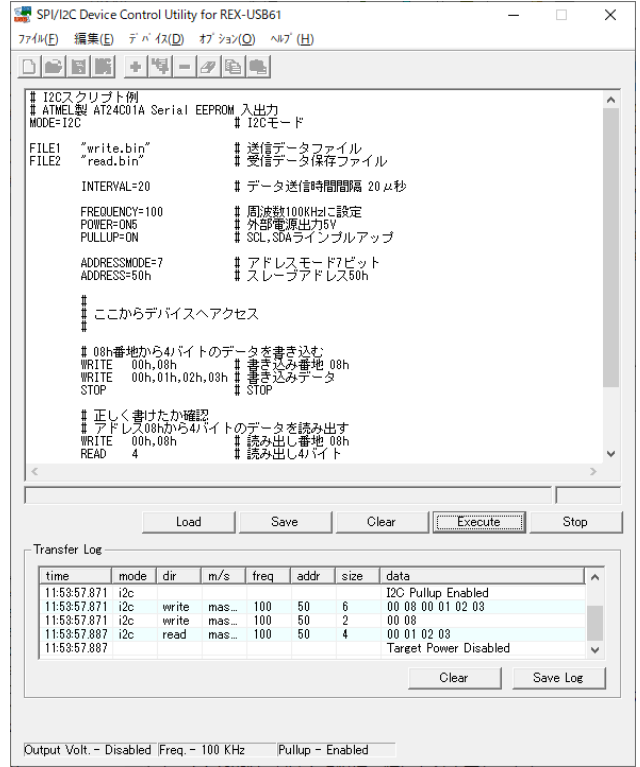
Save ---

Clear ---

Execute ---

Stop ---

Transfer Log



()

I2C SPI (Write / Read)

([Utility]

I2C_script.txt / SPI_script.txt)

I2C (50h 0008h 4 (00h01h
02h 03h)
100KHz/ 5V)

```
# I2C
# ATMEL AT24C01A Serial EEPROM
MODE=I2C # I2C

FILE1 "write.bin" #
FILE2 "read.bin" #

INTERVAL=20 # 20µ
FREQUENCY=100 # 100KHz
POWER=ON5 # 5V
PULLUP=ON # SCL,SDA
ADDRESSMODE=7 # 7
ADDRESS=50h # 50h

#
#
#

# 0008h 4
WRITE 00h,08h # 0008h
WRITE 00h,01h,02h,03h #
STOP # STOP

#
# 0008h 4
WRITE 00h,08h # 0008h
READ 4 # 4
STOP # STOP

# ( )
```

```
# (
#          0008h    FILE1
WRITE 00h,08h    #          0008h
WRITEF          FILE1 #          (FILE1    4          )
STOP            # STOP

#
#          0008h          FILE2
WRITE 00h,08h    #          0008h
READF 04h    FILE2 # 4          FILE2
STOP            # STOP

POWER=OFF      #          0V

END
```

SPI (1500h 4 (00h 01h 02h 03h)

~~3MHz~~ / 5V)

```

# SPI
# ATMEL AT25080 Serial EEPROM
MODE=SPI # SPI

FILE1 "write.bin" #
FILE2 "read.bin" #

POWER=ON5 # 5V
INTERVAL=20 # 20µ
FREQUENCY=3000 # 3MHz
SAMPLING=0 #
FB=MSB #
SS=0 # 0

#
#
#

# 1500h 4
SSSET # SS Low
06h # WREN
SSRESET # SS High
SSSET # SS Low
02h,15h,00h # WRITE
00h,01h,02h,03h #
SSRESET # SS High

# ( )
    
```

```

# (          )
#
# 1500h      4
SSSET      # SS      Low
03h,15h,00h #          READ+
REPEAT=4   #          4
00h        #          1
SSRESET    # SS      High

# 1500h      FILE1
SSSET      # SS      Low
06h        #          WREN
SSRESET    # SS      High
SSSET      # SS      Low
02h,15h,00h #          WRITE
FILE1      # FILE1
SSRESET    # SS      High

#
# 1500h      FILE2
SSSET      # SS      Low
03h,15h,00h #          READ+
FILE1 FILE2 # FILE1
            # FILE2
SSRESET    # SS      High

POWER=OFF  #          0V

END

```

第4章 API関数仕様

(4-1) VC

API REX USB61

API SPI/I2C

Visual C++ usb61def.h

usb61api.lib/64bit LIB\VC_x64

(usb61api.dll)

usb61def.h

usb61def.h

```
#define USB61LIB_API __declspec(dllexport)

USB61LIB_API HANDLE WINAPI usb61_open( RS_STATUS *pStatus );
USB61LIB_API RS_STATUS WINAPI usb61_close( HANDLE hUsb61Device );
USB61LIB_API RS_STATUS WINAPI usb61_power_control( HANDLE hUsb61Device,
                                                    UINT fPowerState );
USB61LIB_API RS_STATUS WINAPI usb61_node_change( HANDLE hUsb61Device,
                                                    UINT fDeviceMode,
                                                    USHORT i2cSlaveAddr );
USB61LIB_API RS_STATUS WINAPI usb61_set_interval( HANDLE hUsb61Device,
                                                    USHORT IntervalCnt );
USB61LIB_API RS_STATUS WINAPI usb61_gpio_write( HANDLE hUsb61Device,
                                                    UINT fPortVal );
USB61LIB_API RS_STATUS WINAPI usb61_get_fw_version( HANDLE hUsb61Device,
                                                    UCHAR* pFWMajorVer,
                                                    UCHAR* pFWMinorVer );
USB61LIB_API RS_STATUS WINAPI usb61_get_dll_version( HANDLE hUsb61Device,
                                                    UCHAR* pDllMajorVer,
                                                    UCHAR* pDllMinorVer );
USB61LIB_API RS_STATUS WINAPI usb61_get_hw_info( HANDLE hUsb61Device,
                                                    PRS_HARDWARE_INFO pHardwareInfo );
USB61LIB_API RS_STATUS WINAPI usb61_i2c_pullup( HANDLE hUsb61Device,
                                                    RS_I2C_PULLUP fI2cPullup );
USB61LIB_API RS_STATUS WINAPI usb61_i2c_bus_reset( HANDLE hUsb61Device );
( )
```


(4-2) VB / Visual C#

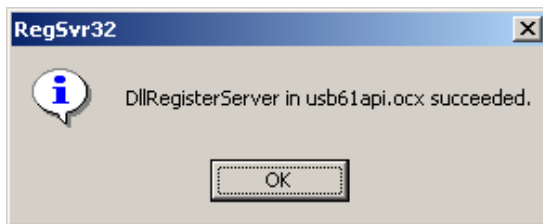
Visual BASIC	Visual C#	ActiveX	ActiveX
ActiveX	32bit	LIB	()
64bit	API		

1 ActiveX

```

2 Windows
  DLL, ActiveX
usb61api.ocx
>regsvr32 usb61api.ocx

```



2 ActiveX

```
>regsvr32 /u usb61api.ocx
```

