How to access CM65xx register through HID report

Rev. 0.4

Revision History

Revision	Date	Description
0.1	2013/9/2	First release
0.2	2013/9/2	Update the link for HID tool source code.
0.3	2013/9/3	Add I2C process flow.
0.4	2013/9/3	Add screen capture for I2C process flow
0.5	2013/10/30	Correct the table in page.3

Index

Summarize: 1
1 HID format for CM65xx: 2
1.1 HID Input Report: 2
1.2 HID Output Report: 3
2 How to read register from HID report: 4
3 How to write register from HID report: 7
4 Example of changing GPIO_06 to control Ring LED. 9
5 I2C Master Mode Control Example: 10
5.1 I2C read data from EEPROM 10
5.2 I2C Write data from EEPROM 14
Appendix A: CM65xx I2C Control Registers 18
Appendix B: CM65xx GPIO Control Registers 21

Summarize:

This is the simple “How-To” document for using a utility “SimpleHIDWrite.exe” to control the CM65xx register through HID report interface. You can learn how to read/write register step by step. And an example of controlling Ring LED on CM6500/CM6502 demo board which is controlled by GPIO_06.

Section 5 gives an example of controlling I2C interface EEPROM through HID report.

Please note that the example shows in this document is based on ROM code (i.e. No FW customization). The HID format night be different as described if customer change their firmware by themselves.

Below documentation is required for access the HID on CM65xx:

1. CM6500_CM6502_CM6510_CM6523_Programming Guide_v1.01.pdf: Given the HID report format for CM65xx platform.
2. CM65xx Registers Definition.pdf: Given the Register address for CM65xx platform.

“SimpleHIDWrite.exe” is a utility provided by Robert Marquardt’s HID Controller component suite for Delphi.

1 HID format for CM65xx:

1.1 HID Input Report:

The 16-bytes input report is defined as the following table. Host will be notified by an input report via interrupt pipe. Host can also get input report with class request “Get Input Report[A1 01 01 01 03 00 10 00]” via control pipe.
Host can read registers of CM65xx by HID input report’s byte 6~byte 15. Start address 35 and length of registers that host reads can be set by sending output report.

	Description	Size
Byte 0	Report ID (Always 1)	1
Byte 1~Byte 2	For defined HID event, and each event occupies one bit (this depends on HID report descriptor)	2
Byte 3	start address of returned data (H-start_addr)	1
Byte 4	start address of returned data (L-start_addr)	1
Byte 5	Interrupt source. Bit 7: Reserved Bit 6: UART_INT Bit 5: GPI_INT Bit 4: SPIS_INT Bit 3: SPIM_INT Bit 2: I2CS_INT Bit 1: I2CM_INT Bit 0: IR_INT	1
Byte 6~Byte15	Register content	10

1.2 HID Output Report:

HID output report is designed for writing registers to CM65xx. It is also used for setting start address and length of registers sent to host in input report.

	Description	Size
Byte 0	Report ID (Always 1)	5
Byte 1	1. 0x00: Set register read in input report 2. start address of returned data (H-start_addr)	1
Byte 2	1. 0xFE: Set register read in input report 2. start address of returned data (L-start_addr)	1
Byte 3	Effective write data length (<= 12) Effective read data length (<= 10)	1
Byte 4	1. If Byte1 is 0x00 and Byte2 is 0xFE, this byte is the value set to “Register Address(H)” in input report. 2. If Byte2 is not 0xFE, this byte is data written to register.	1
Byte 5	1. If Byte1 is 0x00 and Byte2 is 0xFE, this byte is the value set to “Register Address(L)” in input report. 2. If Byte2 is not 0xFE, this byte is data written to register.	1
Byte 6~Byte15	Byte 7~Byte15 are data written to register.	10

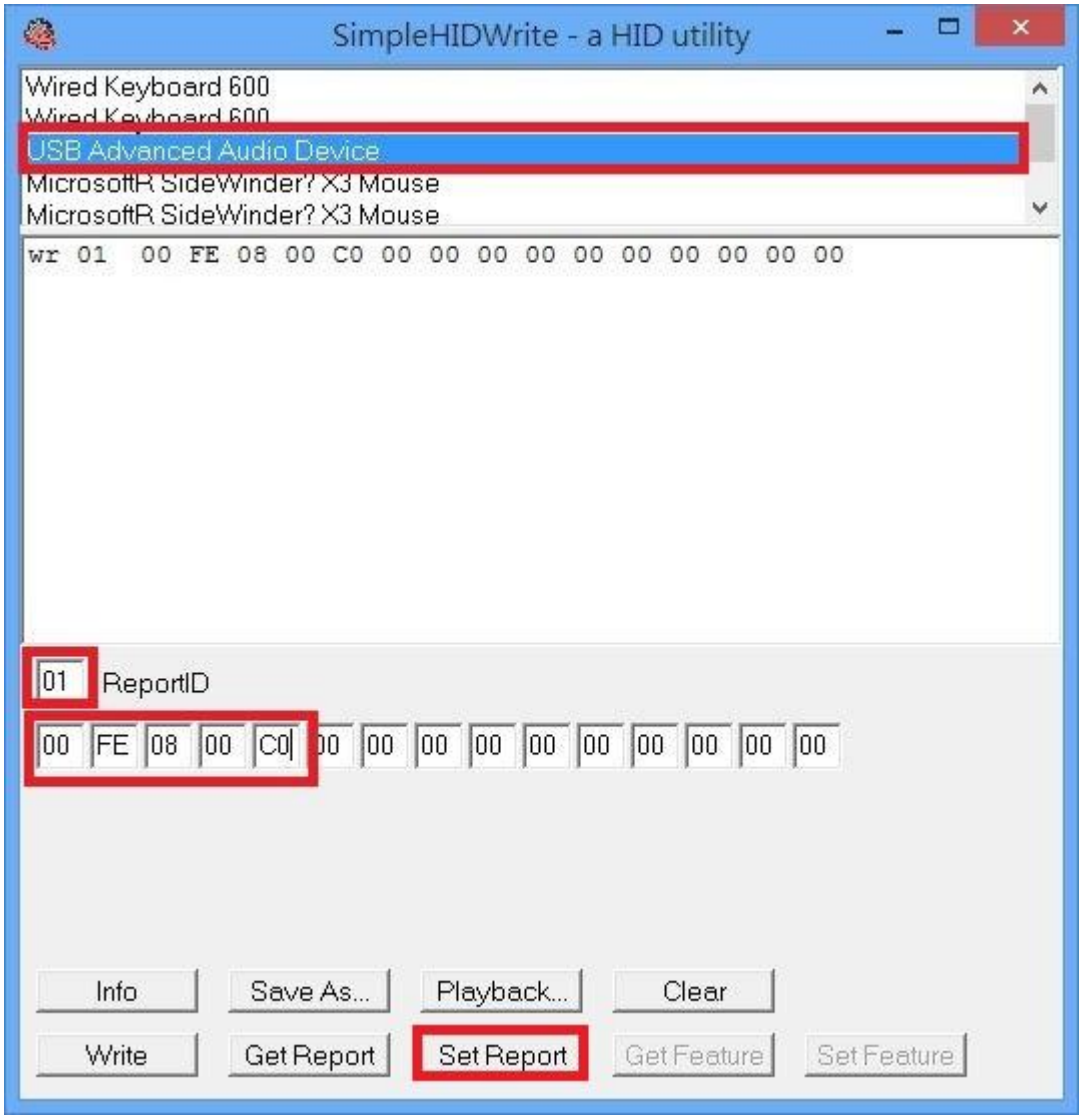
2 How to read register from HID report:

To Read register content with HID report, you have to perform: “Set Output Report[21 09 01 02 03 00 10 00]” and “Get Input Report[A1 01 01 01 03 00 10 00]”.

“Set Output Report” is to define register address and data length you are going to read the register. The start of the byte 1 and byte 2 must be “00 FE” to inform this is a Read Register index.

“Get Input Report” will return the register data in the register you defined in previous valid Read register index “Set Output Report” command.
Below is the example of reading back GPIO status register from address: C0 to C7 (8 bytes data length)

When you running the “SimpleHIDWrite.exe”, you will see the User Interface as below:



Set the report ID =”01”

Byte 1 and Byte 2 = “00 FE”: to perform a register read address set up.

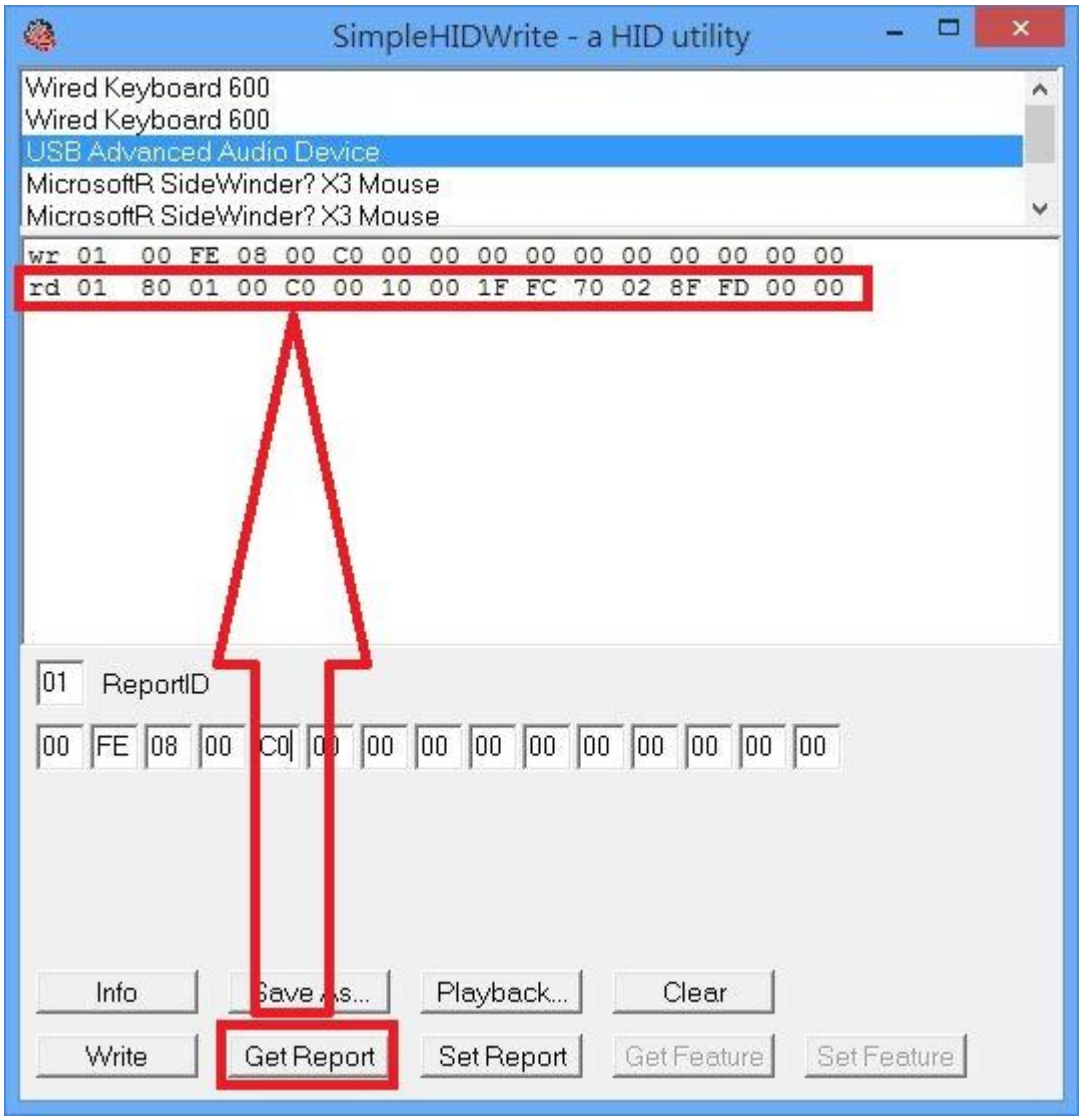
Byte 3= “08”: to define the data length to read back based on the starting address defined as below.

Byte 4 and Byte 5=”00 C0”: high byte and low byte for the starting address.

Others bytes are don’t care when performing register read.

Press “Set Report”.

Press “Get Report” as below figure.



Byte 03 and 04=”00 C0”: Shows the Starting address of register content

Byte 05 should be ignored.

Byte 06~13 are the data returned (Please refer to “CM65xx Registers Definition.pdf” for Register

mapping).

Register address C0= 0x10 → GPO data register for GPO0~GPO7; only GPO4 are set to high.

Register address C1= 0x00 → GPO data register for GPO8~GPO15;

Register address C2= 0x1F → GPI data register for GPI0~GPI7

Register address C3= 0xFC → GPI data register for GPI8~GPI15

Register address C4= 0x70 →GPIO direction control register for GPIO0~7; GPIO4, 5, and 6 are set as Output. Others are input.

Register address C5= 0x02 →GPIO direction control register for GPIO8~15; Only GPIO9 are set as Output. Others are input.

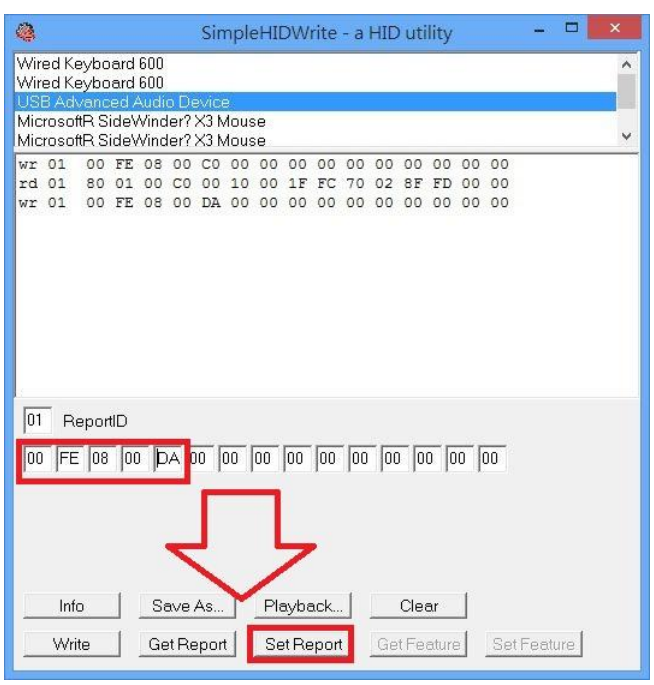
Register address C6= 0x8F → GPIO interrupt enable Mask Register for GPIO0~7.

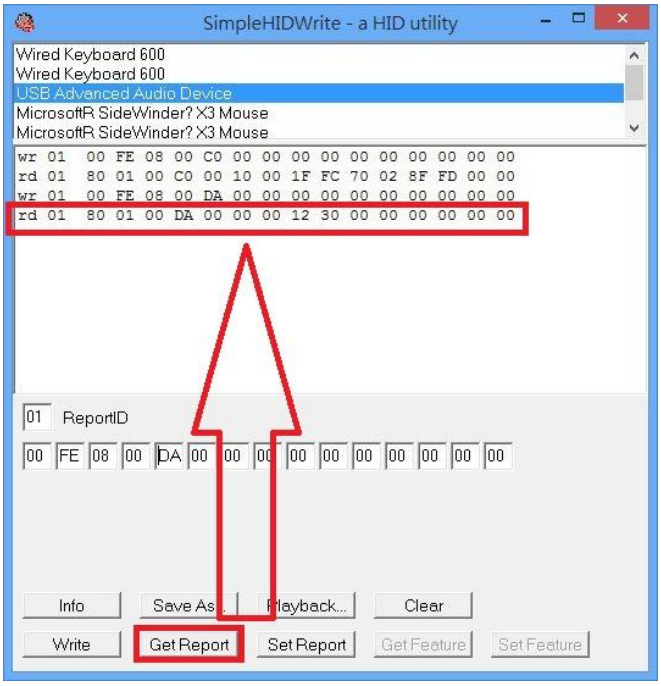
Register address C7= 0xFD→ GPIO interrupt enable Mask Register for GPIO8~15.

3 How to write register from HID report:

To Write register content with HID report, you only have to perform “Set Output Report”. And you can check it with Read Register steps.

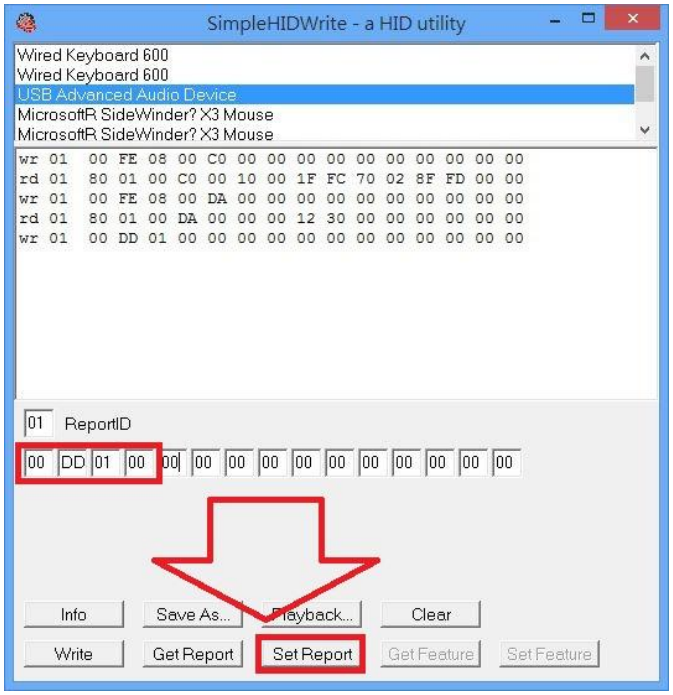
Below is the example of changing GPO mode control by writing “GPO Switch Source register” from address: DA to E0 (8 bytes data length)
Perform a register Read to check the initial value of register DA to E0:



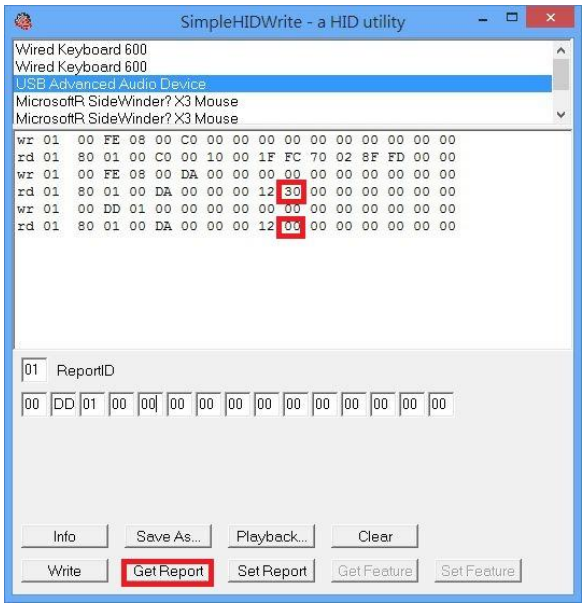


The Default value of register 0xDA to 0xE0 is: 00 00 12 30 00 00 00 00; Where:
0xDC defined GPO4 as Playback LED (LED1), GPO5 as Play Mute LED (LED2)
0xDD defined GPO6 as Record Mute LED (LED3), GPO7 as general purpose GPO
Others are all defined as GPOs.

Perform a Register write to change GPO6 behavior from LED3 to General purpose GPOs.
Byte 1 and Byte 2 = “00 DD” → set the initial register address to write
Byte3 = “01” → set the total data length to write
Byte4 = “00” → set the new value to Register 0xDD.

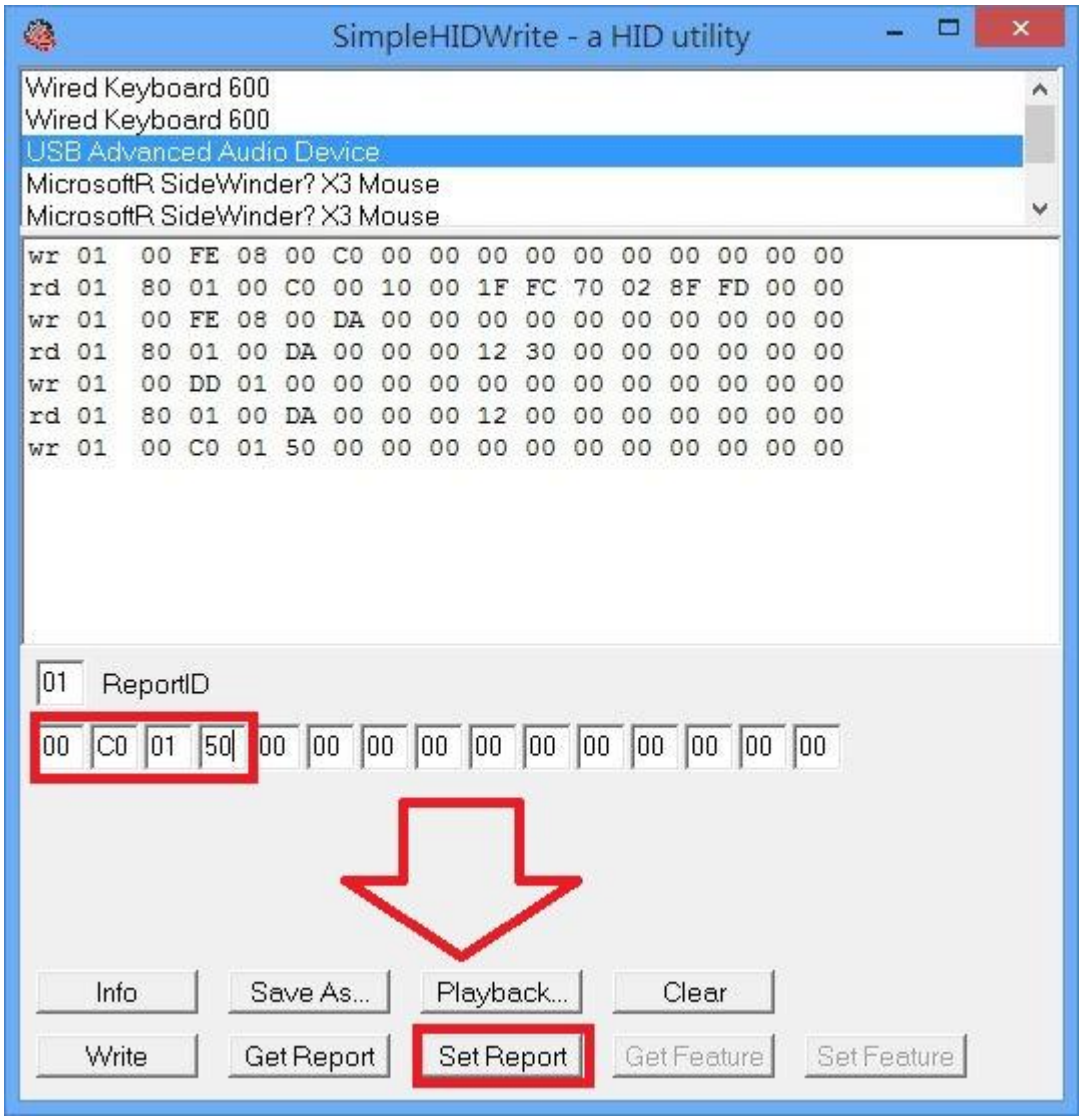


After the command above, the register 0xDD has been changed from 0x30 to 0x00.
Read back to see the change:



4 Example of changing GPIO_06 to control Ring LED.

Following above steps, you know how to read/write registers by HID report. Now, let's try to control GPIO to turn on the Ring LED which is connected to GPIO_06 (pin38 of CM6500/CM6502).
Continue with changing 0xDD register value to 0x00; now, the GPIO_06 is set to General purpose GPO mode. And by default, GPIO_06 direction is set to output. So what we are going to do is to output GPIO_06 high level to turn on the LED:
Byte 1 and Byte 2 = "00 C0": to set Register address to write
Byte 3= "01": To set data length to write.
Byte 4 ="50": To set GPIO_06 high.
Press "Set Report" → now the Ring LED is turn on.



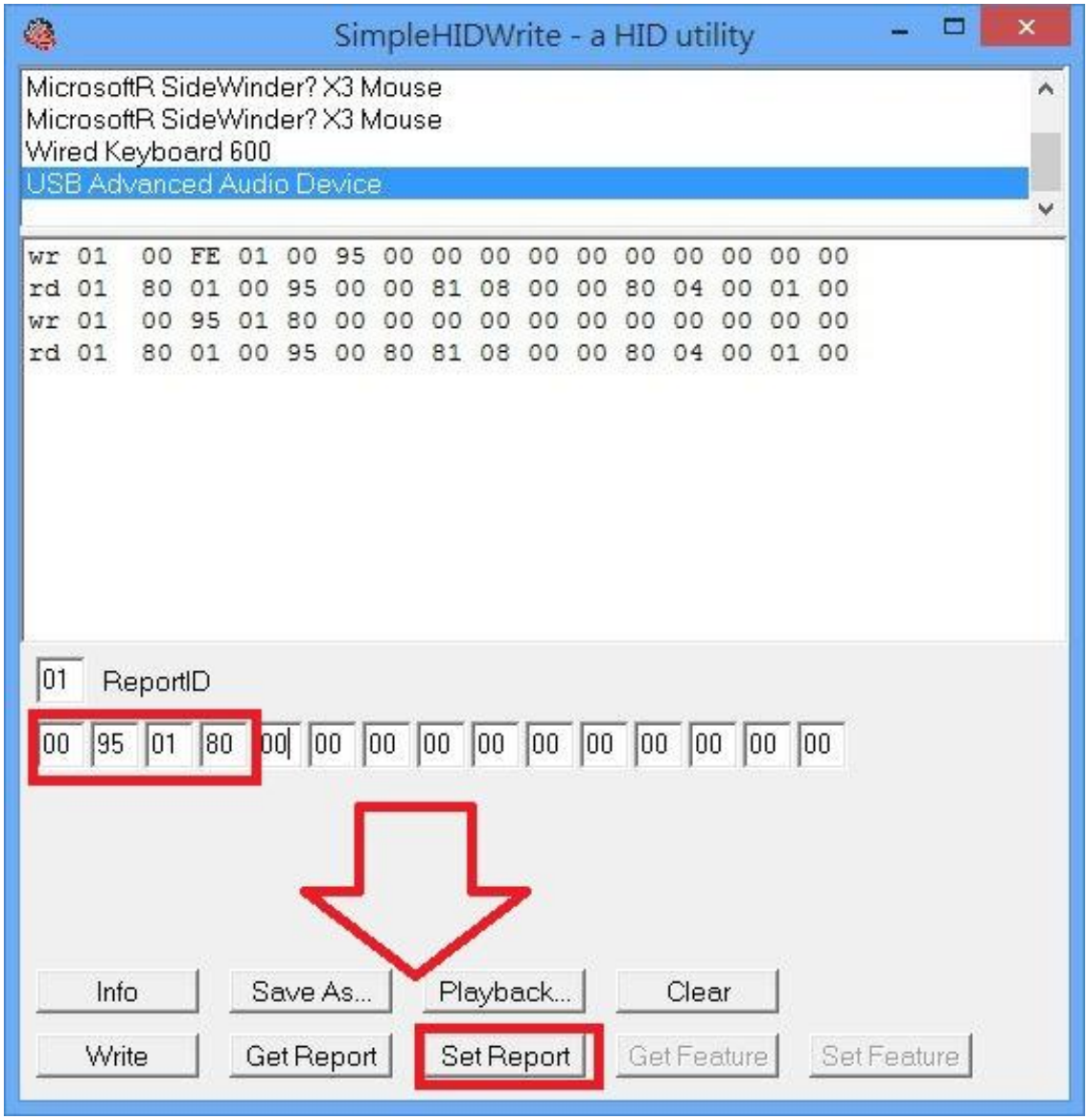
5 I2C Master Mode Control Example:

This section gives an example to access I2C interface EEPROM data through HID report.

5.1 I2C read data from EEPROM

// (initial i2c master mode) 00 95 01 80

STEP 1: Write ADDR 0x95 = 0x80

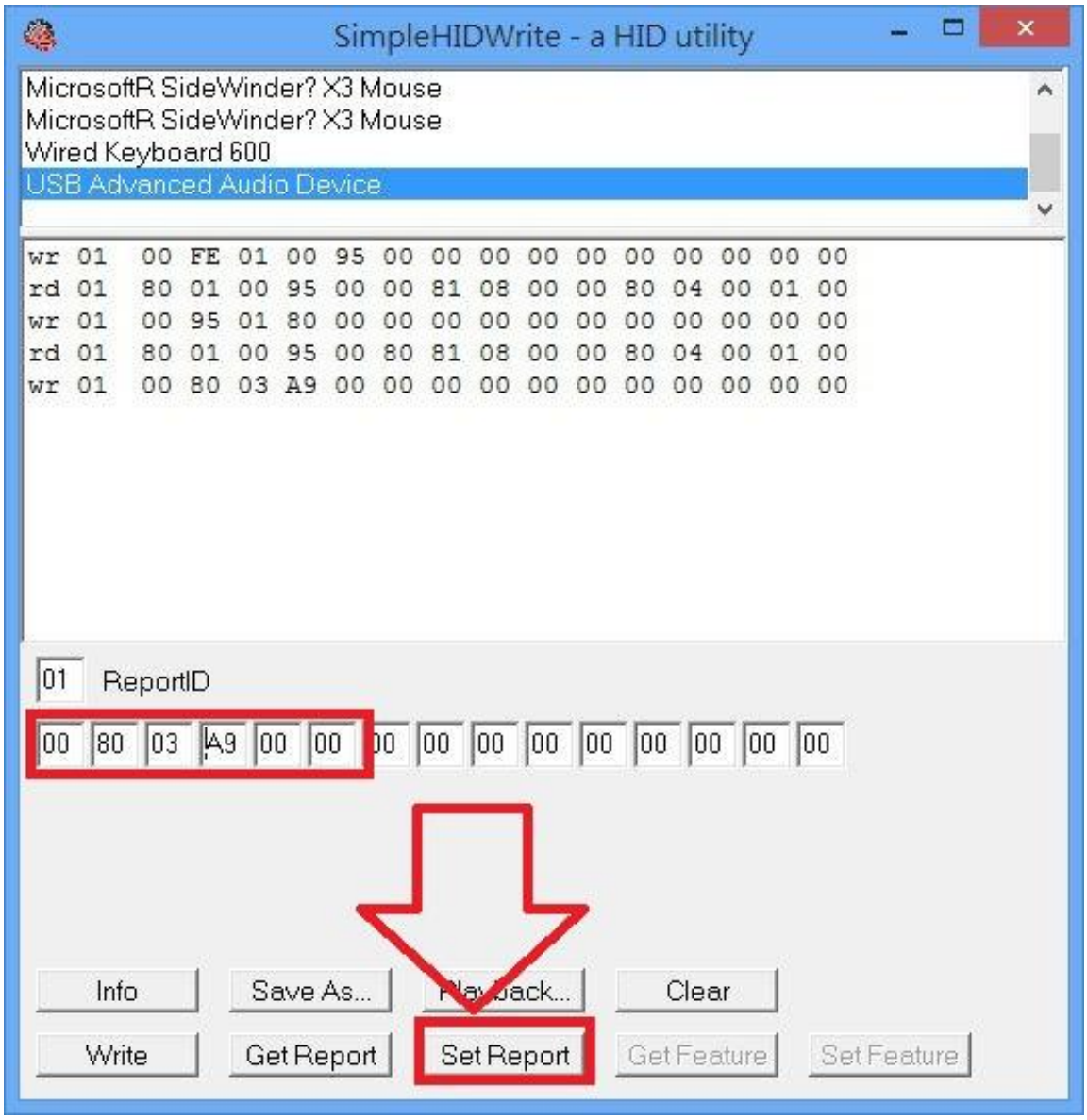


// (initial device address and MAP) 00 80 03 a9 00 00

STEP 2: Write ADDR 0x80 = 0xA9

0x81 = 0x00

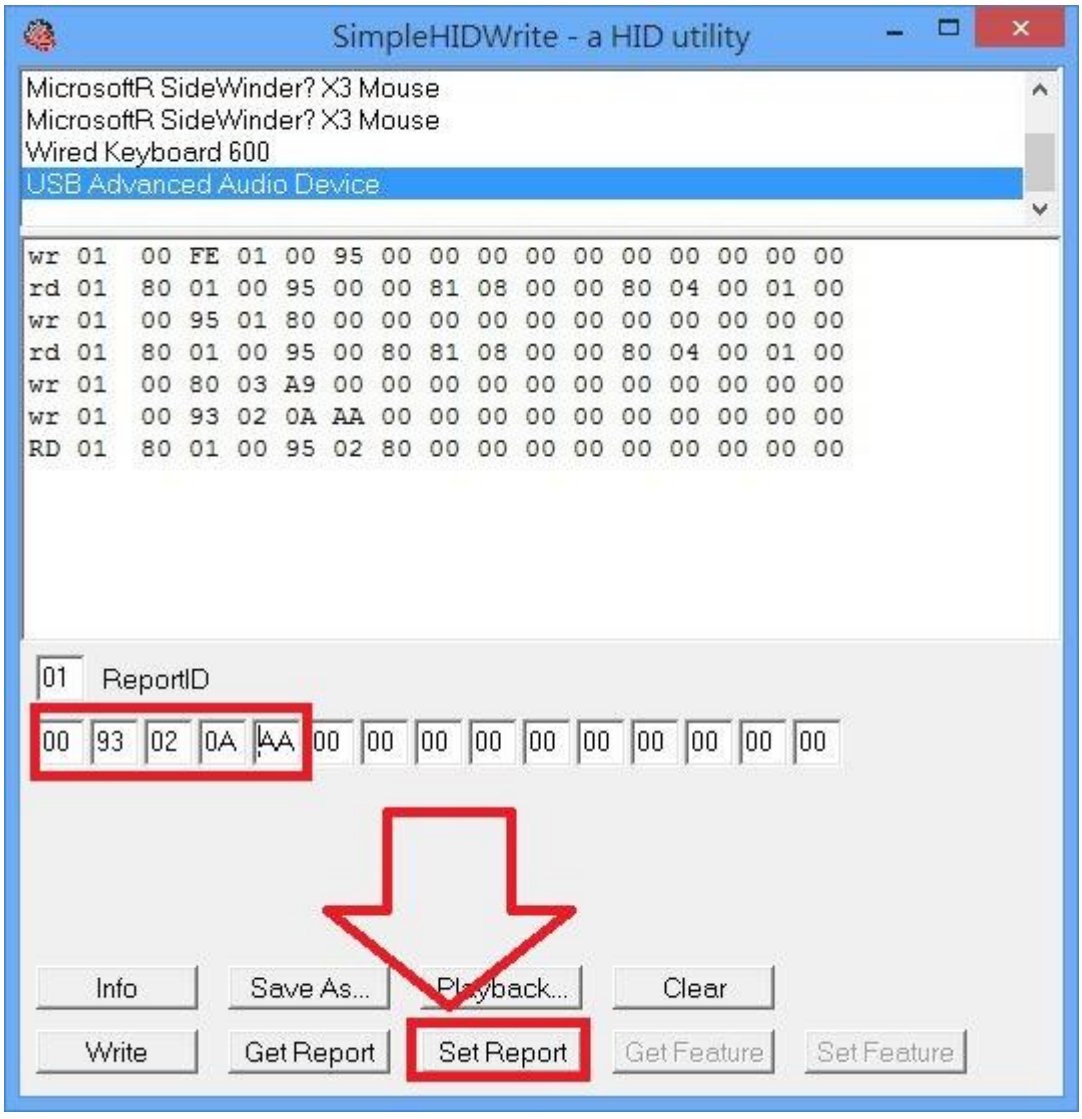
0x82 = 0x00



// (initial data length and i2c start) 00 93 02 0a aa

STEP 3: Write ADDR 0x93 = 0x0A

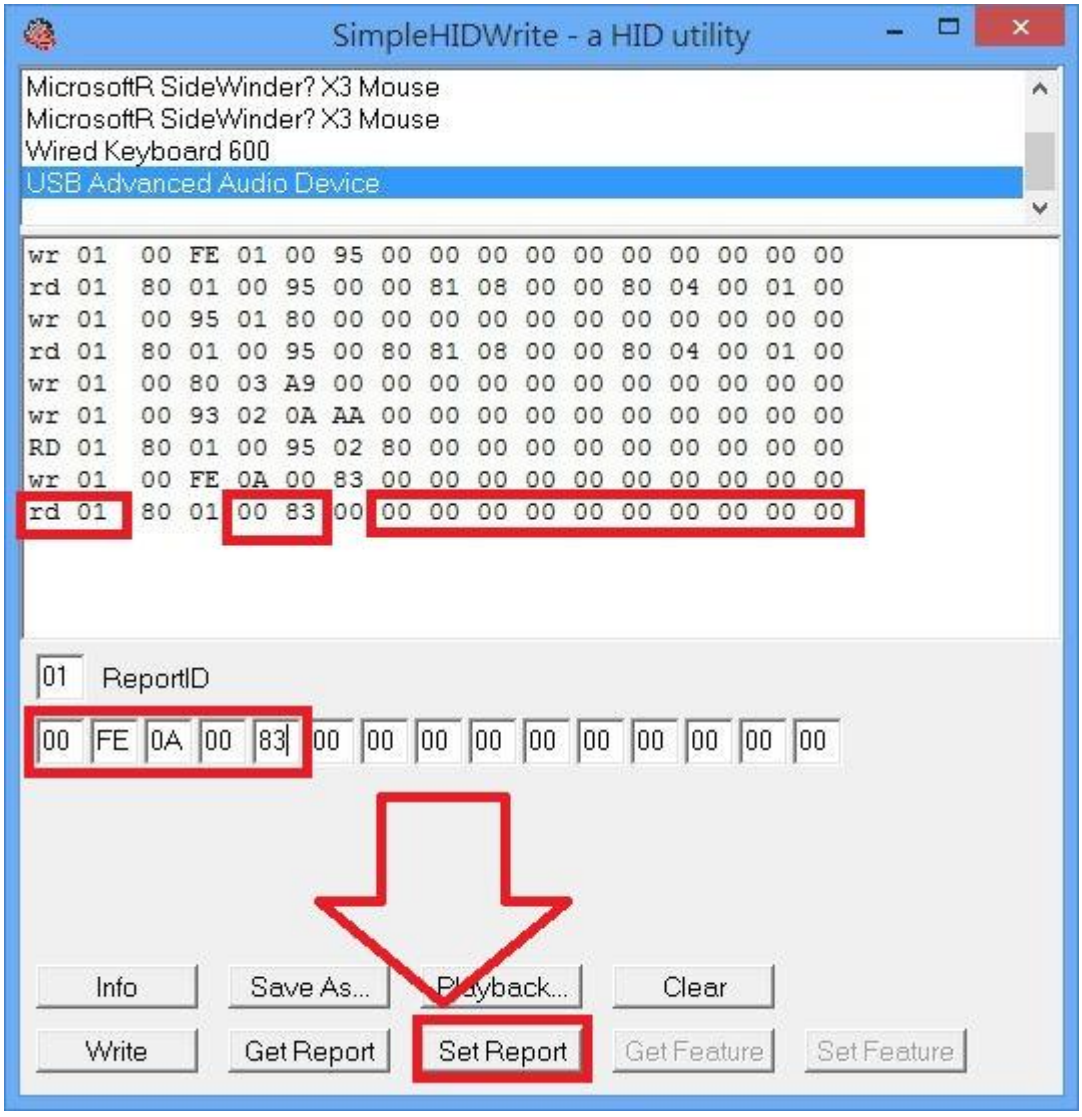
0x94 = 0xAA



// (read hid data) 00 fe 0a 00 83

STEP 4: Use HID Get_Input_Report to read from ADDR 0x00, Length = 10 Bytes (CM65xx HID report limitation for Data Read length)

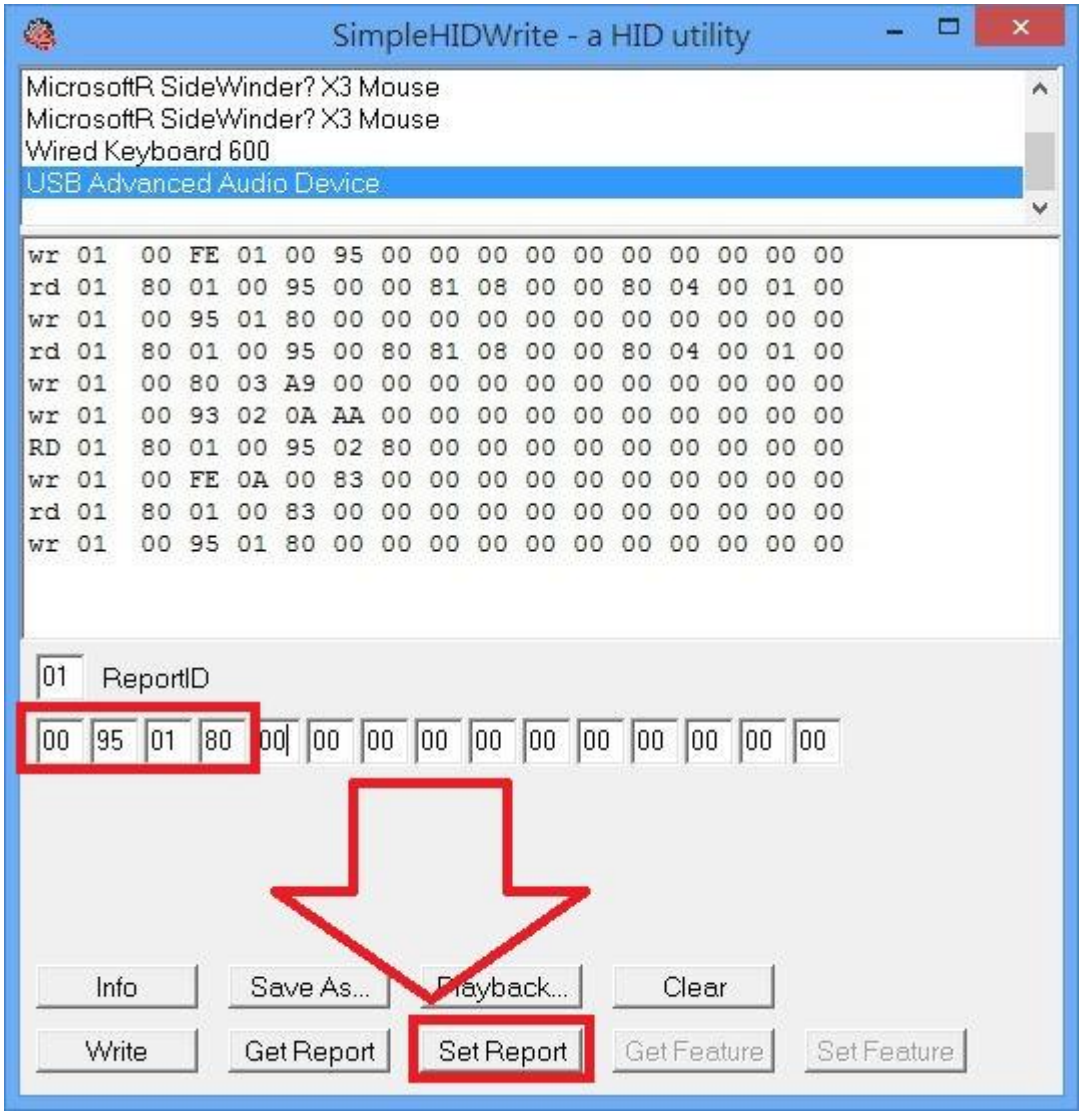
If I2C runs correct, the STEP 3 USB Command will be finished with normal ACK;
if not, you'll get a NAK.



5.2 I2C Write data from EEPROM

// (initial i2c master mode) 00 95 01 80

STEP 1: Write ADDR 0x95 = 0x80 (Select I2C Master)

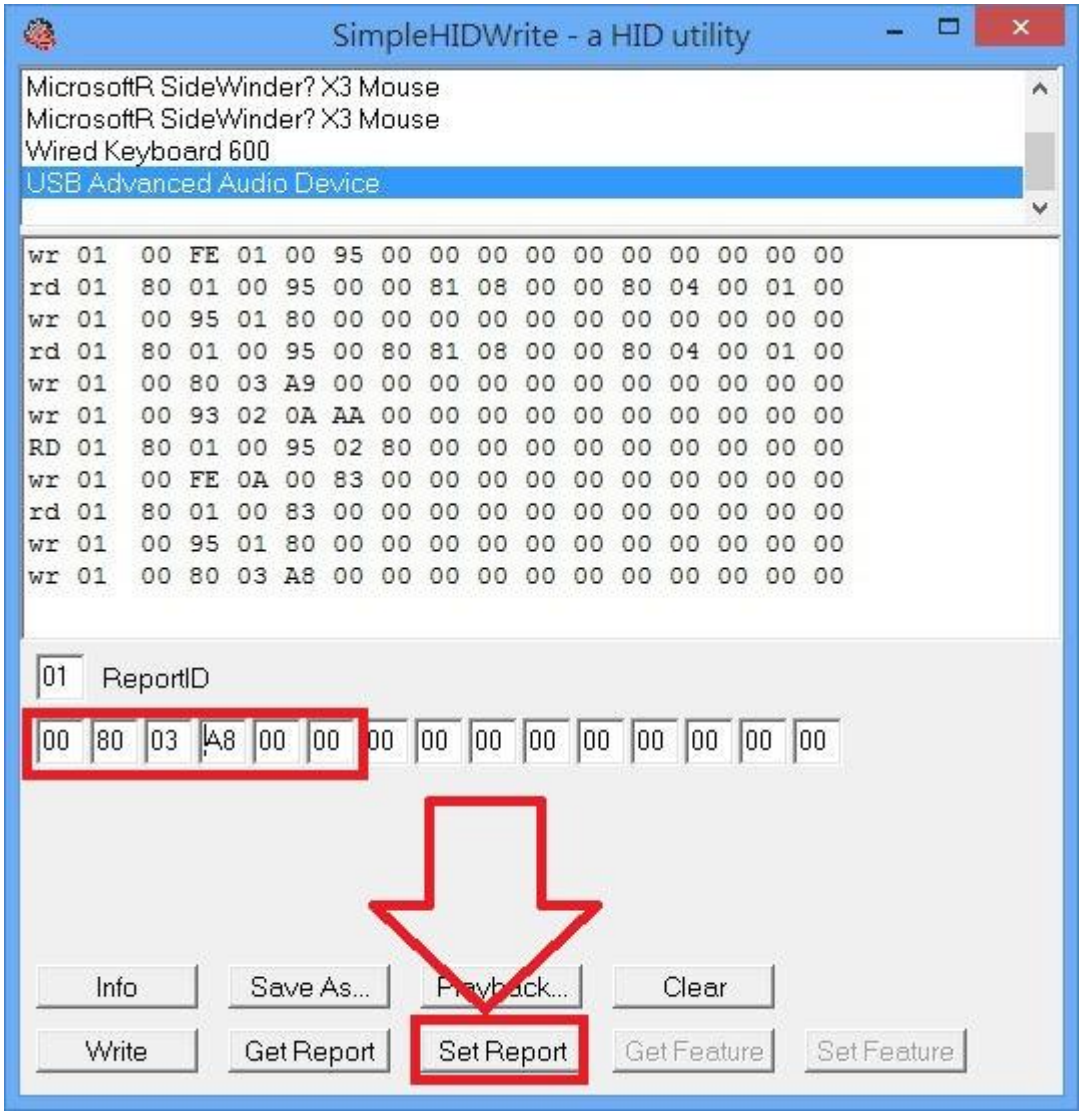


// (initial device address and MAP) 00 80 03 a8 00 00

STEP 2: Write ADDR 0x80 = 0xA8 (Write Address A8)

0x81 = 0x00 (Register low byte of slave device)

0x82 = 0x00 (register high byte of slave device)



// (write 12 bytes data: CM65xx HID report limitation for Write data length) 00 83 0C 11 22 33 44 55 66 77 88 99 aa bb cc

STEP 3: Write ADDR 0x83 = 0x11 (I2C Data for write)

0x84 = 0x22

0x85 = 0x33

0x86 = 0x44

0x87 = 0x55

0x88 = 0x66

0x89 = 0x77

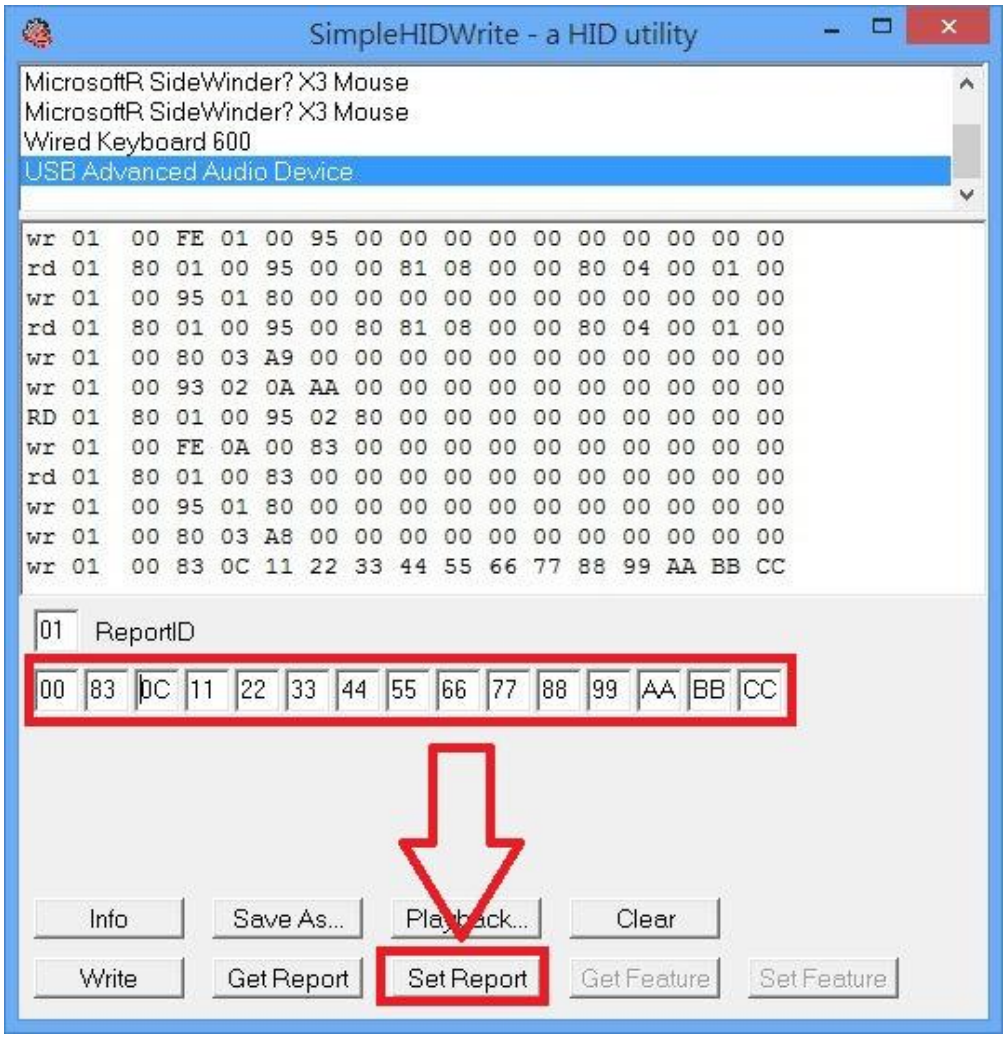
0x8A = 0x88

0x8B = 0x99

0x8C = 0xAA

0x8D = 0xBB

0x8E = 0xCC

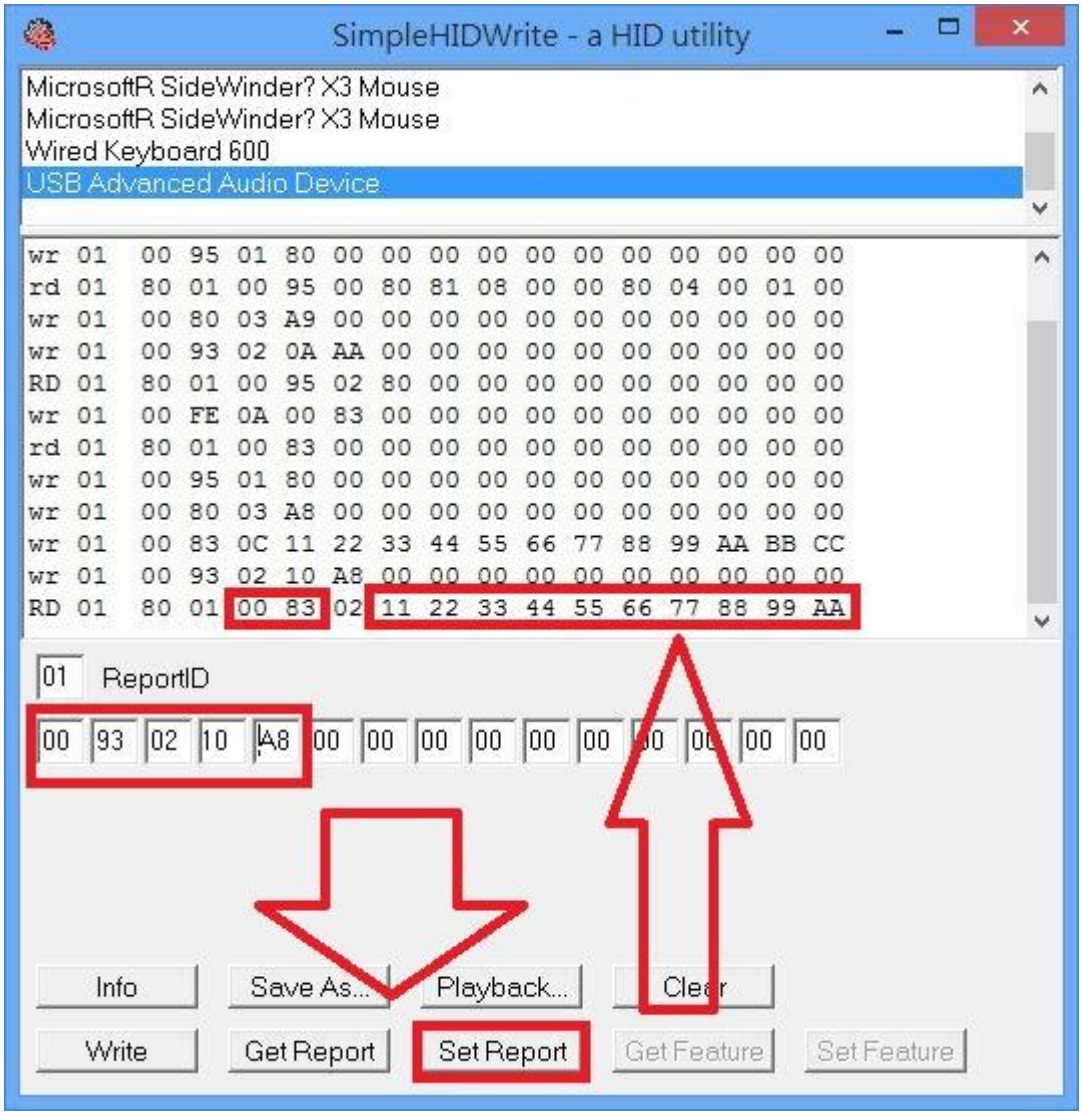


// (initial data length and i2c start) 00 93 02 10 a8

STEP 4: Write ADDR 0x93 = 10 (Control and status register0: 16bytes, maximum length)

0x94 = A8 (Control and status register1: Trigger command in 400KHz fast mode)

If I2C runs correct, the STEP 4 USB Command will be finished with normal ACK;
if not, you'll get a NAK.



Appendix A: CM65xx I2C Control Registers

(master mode)I²C Slave Device Address and Read/Write Control Register (Power Reset)

Address: 0x80

Bits	R/W	Bit Mnemonic	Description	Default
7-1	R/W	SA_reg	The target slave device address.	0xA8(POR)
0	R/W	SA_reg	1: read, 0: write	1'b0(POR)

I²C Memory Address Pointer (MAP) of Slave Device (Power Reset)

Address: 0x81

Bits	R/W	Bit Mnemonic	Description	Default
7-0	R/W	MAP_reg	The register low byte address of salve device to be read or written.	8'b0(POR)

I²C Memory Address Pointer (MAP2) of Slave Device (Power Reset)

Address: 0x82

Bits	R/W	Bit Mnemonic	Description	Default
7-0	R/W	MAP2_reg	The register high byte address of salve device to be read or written.	8'b0(POR)

I²C Data Register (Power Reset)

Address: 0x83 ~ 0x92

Bits	R/W	Bit Mnemonic	Description	Default
7-0	R/W	data0~ data15	The data read from or written to the slave device.	8'b0(POR)

I²C Control and Status Register 0 (Power Reset)

Address: 0x93

Bits	R/W	Bit	Description	Default

		Mnemonic		
7-0	R/W	i2c_ctrl_reg1	Data length of read/write command 8'h1: 1 byte, minimum length 8'h2: 2 bytes … 8'h7: 7 bytes 8'h10: 16 bytes, maximum length Others: Reserved	0x14 (POR)

I²C Control and Status Register 1 (Power Reset)

Address: 0x94

Bits	R/W	Bit Mnemonic	Description	Default
7	R/W	i2c_start	Trigger I2C read/write command 0->1: trigger I2C read/write command. 1->0: I2C interface had completed current task. 0 : I2C interface is idle and ready for work. 1 : I2C interface is running.	1'b0 (POR)
6	R/W	i2c_reset	Reset I2C interface 0 : Not reset I2C interface 1 : Reset I2C interface	1'b0 (POR)
5	R/W	map_len	MAP length 0 : 8-bit MAP 1 : 16-bit MAP	1'b0 (POR)
4	R/W	clk_sync	Clock Synchronization 0: off 1: on, when slave pull-down SCLK, master would pause	1'b1 (POR)
3	R/W	fast_std	I2C speed mode 0 : Standard mode, 100kHz 1 : Fast mode, 400kHz	1'b0 (POR)
2	R/W	map_only	MAP only write command 0 : Write command. 1 : MAP only write command.	1'b0 (POR)
1	R/W	auto_rd	Auto read command 0 : Read command. 1 : Auto read command.	1'b1 (POR)
0	R	i2c_ctrl_reg2	Slave NACK error occur 1 : No error 2 : Slave NACK error occur	1'b0 (POR)

*Note: Write-MAP-Only: An operation which only writes the register MAP the salve device

I²C Download Control and Status (Power Reset)

Address: 0x95

Bits	R/W	Bit Mnemonic	Description	Default
7	R/W	I2c_mas_sel	I2C master/slave select	1'b1 (POR)
6-4	R/W		Reserved	3'b000
3	RO	CHKSUM_ERR	Check sum Error 1. If in LD_PHASE, the check sum value was calculated by I2C load data. 2. If in CHK_PHASE, the check sum value was calculated by SRAM read content.	1'b0
2	RO	CHK_FINISH	CHECK phase done 1: finish download data CHECK	1'b0
1	R/W	CHK_PHASE	MCU select CHECK phase to read SRAM data for check-sum check. 1: enable (after disable LD_PHASE) 0: set 0 after complete	1'b0 (POR)
0	R/W	LD_PHASE	MCU select LOAD phase to access SRAM from download. 1: enable 0: set 0 after complete	1'b0 (POR)

I²C Clock Period Setting (Power Reset)

Address: 0x96

Bits	/W	Bit Mnemonic	Description	Default
7	W	CHG_ENABLE	MCU can program I2C clock; 1'b1: enable	1'b0 (POR)
6	R/W		Reserved	1'b0
5-0	W	CHG_FREQ	Set I2C-master clock period. The clock period=83.3*5*(CHG_FREQ+1) Ex: CHG_FREQ = 6'd48 I2C Clock Period=83.3*5*(48+1)=20408ns HW limitation CHG_FREQ >= 6'h3	6'h0 (POR)

Appendix B: CM65xx GPIO Control Registers

GPO Data Register (Power Reset)

Address Offset: C0-C1h

Bits	R/W	Bit Mnemonic	Description	Default
15:0	R/W	GPO_0_reg GPO_1_reg	GPO data register which represents	16'h0

GPI Data Register (Power Reset)

Address Offset: C2-C3h

Bits	R/W	Bit Mnemonic	Description	Default
15:0	R		GPI data register which represents	16'h0

GPIO Direction Control Register (Power Reset)

Address Offset: C4-C5h

Bits	R/W	Bit Mnemonic	Description	Default
15:0	R/W	GPOE_0 GPOE_1	GPIO output enable register which represents for pin XGPIO[15:0] 1: the corresponding pins are used as output 0: the corresponding pins are used as input	16'h0

GPIO Interrupt Enable Mask Register (Power Reset)

Address Offset: C6-C7h

Bits	R/W	Bit Mnemonic	Description	Default
15:0	R/W	GPI_EN	GPIO_E, GPIO interrupt enable mask which represents for pins, XGPIO[15:0] 1: enable, 0: disable	16'h0

GPIO De-bouncing Register (Power Reset)

Address Offset: C8-C9h

Default Value: 0000h (MSB -> LSB)

Bits	R/W	Bit Mnemonic	Description	Default
15:0	R/W	GPI_Deb	Enable the clock scale of mini-second (32 ms) for de-bouncing, default 1	16'h0

			1: enable, 0: disable	

General Control (Power Reset)

Address Offset: Cah

Bits	R/W	Bit Mnemonic	Description	default
7	R/W		Reserved	1'b0
6	R/W	General_Ctrl[6]	LED3 Control Selector (0: HW, 1: SW)	1'b0
5	R/W	General_Ctrl[5]	LED3 SW Output Enable	1'b0
4	R/W	General_Ctrl[4]	LED2 Control Selector (0: HW, 1: SW)	1'b0
3	R/W	General_Ctrl[3]	LED2 SW Output Enable	1'b0
2	R/W	General_Ctrl[2]	LED1 Control Selector (0: HW, 1: SW)	1'b0
1	R/W	General_Ctrl[1]	LED1 SW Output Enable	1'b0
0	R/W	General_Ctrl[0]	Buzzer Output Enable	1'b0

Buzzer Freq1 (Power Reset)

Address Offset: CBh

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	Buzzer_Freq_1	Buzzer Output Frequency Counter 1 (Base On 23.44 KHz Clock)	8'h8

Buzzer Duty1 (Power Reset)

Address Offset: 0xCCh

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	Buzzer_Duty_1	Buzzer Output Duty Cycle Counter 1 (Base On 23.44 KHz Clock)	8'h4

Buzzer Freq2 (Power Reset)

Address Offset: 0xCDh

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	Buzzer_Freq_2	Buzzer Output Frequency Counter 2 (Base On 23.44 KHz Clock)	8'h4C

Buzzer Duty2 (Power Reset)

Address Offset: 0xCEh

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	Buzzer_Duty_2	Buzzer Output Duty Cycle Counter 1 (Base On 23.44 KHz Clock)	8'h26

Buzzer On Time (Power Reset)

Address Offset: 0xCFh

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	Buzzer_ON_Time	Buzzer Output ON Time	8'h50

Buzzer Cycle Time (Power Reset)

Address Offset: 0XD0h

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	Buzzer_Cycle_Time	Buzzer Output ON Time	8'hC0

LED Freq (Power Reset)

Address Offset: 0XD1h

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	LED_Freq	LED1 Output Frequency	8'h17

LED Duty (Power Reset)

Address Offset: 0Xd2h

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	LED_Duty	LED1 Output Duty Cycle	8'h18

LED1 On Time (Power Reset)

Address Offset: 0xD3h

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	LED1_ON_Time	LED1 Output ON Time	8'h0c

LED1 Cycle Time (Power Reset)

Address Offset: 0xD4h

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	LED1_Cycle_Time	LED1 Output Cycle Time Time unit is ~2.2msec 22msec * 24 = 528msec LED will ON-OFF 2 times per sec	8'h18

LED2 On Time (Power Reset)

Address Offset: 0xD5h

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	LED2_ON_Time	LED2 Output ON Time	8'h31

LED2 Cycle Time (Power Reset)

Address Offset: 0xD6h

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	LED2_Cycle_Time	LED2 Output Cycle Time	8'h30

LED3 On Time (Power Reset)

Address Offset: 0xD7h

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	LED3_ON_Time	LED3 Output ON Time	8'h18

LED3 Cycle Time (Power Reset)

Address Offset: 0xD8h

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	LED3_Cycle_Time	LED3 Output Cycle Time LED will ON-OFF 1 times per sec	8'h30

Bounding Option

Address Offset: 0xD9h

Bits	R/W	Bit Mnemonic	Description	Default
7	R/W	URSTN_MASK	USB Reset Mask: If this bit set to 1, we will ignore USB	2'b0

			reset from host.	
6	R/W	SOF_SEL	SOF select: 0: SOF is coming from host command 1: SOF is coming from IR clock counter	2'b0
5	R	SEL6	Pin SEL6	1'b0
4	R	SEL5	Pin SEL5	1'b0
3	R	SEL4	Pin SEL4	1'b0
2	R	SEL3	Pin SEL3	1'b0
1	R	SEL2	Pin SEL2	1'b0
0	R	SEL1	Pin SEL1	1'b0

GPO Switch Source (Power Reset)

Address Offset: 0xDAh

Bits	R/W	Bit Mnemonic	Description	default
7-4	R/W	GPO_SW0_H	4'h0:GPO[0]=gpo[0] 4'h1:GPO[0]=LED1 4'h2:GPO[0]=LED2 4'h3:GPO[0]=LED3 4'h4:GPO[0]=Buzzer Others: GPO[0]=1'b0	4'h0
3-0	R/W	GPO_SW0_L	4'h0:GPO[1]=gpo[1] 4'h1:GPO[1]=LED1 4'h2:GPO[1]=LED2 4'h3:GPO[1]=LED3 4'h4:GPO[1]=Buzzer Others: GPO[1]=1'b0	4'h0

Address Offset: 0xDBh

Bits	R/W	Bit Mnemonic	Description	default
7-4	R/W	GPO_SW1_H	4'h0:GPO[2]=gpo[2] 4'h1:GPO[2]=LED1 4'h2:GPO[2]=LED2 4'h3:GPO[2]=LED3 4'h4:GPO[2]=Buzzer Others: GPO[2]=1'b0	4'h0
3-0	R/W	GPO_SW1_L	4'h0:GPO[3]=gpo[3] 4'h1:GPO[3]=LED1 4'h2:GPO[3]=LED2 4'h3:GPO[3]=LED3 4'h4:GPO[3]=Buzzer Others: GPO[3]=1'b0	4'h0

Address Offset: 0xDCh

Bits	R/W	Bit Mnemonic	Description	default
7-4	R/W	GPO_SW2_H	4'h0:GPO[4]=gpo[4] 4'h1:GPO[4]=LED1 4'h2:GPO[4]=LED2 4'h3:GPO[4]=LED3 4'h4:GPO[4]=Buzzer Others: GPO[4]=1'b0	4'h0
3-0	R/W	GPO_SW2_L	4'h0:GPO[5]=gpo[5] 4'h1:GPO[5]=LED1 4'h2:GPO[5]=LED2 4'h3:GPO[5]=LED3 4'h4:GPO[5]=Buzzer Others: GPO[5]=1'b0	4'h0

Address Offset: 0xDDh

Bits	R/W	Bit Mnemonic	Description	default
7-4	R/W	GPO_SW3_H	4'h0:GPO[6]=gpo[6] 4'h1:GPO[6]=LED1 4'h2:GPO[6]=LED2 4'h3:GPO[6]=LED3 4'h4:GPO[6]=Buzzer Others: GPO[6]=1'b0	4'h0
3-0	R/W	GPO_SW3_L	4'h0:GPO[7]=gpo[7] 4'h1:GPO[7]=LED1 4'h2:GPO[7]=LED2 4'h3:GPO[7]=LED3 4'h4:GPO[7]=Buzzer Others: GPO[7]=1'b0	4'h0

Address Offset: 0xDEh

Bits	R/W	Bit Mnemonic	Description	default
7-4	R/W	GPO_SW4_H	4'h0:GPO[8]=gpo[8] 4'h1:GPO[8]=LED1 4'h2:GPO[8]=LED2 4'h3:GPO[8]=LED3 4'h4:GPO[8]=Buzzer Others: GPO[8]=1'b0	4'h0
3-0	R/W	GPO_SW4_L	4'h0:GPO[9]=gpo[9] 4'h1:GPO[9]=LED1 4'h2:GPO[9]=LED2 4'h3:GPO[9]=LED3 4'h4:GPO[9]=Buzzer	4'h0

			Others: GPO[9]=1'b0	

Address Offset: 0xDFh

Bits	R/W	Bit Mnemonic	Description	default
7-4	R/W	GPO_SW5_H	4'h0:GPO[10]=gpo[10] 4'h1:GPO[10]=LED1 4'h2:GPO[10]=LED2 4'h3:GPO[10]=LED3 4'h4:GPO[10]=Buzzer Others: GPO[10]=1'b0	4'h0
3-0	R/W	GPO_SW5_L	4'h0:GPO[11]=gpo[11] 4'h1:GPO[11]=LED1 4'h2:GPO[11]=LED2 4'h3:GPO[11]=LED3 4'h4:GPO[11]=Buzzer Others: GPO[11]=1'b0	4'h0

Address Offset: 0XE0h

Bits	R/W	Bit Mnemonic	Description	default
7-4	R/W	GPO_SW6_H	4'h0:GPO[12]=gpo[12] 4'h1:GPO[12]=LED1 4'h2:GPO[12]=LED2 4'h3:GPO[12]=LED3 4'h4:GPO[12]=Buzzer Others: GPO[12]=1'b0	4'h0
3-0	R/W	GPO_SW6_L	4'h0:GPO[13]=gpo[13] 4'h1:GPO[13]=LED1 4'h2:GPO[13]=LED2 4'h3:GPO[13]=LED3 4'h4:GPO[13]=Buzzer Others: GPO[13]=1'b0	4'h0

Address Offset: 0Xe1h

Bits	R/W	Bit Mnemonic	Description	default
7-4	R/W	GPO_SW7_H	4'h0:GPO[14]=gpo[14] 4'h1:GPO[14]=LED1 4'h2:GPO[14]=LED2 4'h3:GPO[14]=LED3 4'h4:GPO[14]=Buzzer Others: GPO[14]=1'b0	4'h0
3-0	R/W	GPO_SW7_L	4'h0:GPO[15]=gpo[15] 4'h1:GPO[15]=LED1 4'h2:GPO[15]=LED2 4'h3:GPO[15]=LED3	4'h0

			4'h4:GPO[15]=Buzzer Others: GPO[15]=1'b0	

GPI Remote Choose

Address Offset: 0xE2~E3h

Bits	R/W	Bit Mnemonic	Description	default
15:0	R/W	GPI_RWL GPI_RWH	D0==1'b1:GPI[0] remote wake up enable D1==1'b1:GPI[1] remote wake up enable D2==1'b1:GPI[2] remote wake up enable D3==1'b1:GPI[3] remote wake up enable D4==1'b1:GPI[4] remote wake up enable D5==1'b1:GPI[5] remote wake up enable D6==1'b1:GPI[6] remote wake up enable D7==1'b1:GPI[7] remote wake up enable D8==1'b1:GPI[8] remote wake up enable D9==1'b1:GPI[9] remote wake up enable D10==1'b1:GPI[10] remote wake up enable D11==1'b1:GPI[11] remote wake up enable D12==1'b1:GPI[12] remote wake up enable D13==1'b1:GPI[13] remote wake up enable D14==1'b1:GPI[14] remote wake up enable D15==1'b1:GPI[15] remote wake up enable	16'h0

GPIO pull-up/down Control Register0

Address Offset: 0xE4

Bits	R/W	Bit Mnemonic	Description	default

7	R/W	GPIO_PD0[7]	GPIO_1 ~ GPIO_7 pad control 1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
6	R/W	GPIO_PD0[6]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
5	R/W	GPIO_PD0[5]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
4	R/W	GPIO_PD0[4]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
3	R/W	GPIO_PD0[3]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
2	R/W	GPIO_PD0[2]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
1	R/W	GPIO_PD0[1]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
0	R/W	GPIO_PD0[0]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1

GPIO pull-up/down Control Register1

Address Offset: 0xE5

Bits	R/W	Bit Mnemonic	Description	default
7	R/W	GPIO_PD1[7]	GPIO_8~GPIO_16 pad control 1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
6	R/W	GPIO_PD1[6]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
5	R/W	GPIO_PD1[5]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
4	R/W	GPIO_PD1[4]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
3	R/W	GPIO_PD1[3]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
2	R/W	GPIO_PD1[2]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
1	R/W	GPIO_PD1[1]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
0	R/W	GPIO_PD1[0]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1

GPIO pull-up/down Control Register2

Address Offset: 0xE6

Bits	R/W	Bit Mnemonic	Description	default
7	R/W	GPIO_PD2[7]	GPIO17~GPIO24 pad control 1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
6	R/W	GPIO_PD2[6]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
5	R/W	GPIO_PD2[5]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
4	R/W	GPIO_PD2[4]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
3	R/W	GPIO_PD2[3]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
2	R/W	GPIO_PD2[2]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
1	R/W	GPIO_PD2[1]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
0	R/W	GPIO_PD2[0]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1

GPIO pull-up/down Control Register3

Address Offset: 0xE7

Bits	R/W	Bit Mnemonic	Description	default
7	R/W	GPIO_PD3[7]	GPIO25~GPIO32 pad control 1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
6	R/W	GPIO_PD3[6]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
5	R/W	GPIO_PD3[5]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
4	R/W	GPIO_PD3[4]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
3	R/W	GPIO_PD3[3]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
2	R/W	GPIO_PD3[2]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
1	R/W	GPIO_PD3[1]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1
0	R/W	GPIO_PD3[0]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1