

---

# ST-NXP Wireless

## IMPORTANT NOTICE

Dear customer,

As from August 2<sup>nd</sup> 2008, the wireless operations of STMicroelectronics have moved to a new company, ST-NXP Wireless.

As a result, the following changes are applicable to the attached document.

- **Company name - STMicroelectronics NV** is replaced with **ST-NXP Wireless**.
- **Copyright** - the copyright notice at the bottom of the last page “© STMicroelectronics 200x - All rights reserved”, shall now read: “© ST-NXP Wireless 200x - All rights reserved”.
- **Web site** - <http://www.st.com> is replaced with <http://www.stnwireless.com>
- **Contact information** - the list of sales offices is found at <http://www.stnwireless.com> under Contacts.

If you have any questions related to the document, please contact our nearest sales office. Thank you for your cooperation and understanding.

ST-NXP Wireless



# AN2533

## Application note

---

### STw481x register access, how to use STw481x indirect addressed registers

---

#### Introduction

STw481x applies to STw4810, STw4811N and STw4811M part numbers.

This document provides some guidance to access STw481x I2C registers and STw481x indirect addressed registers. The two I2C interfaces, main I2C and USB I2C, are described, as well as the procedure to read and write indirect addressed registers.

# 1 Registers access

STw481x uses industry standard I2C bus for registers access. The device supports two I2C bus interfaces. The main interface (SDA,SCL) controls the power management and all the programmable functions. The USB interface (USBSDA, USBSCL) is dedicated to USB control.

STw481x allows to work only with the main I2C interface to control all the functions including the USB, via 'usb\_i2c\_ctrl' bit of power control register (bit 0 in power control register at address 06h).

The device ID is 5Ah (01011010) in read mode and 5Bh (01011011) in write mode.

## Code conventions

Assuming that STw481x I2C registers are accessed through the following software functions:

```
void STw481x_main_I2C_write(unsigned char address, unsigned char
data);
unsigned char STw481x_main_I2C_read(unsigned char address);
void STw481x_USB_I2C_write(unsigned char address, unsigned char
data);
unsigned char STw481x_USB_I2C_read(unsigned char address);
```

**Table 1. STw481x registers**

Address	Description	I2C control
00h to 10h	USB registers	Main I2C or USB I2C
11h	Configuration 1 register	Main I2C
12h to 1Dh	Reserved	
1Eh and 1Fh	Power control registers used for indirect addressing	Main I2C
20h	Configuration 1 register	Main I2C
21h	Vcore_sleep register	Main I2C

## 2 Power management registers (main I2C)

Registers at addresses 11h, 20h and 21h are dedicated to power management and control. Those registers are directly accessed by I2C bus.

Registers 1Eh and 1Fh used for indirect addressing are also accessed by the main I2C bus.

## 3 Power control registers with indirect addressing

The 4-bit wide internal registers are accessed by indirect addressing using registers 1Eh and 1Fh registers from the main I2C bus.

For indirect addressing using 1Eh and 1Fh registers, power control registers address is coded on 5 bits.

To get access to power control registers, both 1Eh and 1Fh registers must be used in a specific order.

- 1Fh register contains MSB (bits 3 and 4) of power control register address, and must be written first.
- 1Eh register contains LSB (bits 0 to 2) of Power control register address, data and EN (write) bit.

Table 2. Power control registers with indirect addressing

I2C register 1Fh						I2C register 1Eh									
Reserved						Power control address		DATA				EN			
						MSB	LSB								
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

### 3.1 Indirect registers write

Indirect registers write is performed in two steps:

- Write MSB of power control register address in I2C register 1Fh
- Write LSB of power control register address, data and EN=1 in I2C register 1Eh

#### 3.1.1 Example

Power control register at address 05h, set Vcore to 1.34V (vcore\_sel = 1100)

Power control register address = 05h = 0000 0101 = 000 00 101

Address[4:3] = 00

-> write 000000 00 = 0000 0000 = **00h at I2C address 1Fh**

Address[2:0] = 101

Data = 1100

EN = 1

-> write 101 1100 1 = 1011 1001 = **B9h at address 1Eh**

### 3.1.2 C code example

```
void STw481x_power_control_write(unsigned char PCR_address,
                                unsigned char data)
{
    unsigned char temp_address = 0;
    unsigned char temp_data = 0;

    temp_address = (PCR_address >> 3) & 0x03;
    STw481x_main_I2C_write(0x1F, temp_address);
    temp_address = (PCR_address << 5) & 0xE0;
    temp_data = (data << 0x01) & 0x1E;
    temp_data |= 0x01;
    temp_data += temp_address;
    STw481x_main_I2C_write(0x1E, temp_data);
}
```

## 3.2 Indirect registers read

As LSB of power control register and of register data are written simultaneously in the same I2C register, it is not easy to re-read data from the power control registers. Register 1Fh must be written with EN=0 (no data write, only LSB of address take into account), then the 1Eh register must be read. This provides in return the corresponding power register address LSB and data.

- Write MSB of power control register address in I2C register 1Fh
- Write LSB of power control register address, dummy data (0000) and EN=0 in I2C register 1Eh
- Read register 1Eh to get the LSB of power control register and data.

### 3.2.1 Example

Power control register at address 09h, read  
 Power control register address = 09h = 0000 1001 = 000 01 001  
 Address[4:3] = 01  
 -> **write** 000000 01 = 0000 0001 = **01h at I2C address 1Fh**  
 Address[2:0] = 001  
 Data = 0000  
 EN = 0  
 -> **write** 001 0000 0 = 0010 0000 = **20h at address 1Eh**  
 -> **Read at I2C address 1Eh**  
 Returned value = 3Eh = 0011 1110 = 001 1111 0  
 Address[2:0] = 001  
 Data = 1111 ->  
 data[3]: vaux\_sleep = 1  
 data[2]: reserved = 1  
 data[1]: vio\_vmem\_sleep = 1  
 data[0]: vcore\_sleep = 1  
 EN = 0

### 3.2.2 C code example

```
unsigned char STw481x_power_control_read(unsigned char PCR_address)
{
    unsigned char temp_address = 0;
    unsigned char temp_data = 0;

    temp_address = (PCR_address >> 3) & 0x03;
    STw481x_main_I2C_write(0x1F, temp_address);
    temp_address = (PCR_address << 5) & 0xE0;
    temp_data = temp_address;
    STw481x_main_I2C_write(0x1E, temp_data);
    temp_data = STw481x_main_I2C_read(0x1E);
    temp_data = (temp_data >> 1) & 0x0F;
    return temp_data;
}
```

## 3.3 Changing an indirect register value

To change some bits of an indirect accessed register without changing undesired bits proceed as follows:

- read the indirect register
- change the desired bits
- read the result which was written in the corresponding indirect register

### 3.3.1 Example: merging USB I2C and main I2C

To merge both I2C buses, set bit `usb_i2c_ctrl` (bit 0 of power control register 06h) to access USB registers through the main I2C interface.

- Read data in 06h power control register,
- write '1' as bit 0 value, and then
- write the corresponding value into power control register 06h.

#### C code example

```
void STw481x_merge_I2C()
{
    unsigned char data = 0;

    data = STw481x_power_control_read(0x06);
    data |= 0x01;
    STw481x_power_control_write(0x06, data);
}
```

### 3.3.2 Example: splitting USB I2C and main I2C

To split the two I2C buses, reset bit `usb_i2c_ctrl` (bit 0 of power control register 06h), to access separately USB registers through the dedicated I2C\_USB bus.

- Read data in 06h power control register,
- write '0' as bit 0 value, and then
- write the corresponding value into power control register 06h.

#### C code example

```
void STw481x_split_I2C()
{
    char data = 0;

    data = STw481x_power_control_read(0x06);
    data &= 0x0E;
    STw481x_power_control_write(0x06, data);
}
```

## 4 USB I2C registers

USB registers (00h to 10h) can be accessed using two methods:

- Using two different I2C buses, USBSDA and USBSCL connected to an I2C controller. Device ID is 5Ah (01011010) in read mode and 5Bh (01011011) in write mode.
- Using the main I2C interface, so that USB and power management registers are accessed using the same interface.

By default, the two I2C interfaces are separated. To enable access to USB registers from the main I2C, `usb_i2c_ctrl` (power control register 06h, bit 0) must be set to 1 using indirect addressing as described in [Chapter 3: Power control registers with indirect addressing](#).

## 5 Revision history

**Table 3. Document revision history**

Date	Revision	Changes
05-Oct-2007	1	Initial release.

**Please Read Carefully:**

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

**UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.**

**UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.**

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2007 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

[www.st.com](http://www.st.com)