

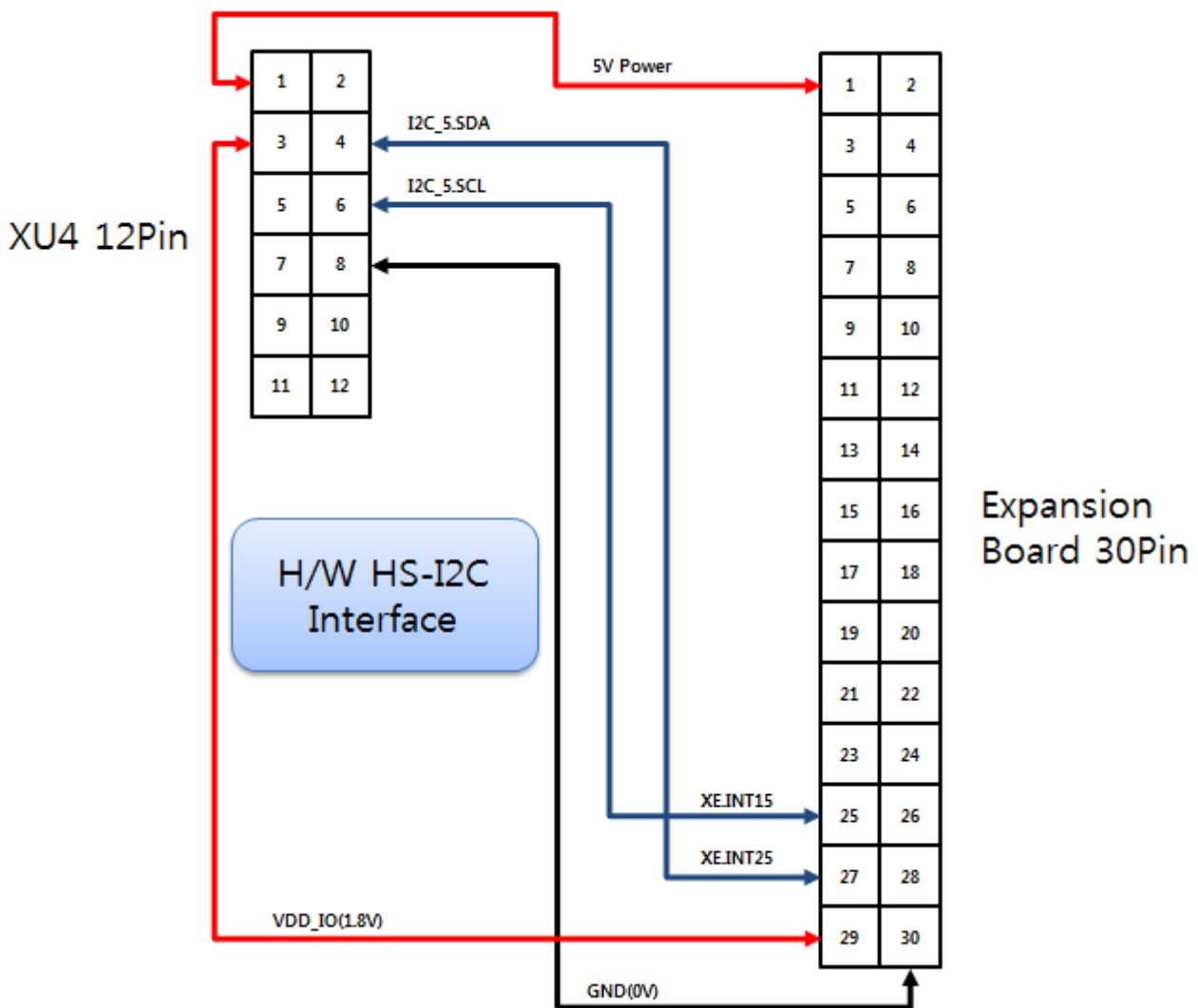
HS-I2C

How to I2C interface using the Expansion Board.

The Expansion Board has two I2C devices.
You can use the devices via HW-HSI2C.

HW-HSI2C Pins

Pin Number	Net Name	Export Number	Description
4	GPA2[2](I2C_5.SDA)	187	SDA
6	GPA2[3](I2C_5.SCL)	188	SCL



**Using the HW-HSI2C(/dev/i2c-1) with Expansion Board. I2C Physical Address : 0x12CB0000
Must Read : Can't use SMBus Quick Write command on this bus.(Not support i2cdetect)**

HW-HSI2C Test Code

[xu4_hsi2c.c](#)

```
//-----  
-----  
//  
// ODR0ID-XU4 with Shifter Shield I2C Test Application.  
// I2C Node : /dev/i2c-3  
//  
// Compile : gcc -o <create excute file name> <source file name>  
// Run : sudo ./<created excute file name>  
//  
//-----  
-----  
#include <stdio.h>  
#include <stdlib.h>  
#include <stdint.h>  
#include <errno.h>  
#include <string.h>  
#include <fcntl.h>  
#include <sys/ioctl.h>  
  
#include <unistd.h>  
#include <string.h>  
#include <time.h>  
  
//-----  
-----  
//  
// Global handle Define  
//  
//-----  
-----  
  
// I2C definitions  
  
#define I2C_SLAVE 0x0703  
#define I2C_SMBUS 0x0720 /* SMBus-level access */  
  
#define I2C_SMBUS_READ 1  
#define I2C_SMBUS_WRITE 0  
  
// SMBus transaction types  
  
#define I2C_SMBUS_QUICK 0  
#define I2C_SMBUS_BYTE 1  
#define I2C_SMBUS_BYTE_DATA 2  
#define I2C_SMBUS_WORD_DATA 3  
#define I2C_SMBUS_PROC_CALL 4  
#define I2C_SMBUS_BLOCK_DATA 5  
#define I2C_SMBUS_I2C_BLOCK_BROKEN 6  
#define I2C_SMBUS_BLOCK_PROC_CALL 7 /* SMBus 2.0 */  
#define I2C_SMBUS_I2C_BLOCK_DATA 8
```

```

// SMBus messages
#define I2C_SMBUS_BLOCK_MAX      32    /* As specified in SMBus
standard */
#define I2C_SMBUS_I2C_BLOCK_MAX  32    /* Not specified but we use
same structure */

// Structures used in the ioctl() calls

union i2c_smbus_data
{
    uint8_t    byte;
    uint16_t   word;
    uint8_t    block [I2C_SMBUS_BLOCK_MAX + 2] ; // block [0] is used for
length + one more for PEC
};

struct i2c_smbus_ioctl_data
{
    char        read_write;
    uint8_t     command ;
    int         size;
    union i2c_smbus_data *data;
};

static inline int i2c_smbus_access (int fd, char rw, uint8_t command,
int size, union i2c_smbus_data *data)
{
    struct i2c_smbus_ioctl_data args ;

    args.read_write = rw ;
    args.command    = command ;
    args.size       = size ;
    args.data       = data ;
    return ioctl (fd, I2C_SMBUS, &args) ;
}

int i2c_smbus_read_byte_data(int fd, int reg)
{
    union i2c_smbus_data data;

    if (i2c_smbus_access (fd, I2C_SMBUS_READ, reg, I2C_SMBUS_BYTE_DATA,
&data))
        return -1 ;
    else
        return data.byte & 0xFF ;
}
//-----
//
// I2C:

```

```
//
//-----
-----
// const char *i2cHandleNode = "/dev/i2c-3";
const char *i2cHandleNode = "/dev/i2c-1";
static int i2c_fd = -1;

//-----
-----
//-----
-----
//
// LCD Update Function:
//
//-----
-----

static void data_update (void)
{
    int val1, val2;
    static unsigned long    count = ;

    if(i2c_fd == -1)    return;

    // set BH1780 Sensor I2C Address
    if(ioctl(i2c_fd, I2C_SLAVE, 0x29) < ) {
        fprintf(stdout, "%s : ioctl BH1780 I2C_SLAVE Setup Error!\n",
__func__);
        return;
    }

    // BH1780 Part ID Read
    val1 = i2c_smbus_read_byte_data(i2c_fd, 0x0A);

    // set Pressure Sensor I2C Address
    if(ioctl(i2c_fd, I2C_SLAVE, 0x77) < ) {
        fprintf(stdout, "%s : ioctl PRESSURE I2C_SLAVE Setup Error!\n",
__func__);
        return;
    }

    // Pressure Sensor CHIP ID Read
    val2 = i2c_smbus_read_byte_data(i2c_fd, 0xD0);

    fprintf(stdout, "%s : count = %ld BH1780 ID = 0x%02X, PRESSURE ID =
0x%02X\n", __func__, count++, val1, val2);
}

//-----
-----
//
```

```
// system init
//
//-----
-----
int system_init(void)
{
    if((i2c_fd = open(i2cHandleNode, O_RDWR)) < ) {
        fprintf(stdout, "%s : %s Open Error!\n", __func__,
i2cHandleNode);
        return -1;
    }
    return ;
}

//-----
-----
//
// Start Program
//
//-----
-----
int main (int argc, char *argv[])
{
    if (system_init() < )
    {
        fprintf (stderr, "%s: System Init failed\n", __func__);
        fflush(stdout);
        return -1;
    }

    for(;;) {
        usleep(100000);
        data_update(); fflush(stdout);
    }

    return ;
}

//-----
-----
//-----
-----
```

From:
<http://wiki.odroid.com/> - **ODROID Wiki**

Permanent link:
http://wiki.odroid.com/old_product/accessory/xu3_expansionboard/hsi2c

Last update: **2017/07/21 09:41**



