

MMA7660_PIC.c

```
/*
*****
*           Rhydo Technologies Pvt Ltd, Golden Plaza, Cochin, Kerala, India
*
*           (c) Copyright 2010, Rhydo Technologies, Cochin, Kerala
*
* All rights reserved. Rhydo Technologies's source code is an unpublished work and the use of a
* copyright notice does not imply otherwise. This source code contains confidential,trade secret
* material of Rhydo Technologies. Any attempt or participation in deciphering,decoding,reverse
* engineering or in any way altering the source code is strictly prohibited, unless the prior written
* consent of Rhydo Technologies is obtained.
*
* File name   : MMA7660_PIC.c
*
* Created    : 12 - May - 2010
*
* Description : Source code for reading the value from Accelerometer Sensor
*****
*
*           Microcontroller -- PIC 16 F877 - 40-pin - 8-bit
*
*           Clock Frequency is 20 MHz -- Period in 200 Nano Seconds
*
*           The program reads the output of the X,Y,Z axis output of MMA7660 IC (3 axis accelerometer)
*           send these values through USART continuously.
*           The X,Y and Z axis values are read using I2C protocol.
*           The output baudrate of the USART is 9600.
*****
*/

#include<16f877.h>                               /*Include header file */
#include delay(clock=2000000)                    /*Set clock frequency */
#include rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7,stream=Rec_Dat) /*Set USART to 9600 Baudrate */
#include I2C(master, slow,sda=PIN_C4, scl=PIN_C3) /*Set I2C Settings */

unsigned int i,I2c_Data[10];

/*
*****
* Function    : WriteData
* Description : Function to write a data to EEPROM using I2C
* Parameters  : address, contains the location and data, stores the data
*****
*/
void WriteData(unsigned char address, unsigned char data)
{
    i2c_start();
    i2c_write(0x98);
    i2c_write(address);
    i2c_write(data);
    i2c_stop();
    delay_ms(100);
}

/*
*****
```

MMA7660_PIC.c
MAIN PROGRAM

```
*
*****
*/
void main()
{
    set_tris_c(0x90);
    printf("X");
    WriteData(0x07,0x08);           /* Initialise I2C Module */
    WriteData(0x06,0x00);
    WriteData(0x08,0x1f);
    WriteData(0x09,0x00);
    WriteData(0x07,0x09);

    while(1)
    {
        i2c_start();               /* Write Mode */
        i2c_write(0x98);           /* Write Device address */
        i2c_write(0x00);
        i2c_stop();
        delay_ms(10);
        i2c_start();               /* Read Mode */
        i2c_write(0x99);
        for(i=0;i<7;i++)
        {
            I2c_Data[i]=i2c_read(1); /* Read Data */
        }
        I2c_Data[i]=i2c_read(0);

        printf("\n\rX=%u",I2c_Data[0]); /* Send X-axis value */
        printf(" Y=%u",I2c_Data[1]); /* Send Y-axis value */
        printf(" Z=%u",I2c_Data[2]); /* Send Z-axis value */
        i2c_stop();
        delay_ms(1000);
    }
}

```