

```
/*
 * 以下是ADS1110.h
 *
 */
#ifndef __ADS1110_H
#define __ADS1110_H

#include "stm32f10x.h"
#include "SysTick.h"
//条件编译 1:使用软件模拟I2C
#define TWI_ENABLE 1

#define TWI_SCL_0      GPIOB->BRR=GPIO_Pin_6
#define TWI_SCL_1      GPIOB->BSRR=GPIO_Pin_6
#define TWI_SDA_0      GPIOB->BRR=GPIO_Pin_7
#define TWI_SDA_1      GPIOB->BSRR=GPIO_Pin_7
#define TWI_SDA_STATE  (GPIOB->IDR&GPIO_Pin_7)

enum ENUM_TWI_REPLY
{
    TWI_NACK=0
    ,TWI_ACK=1
};

enum ENUM_TWI_BUS_STATE
{
    TWI_READY=0
    ,TWI_BUS_BUSY=1
    ,TWI_BUS_ERROR=2
};

#define TWI_RETRY_COUNT 3 //重试次数

void TWI_Initialize(void);
u8 TWI_START(void);
void TWI_STOP(void);
u8 TWI_SendByte(u8 Data);
u8 TWI_ReceiveByte(void);
u8 TWI_WaitAck(void);
void TWI_SendACK(void);
void TWI_SendNACK(void);
void ads1110Config(void);
u16 RD_ADS(void);
u16 get_ad_result(void);

#endif /* __ADS1110_H */

/*
 * 以下是ADS1110.c
 *
 */

#include "ADS1110.h"

#define TWI_NOP TWI_Delay()
```

```
u16 AD_Result[13];
```

```
//-----ADS1110(ED1)地址、配置字-----//
```

```
#define ADS1110_WR_ADDRESS 0x92 // 1001 001 0 写
#define ADS1110_RD_ADDRESS 0x93 // 1001 001 1 读
#define ADS1110_CONFIG_REG 0x8C // 连续转换模式, 16bit精度, PGA=1
```

```
/*-----*/
```

```
* 函数名称: TWI_Delay
```

```
* 描 述: 延时函数
```

```
*
```

```
* 输 入: 无
```

```
* 输 出: 无
```

```
* 返 回: 无
```

```
* 作 者:
```

```
* 修改日期:
```

```
2012年10月20日
```

```
/*-----*/
```

```
void TWI_Delay(void)
```

```
{
    u32 i=15;
    while(i--);
}
```

```
/*-----*/
```

```
* 函数名称:
```

```
TWI_Initialize
```

```
* 描 述:
```

```
I2C初始化函数
```

```
*
```

```
* 输 入: 无
```

```
* 输 出: 无
```

```
* 返 回: 无
```

```
* 作 者:
```

```
* 修改日期:
```

```
2012年10月20日
```

```
/*-----*/
```

```
void TWI_Initialize(void)
```

```
{
    GPIO_InitTypeDef GPIO_InitStructure;
    // RCC_APB2PeriphClockCmd( RCC_APB2Periph_GPIOB, ENABLE);

    GPIO_InitStructure.GPIO_Speed=GPIO_Speed_50MHz;
    GPIO_InitStructure.GPIO_Mode=GPIO_Mode_Out_OD;

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6 | GPIO_Pin_7;

    TWI_SDA_1;
    TWI_SCL_1;
    GPIO_Init(GPIOB, &GPIO_InitStructure);

    ads1110Config();

    //DebugPrint("Software TWI Initializing...\n");
}
```

```
/*-----*/
```

```

* 函数名称: TWI_START
* 描    述: 发送启动
*
* 输    入: 无
* 输    出: 无
* 返    回: 无
* 作    者:
* 修改日期:
2012年10月20日
*****/

u8 TWI_START(void)
{
    TWI_SDA_1;
    TWI_NOP;

    TWI_SCL_1;
    TWI_NOP;

    if(!TWI_SDA_STATE)
    {
        ///DebugPrint("TWI_START:BUSY\n");
        return TWI_BUS_BUSY;
    }
    TWI_SDA_0;
    TWI_NOP;

    if(TWI_SDA_STATE)
    {
        ///DebugPrint("TWI_START:BUS ERROR\n");
        return TWI_BUS_ERROR;
    }

    TWI_SCL_0;
    TWI_NOP;

    return TWI_READY;
}

/*****
* 函数名称: TWI_STOP
* 描    述: 发送停止位
*
* 输    入: 无
* 输    出: 无
* 返    回: 无
* 作    者:
* 修改日期:
2012年10月20日
*****/

void TWI_STOP(void)
{
    TWI_SCL_0;
    TWI_NOP;

    TWI_SDA_0;
    TWI_NOP;

```

```

TWI_SCL_1;
TWI_NOP;

TWI_SDA_1;
TWI_NOP;

// TWI_SCL_0;
// TWI_NOP;

/////DebugPrint("TWI_STOP\n");
}

/*****
* 函数名称: TWI_SendACK
* 描    述:
收到数据,发送ACK
*
* 输    入: 无
* 输    出: 无
* 返    回: 无
* 作    者: 梅川酷子
* 修改日期:
2012年10月20日
*****/
void TWI_SendACK(void)
{
    TWI_SCL_0;
    TWI_NOP;
    TWI_SDA_0;
    TWI_NOP;
    TWI_SCL_1;
    TWI_NOP;
    TWI_SCL_0;
    TWI_NOP;
    /////DebugPrint("TWI_SendACK\n");
}

/*****
* 函数名称: TWI_SendNACK
* 描    述:
收到数据,发送NACK
*
* 输    入: 无
* 输    出: 无
* 返    回: 无
* 作    者:
* 修改日期:
2012年10月20日
*****/
void TWI_SendNACK(void)
{
    TWI_SCL_0;
    TWI_NOP;

    TWI_SCL_1;
    TWI_NOP;

```

```

    TWI_SDA_1;
    TWI_NOP;

    TWI_SCL_0;
    TWI_NOP;
    //DebugPrint("TWI_SendNACK\n");
}

u8 TWI_WaitAck(void)
{
    TWI_SCL_0;
    TWI_NOP;

    TWI_SDA_1;
    TWI_NOP;

    TWI_SCL_1;
    TWI_NOP;

    if(TWI_SDA_STATE)
    {
        TWI_SCL_0;
        return 0x00;
    }

    TWI_SCL_0;
    return 0x01;
}

/*****
* 函数名称: TWI_SendByte
* 描    述: 发送一个字节
*
* 输    入:
要发送的数据 (1字节)
* 输    出: 无
* 返    回:
TWI_ACK/TWI_NACK
* 作    者:
* 修改日期:
2012年10月20日
*****/
u8 TWI_SendByte(u8 Data)
{
    u8 i;
    TWI_SCL_0;
    for(i=0;i<8;i++)
    {
        TWI_SCL_0;
        TWI_NOP;
        //-----数据建立-----
        if(Data&0x80)
        {
            TWI_SDA_1;
        }
        else
        {

```

```

        TWI_SDA_0;
    }
    Data<<=1;
    TWI_NOP;
    //---数据建立保持一定延时----

    TWI_SCL_1;
    TWI_NOP;
}
TWI_SCL_0;

if(TWI_SDA_STATE)
{
//        TWI_SCL_0;
    //DebugPrint("TWI_NACK!\n");
    return TWI_NACK;
}
else
{
//    TWI_SCL_0;
    //DebugPrint("TWI_ACK!\n");
    return TWI_ACK;
}
}

/*****
* 函数名称:
TWI_ReceiveByte
* 描    述: 接收一个字节
*
* 输    入: 无
* 输    出: 无
* 返    回:
返回接收到的数据
* 作    者:
* 修改日期:
2012年10月20日
*****/
u8 TWI_ReceiveByte(void)
{
    u8 i,Dat;

    TWI_SDA_1;
    //TWI_SCL_0;
    Dat=0;

    for(i=0;i<8;i++)
    {
        //TWI_SCL_1;//产生时钟上升沿[正脉冲],让从机准备好数据
        //TWI_NOP;
        Dat<<=1;
        TWI_SCL_0;
        TWI_NOP;
        TWI_SCL_1;
        TWI_NOP;
        if(TWI_SDA_STATE) //读引脚状态
        {

```

```

        Dat |= 0x01;
    }

    TWI_SCL_0; //准备好再次接收数据
    //TWI_NOP; //等待数据准备好
}
/////DebugPrint("TWI_Dat:%x\n",Dat);
return Dat;
}

/*****
* 函数名称:
ads1110Config
* 描    述:
对ADS1110进行配置
*
* 输    入: 无
* 输    出: 无
* 返    回: 无
* 作    者:
* 修改日期:
2012年10月20日
*****/
void ads1110Config(void)
{
    TWI_START();
    TWI_SendByte(ADS1110_WR_ADDRESS);
    TWI_SendNACK();
    TWI_SendByte(ADS1110_CONFIG_REG);
    TWI_SendNACK();
    TWI_STOP();
}

/*****
* 函数名称: RD_ADS
* 描    述:
读取ADS1100数据子程序
*
* 输    入: 无
* 输    出: 无
* 返    回: 转换后的数据
* 作    者:
* 修改日期:
2012年10月20日
*****/
u16 RD_ADS(void)
{
    // u8 temp;
    u16  W_B1byte_high, W_B1byte_low, W_B1_word;

    TWI_START();

    TWI_SendByte(ADS1110_RD_ADDRESS);

    TWI_SendNACK();
//

```

```

//  if(!temp)
//  {
    W_Blbyte_high /*= TMR_H*/ = TWI_ReceiveByte();

    TWI_SendACK();

    W_Blbyte_low /*= TMR_L*/ = TWI_ReceiveByte();
    TWI_SendACK();

    W_Bl_word = TWI_ReceiveByte();

    TWI_STOP();

    W_Bl_word = (W_Blbyte_high << 8)+ W_Blbyte_low;

//  if (W_Bl_word > 0x7fff)
//      W_Bl_word = 0;

    return W_Bl_word;
//  }
//  else
//      return 0x0000;
}

/*****
* 函数名称:
get_ad_result
* 描 述:
16位AD转换, 连续取值13次, 最后采用中间值
*
* 输 入: 无
* 输 出: 无
* 返 回:
中值滤波后的取值
* 作 者:
* 修改日期:
2012年10月20日
*****/
u16 get_ad_result(void)
{
    u8 i,j;
    u16 temp;

    for(i = 0; i < 13; i++) // 连续取值13次
    {
        AD_Result[i] = RD_ADS();
        Delay_us(10);
    }

    for(i = 1; i < 13; i++) // 插入法对取的13个值进行排序
    {
        temp = AD_Result[i]; // store the original sorted array
        in temp
        for(j=i ; j>0 && temp < AD_Result[j-1] ; j--) // compare the new array with temp

        {

```



```
        AD_Result[j] = AD_Result[j-1];           // all larger elements are moved one
        pot to the right
    }
    AD_Result[j] = temp;
}

return AD_Result[7];
}
```