/\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* @file USART/Printf/main.c

\* @author MCD Application Team

\* @version V3.3.0

\* @date 04/16/2010

\* @brief Main program body

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* @copy

\*

\* THE PRESENT FIRMWARE WHICH IS FOR GUIDANCE ONLY AIMS AT PROVIDING CUSTOMERS

\* WITH CODING INFORMATION REGARDING THEIR PRODUCTS IN ORDER FOR THEM TO SAVE

\* TIME. AS A RESULT, STMICROELECTRONICS SHALL NOT BE HELD LIABLE FOR ANY

\* DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARISING

\* FROM THE CONTENT OF SUCH FIRMWARE AND/OR THE USE MADE BY CUSTOMERS OF THE

\* CODING INFORMATION CONTAINED HEREIN IN CONNECTION WITH THEIR PRODUCTS.

\*

\* <h2><center>&copy; COPYRIGHT 2010 STMicroelectronics</center></h2>

\*/

/\* Includes ------------------------------------------------------------------\*/

#include "stm32f10x.h"

#include "stm32\_eval.h"

#include "flash.h"

#include "delay.h"

#include "ADS1256.h"

#include <stdio.h>

#include "ads1256\_exti.h"

/\*\* @addtogroup STM32F10x\_StdPeriph\_Examples

\* @{

\*/

/\*\* @addtogroup USART\_Printf

\* @{

\*/

/\* Private typedef -----------------------------------------------------------\*/

/\* Private define ------------------------------------------------------------\*/

/\* Private macro -------------------------------------------------------------\*/

/\* Private variables ---------------------------------------------------------\*/

/\* Values magic to the Board keys \*/

#define NOKEY 0

#define KEY1 1

#define KEY2 2

#define KEY3 3

#define KEY4 4

unsigned char data[16];

extern void ads1256\_init(void);

// long ulResult; //调试用

// long double ldVolutage;

/\* Private function prototypes -----------------------------------------------\*/

#ifdef \_\_GNUC\_\_

/\* With GCC/RAISONANCE, small printf (option LD Linker->Libraries->Small printf

set to 'Yes') calls \_\_io\_putchar() \*/

#define PUTCHAR\_PROTOTYPE int \_\_io\_putchar(int ch)

#else

#define PUTCHAR\_PROTOTYPE int fputc(int ch, FILE \*f)

#endif /\* \_\_GNUC\_\_ \*/

/\* Private functions ---------------------------------------------------------\*/

void Init\_UART2()

{

USART\_InitTypeDef USART\_InitStructure;

/\* USARTx configured as follow:

- BaudRate = 115200 baud

- Word Length = 8 Bits

- One Stop Bit

- No parity

- Hardware flow control disabled (RTS and CTS signals)

- Receive and transmit enabled

\*/

USART\_InitStructure.USART\_BaudRate = 9600;//波特率

USART\_InitStructure.USART\_WordLength = USART\_WordLength\_8b;//不需要校验，字长选择8位

USART\_InitStructure.USART\_StopBits = USART\_StopBits\_1;//停止位

USART\_InitStructure.USART\_Parity = USART\_Parity\_No;//校验位

USART\_InitStructure.USART\_HardwareFlowControl = USART\_HardwareFlowControl\_None;//硬件流控制，不使用

USART\_InitStructure.USART\_Mode = USART\_Mode\_Rx | USART\_Mode\_Tx;//串口模式：即发送又接收

STM\_EVAL\_COMInit(COM1, &USART\_InitStructure);//完成串口COM1时钟配置、GPIO配置，根据上述参数初始化并使能

}

/\*\*

\* @brief Main program

\* @param None

\* @retval None

\*/

int main(void)

{

unsigned char i=0;

long ulResult;

long double ldVolutage;//多精度浮点类型

ads1256\_init();

Init\_UART2();

Init\_ADS1256\_GPIO(); //初始化ADS1256 GPIO管脚

Delay(0x1ffFF);

GPIO\_SetBits(GPIOB, GPIO\_Pin\_11 );

ADS1256\_Init();

Reset\_Trans\_State ();

while(1)

{

// ulResult = ADS\_sum( ADS1256\_MUXP\_AIN0 | ADS1256\_MUXN\_AINCOM);

if(Get\_Trans\_State() == SET ) //读取数据传输状态，判断数据是否从ADS1256传输至单片机，传输完成则进行数据转换和打印

{

ulResult = Get\_ulResult\_exti();

if( ulResult & 0x800000 ) //判断采集数据是否为负数，如果是负数则进行补码转换，并把负数的原码赋值给ulResult

{ //正数补码与原码相同不需要进行补码转换

ulResult = ~(unsigned long)ulResult;

ulResult &= 0x7fffff;

ulResult += 1;

ulResult = -ulResult;

}

ldVolutage = (long double)ulResult\*0.59604644775390625; //采集电压值，单位uV采集数据乘以5/2的23次方，采集数据转电压

ldVolutage = ldVolutage \* 0.000001; //采集的电压值，单位为V

ldVolutage = ldVolutage \*(1000/5); //（x/5）x为5V电压对应测量值即最大测量值 浊度1000 温度100，电压转测量值，后面可以加标定方程

printf("%.1lf",ldVolutage); //double

printf("\n");

Reset\_Trans\_State (); //传输完成标志位复位

GPIO\_ResetBits(GPIOB, GPIO\_Pin\_12); //片选引脚复位，进行新一轮的电压采集

}

}

}

/\*\*

\* @brief Retargets the C library printf function to the USART.

\* @param None

\* @retval None

\*/

PUTCHAR\_PROTOTYPE//发送一个字符协议，借助此函数测试串口是否正常

{

/\* Place your implementation of fputc here \*/

/\* e.g. write a character to the USART \*/

USART\_SendData(EVAL\_COM1, (uint8\_t) ch);//发送函数，这是stm32库函数

/\* Loop until the end of transmission \*/ //循环直到发送完成

while (USART\_GetFlagStatus(EVAL\_COM1, USART\_FLAG\_TC) == RESET)//如果发送未完成，标志位未置位，则循环等待

{}

return ch;

}

#ifdef USE\_FULL\_ASSERT

/\*\*

\* @brief Reports the name of the source file and the source line number

\* where the assert\_param error has occurred.

\* @param file: pointer to the source file name

\* @param line: assert\_param error line source number

\* @retval None

\*/

void assert\_failed(uint8\_t\* file, uint32\_t line)

{

/\* User can add his own implementation to report the file name and line number,

ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) \*/

/\* Infinite loop \*/

while (1)

{

}

}

#endif

/\*\*

\* @}

\*/

/\*\*

\* @}

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* (C) COPYRIGHT 2010 STMicroelectronics \*\*\*\*\*END OF FILE\*\*\*\*/