

MQ2 AO 口模拟电压采样

1. 实验目的

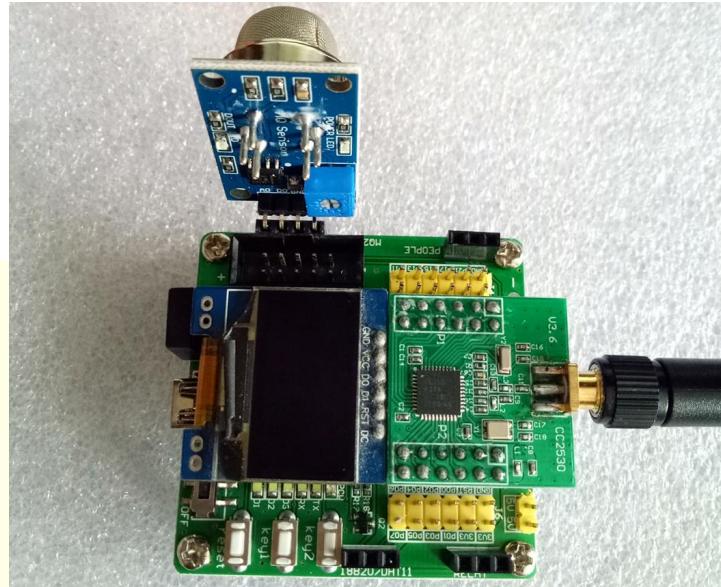
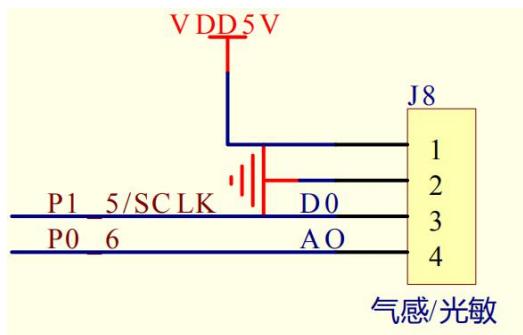
- 1) 了解配置 IO 口
- 2) 了解 ADC 办理
- 3) 了解 ADC 计算电压的方法

2. 实验设备

硬件: PC 机	一台
EB2530 (底板、核心板、仿真器、USB 线)	一套
MQ2	一个
软件: 2000/XP/win7 系统, IAR 8.10 集成开发环境	

3. 接线方法

在这里我们采集的是 MQ2 的浓度对应的电压值, 传感器的 AO 是直接接到 CPU 管脚 P0_6 上, 所以我们只要采集 P06 上的电压即可。



接线原则:

- 1、VCC:接电源正极 (5V)
- 2、GND:接电源负极
- 3、D0: (本实验中没有用)
- 4、AO: 模拟信号输出, 本实验间接接在 P06 上。

自己购买的模块请仔细核对一下引脚, 确保连接正确

MQ-2 传感器简介:

MQ-2 气体传感器所使用的气敏材料是在清洁空气中电导率较低的二氧化锡(SnO₂)。当传感器所处环境中存在可燃气体时, 传感器的电导率随空气中可燃气体浓度的增加而增大。使用简单的电路即可将电导率的变化转换为与该气体浓度相对应的输出信号。MQ-2 气体传感器对液化气、丙烷、氢气的灵敏度高,

深圳市亿研电子有限公司

技术交流 QQ:89339784

咨询客服获取最新 QQ 群

网址: <http://sz-yv.taobao.com/>

售后电话:13798292327

对天然气和其它可燃蒸汽的检测也很理想。这种传感器可检测多种可燃性气体，是一款适合多种应用的低成本传感器。

4. 使用到的寄存器

APCFG (0xF2) - 模拟外设 I/O 配置

位	名称	复位	R/W	描述
7:0	APCFG[7:0]	0x00	R/W	模拟外设I/O配置。APCFG[7:0]选择P0.7 - P0.0作为模拟I/O 0: 模拟I/O禁用 1: 模拟I/O使能

ADCCON3 (0xB6) - ADC 控制 3

位	名称	复位	R/W	描述
7:6	EREF[1:0]	00	R/W	选择用于额外转换的参考电压 00: 内部参考电压 01: AIN7 引脚上的外部参考电压 10: AVDD5 引脚 11: 在 AIN6-AIN7 差分输入的外部参考电压
5:4	EDIV[1:0]	00	R/W	设置用于额外转换的抽取率。抽取率也决定了完成转换需要的时间和分辨率。 00: 64 抽取率(7 位 ENOB) 01: 128 抽取率(9 位 ENOB) 10: 256 抽取率(10 位 ENOB) 11: 512 抽取率(12 位 ENOB)
3:0	ECH[3:0]	0000	R/W	单个通道选择。选择写 ADCCON3 触发的单个转换所在的通道号码。 当单个转换完成，该位自动清除。 0000: AIN0 0001: AIN1 0010: AIN2 0011: AIN3 0100: AIN4 0101: AIN5 0110: AIN6 0111: AIN7 1000: AIN0-AIN1 1001: AIN2-AIN3 1010: AIN4-AIN5 1011: AIN6-AIN7 1100: GND 1101: 正电压参考 1110: 温度传感器 1111: VDD/3

ADCCON1 (0xB4) – ADC 控制 1

位	名称	复位	R/W	描述
7	EOC	0	R/ H0	转换结束。当 ADCH 被读取的时候清除。如果已读取前一数据之前，完成一个新的转换，EOC 位仍然为高。 0： 转换没有完成 1： 转换完成
6	ST	0		开始转换。读为1，直到转换完成 0： 没有转换正在进行 1： 如果 ADCCON1.STSEL = 11 并且没有序列正在运行就启动一个转换序列。
5:4	STSEL[1:0]	11	R/W1	启动选择。选择该事件，将启动一个新的转换序列。 00： P2.0引脚的外部触发。 01： 全速。不等待触发器 10： 定时器1通道0比较事件 11： ADCCON1.ST = 1
3:2	RCTRL[1:0]	00	R/W	控制 16 位随机数发生器（第 13 章）。当写 01 时，当操作完成时设置将自动返回到 00。 00： 正常运行。(13X 型展开) 01： LFSR 的时钟一次(没有展开). 10： 保留 11： 停止。关闭随机数发生器
1:0	-	11	R/W	保留。一直设为 11 。

ADCL (0xBA) – ADC 数据低位

位	名称	复位	R/W	描述
7:2	ADC[5:0]	0000 00	R	ADC转换结果的低位部分
1:0	-	00	R0	没有使用。读出来一直是 0

ADCH (0xBB) – ADC 数据高位

位	名称	复位	R/W	描述
7:0	ADC[13:6]	0x00	R	ADC转换结果的高位部分。

5. 代码分析

```
#include <ioCC2530.h>
#include <string.h>
#include "LCD.h"
#include "adc.h"
```

```
#define uint16 unsigned int
#define uint8 unsigned char
```

```
void InitUart0(void)
{
```

深圳市亿研电子有限公司
技术交流 QQ:89339784

网址：<http://sz-yv.taobao.com/>
咨询客服获取最新 QQ 群 售后电话:13798292327

```

PERCFG = 0x00;           //外设控制寄存器 USART 0 的 IO 位置:0 为 P0 口位置 1
P0SEL |= 0x0c;           //P0_2, P0_3 用作串口 (外设功能)
P2DIR &= ~0xC0;          //P0 优先作为 UART0

U0CSR |= 0x80;           //设置为 UART 方式
U0GCR |= 11;              //波特率设为 115200
U0BAUD |= 216;            //UART0 TX 中断标志初始置位 0
UTX0IF = 0;                //允许接收
U0CSR |= 0x40;            //开总中断允许接收中断
IEN0 |= 0x84;

}

void Uart0SendString(char *Data, int len)
{
    uint16 i;
    for(i=0; i<len; i++)
    {
        U0DBUF = *Data++;
        while(UTX0IF == 0);
        UTX0IF = 0;
    }
}

void InitClockTo32M(void)
{
    CLKCONCMD &= ~0x40;           //设置系统时钟源为 32MHZ 晶振
    while(CLKCONSTA & 0x40);      //等待晶振稳定
    CLKCONCMD &= ~0x47;           //设置系统主时钟频率为 32MHZ
}

//ADC 采样主要是这个函数
//channel: 采样的通道,
//resolution: 分辨率,
uint16 readV(uint8 channel,uint8 resolution)
{
    uint16 value ;

    APCFG |= 1 << channel ; //注意这里是设置 ADC 输入通道!!
    ADC_ENABLE_CHANNEL(channel); //使能 ADC 的采样通道

    ADC_SINGLE_CONVERSION(ADC_REF_AVDD | resolution | channel); //片上 3.3V 参考电压, 12 位, 6
    通道
    ADC_SAMPLE_SINGLE(); //开始转换

    //等待转换完成

```

```

while (0==(ADCCON1 & 0x80));

// ADCCFG &= (0x40 ^ 0xFF);

value = ADCL ;
value |= ((uint16) ADCH) << 8 ;//这里注意一下

if(resolution == ADC_7_BIT)//7 位分辨率
{
    value >= 9 ;
}
else if(resolution == ADC_9_BIT) //9 位分辨率
{
    value >= 7 ;
}
else if(resolution == ADC_10_BIT) //10 位分辨率
{
    value >= 6 ;
}
else if(resolution == ADC_12_BIT) //12 位分辨率
{
    value >= 4;
}

return value;
}

void main( void )
{
    uint16 adc=0;//adc 采样值
    uint16 vol=0; //adc 采样电压
    uint8 buff[10]=0; //adc 采样字符串

    //初始化
    InitClockTo32M();
    InitUart0();
    LCD_Init();
    LCD_P6x8Str(0, 2, "MQ2 AO VOLTAGE");

    while(1)
    {
        adc = readV(ADC_AIN6,ADC_12_BIT); //通道 6, 12 位分辨率
        //12 位的分辨率最大为 2048
    }
}

```

```

ifadc>2048) continue;

//mq2 Ao 口输出电压
vol=(uint16)((float)adc/2048.0)*3300;

//变成可视的字符输出
sprintf(buff, "v:%04d mv, adc:%04d", vol,adc);

//串口输出
Uart0SendString(buff, strlen(buff));
Uart0SendString("\r\n", 2);

//lcd 显示
LCD_P6x8Str(0, 4, buff);

DelayMS(2000);
}
}

```

6. 程序步骤及效果

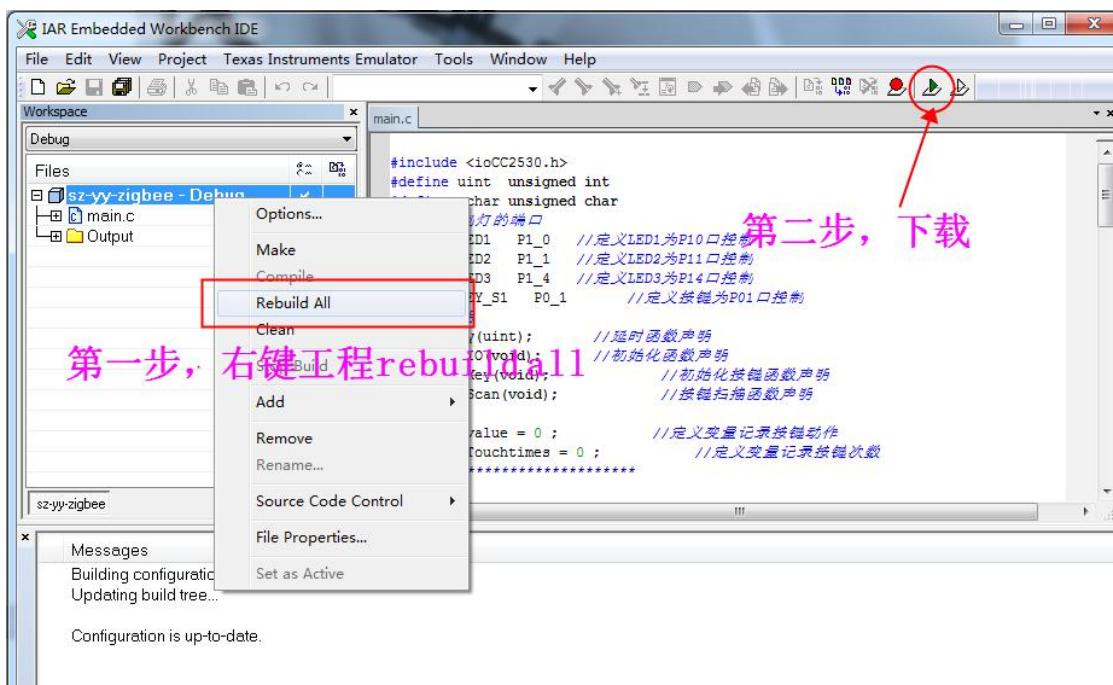
1)、双击打开代码工程。



2)通过仿真器把开发板接到电脑上。



3) 编译下载。



程序已下载完成，可以运行：

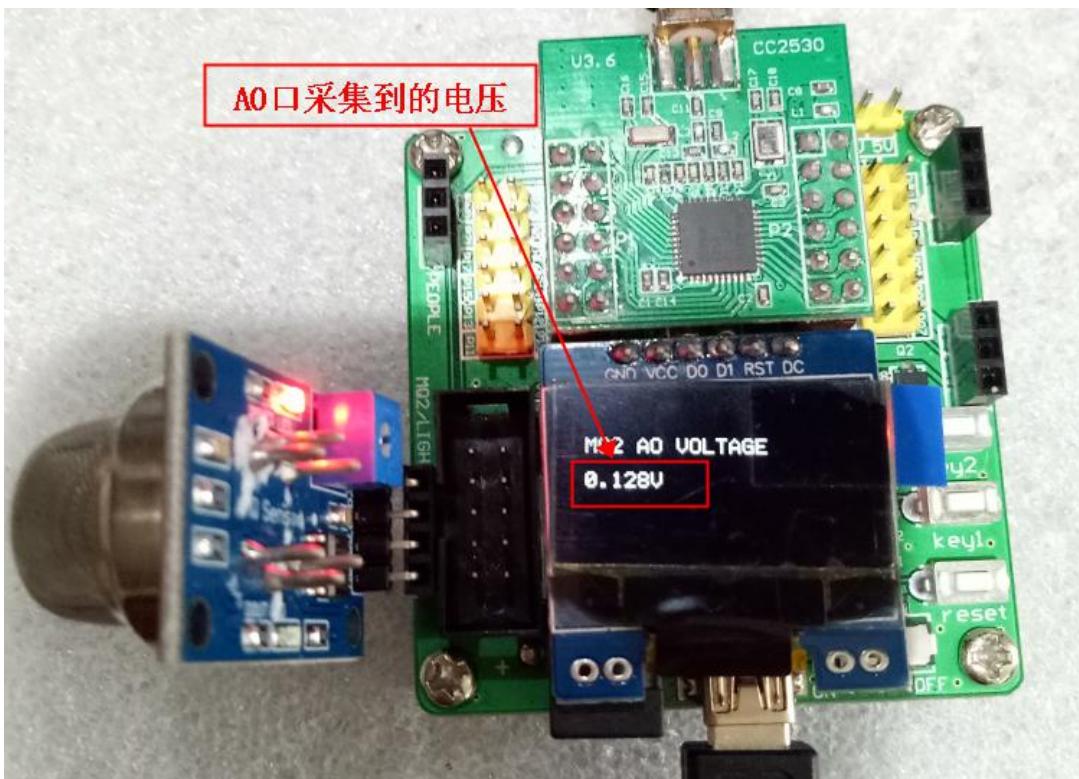
```

File Edit View Project Debug Texas Instruments Emulator Tools Window Help
Workspace x
Debug
Files sz-yy-zigbee -> Debug
main.c
if(KEY_S1 == 0) //低电平表示按下
{
    Delay(500);
    if(KEY_S1 == 0) //还是低电平
    {
        while(KEY_S1==0); //直到松开按钮
        return(1);
    }
}
return(0);
}
***** 主函数 *****
void main(void)
{
    InitIO(); // 初始化LED灯控制IO口
    InitKey(); // 初始化按键控制IO口
    while(1)
    {
        Keyvalue = KeyScan(); //读取按钮动作

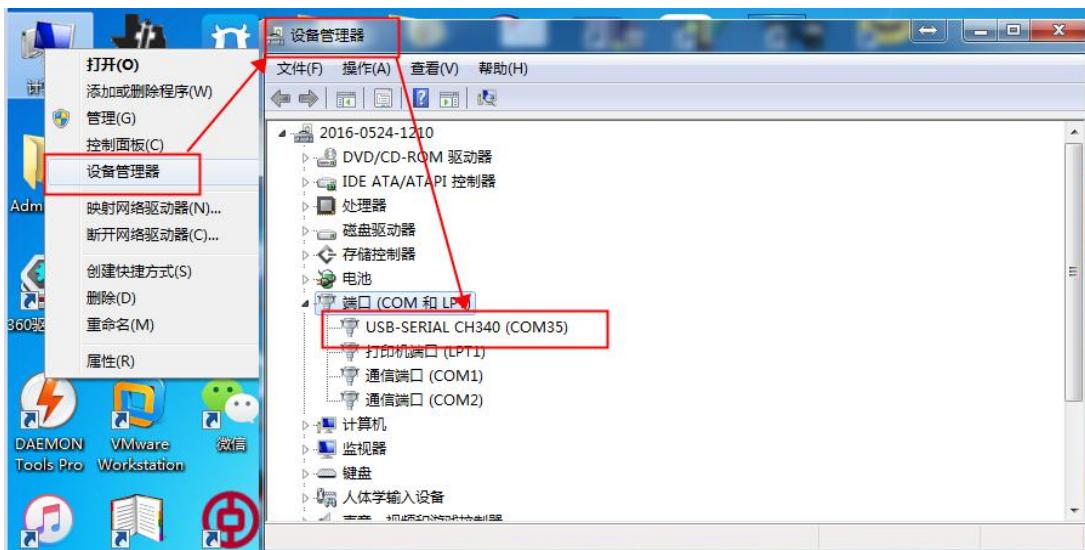
        if(Keyvalue == 1) //按下按钮S1, 弹起时流水方向: LED3->LED2->LED1(倒序流水闪烁
        {
            LED3 = !LED3;
            Delay(2000);
            LED2 = !LED2;
        }
    }
}

```

4)、按上面提到的接线方法，把传感器插上，同时把 LCD 插上。USB 线直接接到电脑上，**开关处于开的位置**。



5)、查看生成的串口号。查看方法“我的电脑”->“设备管理器”，如图：



打开生面的串口 COM35:，可以看到有以下数据输出：

