实验五 AD 转换及 PWM 控制

实验目的:

(1) 掌握 SPI 总线的使用方式
(2) 掌握 xpt2046 AD 转换芯片的工作原理
(3) 掌握 SPI 总线方式实现基于 xpt2046 的 AD 转换
(4) 掌握 PWM 控制功率的方式

实验内容:

学习 xpt2046 AD 转换芯片的工作原理，利用 SPI 总线实现基于该芯片的 AD 转换，调节滑动变阻器将 AD 转换的结果显示在数码管上，同时利用 PWM 控制方式实现 LED 灯的亮度联动，当 AD 结果增大时，亮度增加，反之，亮度减小。

参考资料：芯片手册文档

实验步骤:

(1) 编写 SPI 总线通信程序和数码管显示程序
(2) 编写 xpt2046 控制程序，实现 AD 转换。
(3) 编写 PWM 控制，实现 LED 灯亮度联动。

实验要求:

编写实验报告，主要包括关键步骤的实现和效果截屏，并分析实验过程中出现的问题和分析解决方法。

代码

```
/*
（1）编写 SPI 总线通信程序和数码管显示程序
（2）编写 xpt2046 控制程序，实现 AD 转换。
（3）编写 PWM 控制，实现 LED 灯亮度联动。
*/

#include "reg52.h"
#include "XPT2046.h"



uchar DisplayData[8];
sbit LSA=P2^2;                  //LSA、LSB、LSC 共同控制显示数码管
sbit LSB=P2^3;
sbit LSC=P2^4;

sbit PWM=P2^0;//定义使用的 IO 口

uchar step=50;      //总共级别数
int result=0; //获取的电阻的值
```

```
int timer=0; //中断次数计数器变量

uchar code smgduan[10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};
//0-9

void delay(int i)          //延时
{
        while(i--);
}

/*******************************************************************************
*函数名：initTimer()
*输  入：无
*输  出：
*功  能：初始化定时器
*******************************************************************************/
void initTimer()
{
        TMOD=0X01;//晶振 11.0592，定时器定时方式 1
        TH0=0XFF;  //250ms
        TL0=0X06;
        EA=1;//开总中断
        ET0=1;//开定时器中断
        TR0=1;//开定时器
}

/*******************************************************************************
*函数名：timer0() interrupt 1 using 3
*输  入：无
*输  出：
*功  能：定时器中断函数
*******************************************************************************/
void timer0() interrupt 1 using 3
{
        int grade=result/80;  //共分为 4000/80=50 个级别，用 grade 表示当前级别
        TH0=0xFF; //恢复定时器初始值
        TL0=0x06;
        timer++;   //定时器自加
        if(timer>grade)
        {
                PWM=1;   //灯灭（查看指导手册，高电平灭）
        }
        else
                PWM=0;
```

```
        if(timer==step)  //当 timer 达到级别数后，timer 重置为 0
        {
                timer=0;
        }
}

/*****************************************************************************
*函数名：datapros(int result)
*输  入：无
*输  出：
*功  能：将滑动变阻器的结果转换到数码管上
*****************************************************************************/
void datapros(int result)
{

        DisplayData[0] = smgduan[result / 1000 % 10];
        DisplayData[1] = smgduan[result / 100 % 10];
        DisplayData[2] = smgduan[result / 10 % 10];
        DisplayData[3] = smgduan[result % 10];
}

/*****************************************************************************
*函数名：DigDisplay()
*输  入：无
*输  出：
*功  能：显示数码管上内容
*****************************************************************************/
void DigDisplay()
{
        uchar i;
        for(i=0;i<4;i++)
        {
                switch(i)          //位选，选择点亮的数码管，
                {
                        case(0):
                                LSA=0;LSB=0;LSC=0; break;//显示第 0 位
                        case(1):
                                LSA=1;LSB=0;LSC=0; break;//显示第 1 位
                        case(2):
                                LSA=0;LSB=1;LSC=0; break;//显示第 2 位
                        case(3):
                                LSA=1;LSB=1;LSC=0; break;//显示第 3 位

                }
```

```
                P0=DisplayData[3-i];//发送数据
                delay(100); //间隔一段时间扫描
                P0=0x00;//消隐
        }
}

/*********************************************************************************
*函数名：main()
*输  入：无
*输  出：
*功
```

```
#include"XPT2046.h"

/*********************************************************************************
*函数名：SPI_Write
*输  入：dat：写入数据
*输  出：无
*功  能：使用SPI写入数据
*********************************************************************************/

void SPI_Write(uchar dat)
{
        uchar i;
        CLK = 0;
        for(i=0; i<8; i++)
        {
                DIN = dat >> 7;  //放置最高位
                dat <<= 1;
                CLK = 0;                  //上升沿放置数据

                CLK = 1;

        }
}
```

```
/*******************************************************************************
*函数名：SPI_Read
*输  入：无
*输  出：dat：读取 到的数据
*功  能：使用 SPI 读取数据
*******************************************************************************/

uint SPI_Read(void)
{
        uint i, dat=0;
        CLK = 0;
        for(i=0; i<12; i++)                 //接收 12 位数据
        {
                dat <<= 1;

                CLK = 1;
                CLK = 0;

                dat |= DOUT;

        }
        return dat;
}

/*******************************************************************************
*函数名：Read_AD_Data
*输  入：cmd：读取的 X 或者 Y
*输  出：endValue：最终信号处理后返回的值
*功  能：读取触摸数据
*******************************************************************************/
uint Read_AD_Data(uchar cmd)
{
        uchar i;
        uint AD_Value;
        CLK = 0;
```

```
        CS  = 0;
        SPI_Write(cmd);
        for(i=6; i>0; i--);        //延时等待转换结果
        CLK = 1;   //发送一个时钟周期，清除 BUSY
        _nop_();
        _nop_();
        CLK = 0;
        _nop_();
        _nop_();
```

```
#ifndef   __XPT2046_H_
#define   __XPT2046_H_

//---包含头文件---//
#include<reg52.h>
#include<intrins.h>

//---重定义关键词---//
#ifndef uchar
#define uchar unsigned char
#endif

#ifndef uint
#define uint  unsigned int
#endif

#ifndef ulong
#define ulong  unsigned long
#endif

//---定义使用的 IO 口---//
sbit DOUT = P3^7;  //输出
sbit CLK  = P3^6;  //时钟
sbit DIN  = P3^4;  //输入
sbit CS   = P3^5;  //片选

uint Read_AD_Data(uchar cmd);
uint SPI_Read(void);
void SPI_Write(uchar dat);
```

演示