

《单片机与接口技术》课程实验指导书

编写教师：张永安

内蒙古农业大学
计算机与信息工程学院
计算机科学系

二零一八年七月

目 录

实验一	点亮第一个 LED 实验	1
实验二	LED 流水灯实验	3
实验三	独立按键实验	6
实验四	定时器实验	9
实验五	8*8LED 点阵实验	11
实验六	矩阵按键实验	15
实验七	串行通信实验	19
附录：	烧写软件使用说明	21

实验一 点亮第一个 LED 实验

一、实验目的

- (1) 熟悉 C51 编程环境;
- (2) 熟悉实验箱结构, 硬件连线;
- (3) 会查看实验箱硬件连线图;
- (4) 利用 C51 编程, 控制单片机输出引脚点亮 LED;

二、实验电路

如图 1 所示, 根据实验箱 LED 和单片机引脚接口图, 选择一个 LED, 利用程序控制其点亮。

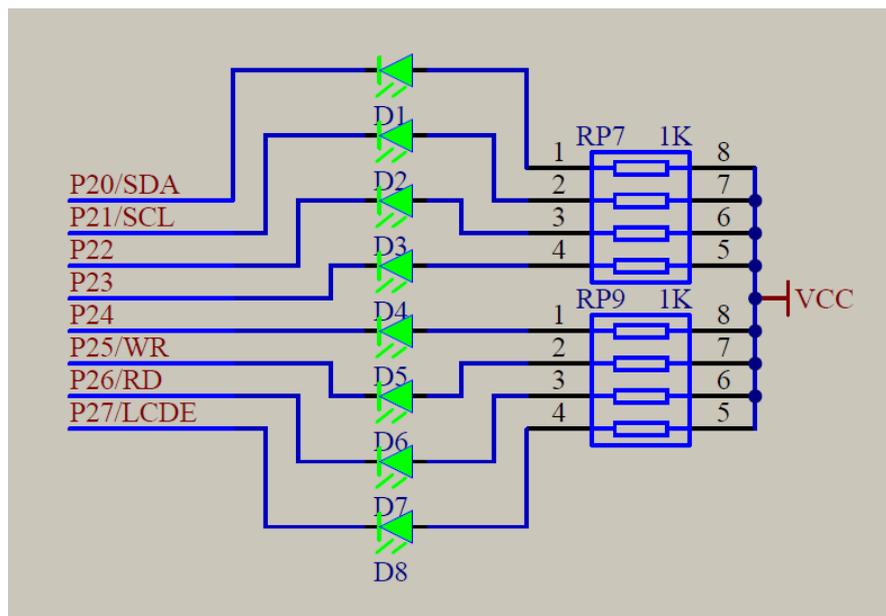


图 1: LED 接口图

三、实验内容

编写程序, 控制 LED 指示灯 D1~D8 中的一个点亮。

四、参考程序

```
/*
*****
*           点亮第一个 LED 实验           实现现象：下
载程序后 D1~D8 中的一个指示灯点亮
*****/

#include "reg52.h" //此文件中定义了单片机的一些特殊功能寄存器
sbit led=P2^0;    //将单片机的 P2.0 端口定义为 led, 根据自己确定的 LED 定义该接口
/*
*****
* 函数名      : main
* 函数功能    : 主函数
*****/

void main()
{
    while(1)
    {
        led=0; //P2.0 端口设置为低电平
    }
}
```

五、实验思考题

- 1、如果要控制多个 LED 同时点亮，程序应如何改动？
- 2、试着实现 LED 的点亮与熄灭？

实验二 LED 流水灯实验

一、实验目的

- (1) 学会使用延时函数及其延时时间的计算;
- (2) 掌握LED灯循环点亮的控制方法;

二、实验电路

如图 1 所示，根据实验箱 LED 和单片机引脚接口图，实现 LED 流水灯循环点亮效果。

三、实验内容

利用 C51 编程实现 LED 流水灯循环点亮效果。

四、参考程序

```
/*  
*****  
*  
*          LED 流水灯实验  
*实现现象：下载程序后 LED 呈现流水灯效果进行左移右移滚动  
*****/  
#include "reg52.h"          //此文件中定义了单片机的一些特殊功能寄存器  
#include<intrins.h>        //因为要用到左右移函数，所以加入这个头文件  
typedef unsigned int u16;    //对数据类型进行声明定义  
typedef unsigned char u8;  
#define led P2              //将 P2 口定义为 led 后面就可以使用 led 代替 P2 口  
  
/*  
*****  
* 函 数 名          : delay  
* 函数功能          : 延时函数，i=1 时，大约延时 10us  
*****  
*/
```

```

*****/
void delay(u16 i)
{
    while(i--);
}

/*****
* 函数名      : main
* 函数功能    : 主函数
*****/
void main()
{
    u8 i;
    led=0xfe;
    delay(50000); //大约延时 450ms
    while(1)
    {
/*      for(i=0;i<8;i++)
        {
            P2=~(0x01<<i);    //将 1 右移 i 位，然后将结果取反赋值到 P2 口
            delay(50000); //大约延时 450ms
        }
*/
        for(i=0;i<7;i++) //将 led 左移一位
        {

```

```
        led=_crol_(led,1);
        delay(50000); //大约延时 450ms
    }
    for(i=0;i<7;i++) //将 led 右移一位
    {
        led=_cror_(led,1);
        delay(50000); //大约延时 450ms
    }
}
}
```

五、实验思考题

- 1、尝试不用_crol_()函数和_crор_()函数实现循环移位?
- 2、试着改变发光管的移动速度?

实验三 独立按键实验

一、实验目的

- (1) 学习单片机输入输出的控制方法;
- (2) 掌握按键和LED的控制;
- (3) 掌握键盘的延时去抖概念和编程实现;

二、实验电路

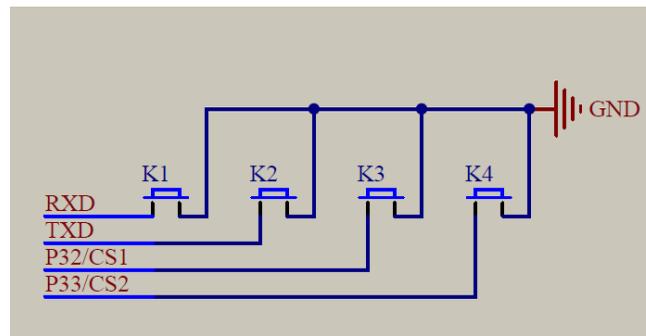


图 2: 独立按键接口电路

结合图 2 的独立按键接口和图 1 的 LED 接口, 实现利用一个按键控制一个 LED 灯点亮和熄灭的方法。

三、实验内容

- (1) 查看实验箱硬件连线图, 确定 LED、按键和单片机引脚的连接;
- (2) 编写 c51 程序, 实现利用按键控制相应 LED 灯的亮灭;

四、参考程序

```
/*****
```

```
*                独立按键实验                *
```

```
实现现象: 下载程序后按下 K1 按键可以对 D1 小灯状态取反。
```

```

*****/
#include "reg52.h"      //此文件中定义了单片机的一些特殊功能寄存器
typedef unsigned int u16;    //对数据类型进行声明定义
typedef unsigned char u8;
sbit k1=P3^1;  //定义 P31 口是 k1
sbit led=P2^0;    //定义 P20 口是 led
/*****
* 函数名      : delay
* 函数功能    : 延时函数, i=1 时, 大约延时 10us
*****/
void delay(u16 i)
{
    while(i--);
}
/*****
* 函数名      : keypros
* 函数功能    : 按键处理函数, 判断按键 K1 是否按下
*****/
void keypros()
{
    if(k1==0)      //检测按键 K1 是否按下
    {
        delay(1000);    //消除抖动 一般大约 10ms
        if(k1==0)    //再次判断按键是否按下
        {

```

```

        led=~led;    //led 状态取反
    }
    while(!k1);    //检测按键是否松开
}
}
/*****
* 函数名      : main
* 函数功能    : 主函数
*****/
void main()
{
    while(1)
    {
        keypros(); //按键处理函数
    }
}

```

五、实验思考题

如果将去抖动及等待按键释放的功能取消，程序的运行情况将会如何？

实验四 定时器实验

一、实验目的

- (1) 学习定时器的使用;
- (2) 掌握定时初值的计算、定时器控制寄存器的设置;
- (3) 掌握定时中断的程序设计方法;

二、实验电路

LED 控制电路参照图 1。

三、实验内容

- (1) 查看实验箱硬件连线图，确定所控制的 LED 和单片机引脚的连接;
- (2) 计算定时初值;
- (3) 编程实现定时器 0 的 1ms 定时，使 LED 每隔 1 秒闪烁一次;

四、参考程序

```
/*  
*  
* 定时器 0 实验  
*实现现象：下载程序后 D1 小灯循环点亮 1 秒，熄灭 1 秒。使用单片机内部定时器可以实现  
准确延时。  
*/  
  
#include "reg52.h" //此文件中定义了单片机的一些特殊功能寄存器  
  
typedef unsigned int u16; //对数据类型进行声明定义  
typedef unsigned char u8;  
  
sbit led=P2^0; //定义 P20 口是 led  
  
/*  
* 函数名 : Timer0Init  
* 函数功能 : 定时器 0 初始化  
*/  
void Timer0Init()  
{  
TMOD|=0X01;//选择为定时器 0 模式，工作方式 1，仅用 TR0 打开启动。
```

```

    TH0=0XFC; //给定时器赋初值, 定时 1ms
    TL0=0X18;
    ET0=1;//打开定时器 0 中断允许
    EA=1;//打开总中断
    TR0=1;//打开定时器
}

/*****
* 函数名      : main
* 函数功能    : 主函数
*****/
void main()
{
    Timer0Init(); //定时器 0 初始化
    while(1);
}

/*****
* 函数名      : void Timer0() interrupt 1
* 函数功能    : 定时器 0 中断函数
*****/
void Timer0() interrupt 1
{
    static unsigned int i;
    TH0=0XFC; //给定时器赋初值, 定时 1ms
    TL0=0X18;
    i++;
    if(i==1000)
    {
        i=0;
        led=~led;
    }
}

```

五、实验思考题

该实验如果用定时器方式 1 来实现, 程序应作如何修改?

实验五 8*8LED 点阵实验

一、实验目的

- (1) 了解 74HC595 芯片的原理和接口；
- (2) 掌握 8*8LED 显示数字的原理和编程方法；

二、实验电路

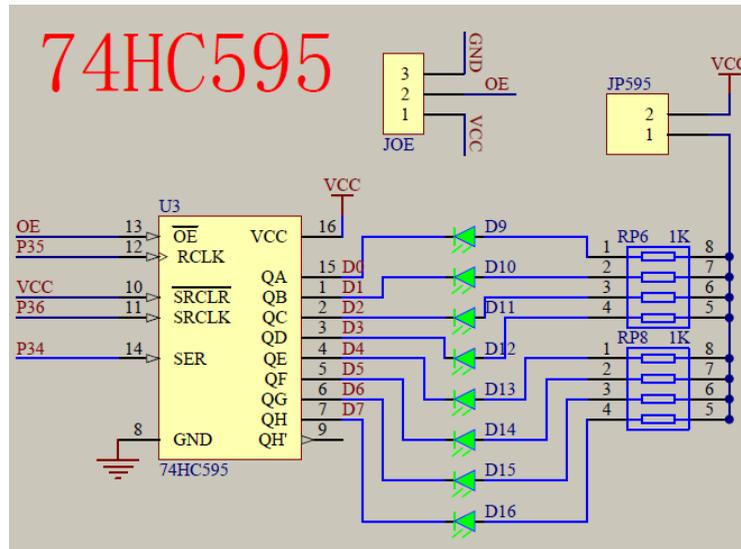
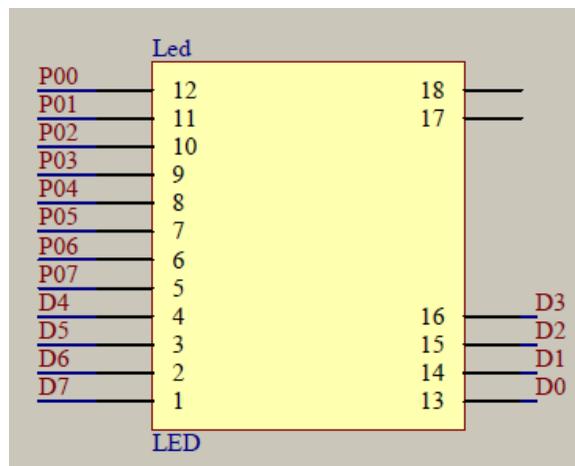


图 3: 8*8LED 点阵模块控制电路与 74HC595 控制电路接口

三、实验内容

- (1) 了解字符显示与消隐的原理和程序设计;
- (2) 查看实验箱硬件连线图, 了解 8*8LED 点阵和单片机引脚的连接;
- (3) 利用 C51 编程实现 8*8LED 点阵的数字显示;

四、参考程序

```
/*  
* 8*8LED 点阵——显示数字实验  
*实现现象: 下载程序后点阵上显示数字 0
```

注意事项: 一定要将 JOE 短接片短接到 GND 端, 如果不需要 595 后面的小灯点亮可以拔掉 JP595 短接片。

```
*/
```

```
#include "reg51.h" //此文件中定义了单片机的一些特殊功能寄存器  
#include<intrins.h>
```

```
typedef unsigned int u16; //对数据类型进行声明定义  
typedef unsigned char u8;
```

```
sbit SRCLK=P3^6;  
sbit RCLK=P3^5;  
sbit SER=P3^4;
```

```
u8 ledduan[]={0x00, 0x00, 0x3e, 0x41, 0x41, 0x41, 0x3e, 0x00};  
u8 ledwei[]={0x7f, 0xbf, 0xdf, 0xef, 0xf7, 0xfb, 0xfd, 0xfe};
```

```
/*
```

```
* 函数名 : delay  
* 函数功能 : 延时函数, i=1 时, 大约延时 10us
```

```
*/
```

```
void delay(u16 i)  
{  
    while(i--);  
}
```

```
/*
```

```
* 函数名 : Hc595SendByte(u8 dat)
```

```

* 函数功能      : 向 74HC595 发送一个字节的数
*****/
void Hc595SendByte(u8 dat)
{
    u8 a;
    SRCLK=0;
    RCLK=0;
    for(a=0;a<8;a++)
    {
        SER=dat>>7;
        dat<<=1;

        SRCLK=1;
        _nop_();
        _nop_();
        SRCLK=0;
    }

    RCLK=1;
    _nop_();
    _nop_();
    RCLK=0;
}

/*****
* 函数名      : main
* 函数功能    : 主函数
*****/
void main()
{
    u8 i;
    while(1)
    {
        P0=0x7f;
        for(i=0;i<8;i++)
        {
            P0=ledwei[i];      //位选

```

```
Hc595SendByte(leduan[i]); //发送段选数据
delay(100); //延时
Hc595SendByte(0x00); //消隐
}
}
}
```

五、实验思考题

尝试用点阵显示简单汉字，并探索汉字显示原理？

实验六 矩阵按键实验

一、实验目的

- (1) 掌握矩阵键盘编码原理与键值获取方法；
- (2) 掌握键盘延时去抖概念和原理；
- (3) 掌握 7 段数码管的显示原理与程序设计；

二、实验电路

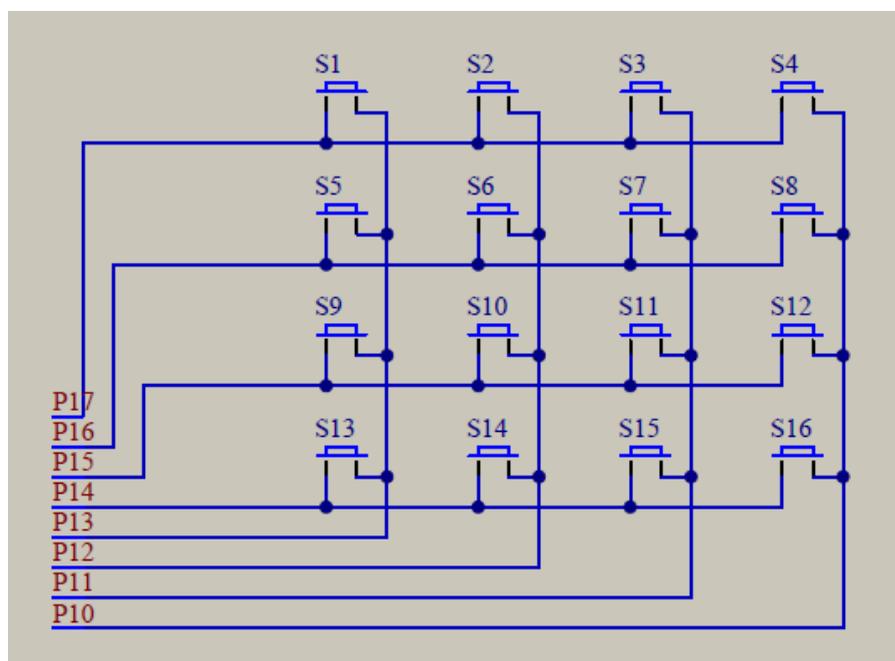


图 4: 矩阵按键模块接口电路

三、实验内容

(1) 查看硬件电路连线图，了解 4*4 矩阵键盘和 74HC595 的接口，以及 74HC595 和单片机的接口；

(2) 分配按键对应的显示数字：参考显示如下，学生可自定义

S1-S4: 0-3

S5-S8: 4-7

S9-S12: 8-B

S13-S16: C-F

(3) 巩固按键去抖程序;

(4) 利用 C51 编程, 实现矩阵键盘的键值读取, 以及按下相应键值在 LED 上显示对应数字。

四、参考程序

```
*****
*                               矩阵按键实验                               *
实现现象: 下载程序后数码管显示 0, 按下矩阵按键上的按键显示对应的数字
          S1-S4: 0-3
          S5-S8: 4-7
          S9-S12: 8-B
          S13-S16: C-F。
注意事项: 如果不想让点阵模块显示, 可以将 74HC595 模块上的 JP595 短接片拔掉。

*****/

#include "reg52.h"          //此文件中定义了单片机的一些特殊功能寄存器

typedef unsigned int u16;   //对数据类型进行声明定义
typedef unsigned char u8;

#define GPIO_DIG P0
#define GPIO_KEY P1

sbit LSA=P2^2;
sbit LSB=P2^3;
sbit LSC=P2^4;

u8 KeyValue; //用来存放读取到的键值

u8 code smgduan[17]={0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07,
                    0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71}; //显示 0~F 的值

/*****
* 函数名      : delay
* 函数功能    : 延时函数, i=1 时, 大约延时 10us
*****/
void delay(u16 i)
```

```

{
    while(i--);
}
/*****
* 函数名      : KeyDown
* 函数功能    : 检测有按键按下并读取键值
*****/
void KeyDown(void)
{
    char a=0;
    GPIO_KEY=0x0f;
    if (GPIO_KEY!=0x0f)//读取按键是否按下
    {
        delay(1000);//延时 10ms 进行消抖
        if (GPIO_KEY!=0x0f)//再次检测键盘是否按下
        {
            //测试列
            GPIO_KEY=0X0F;
            switch (GPIO_KEY)
            {
                case (0X07):   KeyValue=0;break;
                case (0X0b):   KeyValue=1;break;
                case (0X0d):   KeyValue=2;break;
                case (0X0e):   KeyValue=3;break;
            }
            //测试行
            GPIO_KEY=0XF0;
            switch (GPIO_KEY)
            {
                case (0X70):   KeyValue=KeyValue;break;
                case (0Xb0):   KeyValue=KeyValue+4;break;
                case (0Xd0):   KeyValue=KeyValue+8;break;
                case (0Xe0):   KeyValue=KeyValue+12;break;
            }
            while ((a<50)&&(GPIO_KEY!=0xf0)) //检测按键松手检测
            {
                delay(1000);
            }
        }
    }
}

```


实验七 串行通信实验

一、实验目的

- (1) 了解串口通信原理;
- (2) 掌握单片机串口通信相关寄存器设置;
- (3) 编程实现单片机串口通信。

二、实验电路

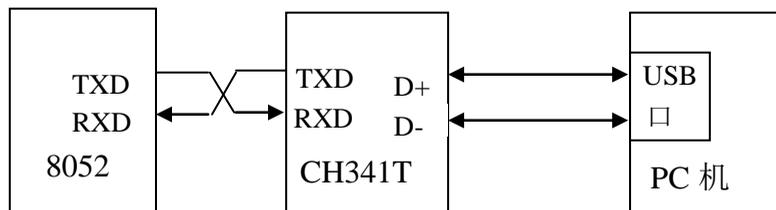


图 5: 串口接线原理图

三、实验内容

- (1) 熟悉中断编程方法;
- (2) 熟悉串口控制寄存器的设置方法;
- (3) 利用 C51 编程实现串口通信;

四、参考程序

```
/*  
*          串口通信实验          *  
实现现象：下载程序后打开串口调试助手，将波特率设置为 4800，选择发送的数据就可以显示在串口助手上。  
*/  
#include "reg52.h" //此文件中定义了单片机的一些特殊功能寄存器  
typedef unsigned int u16; //对数据类型进行声明定义  
typedef unsigned char u8;  
/*  
* 函数名      :UsartInit()  
* 函数功能    :设置串口  
*/  
void UsartInit()
```

```

{
    SCON=0X50;        //设置为工作方式 1
    TMOD=0X20;       //设置计数器工作方式 2
    PCON=0X80;       //波特率加倍
    TH1=0XF3;        //计数器初始值设置，注意波特率是 4800 的
    TL1=0XF3;
    ES=1;            //打开接收中断
    EA=1;            //打开总中断
    TR1=1;           //打开计数器
}
/*****
* 函数名      : main
* 函数功能    : 主函数
*****/
void main()
{
    UsartInit(); // 串口初始化
    while(1);
}
/*****
* 函数名      : Usart() interrupt 4
* 函数功能    : 串口通信中断函数
*****/
void Usart() interrupt 4
{
    u8 receiveData;
    receiveData=SBUF;//出去接收到的数据
    RI = 0;          //清除接收中断标志位
    SBUF=receiveData;//将接收到的数据放入到发送寄存器
    while(!TI);     //等待发送数据完成
    TI=0;           //清除发送完成标志位
}

```

附录：烧写软件使用说明

PZ-ISP 下载软件配置

1、打开 PZ-ISP V1.72 软件，界面如图 6 所示。

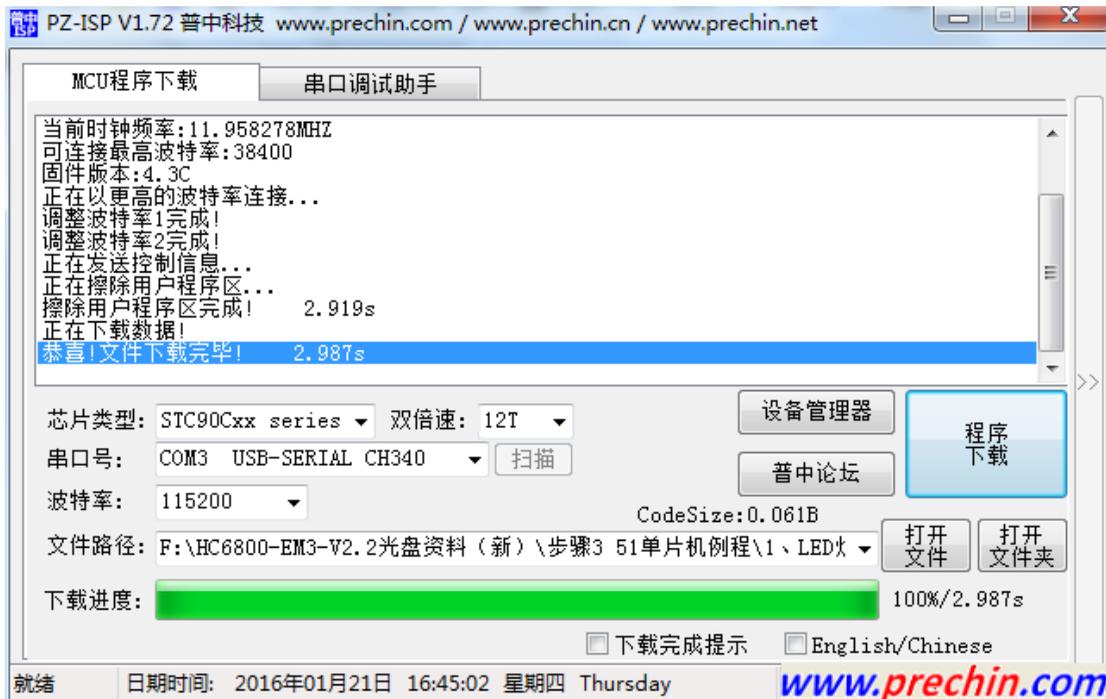
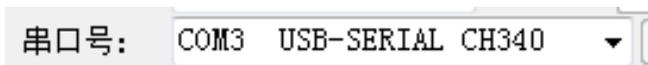


图 6: PZ-ISP V1.72 软件界面图

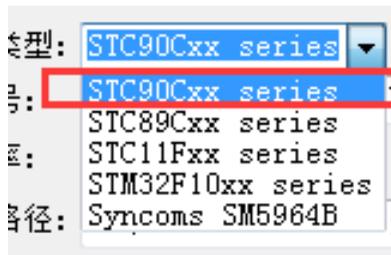
注：在这里打开软件后首先要确认实验箱是否连接到电脑。
看串口号有 ch340 就说明已经被识别的



或者是在设备管理器的端口下查看，如果是未能识别串口就先把驱动安装上



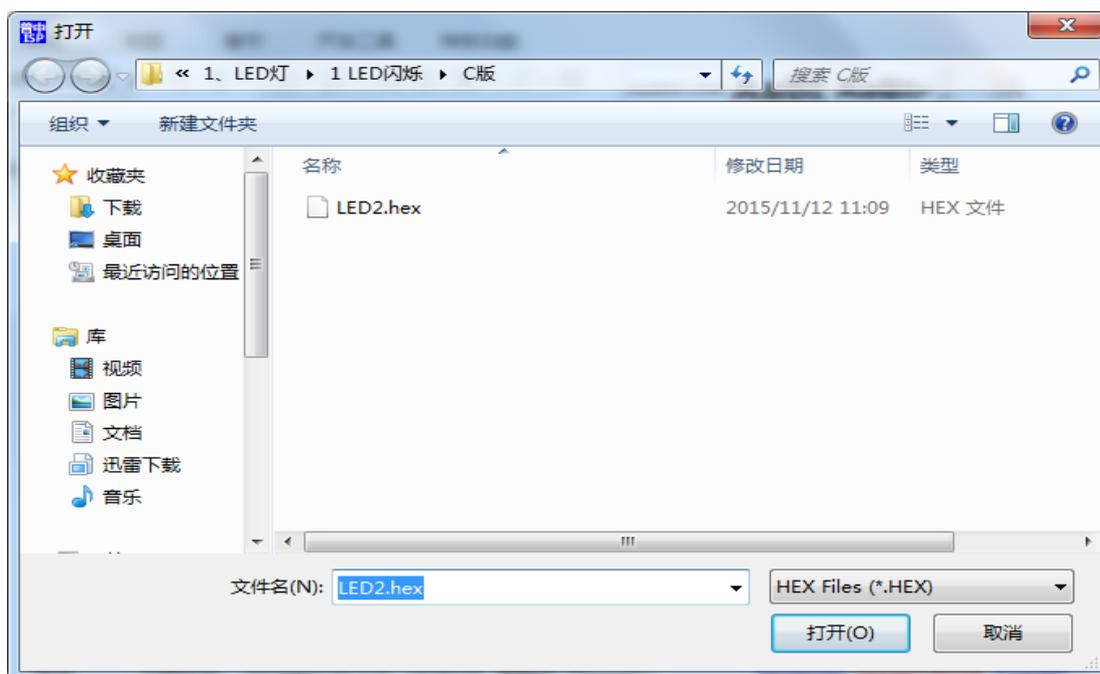
2、选择型号，根据实验箱上单片机的型号选择 STC89cxx series 系列芯片。



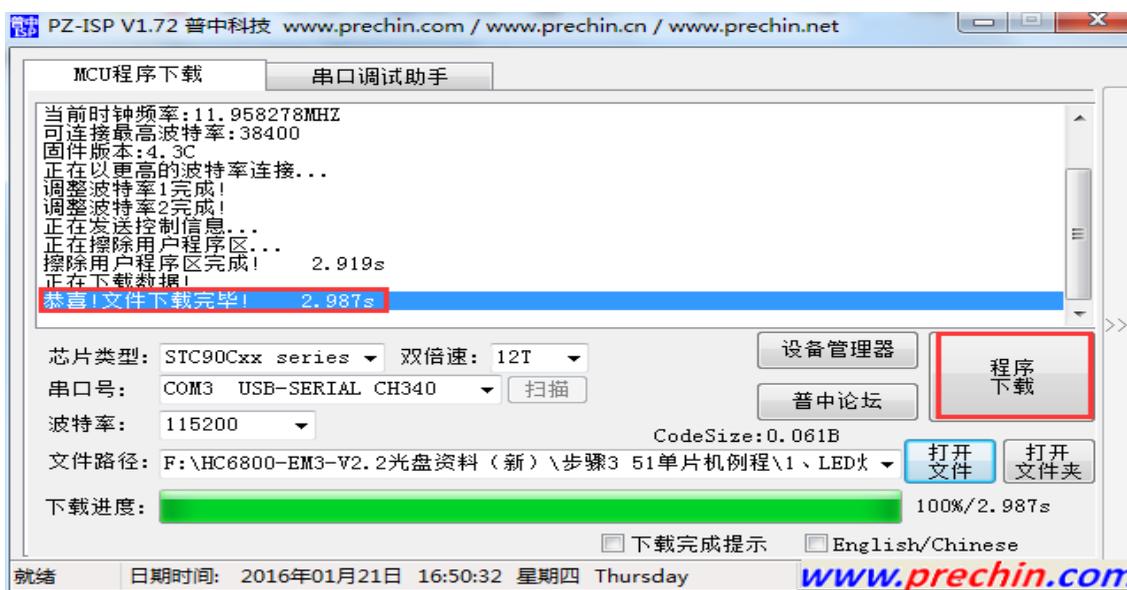
3、选择波特率，波特率选择 115200，如果出现报错可以把波特率调低一点



4、点击打开文件找到光盘的步骤三选择需要下载的 hex 文件。（如果出现不能打开文件夹就把 hex 文件拖到桌面，把桌面上面的 hex 加进去。）



5、Hex 文件添加进去之后点击下载程序等待下载完成。



6、在试验箱上查看实验效果，如为达到实验要求，可修改软件重新编译下载，直至实验完成。