

《嵌入式与接口技术》

实验指导书

武昌首义学院
信息工程学院 计算机系
2025 年 9 月

目 录

实验一 单片机存储器实验	1
一、实验目的	1
二、实验平台	1
三、实验要求	1
四、实验内容和步骤	1
实验内容 1: 使用 Keil 软件熟悉编程环境	1
实验内容 2: 理解单片机的存储器	9
五、实验总结与心得	13
实验二 C51 语言编程实验	14
一、实验目的	14
二、实验平台	14
三、实验要求	14
四、实验内容和步骤	14
实验内容 1: 使用 Proteus 完成电路图设计	14
实验内容 2: 使用 Keil 软件完成 C 语言编程	16
五、实验总结与心得	16
实验三 I/O 接口实验	17
一、实验目的	17
二、实验平台	17
三、实验要求	17
四、实验内容和步骤	17
实验内容 1: 使用 Proteus 完成电路图设计	17
实验内容 2: 编程实现 I/O 口控制	19
五、实验总结与心得	22
实验四 定时器应用实验	23
一、实验目的	23
二、实验平台	23
三、实验要求	23
四、实验内容和步骤	23
实验内容 1: 完成定时器定时功能	23
实验内容 2: 完成秒表的设计	24
实验内容 3 (扩展功能) : 完成智能校车的 PWM 调速实验	25
五、实验总结与心得	25

实验一 单片机存储器实验

一、实验目的

1. 掌握 C51 单片机 Keil 编程环境;
2. 掌握 C51 单片机内存 RAM、ROM 特点;
3. 掌握访问单片机内存 RAM、ROM 方法。

二、实验平台

1. 计算机;
2. 安装 KEIL 软件、Proteus 软件、STC 烧录软件;

三、实验要求

1. 熟悉 C51 单片机内存 RAM 和 ROM 知识;
2. 阅读 keil 使用手册, 熟练使用 keil 编程调试方法;
3. 掌握使用 data、bdata、idata、pdata、xdata 和 code 存储器类型使用 C51 内存, 编写程序, 分析关键语句和关键程序段功能, 调试并观察存储器变化。
4. 按照要求认真记录并规范如实填写实验报告, 提交实验报告 (电子版+打印版), 电子版命名方式“学号+班级+姓名+实验一”。

四、实验内容和步骤

实验内容 1: 使用 Keil 软件熟悉编程环境

1. 添加型号、头文件和 STC 仿真驱动

Keil 本来不支持 STC 系列的芯片, 需要自己手动添加相应的文件。使用宏晶官网提供的 STC-ISP 工具根据项目使用的单片机芯片添加型号、头文件和 STC 仿真驱动, 添加成功后, 就能在 Keil 新建工程的时候选择 STC MCU。

(1) 打开 STC-ISP 下载软件, 在 STC-ISP 右上方选择“Keil 仿真设置”标签, 单击“添加型号和头文件到 Keil 中”按钮, 如图 1 所示;

(2) 在“浏览文件夹”对话框里, 选择 Keil 的安装目录, 确定目录下有 C51 和 UVx 目录, 点击“确定”按钮, 如图 2 所示。然后 STC-ISP 弹出“STC MCU 型号添加成功! ”。添加成功后, “C:\Keil_v5\C51\BIN”文件夹中出现 STC Monitor51 仿真驱动程序

“stcmon51.dll”, 同时“C:\Keil_v5\C51\INC\STC”文件夹中出现 STC 头文件, 在 Keil 中新建工程选择芯片型号时, 便会有“STC MCU Database”选项, 从 MCU 列表中选择 MCU 型号。

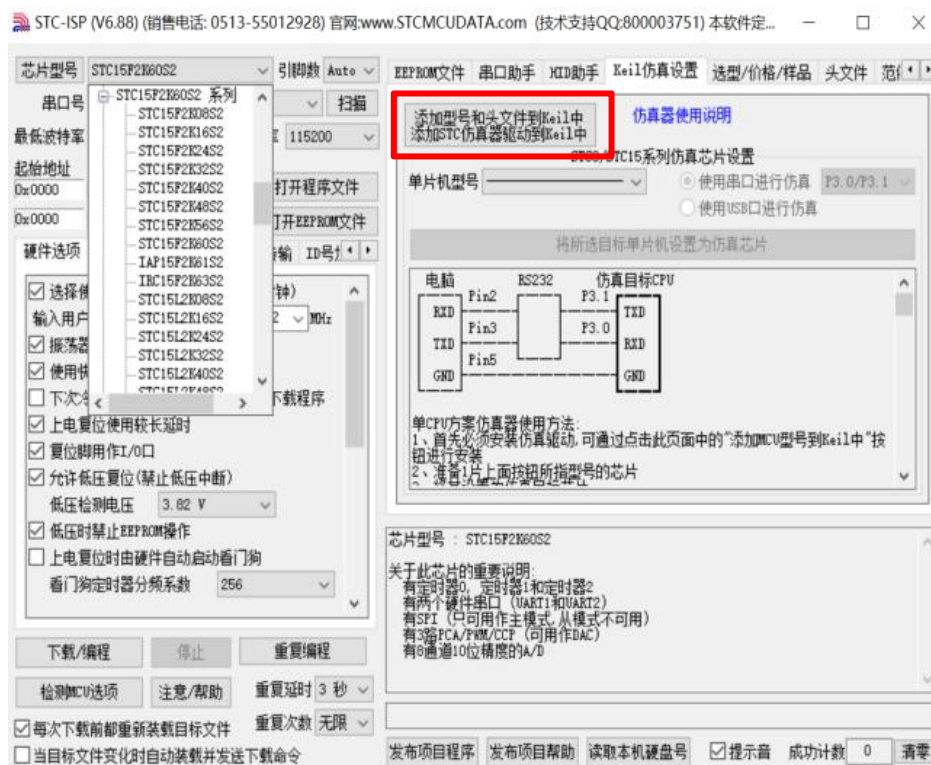


图 1 Keil 仿真设置界面

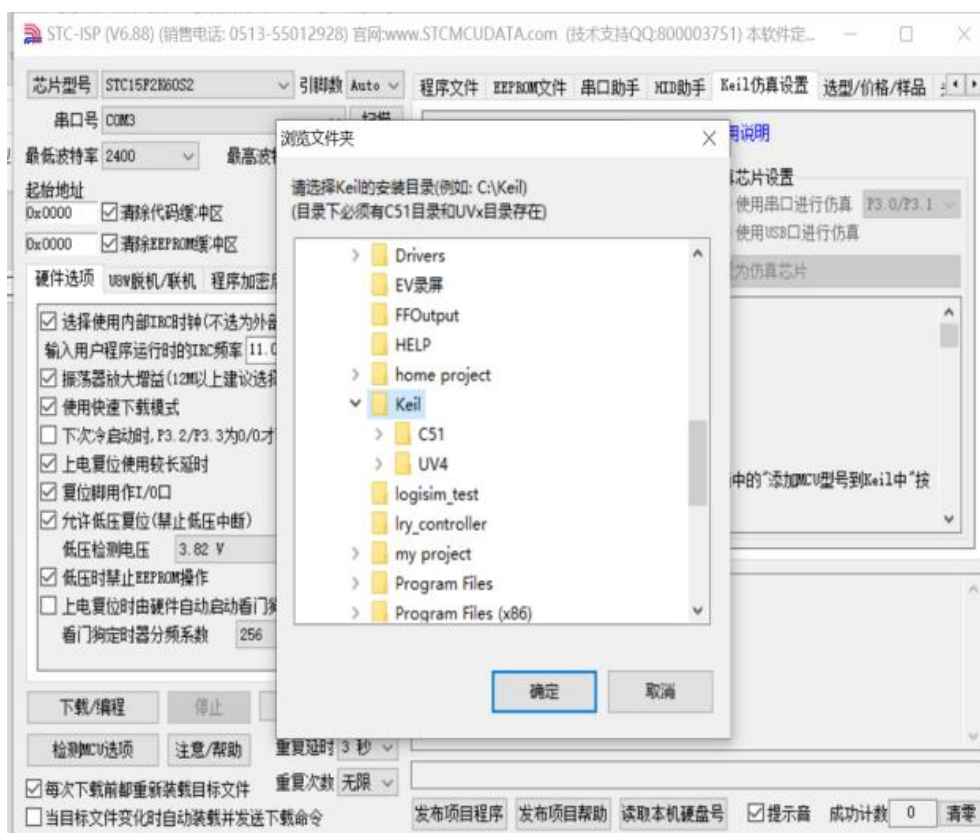


图 2 路径选择界面

2. 新建一个工程

- (1) 运行 keil;
- (2) 点击 Project –> New uvision –> 选择自己要新建的项目路径，新建一个项目，选择“STC MCU Database” –> 选择 C51 对应型号芯片，如 “STC15F2K60S2 Series”。

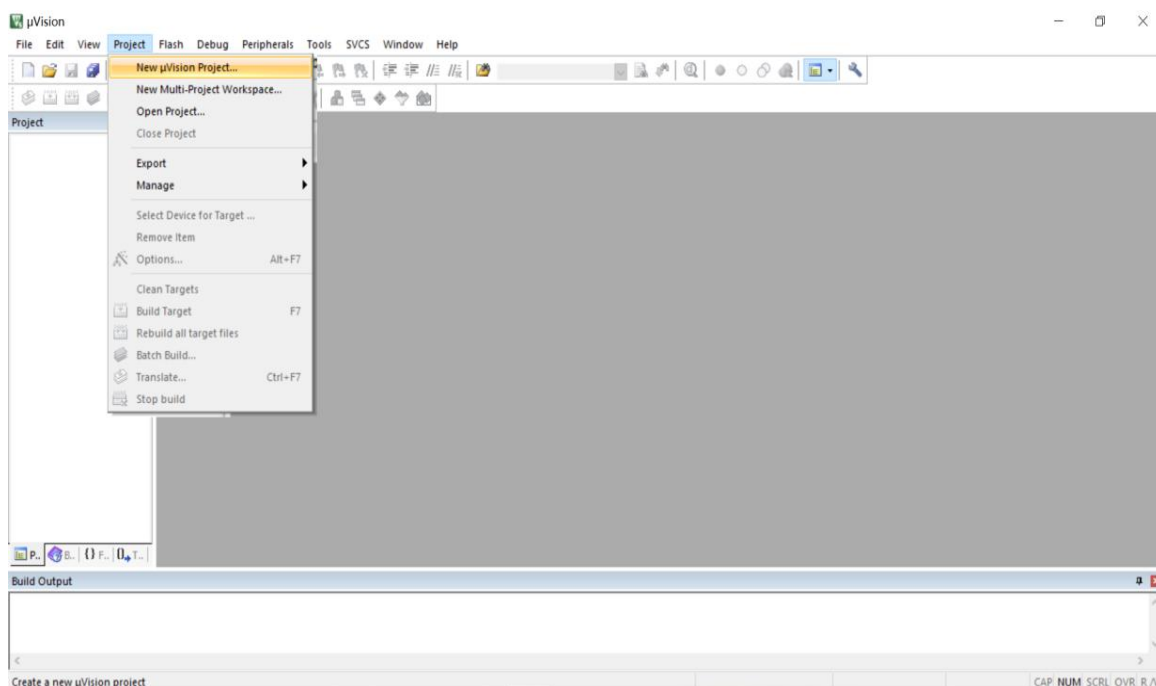


图 3 新建 keil 工程

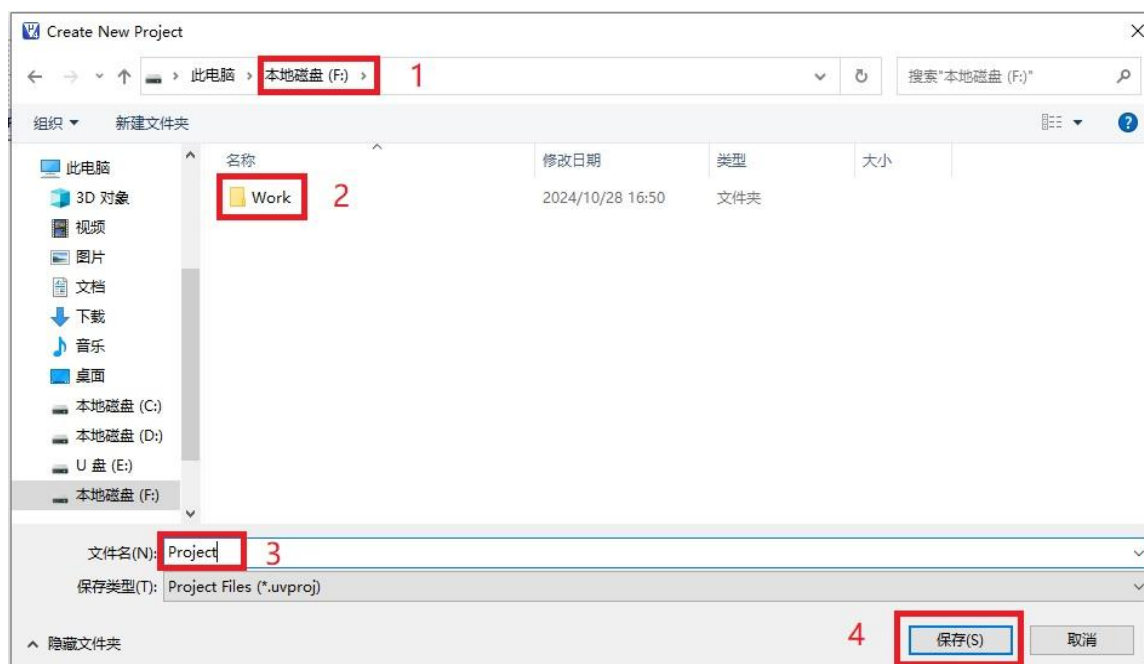


图 4 创建 Project 工程文件

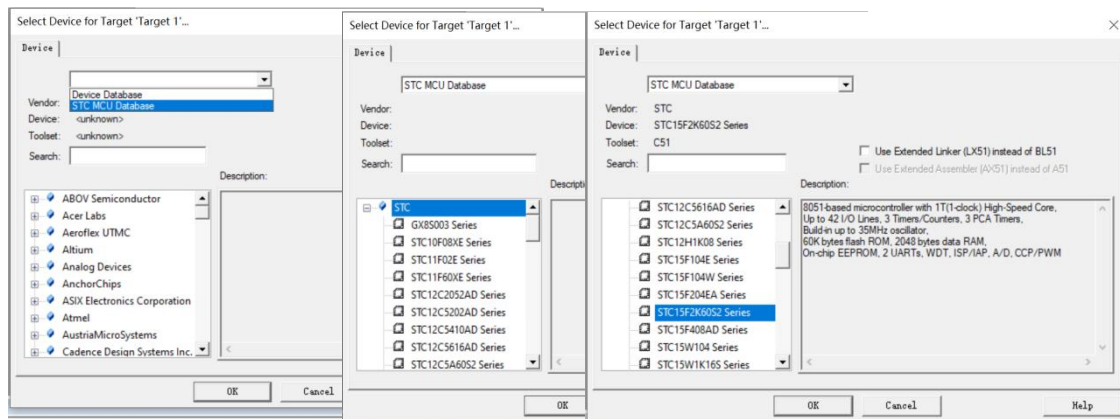


图 5 选择 STC15F2K60S2 芯片

(3) 在工程界面中，点击菜单“File”中的“New...”或界面快捷图标，新建一个文件，用于编写代码。单击“File”菜单下的“Save”菜单项或文件工具栏中的“Save”按钮，在打开的对话框中输入文件名“main.c”，点击“保存 (S)”按钮，如图所示。当保存完文件以后，关键字则会特殊的颜色出现，方便用户检查拼写的错误。

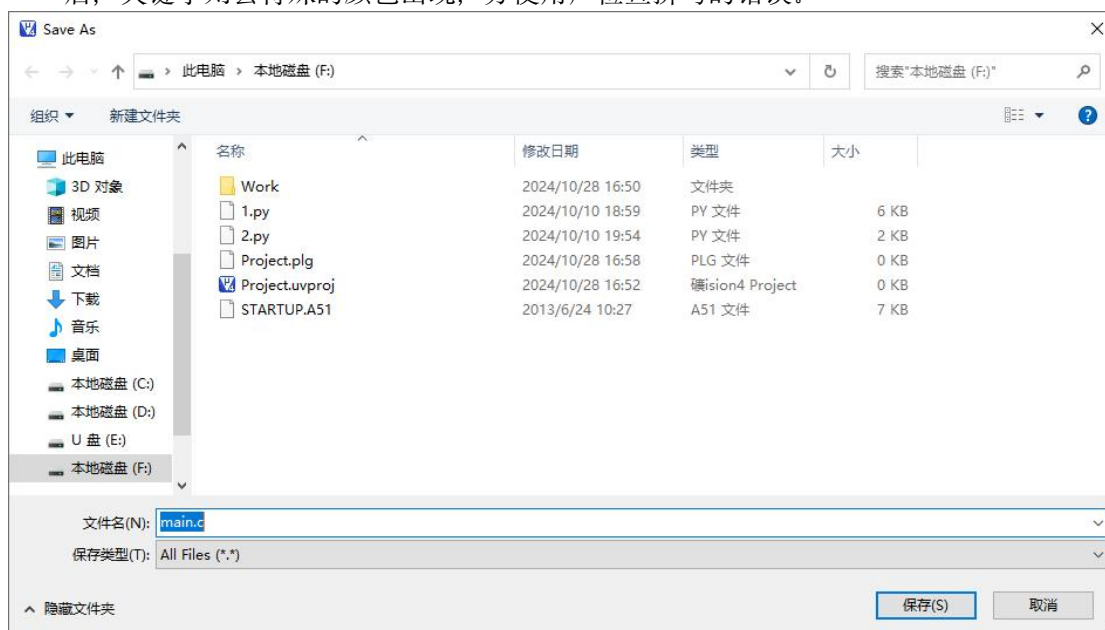


图 6 新建 main.c 文件

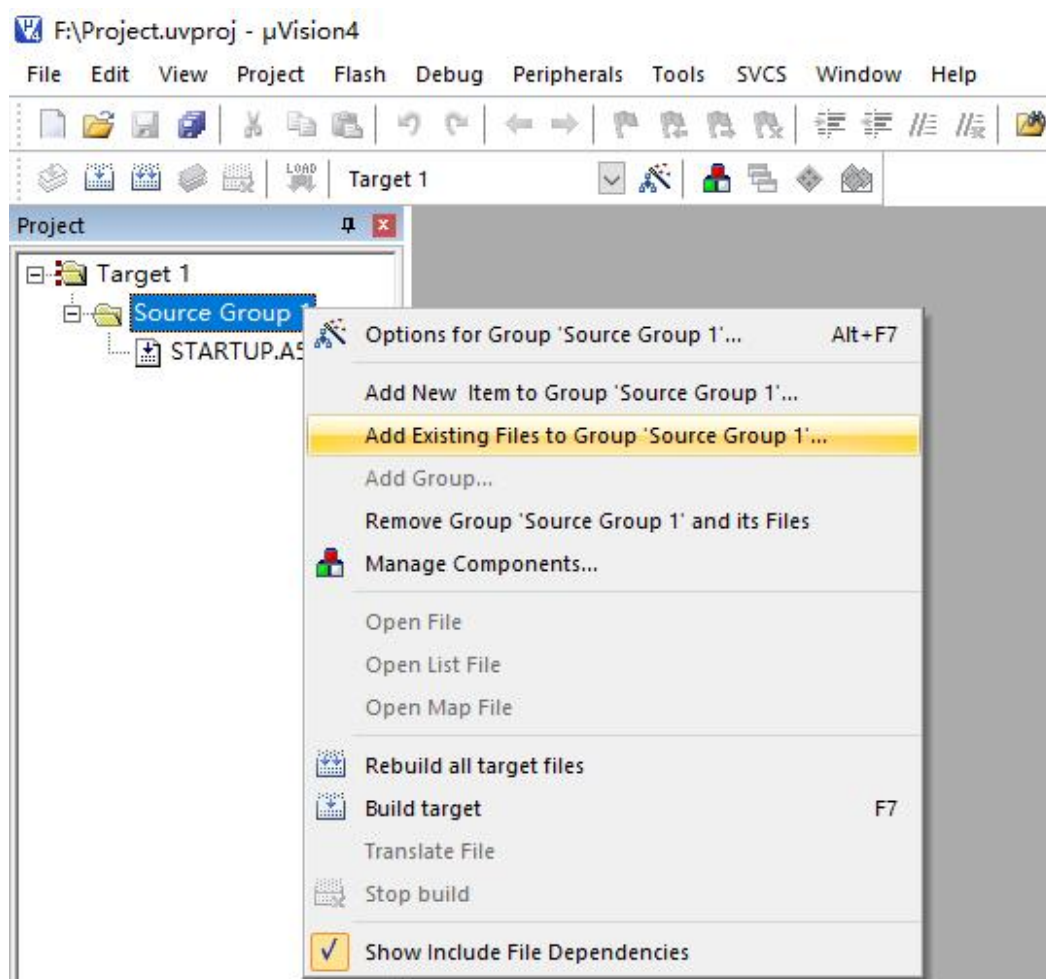


图 7 添加工程文件

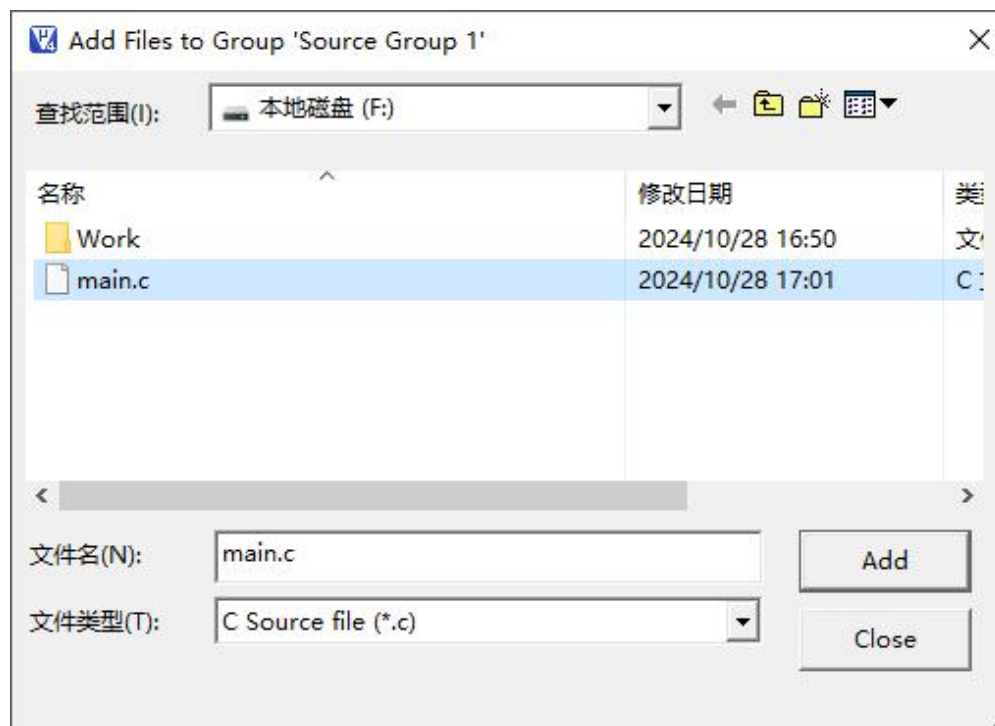


图 8 添加 main.c 文件



图 9 添加 main.c 文件后效果

```

1 #include <reg52.h>
2 #include <absacc.h>
3
4 #define uchar unsigned char
5 uchar var;
6 void main()
7 {
8     unsigned char xx;
9     unsigned char y;
10    data uchar *P;
11    SP = 0x59;
12    DBYTE[0x00] = 0x00;
13    DBYTE[0x01] = 0x01;
14    DBYTE[0x02] = 0x02;
15    DBYTE[0x03] = 0x03;
16    DBYTE[0x04] = 0x04;
17    DBYTE[0x05] = 0x05;
18    DBYTE[0x06] = 0x06;
19    DBYTE[0x07] = 0x07;
20    DBYTE[0x30] = 0xA0;
21    DBYTE[0x31] = 0xA1;
22    DBYTE[0x32] = 0xA2;
23    DBYTE[0x33] = 0xA3;
24    DBYTE[0x34] = 0xA4;
25    DBYTE[0x35] = 0xA5;
26    DBYTE[0x36] = 0xA6;
27    DBYTE[0x37] = 0xA7;
28    xx = DBYTE[0x36];
29    B = 0x99;
30    CY = 1;
31    RSO = 0;
32    RS1 = 1;
33    y = ACC;
34    P = 0x20;
35    *P = 0x38;
36    DBYTE[0x20] = 0x88;
37    DBYTE[0x21] = 0x99;
38    var = 0x21;
39
40 }

```

图 10 在 main.c 文件中编写程序

(4) 点击 file -> new, 新建一个 Text 文件并另存为“main.c”, 在“main.c”文件里练习所给代码, “Build”编译 -> debug 模式, 用 F10 单步执行, 其他操作参考 keil 使用手册。

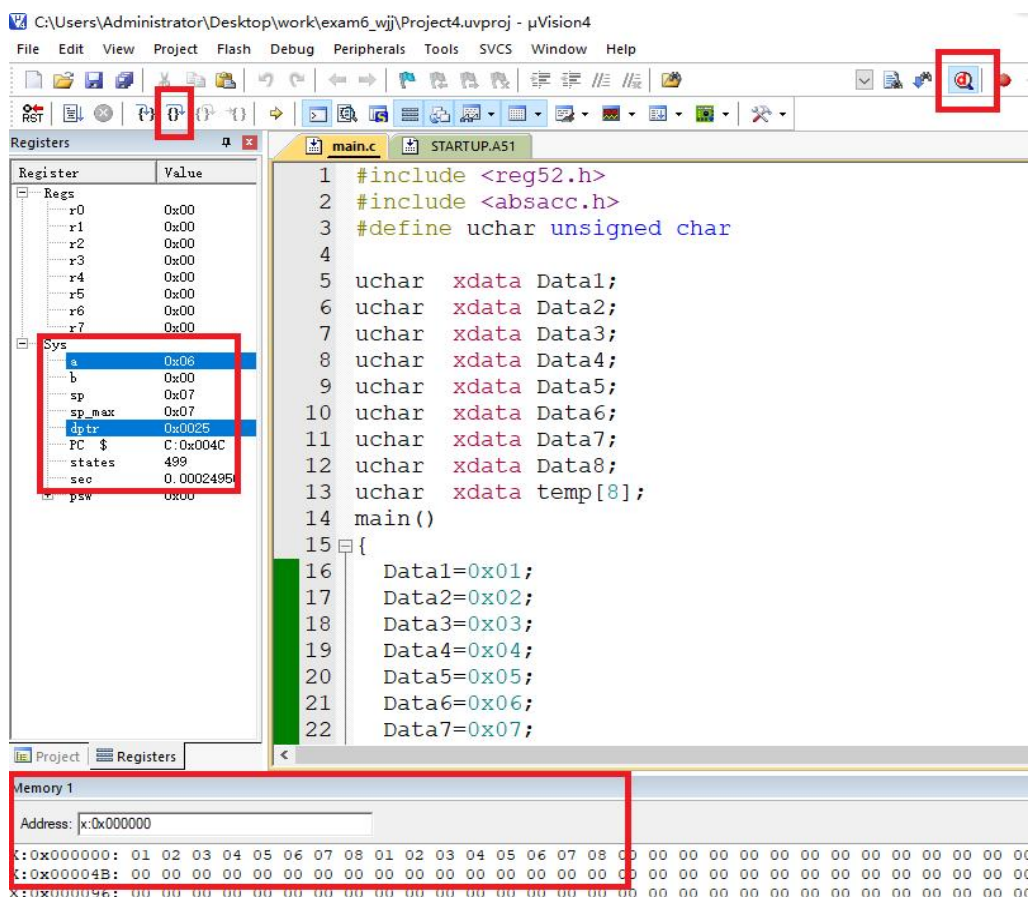


图 11 C51 单片机内存里变化

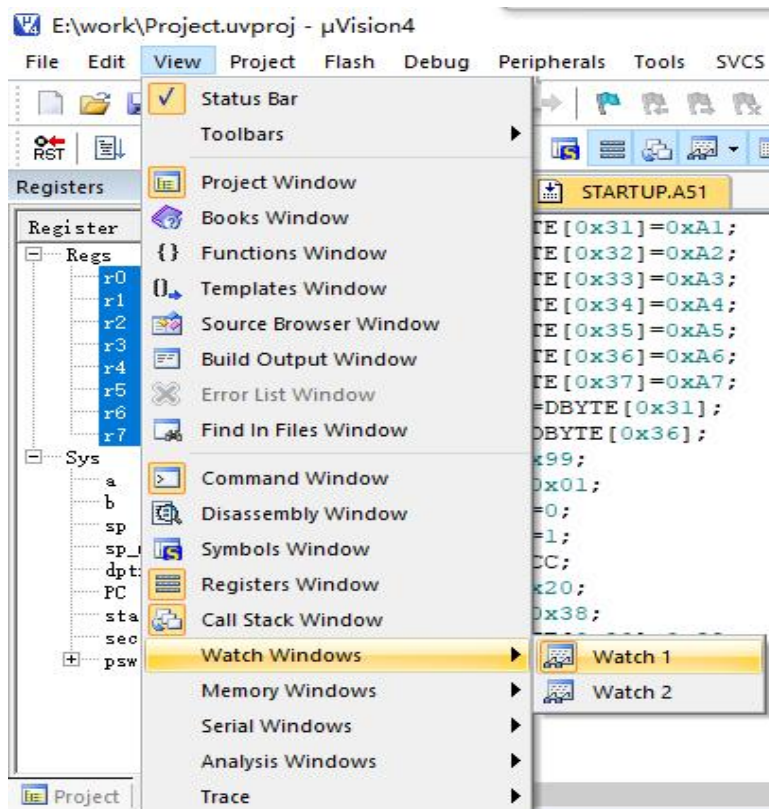


图 12 调试环境下调观察窗口

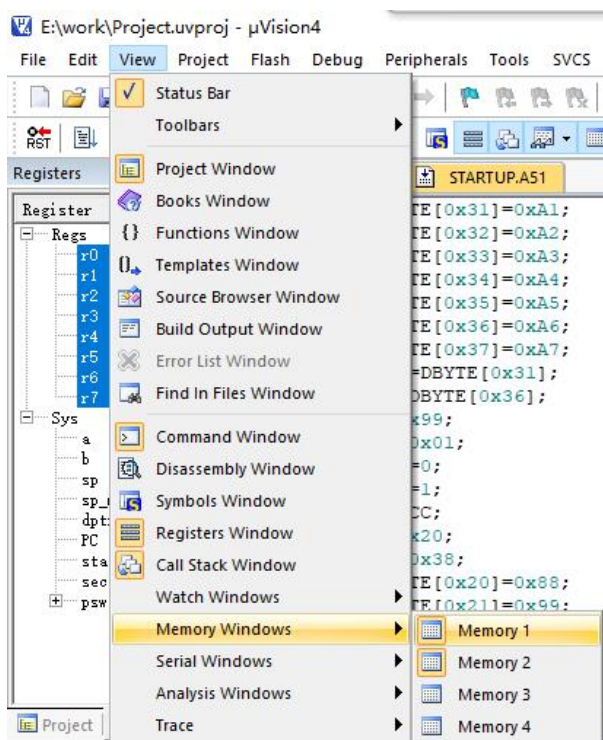


图 13 调试环境下调存储器窗口

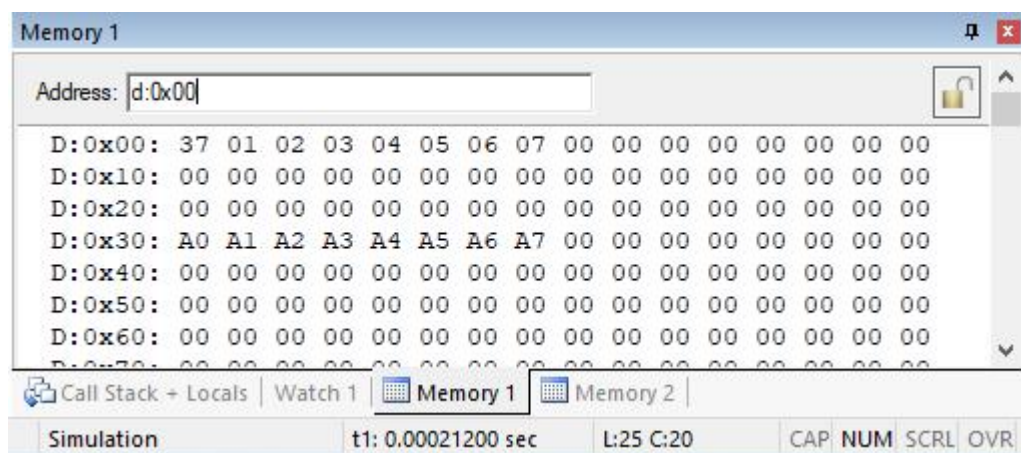


图 14 调试环境下存储器窗口

打开在此编辑框内我们输入“字母:数字”,其中字母代表寻址空间,所在寻址空间的地址。字母代表寻址空间含义分别如下所示。

D:代表直接寻址的片内 RAM

I:代表间接寻址的片内 RAM

X:代表扩展的外部 RAM

C:代表程序存储器 ROM

问题:

1.请说明 SP=0x59;这条语句的功能, 栈指针 SP 在 51 单片机中起什么作用?

2.分析 DBYTE[0x00]=0x00;到 DBYTE[0x37]=0xA7;这些语句的作用, DBYTE 在这里代表什么存储区域?

3.变量 xx 被赋值为 DBYTE[0x36];, 请说出 xx 的最终值是多少?

4.说明 RS0=0; RS1=1 中 RS0 和 RS1 的作用, 改变它的值会产生什么变化?

5.分析指针操作 p=0x20; *p=0x38 的功能, 这里的 0x20 地址属于哪个存储区域?

6.语句 DBYTE[0x20]=0x88; DBYTE[0x21]=0x99;与前面的指针操作有什么联系和区别?

7.变量 var 被赋值为 0x21;, 请思考 var 在程序中的数据类型以及它的存储方式。

结合整个代码, 简述该程序在 51 单片机中运行时, 理解内部数据存储器的操作流程和关键节点。

实验内容 2:理解单片机的存储器

1. 分别使用 data、bdata、idata、pdata、xdata 和 code 类型变量, 编写程序调用 C51 相应的存储器, 并观察存储器变化, 掌握基本的数据读写操作。将自己的结果截图提交。

(1) 使用 data 定义变量, 读写内部存储区 0x00-0x7F 128 字节存储单元;

```
#include <reg52.h>
#define uchar unsigned char

uchar  data Data1  _at_ 0x30;
uchar  data Data2  _at_ 0x31;
uchar  data Data3  _at_ 0x32;
uchar  data Data4  _at_ 0x33;
uchar  data Data5  _at_ 0x34;
uchar  data Data6  _at_ 0x35;
uchar  data Data7  _at_ 0x36;
uchar  data Data8  _at_ 0x37;
uchar  data temp[8] _at_ 0x40;
main()
{
    Data1=0x01;
    Data2=0x02;
    Data3=0x03;
    Data4=0x04;
    Data5=0x05;
    Data6=0x06;
    Data7=0x07;
    Data8=0x08;

    temp[0]=Data1;
    temp[1]=Data2;
    temp[2]=Data3;
    temp[3]=Data4;
    temp[4]=Data5;
    temp[5]=Data6;
    temp[6]=Data7;
    temp[7]=Data8;
    while(1);
}
```

(2) 使用 bdata 定义位寻址变量，读写位寻址区域 0x20-0x2F 16 字节存储区；

```
#include <reg52.h>
#define uchar unsigned char

bit    bit0;
bit    bit1;
bit    bit2;
bit    bit3;
bit    bit4;
bit    bit5;
bit    bit6;
bit    bit7;
uchar  bdata  bdata1;
main()
{
    bit0=1;
    bit1=0;
    bit2=1;
    bit3=0;
    bit4=1;
    bit5=0;
    bit6=1;
    bit7=0;
    bdata1=1;

    while(1);
}
```

(3) 使用 idata 定义变量，读写内部存储区 0x00-0xFF 256 字节存储单元；

```
#include <reg52.h>
#define uchar unsigned char

uchar  idata Data1  _at_ 0x80;
uchar  idata Data2  _at_ 0x81;
uchar  idata Data3  _at_ 0x82;
uchar  idata Data4  _at_ 0x83;
uchar  idata Data5  _at_ 0x84;
uchar  idata Data6  _at_ 0x85;
uchar  idata Data7  _at_ 0x86;
uchar  idata Data8  _at_ 0x87;
uchar  idata temp[8] _at_ 0x90;
main()
{
    Data1=0x01;
    Data2=0x02;
    Data3=0x03;
    Data4=0x04;
    Data5=0x05;
    Data6=0x06;
    Data7=0x07;
    Data8=0x08;

    temp[0]=Data1;
    temp[1]=Data2;
    temp[2]=Data3;
    temp[3]=Data4;
    temp[4]=Data5;
    temp[5]=Data6;
    temp[6]=Data7;
    temp[7]=Data8;
    while(1);
}
```

- (4) 使用 xdata 定义变量，读写外部存储区 0x0000-0xFFFF 65536 字节存储单元；

```
#include <reg52.h>
#include <absacc.h>
#define uchar unsigned char

uchar xdata Data1;
uchar xdata Data2;
uchar xdata Data3;
uchar xdata Data4;
uchar xdata Data5;
uchar xdata Data6;
uchar xdata Data7;
uchar xdata Data8;
uchar xdata temp[8];
main()
{
    Data1=0x01;
    Data2=0x02;
    Data3=0x03;
    Data4=0x04;
    Data5=0x05;
    Data6=0x06;
    Data7=0x07;
    Data8=0x08;

    temp[0]=Data1;
    temp[1]=Data2;
    temp[2]=Data3;
    temp[3]=Data4;
    temp[4]=Data5;
    temp[5]=Data6;
    temp[6]=Data7;
    temp[7]=Data8;

    XBYTE[0x20]=Data1;
    XBYTE[0x21]=Data2;
    XBYTE[0x22]=Data3;
    XBYTE[0x23]=Data4;
    XBYTE[0x24]=Data5;
    XBYTE[0x25]=Data6;
    XBYTE[0x26]=Data7;
    XBYTE[0x27]=Data8;

    while(1);
}
```

(5) 使用 code 定义数据, 读程序存储区 0x0000-0xFFFF 65536 字节存储单元(此区域不可写)。

五、实验总结与心得

(建议写自己遇到的问题及解决方法、实验总结的经验和心得体会, 每个人不一样)

实验二 C51 语言编程实验

一、实验目的

1. 掌握 51 单片机最小系统的搭建;
2. 熟悉 Protues 仿真软件, 能够绘制简单的电路原理图, 掌握仿真的方法;
3. 熟悉 Keil 开发环境, 掌握 C51 的编程方法, 熟悉工程创建、编辑文件、编译调试等步骤;
4. 掌握嵌入式开发的流程, 能够用烧录软件将程序写入单片机。

二、实验平台

1. 计算机;
2. 安装 KEIL 软件、Proteus 软件、STC 烧录软件;

三、实验要求

1. 阅读 Proteus 软件使用手册, 了解 Proteus 的用途、界面等信息;
2. 阅读 KEIL 软件使用手册, 熟悉其操作与使用方法;
3. 熟悉相关知识: C51 语言编程知识;
4. 按照要求认真记录并规范如实填写实验报告, 提交实验报告(电子版+打印版), 电子版命名方式“学号+班级+姓名+实验二”。

四、实验内容和步骤

实验内容 1: 使用 Proteus 完成电路图设计

1. 启动 Proteus 软件。
 2. 选择“File”→“New Design...”命令, 创建一个新工程。
 3. 选择工程所在的路径(注意全英文), 设置工程名字(注意使用英文字符)。
- 选择电路元器件, 点击如下图所示图标(或选择菜单栏 library->Pick Parts), 依次选择元器件“STC15W4K32S4”、“CAP”、“CAP-ELEC”、“CRYSTAL”、“LED-RED”、“RES”和“SWITCH”。

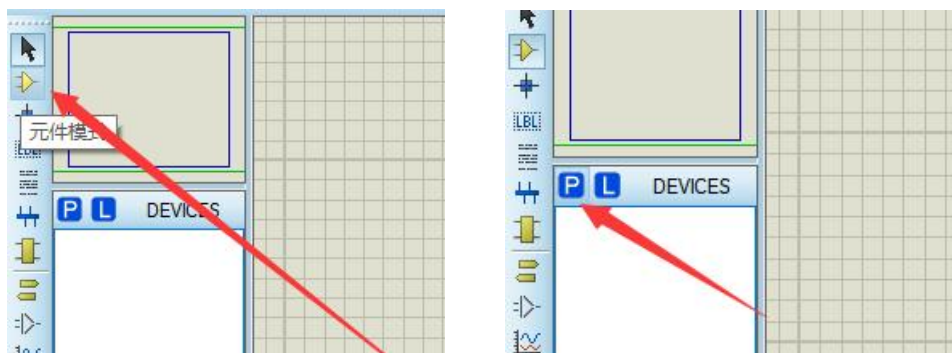


图 2-1 添加元器件 (1)

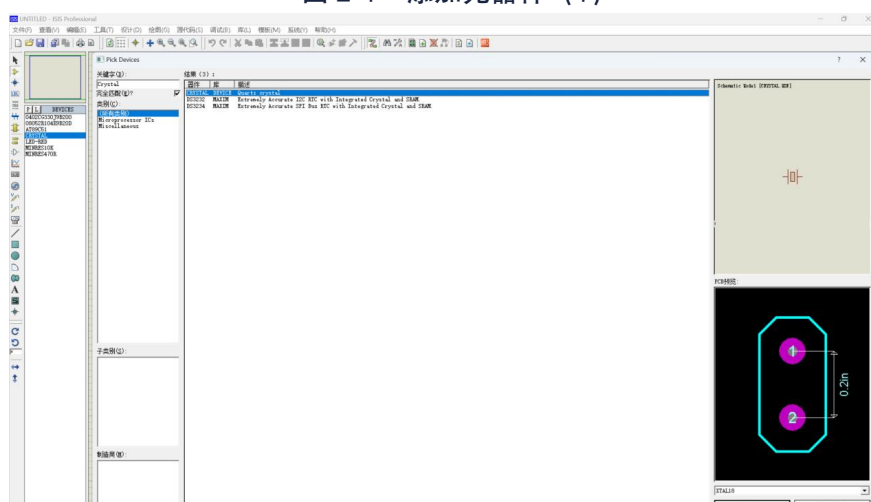


图 2-2 添加元器件 (2)

4. 完成如下电路图的绘制。

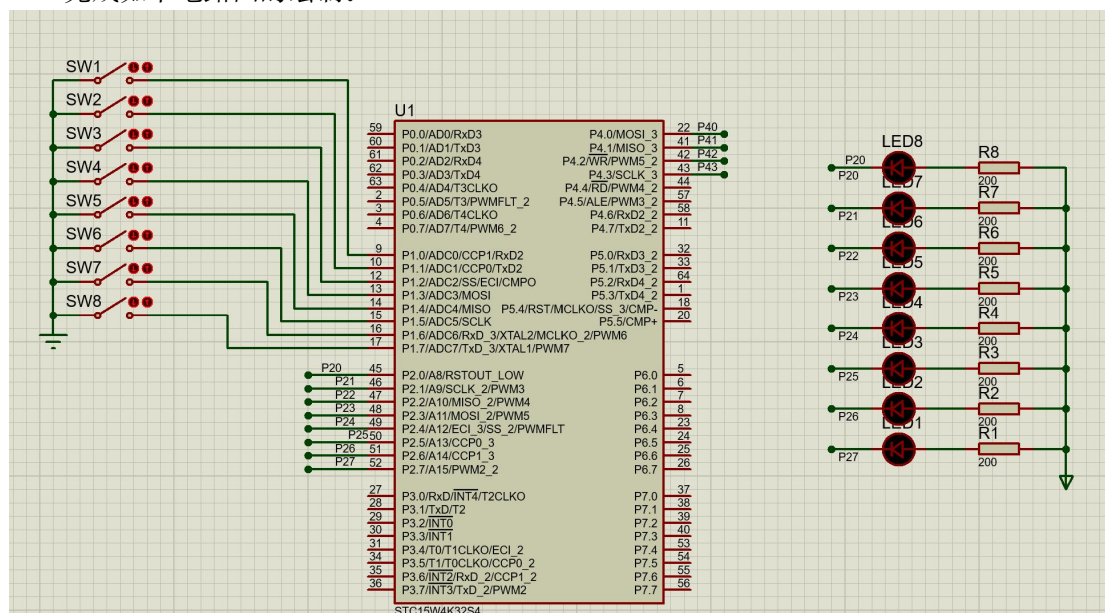


图 2-3 绘制电路图

5. 加载提供的“实验 2.hex”文件，检查电路的正确性，熟悉仿真功能。
6. 将验证通过的电路图文件以“学号+电路图”命名保存（电子档），实验报告中放仿真运行正常的状态的截图，并用文字说明。

实验内容 2：使用 Keil 软件完成 C 语言编程

1. 打开 keil 软件，进入 Keil μ Vision5 集成开发环境。
2. 新建工程，可以点击菜单的“Project”的“New μ Vision Project...”按键，工程文件、工程文件夹用英文命名。
3. 选择项目所用的单片机型号。在出现的对话框里，下拉选项，选择“STC MCU Database”--“STC15F2K60S2 Series”，最后点击“确定”按键。

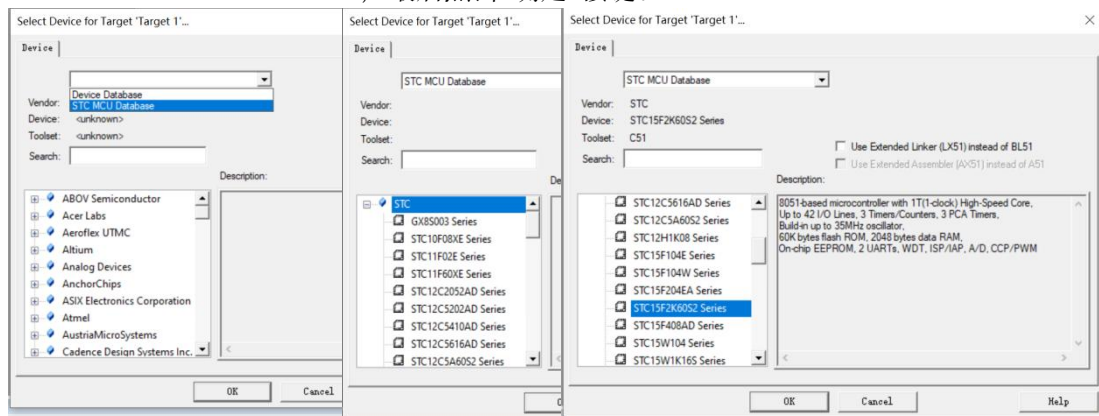


图 2-4 选择单片机型号

4. 新建源程序文件，输入“`#include <STC15F2K60S2.h>`”，然后保存为“main.c”。
5. 将新建的源文件添加到项目工程中。
6. 编写程序完成功能，依次完成下列功能。
 - (1) 将 RAM 的地址空间“30H ~ 60H”的内容清 0；
 - (2) 编程实现 BCD 码转 ASCII 码；
 - (3) 编程实现双字节的 BCD 码的加法运算；
 - (4) 编程实现内存块移动，将片外 RAM 1000H 单元开始的内容送片内 RAM 20H 单元，长度为 40 个字节。
7. 在 keil 软件中编译、调试。
8. 执行该程序，观察结果。
9. 完成实验报告，将编程分析（可以流程图、文字）和程序代码（需注释代码）撰写在报告中。

五、实验总结与心得

（建议写自己遇到的问题及解决方法、实验总结的经验和心得体会，每个人不一样）

实验三 I/O 接口实验

一、实验目的

1. 熟悉 Proteus、KEIL 软件的使用；
2. 掌握 C51 单片机 I/O 口的读写方法；
3. 掌握数码管动态显示原理。

二、实验平台

1. 计算机；
2. 单片机开发板、下载数据线；
3. 安装 KEIL 软件、Proteus 软件、STC 烧录软件。

三、实验要求

1. 熟悉流水灯高低电平控制 LED 灯亮灭原理、数码管动态显示原理；
2. 绘制流水灯、数码管电路图；
3. 编写 C 语言程序实现 I/O 接口读写功能；
4. 按照要求认真记录并规范如实填写实验报告，提交实验报告（电子版+打印版），电子版命名方式“学号+班级+姓名+实验三”。

四、实验内容和步骤

实验内容 1：使用 Proteus 完成电路图设计

1. 打开工程文件，添加数码管元器件“7SEG-MPX4-CA”。

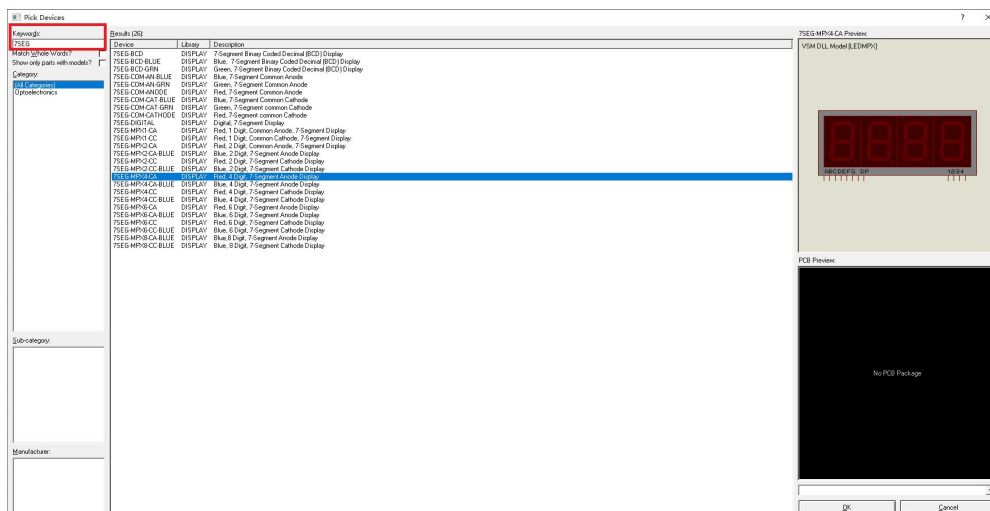


图 3-1 添加元器件数码管

2. 综合运用上节课知识，结合开发板电路原理图 3-2，绘制本次实验电路图 3-3。

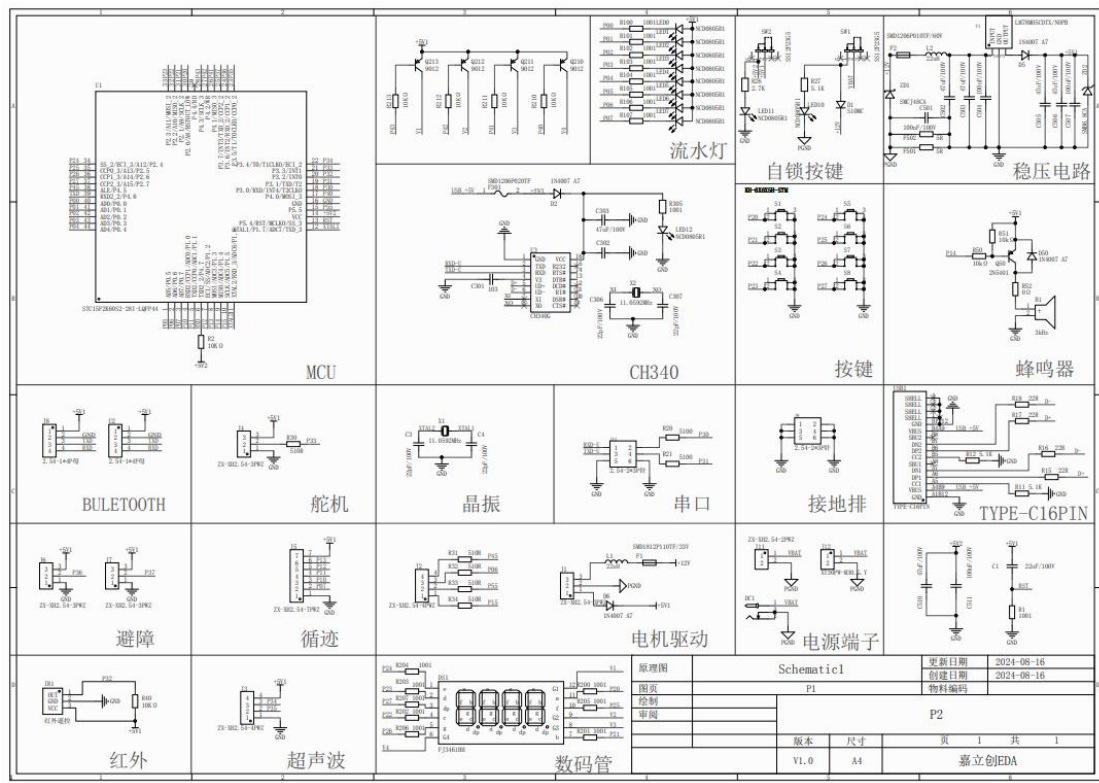


图 3-2 开发板电路原理图

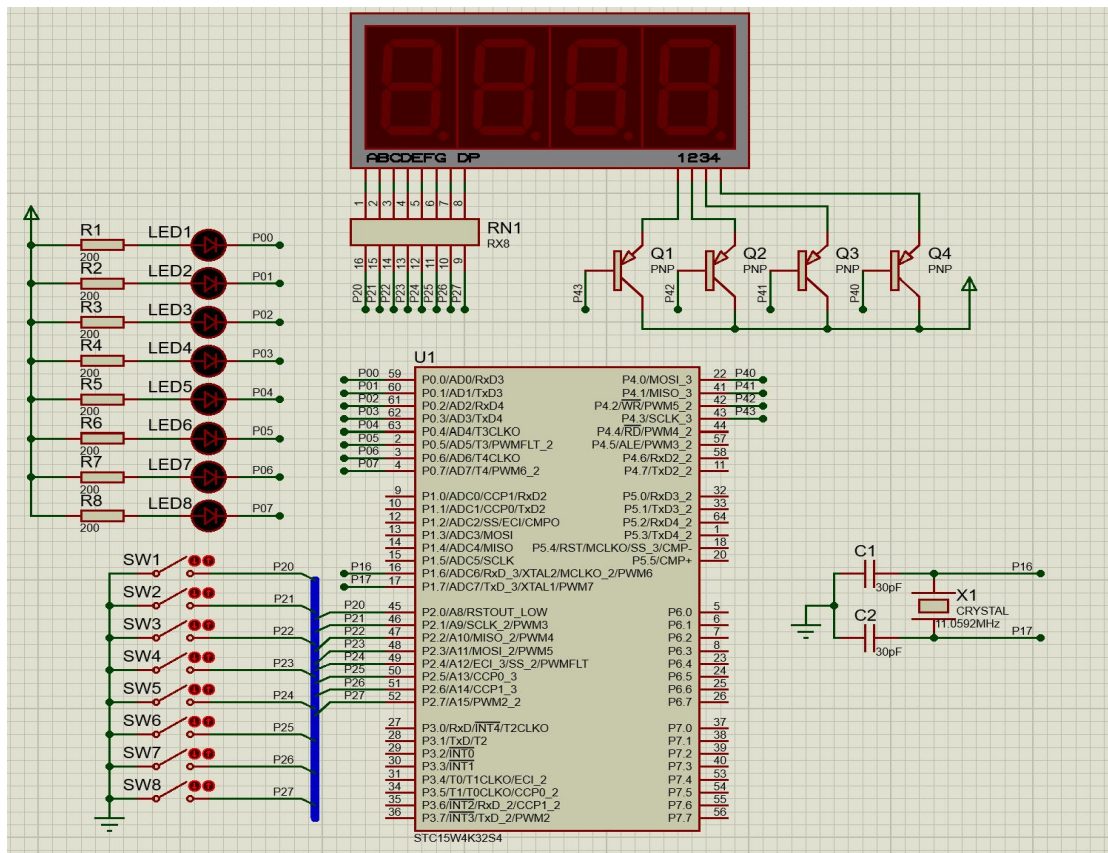


图 3-3 实验 3 电路图

实验内容 2：编程实现 I/O 口控制

1. KEIL 环境编写 C 语言程序实现：

- (1) 8 只发光二极管先整体闪烁 3 次；
- (2) 后根据开关状态控制相应灯亮或灯灭；
- (3) 在第一问的基础上改进功能，实现程序开始后高 4 位和低 4 位发光二极管先分别闪烁 3 次，然后根据开关状态反向控制相应灯亮或灯灭：即开关闭合相应灯灭，反之则灯亮。
- (4) 实现数码管显示“学号后 4 位”，要求从左向右显示，并设置时间消除抖动；
- (5) 拓展应用：用 STC15F2K60S2 开发板实现相应功能，如图 3-4 所示。

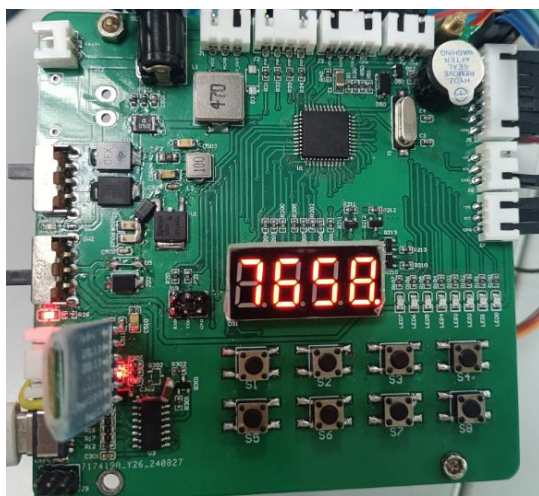


图 3-4 开发板实现数码管显示

(6) 下载程序步骤：

开发板使用数据 USB 线连接，USB 一端连接电脑另一端连接开发板，开发板状态如图 3-5 所示。

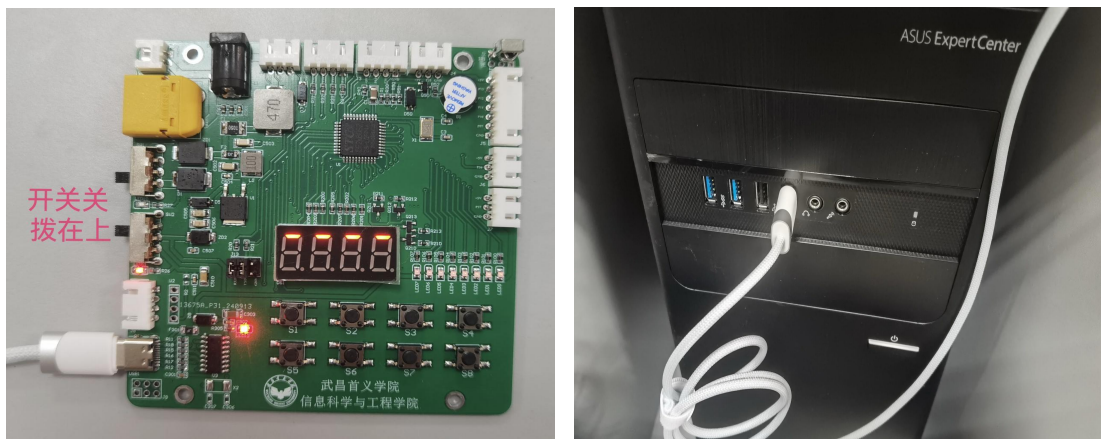


图 3-5 开发板连接

开发板 SW2 开关拨在上方，关闭单片机的供电。SW1 开关随意放置。

电脑端安装 USB 转串口驱动，安装完驱动在设备管理器有串口号，如下 3-6 所示。

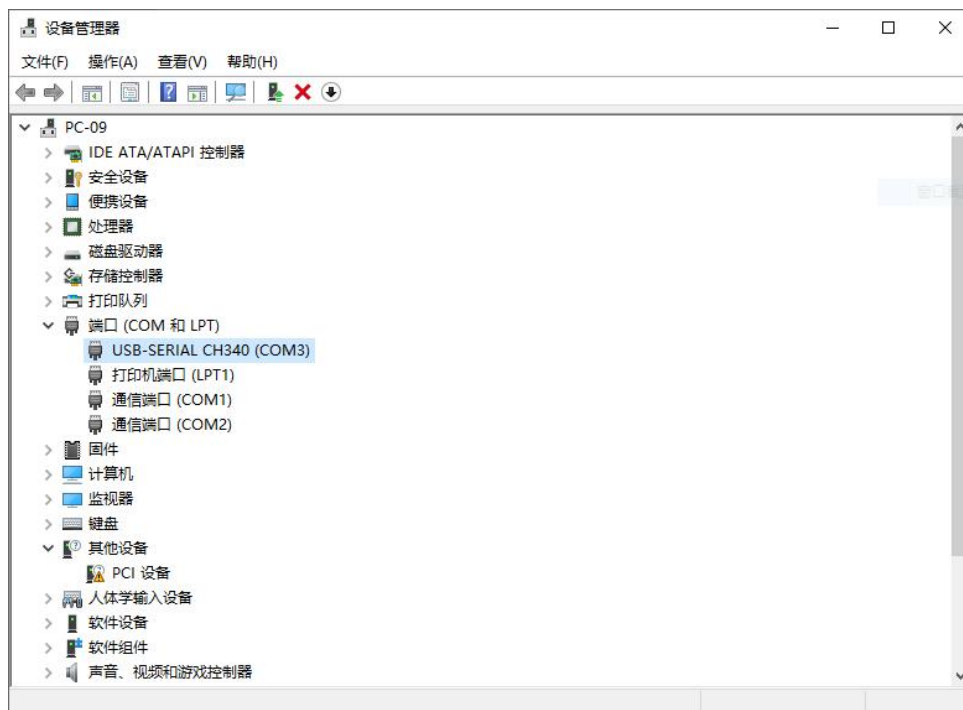


图 3-6 设备管理器查看驱动安装成功

打开 STC-ISP 软件，按照图 3-7 设置，注意串口号为之前驱动安装后的串口号（自动识别），点击“打开程序文件”选择要下载的程序，如图 3-8 所示。然后点击“下载/编程”按键，迅速将单片机电源开关打开，如图 3-10 所示。

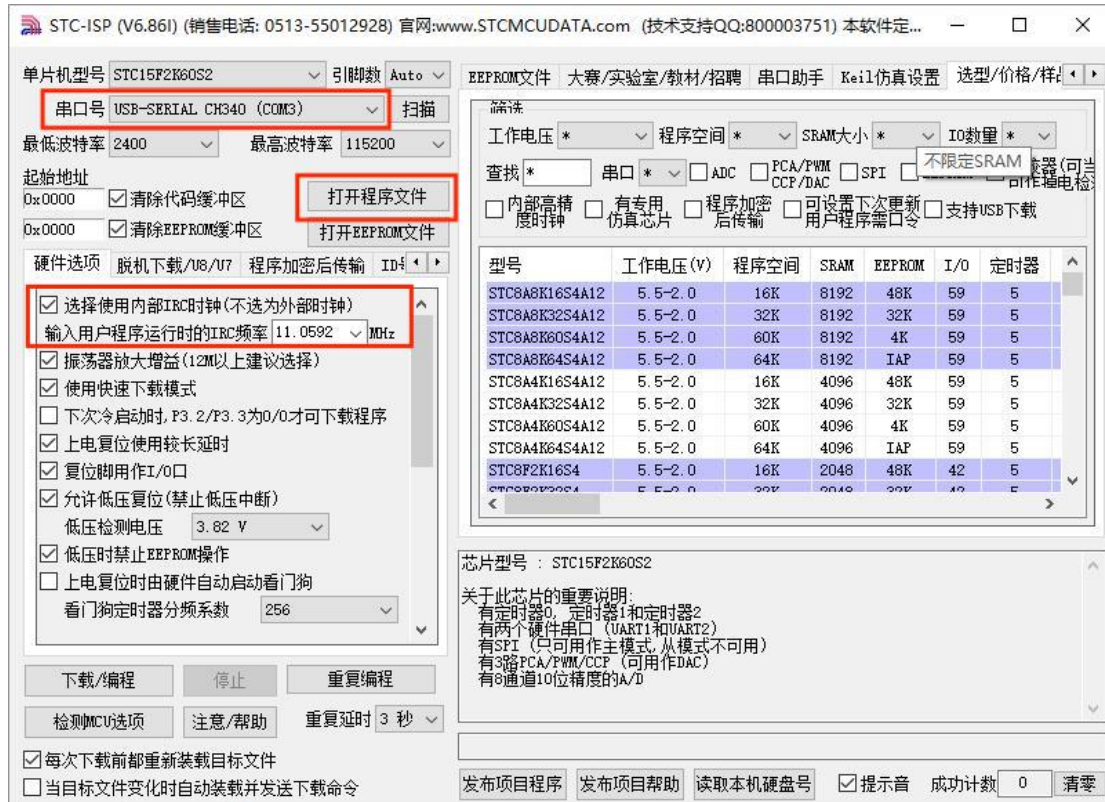


图 3-7 下载设置

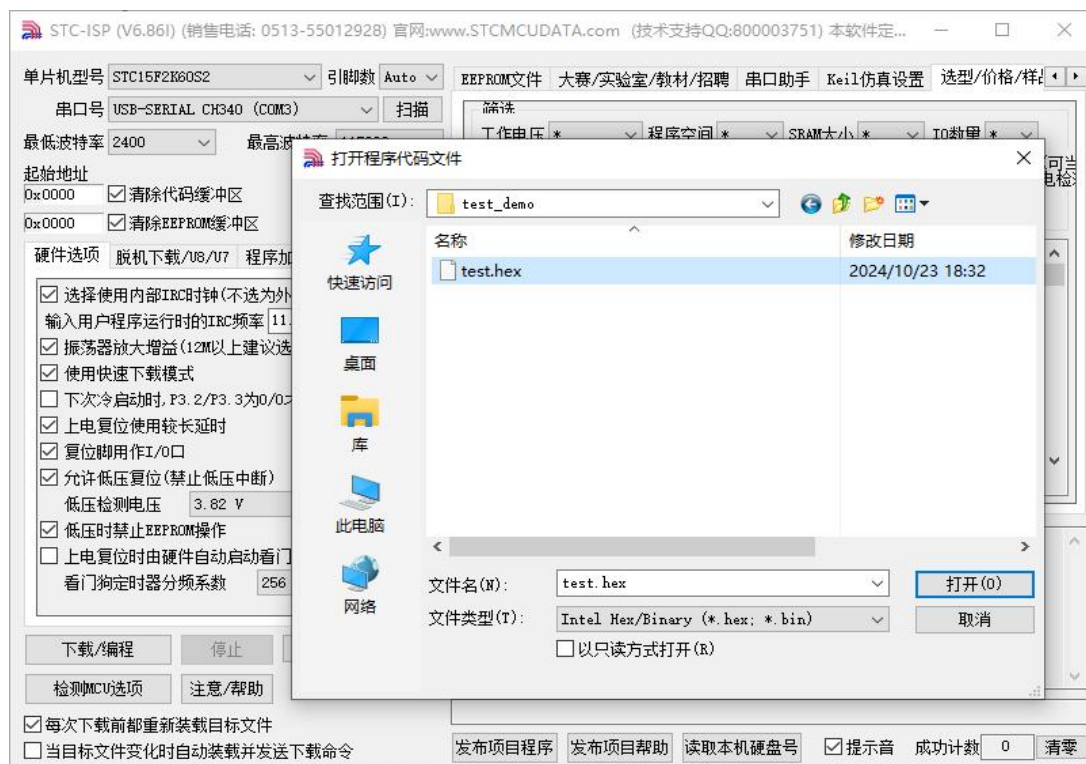


图 3-8 选择下载文件

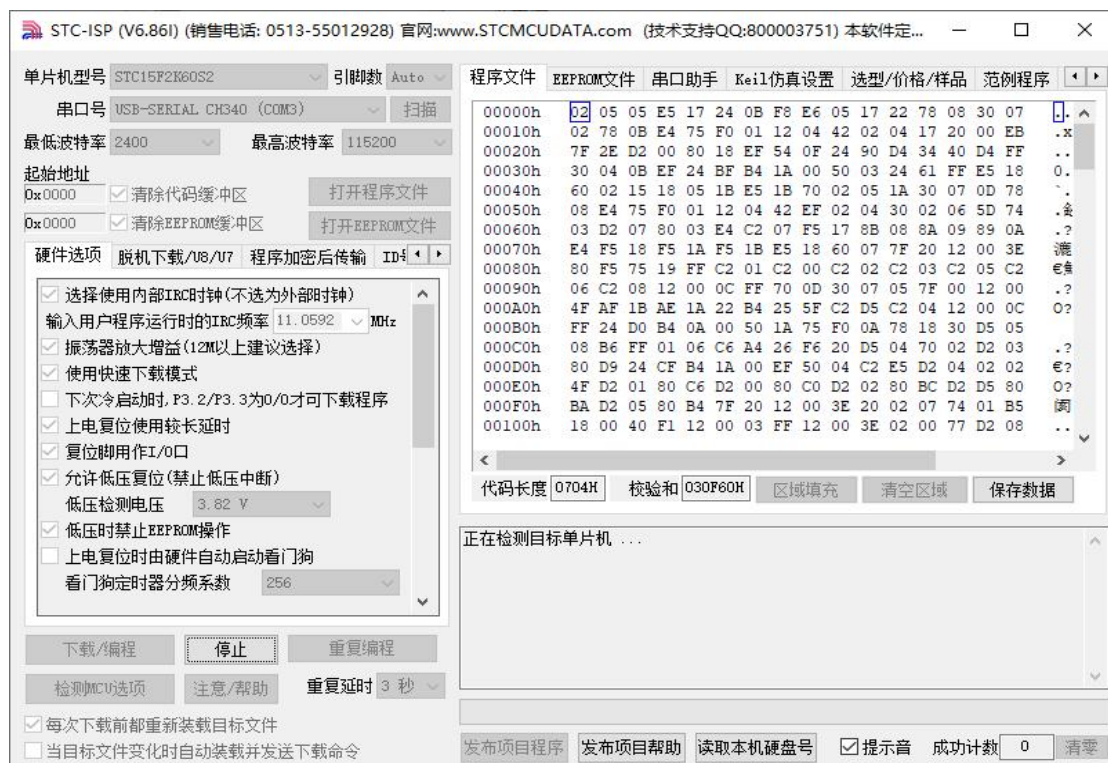


图 3-9 点击“下载/编程”按钮

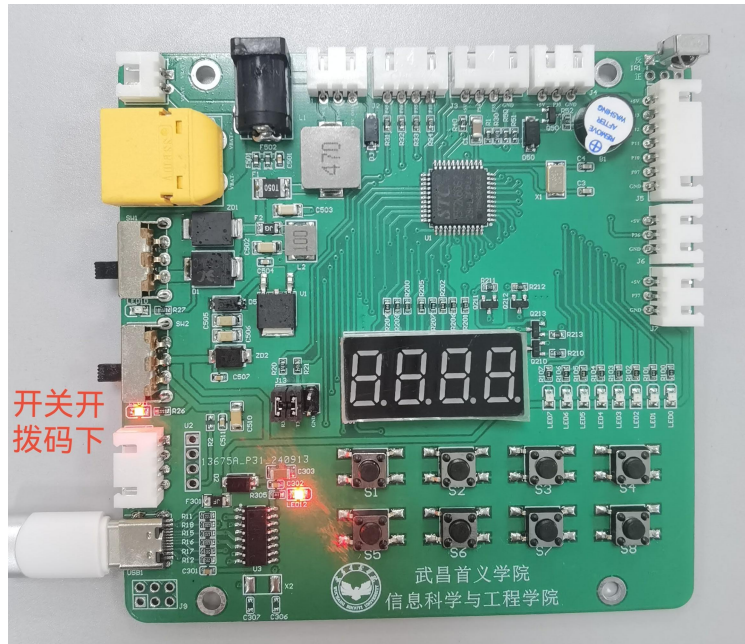


图 3-10 打开开发板电源开关

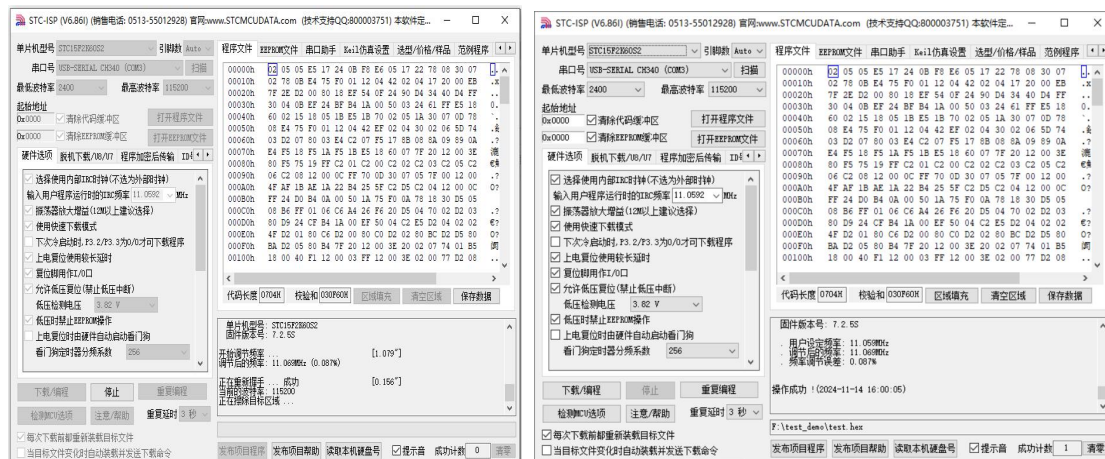


图 3-11 下载程序成功

五、实验总结与心得

(建议写自己遇到的问题及解决方法、实验总结的经验和心得体会，每个人不一样，做了拓展应用的同学可在实验报告里附效果图)

实验四 定时器应用实验

一、实验目的

1. 加深对定时器和中断概念的理解，掌握定时/计数器的工作原理；
2. 熟悉 51 单片机定时器的内部结构；
3. 熟悉定时/计数器查询方式编程与调试；
4. 熟悉定时/计数器中断法编程与调试。

二、实验平台

1. 计算机；
2. 单片机开发板、下载数据线；
3. 安装 KEIL 软件、Proteus 软件、STC 烧录软件。

三、实验要求

1. 复习定时器的原理、计数的方式、相关控制寄存器的使用；
2. 复习中断响应和中断处理的过程、中断函数定义、相关控制寄存器的使用；
3. 按照要求认真记录并规范如实填写实验报告（实验报告每人一份），提交实验报告（电子版+打印版），电子版命名方式“学号+班级+姓名+实验四”。

四、实验内容和步骤

实验内容 1：完成定时器定时功能

1. 打开 keil 软件，进入 Keil μ Vision5 集成开发环境，在 Keil 中新建工程。
2. 给出电路图原理图如 4-1 所示，利用 keil5 开发环境，编程实现下述功能。采用 12MHz 晶振，T0 定时方式 0，利用中断方式编程实现如下功能：程序运行开始时 LED 灯为熄灭状态，之后每隔 1 秒钟亮灭一次，1 分钟后为常亮状态。
3. 在 keil 软件中编译、调试。
4. 执行该程序，观察结果。
5. 完成实验报告，将编程分析（可以流程图、文字）和程序代码（需注释代码）撰写在报告中。

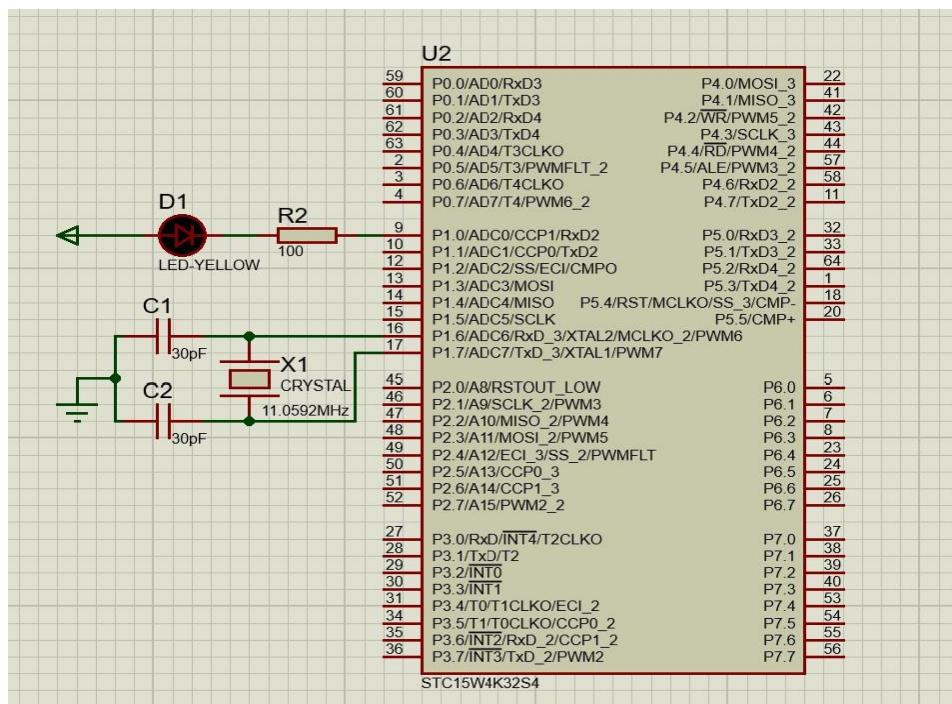


图 4-1 电路原理图

实验内容 2：完成秒表的设计

1. keil 软件完成秒表的程序编写，将程序下载在开发板中。
2. 给出电路图原理图如 4-2 所示。

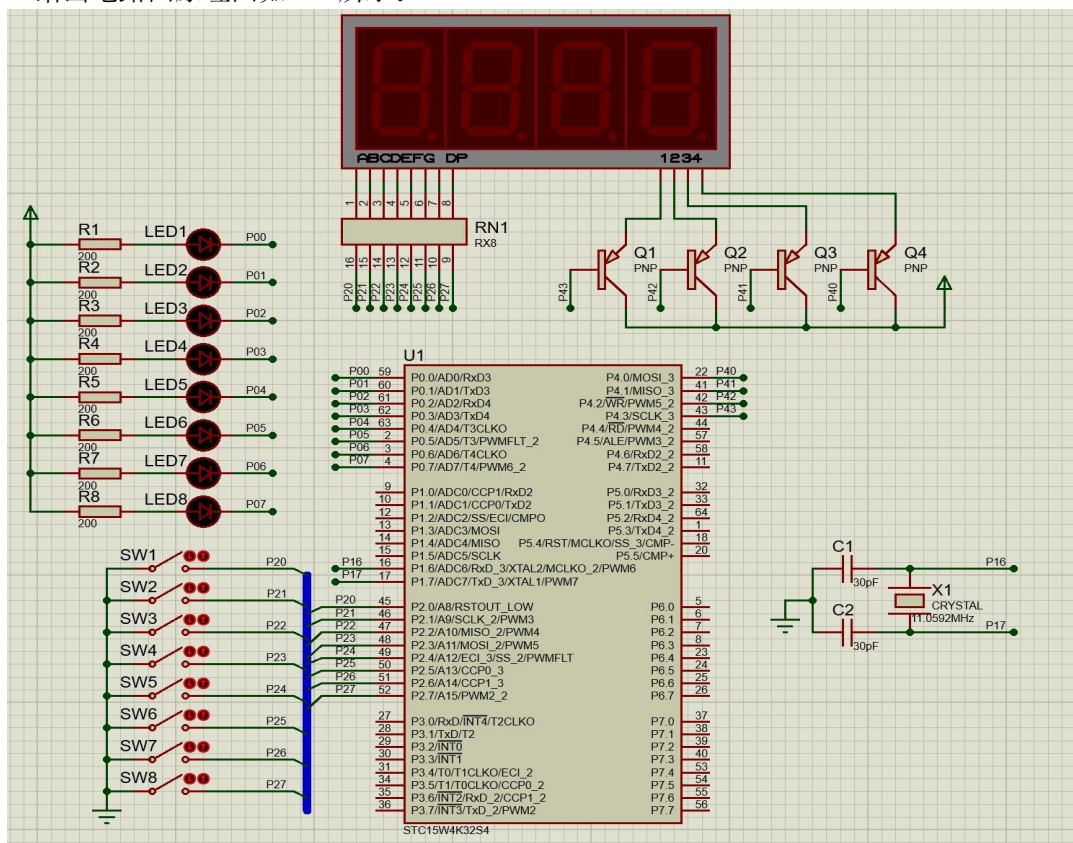


图 4-2 秒表设计

3. 具体功能如下:
 - (1) 按键 1 按下, 数码管的数字清零;
 - (2) 按键 2 按下, 数码管的数字开始秒表计时, 每一秒钟数字加 1;
 - (3) 按键 3 按下, 数码管的数字暂停秒表计时;
 - (4) 每 60 秒, 分针数加 1 (选做)。
4. 调试程序, 下载到开发板上, 观察实验结果;
5. 完成实验报告, 将编程分析 (可以流程图、文字)、程序代码 (需注释代码) 和实验结果 (开发板运行状态可以拍照) 撰写在报告中。

实验内容 3 (扩展功能) : 完成智能车的 PWM 调速实验

在开发板上完成以下功能: 编写程序实现小车车速控制。控制 L298N 电机驱动芯片, 调节控制管输出波形的占空比, 达到控制车速的目的。

五、实验总结与心得

(建议写自己遇到的问题及解决方法、实验总结的经验和心得体会, 每个人不一样)