**南京信息工程大学 实验（实习）报告**

实验（实习）名称 LCD的控制 日期 2024.6.5 得分 指导教师 谢胜东

学院 计算机学院 专业 计算机科学与技术 年级 2021班次 5 姓名 刘祥宇 学号 202183290006

一、实验目的

（1）理解LCD的基本原理：通过实验，了解液晶显示器（LCD）的基本工作原理，包括液晶分子的排列如何受到电场的影响而改变，从而实现显示效果。

（2）掌握LCD的驱动原理：学习LCD控制器的工作原理，包括如何通过发送特定的信号来控制像素的状态，以显示所需的图像或文本。

二、实验内容

本实验旨在使用Keil uVision3和Proteus 8 Professional工具控制LCD显示屏。在Keil uVision3中，编写并编译一个程序，使用C语言控制LCD显示特定字符或信息。在Proteus 8 Professional中，搭建电路仿真模型，包含微控制器、LCD模块及必要的连接。程序步骤包括初始化LCD、设置显示模式、清屏、显示字符和字符串等功能。通过在Proteus中运行仿真，验证LCD显示效果是否符合预期。

三、实验设备及工具

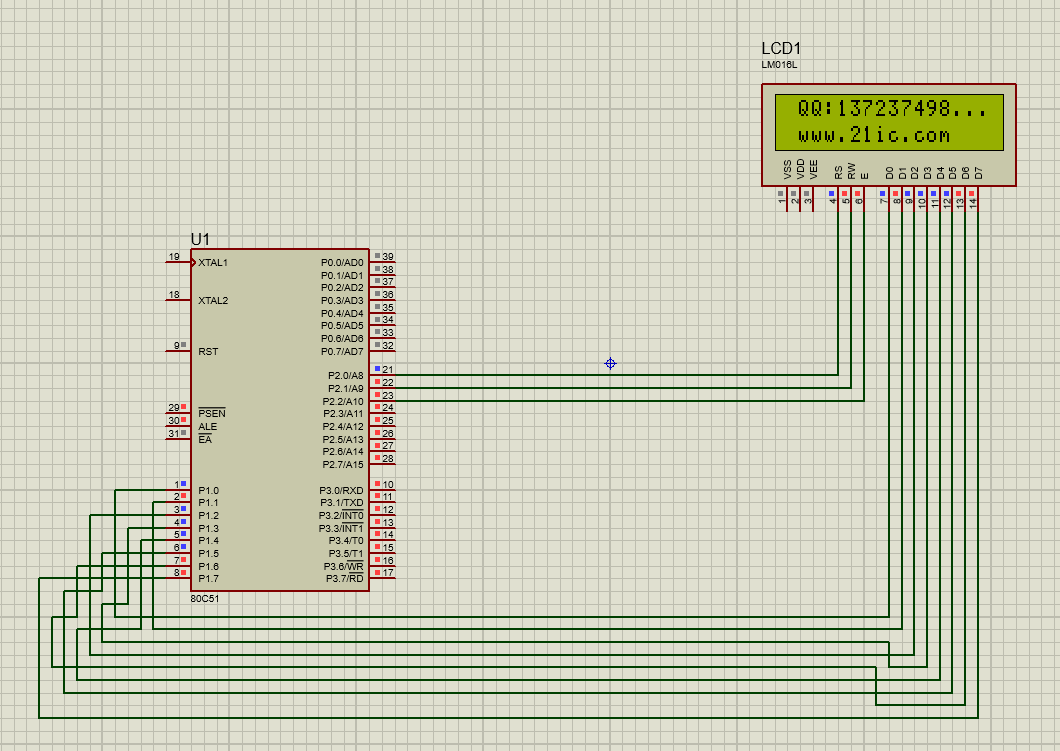
Keil uVision3（用于编写和调试程序）

Proteus 8 Professional（用于电路仿真）

计算机（Windows11系统，用于运行Keil uVision3和Proteus 8 Professional）

四、实验步骤

设计电路原理图



生成一个c文件并编写代码，编写完代码后进行编译，生成一个hex文件

具体代码如下：

#include <reg51.h>

#include <intrins.h>

#define uchar unsigned char

#define dataport P1

sbit RS = P2 ^ 0;

sbit RW = P2 ^ 1;

sbit EN = P2 ^ 2;

void delay(uchar t) {

uchar i;

for (; t > 0; t--) {

for (i = 50; i > 0; i--);

}

}

void write\_com(uchar com) {

RS = 0;

RW = 0;

dataport = com;

EN = 0;

EN = 1;

delay(1);

}

void write\_data(uchar dat) {

RS = 1;

RW = 0;

dataport = dat;

EN = 0;

EN = 1;

delay(1);

}

uchar read\_data() {

RS = 0;

RW = 1;

EN = 1;

\_nop\_();

return dataport;

}

bit read\_state() {

RS = 0;

RW = 1;

EN = 1;

\_nop\_();

return (bit)(dataport & 0x80);

}

uchar read\_char(uchar addr) {

while (read\_state());

write\_com(addr);

return read\_data();

}

void write\_str(char x, char y, uchar \*p) {

while (read\_state());

if (y) {

while (\*p) {

if (x == 16) break;

write\_com(0xc0 + x);

write\_data(\*p);

p++;

x++;

}

} else {

while (\*p) {

if (x == 16) break;

write\_com(0x80 + x);

write\_data(\*p);

x++;

p++;

}

}

}

void write\_char(char x, char y, uchar dat) {

while (read\_state());

if (y) {

if (x == 16) {

// Do nothing

} else {

write\_com(0xc0 + x);

write\_data(dat);

}

} else {

if (x == 16) {

// Do nothing

} else {

write\_com(0x80 + x);

write\_data(dat);

}

}

}

void LCD\_CLEAR() {

while (read\_state());

write\_com(0x01);

}

void init() {

write\_com(0x38);

delay(10);

write\_com(0x38);

delay(10);

write\_com(0x38);

delay(10);

write\_com(0x0c);

write\_com(0x06);

}

void main() {

uchar \*p1 = "202183290006";

uchar \*p2 = "1234"; // Assuming the LCD supports these characters

init();

write\_str(0, 0, p1); // Write "202183290192" to the first row starting at column 0

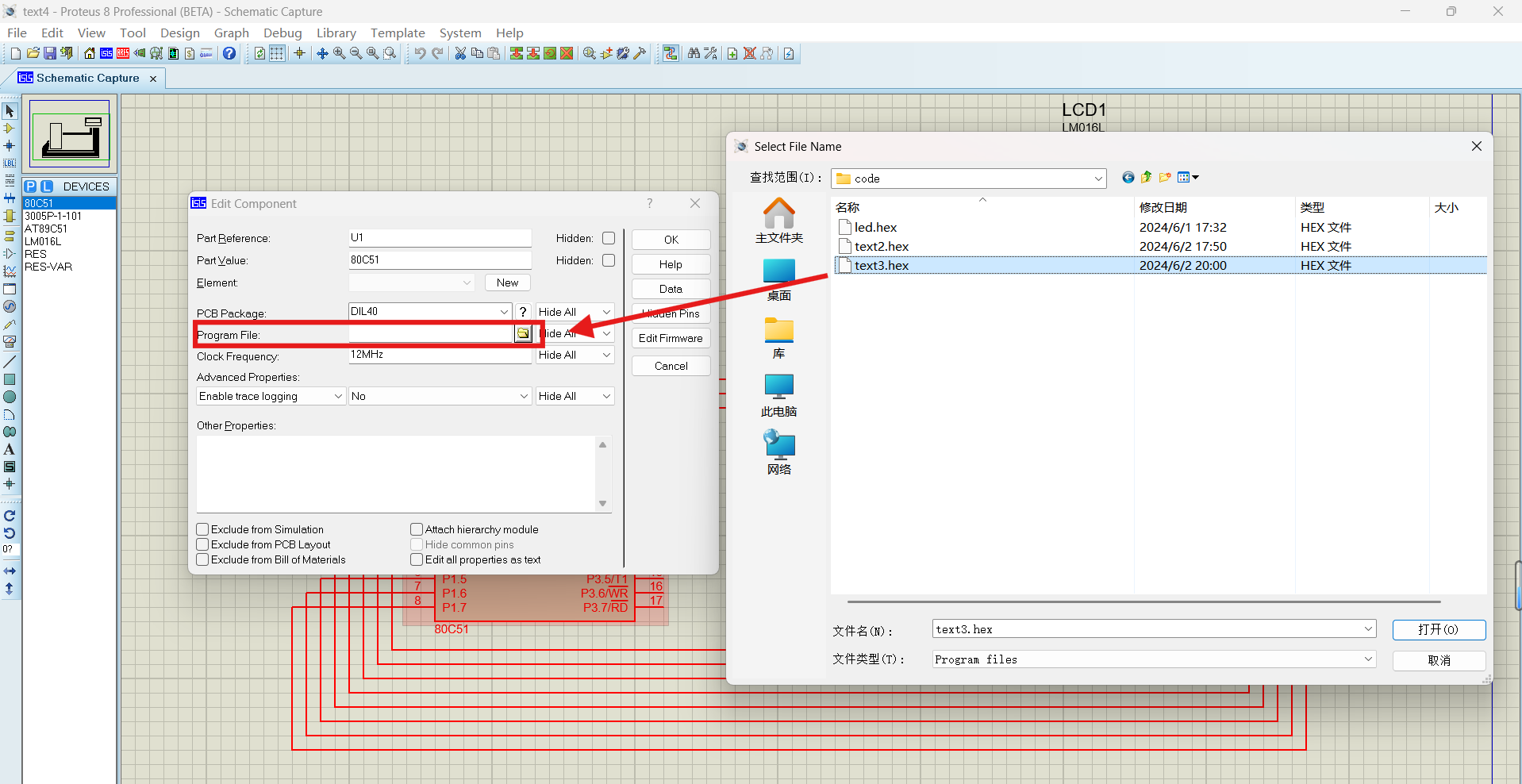
write\_str(0, 1, p2); // Write "石泽华" to the second row starting at column 0

while(1); // Keep running the program

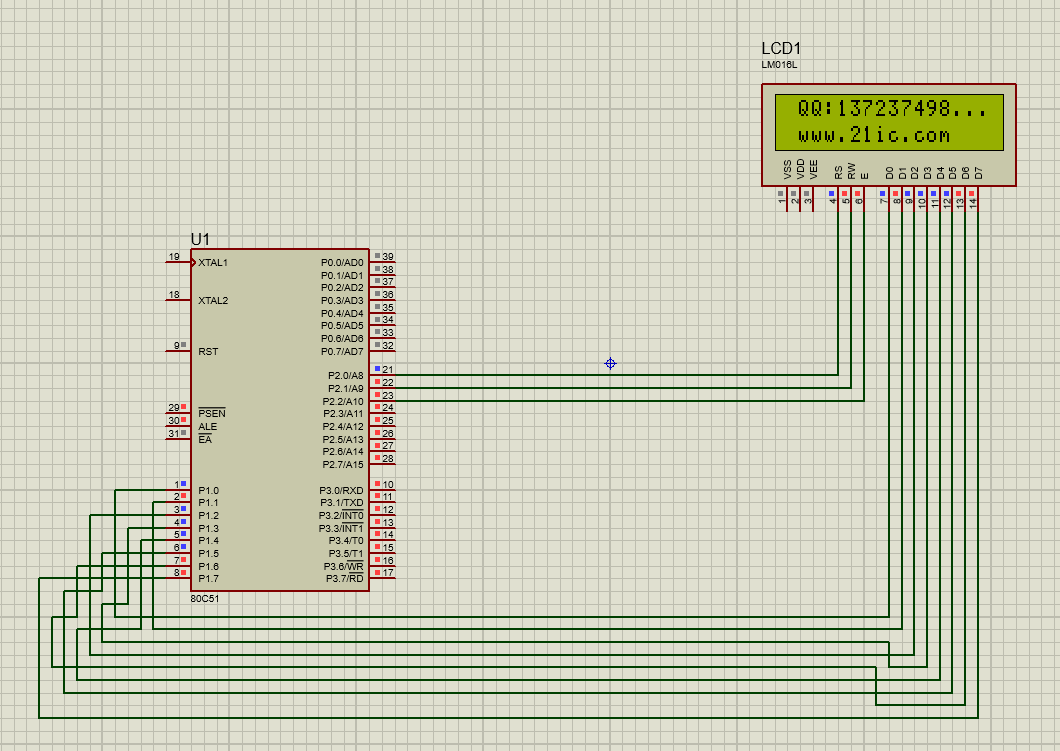
}

回到proteus中，在51单片机上导入生成的hex文件，并进行仿真

1. 导入hex文件



1. 进行仿真



从上图可知，按钮可以成功的控制，仿真效果完美实现，至此，此次实验成功完成。

五、实验总结

通过此次实验，使用Keil uVision3和Proteus 8 Professional工具对LCD显示屏进行了基本控制，掌握了LCD初始化、字符显示和字符串显示等操作。在实验中，首先编写了初始化函数，将LCD设置为8位数据接口、两行显示和5x7点阵模式。然后，通过延时函数确保指令和数据的写入有足够的时间间隔。通过写命令和写数据函数，分别实现了对LCD命令寄存器和数据寄存器的操作。读取LCD状态和数据的函数用于检测LCD是否忙碌并获取显示的数据。在显示字符和字符串的过程中，使用了两个函数，一个用于在指定位置显示字符串，另一个用于显示单个字符。主函数中，首先对LCD进行初始化，然后在第一行显示字符串"202183290192"，在第二行显示字符串"shizehua"。通过Proteus仿真，验证了LCD能够正确显示预期内容。本实验成功地实现了对LCD的基本控制，增强了嵌入式系统编程和硬件调试的理解和实践能力。