

合肥工业大学

计算机与信息类创新实践

实践报告

开 课 学 院 _____

专 业 班 级 _____

学 生 姓 名 及 学 号 _____

指 导 教 师 _____

2023 年 4 月 29 日

目录

实验一 以多种方式点亮 led 灯	4
1.实验目的	4
2.基本原理	4
2.1 led 闪烁原理	4
2.2 矩阵键盘工作原理	5
2.3 定时器中断原理	5
3.实验步骤	6
4.实验结果	7
实验二 电机控制	10
1.实验目的	10
2.基本原理	10
2.1 直流电机驱动控制原理	10
2.2 PWM 基本原理	11
3.实验步骤	11
4.实验结果	12
实验三 用显示屏显示自己的学号	14
1.实验目的	14
2.基本原理	14
2.1 TFTLCD 屏显示原理	14
3.实验步骤	15
4.实验结果	16
实验四 通过 A/D 采集模拟量的输入并显示	18
1.实验目的	18
2.基本原理	18
2.1 温湿度传感器原理	18
2.2 超声波模块 HC-SR04 测距原理	19
2.3 A/D 转换器原理	20
3.实验步骤	20
4.实验结果	21
实验五 温湿度报警与键盘设置	22
1.实验目的	22
2.基本原理	22
2.1 蜂鸣器工作原理	22
3.实验步骤	23
4.实验结果	24
考核及成绩	26

实验一 以多种方式点亮 led 灯

1.实验目的

- (1) 理解并掌握 led 灯的点亮原理
- (2) 理解并掌握单片机位操作与端口操作的方法
- (3) 掌握 4*4 矩阵键盘的工作原理
- (4) 理解并掌握单片机定时器中断的原理与使用方法

2.基本原理

2.1 led 闪烁原理

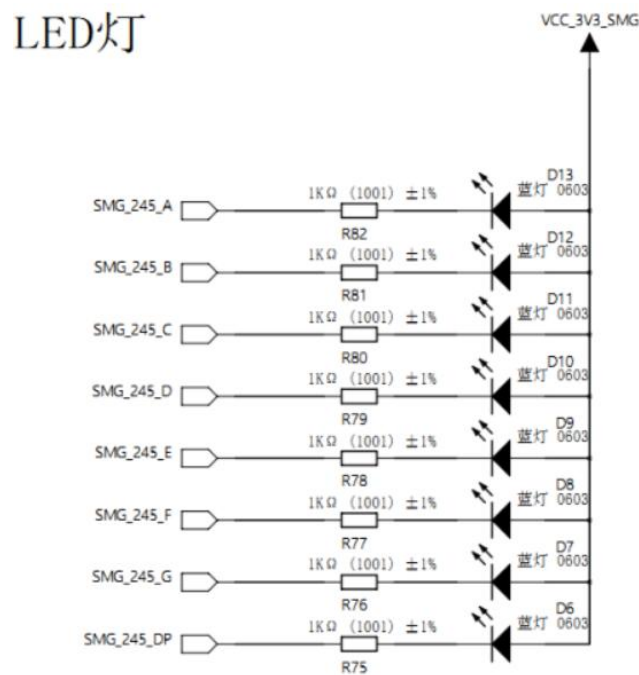


图 1 场景功能板 LED 电路原理图

场景功能板上的 IO 口电路图如图 1 所示，SMG_245_A 到 SMG_245_DP 连接单片机的 IO 口。

当单片机引脚为低电平时 LED 被点亮，高电平时 LED 灯熄灭。LED 点亮一段时间后，熄灭即可实现闪烁功能。

点亮 LED，需要初始化 IO 口，这里将 P0 到 P7 IO 口全部初始化为准双向 IO 口，弱上拉模式。

2.2 矩阵键盘工作原理

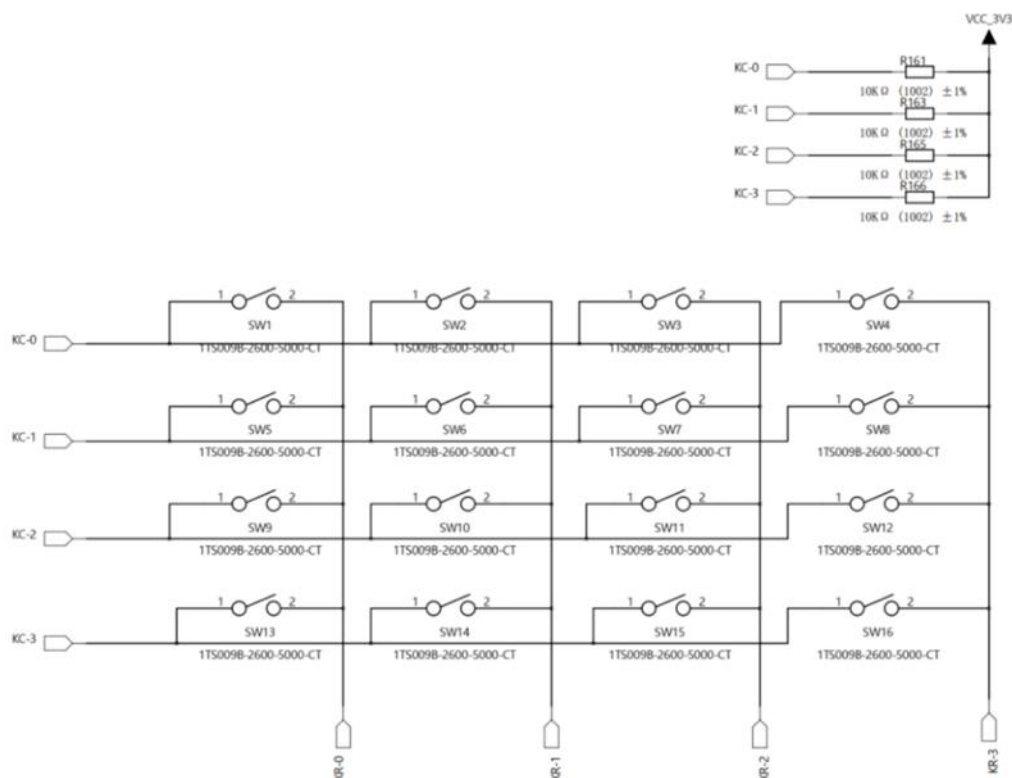


图 2 4*4 矩阵键盘电路原理图

矩阵键盘的电路原理图如图 2 所示，基本原理为，依次令 KR-0、KR-1、KR-2、KR-3 为低电平，然后依次读取 KC-0、KC-1、KC-2、KC-3 的值，当某一列为低电平时（如 KR-0），若读取到某一行低电平（如 KC-1），则表明该行与该列相交的那一点的按键被按下了（SW5）。当反复进行上述工作时，便完成了矩阵键盘的扫描。矩阵键盘的行列式结构能够有效提高单片机系统中的 IO 口利用率，节约单片机 IO 资源。

2.3 定时器中断原理

STC15W4K32S4 系列单片机内部设置了 5 个 16 位定时器/计数器:16 位定时器/计数器 T0 和 T1、T2、T3 以及 T4。5 个 16 位定时器 T0、T1、T2、T3 和 T4 都具有计数方式和定时方式两种工作方式。

定时器/计数器的核心部件是一个加法计数器，其本质是对脉冲进行计数。只是计数脉冲来源不同:如果计数脉冲来自系统时钟，则为定时方式，此时定时器/计数器每 12 个时钟或者每 1 个时钟得到一个计数脉冲，计数值加 1;如果计数脉冲来自单片机外部引脚(T0 为 P3.4，T1 为 P3.5，T2 为 P3.1，T3 为 P0.7，T4 为 P0.5)，则为计数方式，每来一个脉冲加 1。

中断系统是为使 CPU 具有对外界紧急事件的实时处理能力而设置的。

51 单片机定时器中断是指在 51 单片机中，定时器计数到一定值后，会产生一个中断信号，从而触发中断程序的执行。

本实验使用定时器 0 作为自动重装的 16 位计数器，其原理图如图 3 所示：

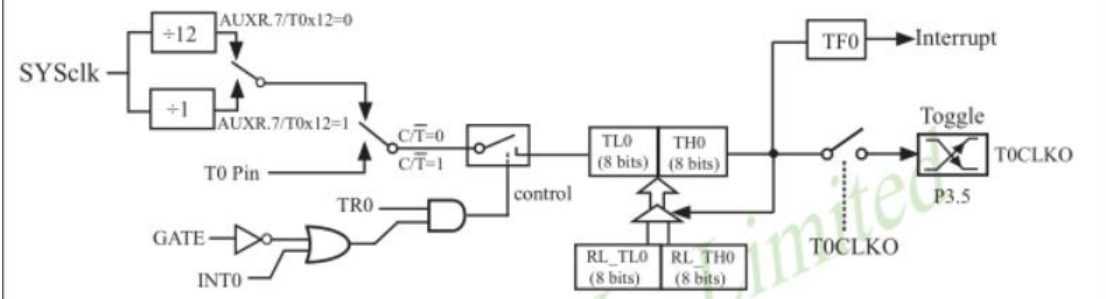


图 3 定时器 0 的模式 0:16 位自动重装内部原理图

3.实验步骤

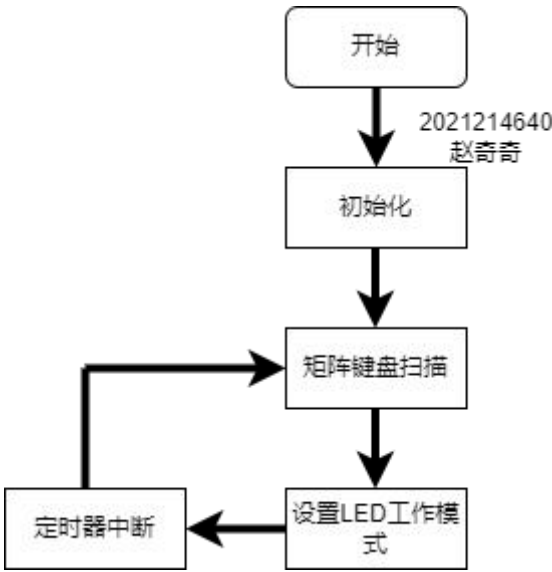


图 4 程序流程图

本实验程序流程图如图 4 所示，首先进行初始化，包括 IO 口初始化、定时器 0 初始化、矩阵键盘初始化等，之后矩阵键盘扫描，判断所按键，根据所按键来设置 LED 的工作模式设置定时器 0 中断时间为 5ms，并且每 5ms 扫描一次矩阵键盘，以此来提高按键判断的灵敏性与准确性。

本实验程序主函数如图 5 所示。

```
//主函数2021214640
void main(void)
{
    IO_init(); //IO口初始化
    delaylms(10);
    KeyCode = 0; //给用户使用的键码, 1~16有效
    IO_KeyState = 0;
    IO_KeyState1 = 0;
    IO_KeyHoldCnt = 0;

    while(1);
}
```

图 5 程序主函数图

完成代码编写后生成 HEX 文件，并使用 STC-ISP 将该 HEX 文件烧录到单片机中。烧录之前要设置 STC-ISP，具体设置如图 6 所示：

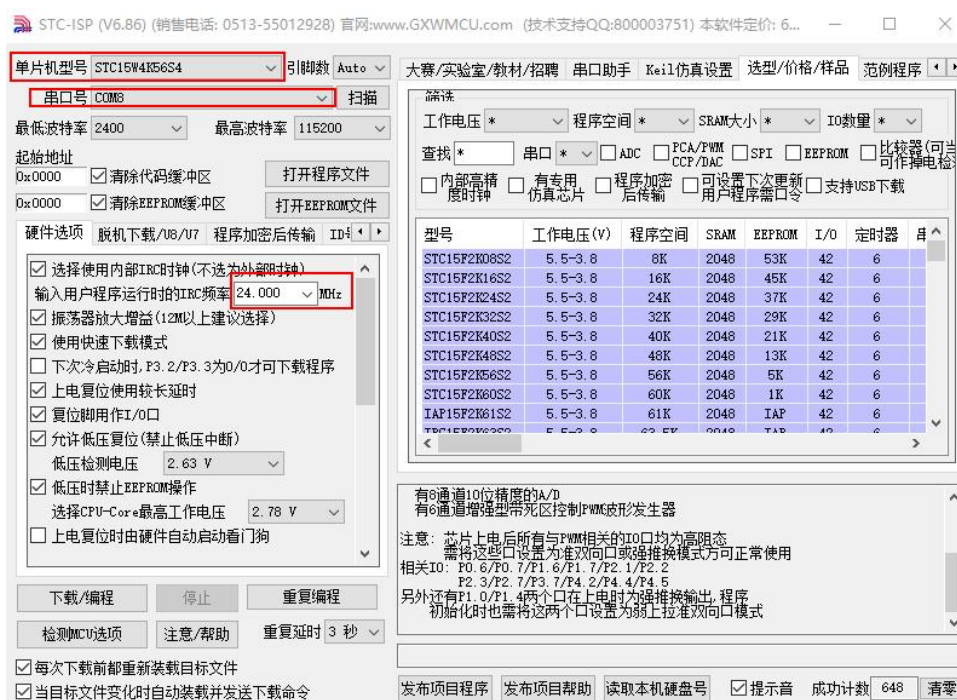


图 6 STC-ISP 设置图

4.实验结果

按下按键 1 所有 LED 同步闪烁，如图 7 所示：

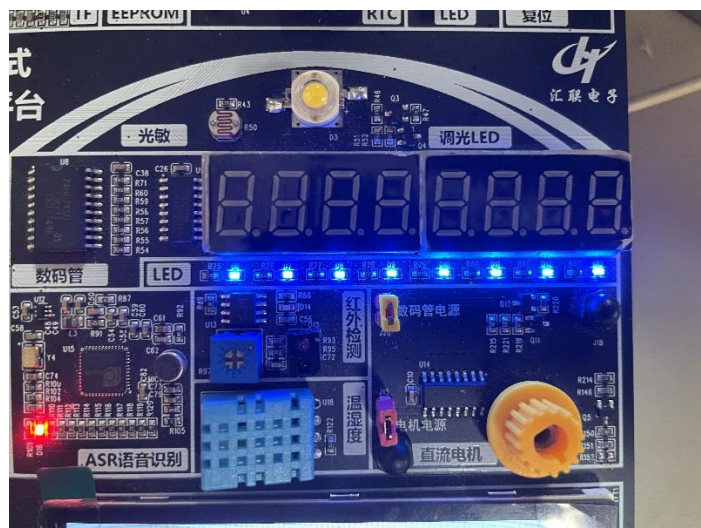


图 7 LED 同步闪烁图

按下按键 2 依次点亮 LED，如图 8 所示：

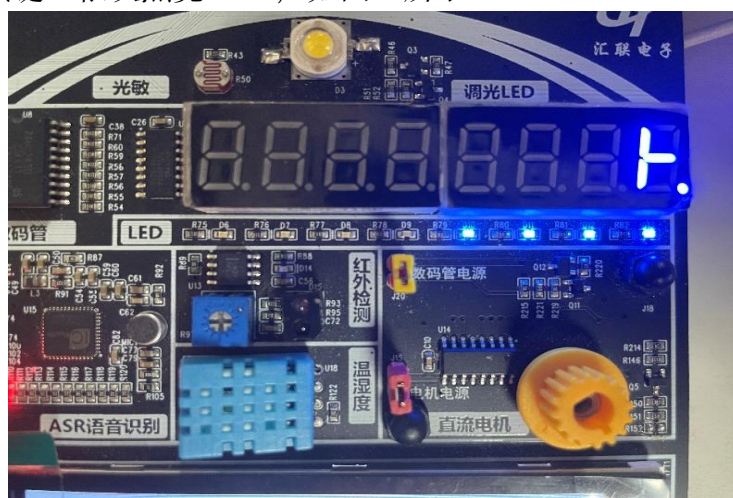


图 8 LED 依次闪烁图

按下按键 3 依次点亮 LED，如图 9 所示：

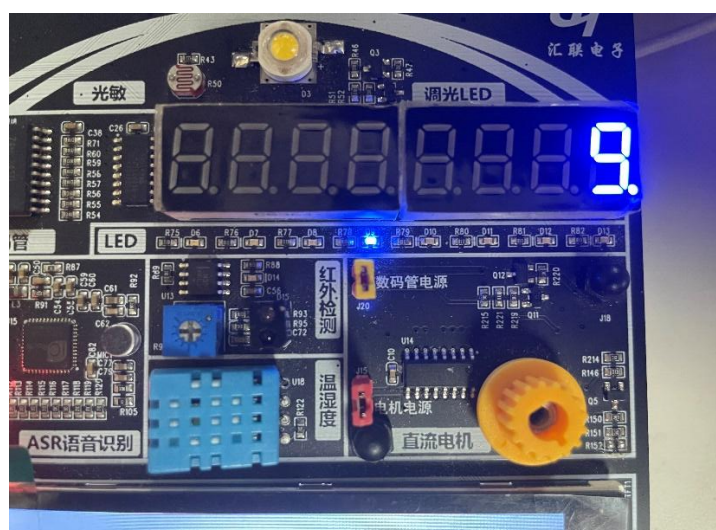


图 9 LED 依次闪烁图

按下按键 4 后 LED 从两侧向中间循环点亮，如图 10 所示：

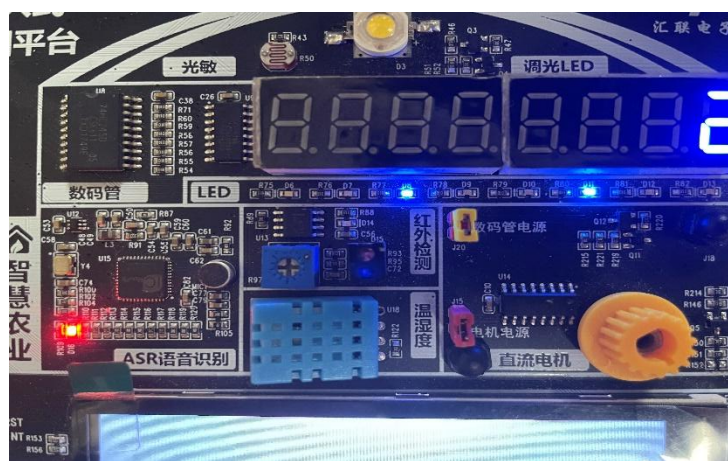


图 10 LED 从两侧向中间循环点亮图

实验二 电机控制

1.实验目的

- (1) 掌握直流电机驱动控制原理
- (2) 掌握 PWM 基本原理内容
- (3) 掌握 4*4 矩阵键盘的工作原理
- (4) 掌握单片机定时器中断的原理与使用方法

2.基本原理

4*4 矩阵键盘与定时器中断的原理及使用方法在实验一已经进行了详细地叙述，所以此处就不再赘述。

2.1 直流电机驱动控制原理

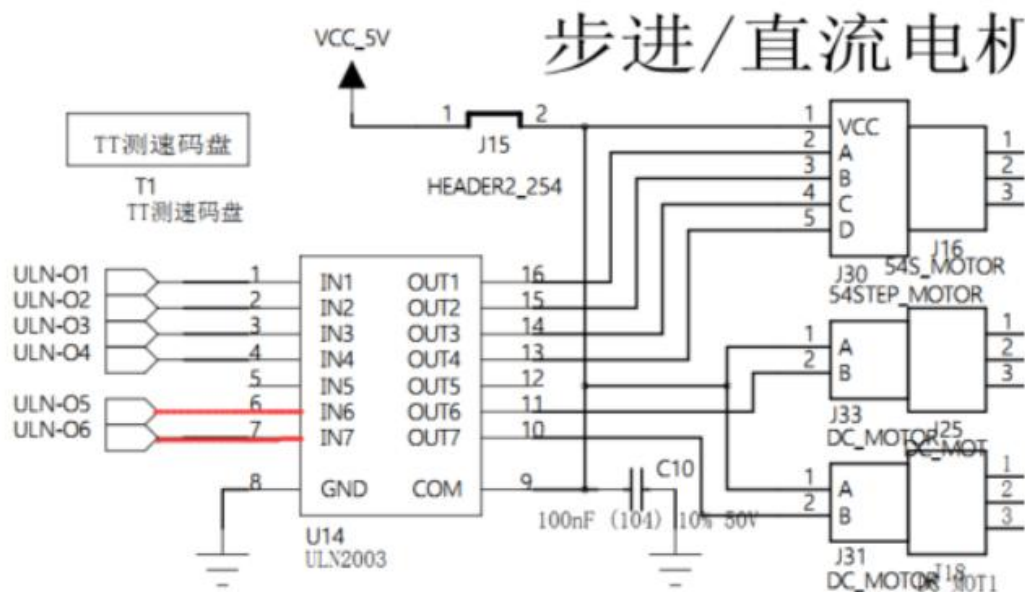


图 1 直流电机驱动原理图

直流电机通过 ULN2003 芯片驱动，由图 1 原理图中可以得到直流电机的控制引脚分别为 ULN-O5 ULN-O6。

如上图，ULN-O5 在选择“智能小车”场景时，模拟开关切换将其会连接至 P16 ULN-O6 在选择“智慧农业”场景时，模拟开关切换将其会连接至 P15。

因此在本实验中，首先要确认场景，之后可以通过将 P15 或者 P16 置 1 或者置 0 开控制直流电机的状态。

2.2 PWM 基本原理

PWM (Pulse Width Modulation) 简称脉宽调制, 是利用微处理器的数字输出来对模拟电路进行控制的一种非常有效的技术, 广泛应用在测量、通信、工控等方面。

本实验中使用定时器 0 模拟 PWM, 首先确定 PWM 的周期 T 和占空比 D , 确定了这些以后, 就可以用定时器产生一个时间基准 t , 比如定时器溢出 n 次的时间是 PWM 的高电平的时间, 则 $D \cdot T = n \cdot t$, 类似的可以求出 PWM 低电平时间需要多少个时间基准 n 。

因为这里我们是产生周期为 1ms (1kHz) 的 PWM, 所以可设置中断的时间基准为 0.01ms , 然后中断 100 次即为 1ms 。在中断子程序内, 可设置一个变量如 time , 在中断子程序内, 有三条重要的语句:

- (1) 当 $\text{time} \geq 100$ 时, time 清零(此语句保证频率为 1kHz);
- (2) 当 $\text{time} > n$ 时(n 应该在 $0 - 100$ 之间变化开), 让单片相应的 I/O 口输出低电平;
- (3) 当 $\text{time} \leq n$ 时, 让单片相应的 I/O 口输出高电平, 此时占空比就为 $\%n$ 。

通过不同占空比的 PWM 来控制直流电机的转速。

3. 实验步骤

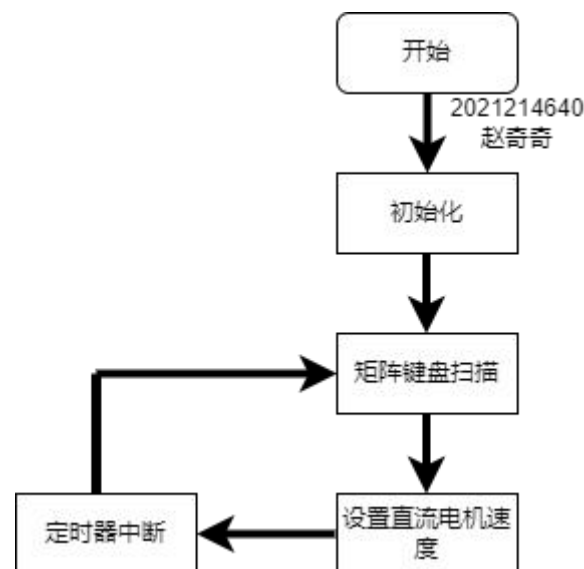


图 2 程序流程图

本实验程序流程图如图 2 所示, 首先进行初始化, 包括 IO 口初始化、定时器 0 初始化、矩阵键盘初始化、直流电机初始化等。特别的, 直流电机初始化需要将对应 IO 口设置为推挽输出状态。

之后矩阵键盘扫描，判断所按键，根据所按键来设置直流电机的工作模式，设置定时器 1 中断时间为 5ms，并且每 5ms 扫描一次矩阵键盘，以此来提高按键判断的灵敏性与准确性。

利用定时器 0 产生 PWM，以此来实现直流电机的加速与减速功能。

程序主函数如 3 所示。

```
//主函数2021214640
void main(void)
{
    IO_init(); //IO口初始化
    UartInit(); //串口初始化
    KeyCode = 0; //给用户使用的键码，1~16有效
    IO_KeyState = 0;
    IO_KeyStatel = 0;
    IO_KeyHoldCnt = 0;

    //设置STC 单片机的P15 P16为推挽输出
    P1M1 &= ~(1<<5), P1M0 |= (1<<5);
    P1M1 &= ~(1<<6), P1M0 |= (1<<6);

    while(1)
    {
        motol=0;
        moto2=0;
        IO_KeyScan();
        if(KeyCode==1){
            while(1){
                motol=1;moto2=1;
                delay1ms(2);
                motol=0;moto2=0;
                delay1ms(1);
                IO_KeyScan();
                if(KeyCode!=1) break;
            }
        }
        if(KeyCode==2){
            while(1){
                motol=0;
                moto2=0;
                IO_KeyScan();
                if(KeyCode!=2) break;
            }
        }
        if(KeyCode==3){ //加速
            while(1){
                motol=1;moto2=1;
                IO_KeyScan();
                if(KeyCode!=3) break;
            }
        }
        if(KeyCode==4){ //减速
            motol=1;moto2=1;
            delay1ms(1);
            motol=0;moto2=0;
            delay1ms(1);
            IO_KeyScan();
            if(KeyCode!=4) break;
        }
    }
}
```

图 3 程序主函数

完成代码编写后生成 HEX 文件，并使用 STC-ISP 将该 HEX 文件烧录到单片机中。

观察实验结果，并拍照记录。

4.实验结果

按下按键 1 控制直流电机打开，如图 4 所示。



图 4 直流电机打开示意图

按下按键 2 控制直流电机关闭，如图 5 所示。

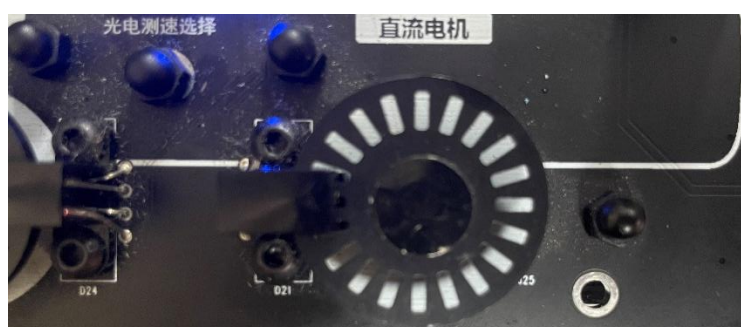


图 5 直流电机关闭示意图

按下按键 3 控制直流电机加速，如图 6 所示。



图 6 直流电机加速示意图

按下按键 4 控制直流电机减速，如图 7 所示。



图 7 直流电机减速示意图

实验三 用显示屏显示自己的学号

1.实验目的

- (1) 掌握 TFT LCD 屏的显示
- (2) 掌握 4*4 矩阵键盘的工作原理
- (3) 掌握单片机定时器中断的原理与使用方法

2.基本原理

4*4 矩阵键盘与定时器中断的原理及使用方法在实验一已经进行了详细地叙述，所以此处就不再赘述。

2.1 TFTLCD 屏显示原理

TFTLCD 是一种液晶显示屏，它的显示原理是利用薄膜晶体管来驱动每一个像素点，这样可以提高显示屏的响应速度，同时可以精确控制显示色阶，所以 TFT 液晶的色彩更逼真，其电路原理图如图 1 所示。

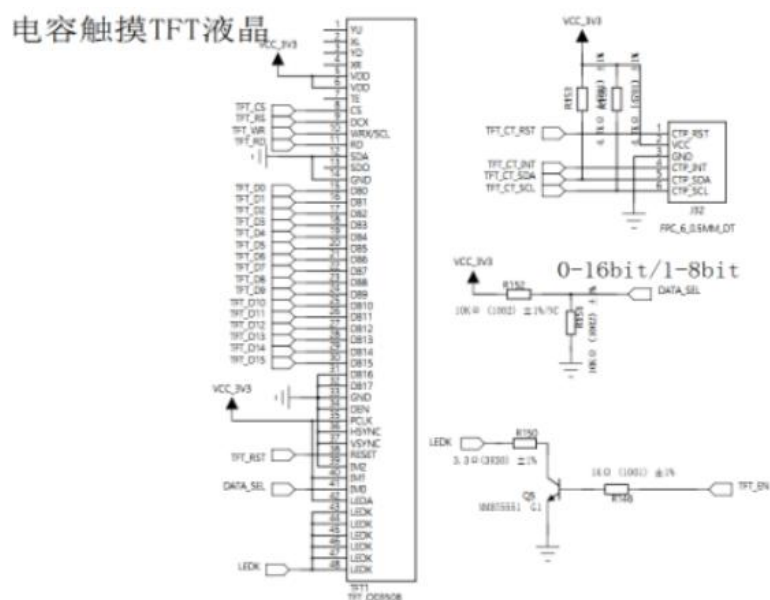


图 1 TFTLCD 电路原理图

TFTLCD 屏信号线介绍:

CS : TFT 片选信号。WR : 向 TFT 写入数据。RD : 从 TFT 读取数据。

RS : 命令/数据标志 (0, 读写命令; 1, 读写数据)。

DB[15 : 0] : 16 位双向数据线。

RESET：硬复位 TFT。

事实上，在本实验中，只需要用到相关函数 LCD_ShowNum(u16 x,u16 y,u32 num,u8 len,u8 size)，其中 x 代表横坐标，y 代表纵坐标，num 代表待显示的数字，len 代表待显示数字的长度，size 代表待显示数字的大小。

3.实验步骤

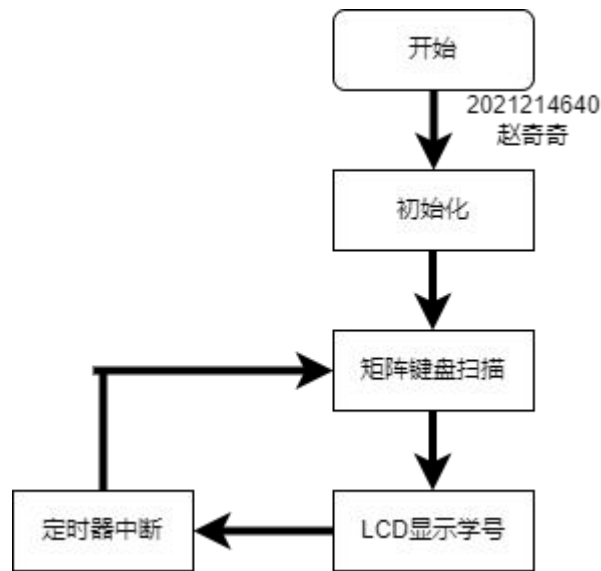


图 2 实验流程图

本实验程序流程图如图 2 所示，首先进行初始化，包括 IO 口初始化、定时器 0 初始化、矩阵键盘初始化、LCD 初始化等。

其次矩阵键盘扫描，判断被按下的按键，根据按键的不同，LCD 通过不同的方式显示学号，本实验用到了定时器 0，以实现矩阵键盘的定时扫描，这种方式可以提高矩阵键盘扫描的准确性与及时性。

程序部分主函数如图 3 所示。

```

//主函数2021214640
void main(void)
{
    IO_init(); //IO口初始化
    LCD_Init(); //液晶屏初始化
    delayms(10);
    LCD_Clear(WHITE); //清屏白底
    Timer0Init();

    KeyCode = 0; //给用户使用的键码, 1~16有效
    IO_KeyState = 0;
    IO_KeyState1 = 0;
    IO_KeyHoldCnt = 0;

    while(1)
    {
        if (KeyCode==1)
        {
            LCD_Clear(WHITE);
            for(i=0;i<11;i++){
                for(j=0;j<i;j++){
                    LCD_ShowNum(10*(i-j), 0, a[j], 1, 16);
                }
                if (KeyCode!=1) {
                    break;
                }
                delayms(500);
            }
            while(KeyCode==1) {
                IO_KeyScan();
            }
        }
        if (KeyCode==2)
        {
            LCD_Clear(WHITE);
            for(i=0;i<11;i++){
                for(j=0;j<i;j++){
                    LCD_ShowNum(10*(i-j), 0, a[j], 1, 16);
                }
                if (KeyCode!=2) {
                    break;
                }
                delayms(200);
            }
        }
    }
}

```

图 3 程序部分主函数

完成代码编写后生成 HEX 文件，并使用 STC-ISP 将该 HEX 文件烧录到单片机中。

观察实验结果，并拍照记录。

4.实验结果

按下按键 1 学号从左侧移入，如图 4 所示：

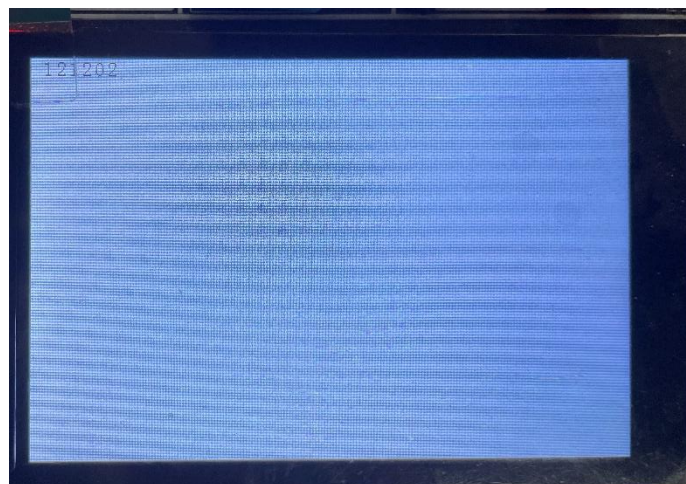


图 4 学号从左侧移入示意图

按下按键 3 学号从右侧移入，如图 5 所示：

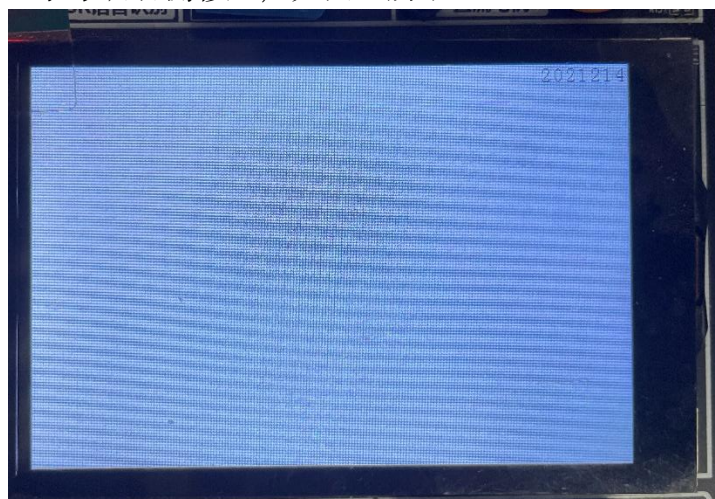


图 5 学号从右侧移入示意图

按下按键 2 学号从左侧以 200ms 间隔移入与按下按键 3 学号从右侧以 200ms 间隔移入的图片效果与另外两者相同，就不在此占用篇幅。

实验四 通过 A/D 采集模拟量的输入并显示

1.实验目的

- (1) 单片机控制 DHT11 温湿度传感器
- (2) 掌握单片机控制超声波模块 HC-SR04
- (3) 掌握单片机内部的 A/D 转换器的使用
- (4) 掌握 TFT LCD 屏的显示

2.基本原理

TFT LCD 屏的显示已经在实验三进行过详细介绍, 此处就不再进行赘述。

2.1 温湿度传感器原理

本实验使用通过读写 DHT11 的寄存器实现对温湿度实时检测，其电路原理图如图 1 所示。

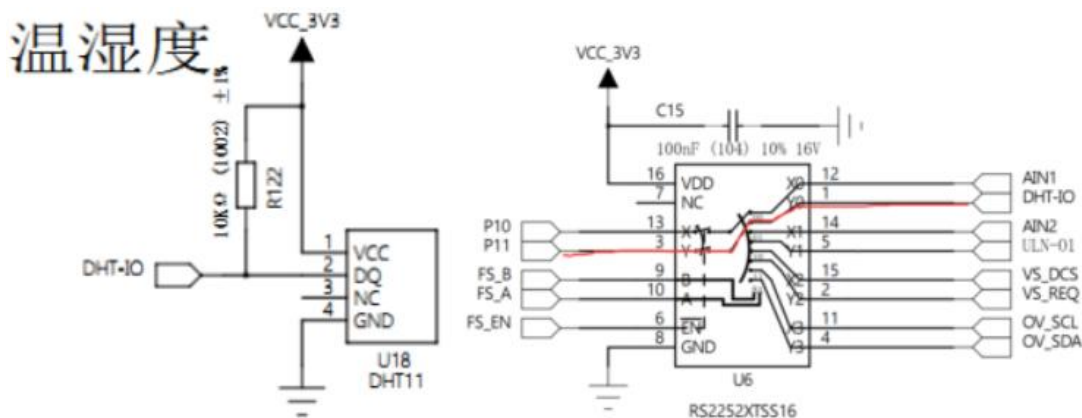


图 1 DHT11 外围电路原理图

DHT11 温湿度传感器 DHT-IO 与单片机 P5.5 相连。此时需要将核心板上拨码开关，选择场景为“智慧农业”。

DHT11 数字温湿度传感器是一款含有已校准数字信号输出的温湿度复合传感器。相比于 DS18B20 只能测量温度，DHT11 既能检测温度又能检测湿度，不过 DHT11 的精度和测量范围都要低于 DS18B20，其温度测量范围为 0~50℃，误差在 $\pm 2^{\circ}\text{C}$ ；湿度的测量范围为 20%~90%RH（Relative Humidity 相对湿度——指空气中水汽压与饱和水汽压的百分比），误差在 $\pm 5\%\text{RH}$ 。

DHT11 采用单总线协议与单片机通信，单片机发送一次复位信号后，DHT11 从低功耗模式转换到高速模式，等待主机复位结束后，DHT11 发送响应信号，并拉高总线准备传输数据。一次完整的数据为 40bit，按照高位在前，低位在后的顺序传输。

数据格式为：8bit 湿度整数数据+8bit 湿度小数数据+8bit 温度整数数据+8bit 温度小数数据+8bit 校验和，一共 5 字节（40bit）数据。由于 DHT11 分辨率只能精确到个位，所以小数部分是数据全为 0。校验和为前 4 个字节数据相加，校验的目的是为了保证数据传输的准确性。

DHT11 只有在接收到开始信号后才触发一次温湿度采集，如果没有接收到主机发送复位信号，DHT11 不主动进行温湿度采集。当数据采集完毕且无开始信号后，DHT11 自动切换到低速模式。

2.2 超声波模块 HC-SR04 测距原理

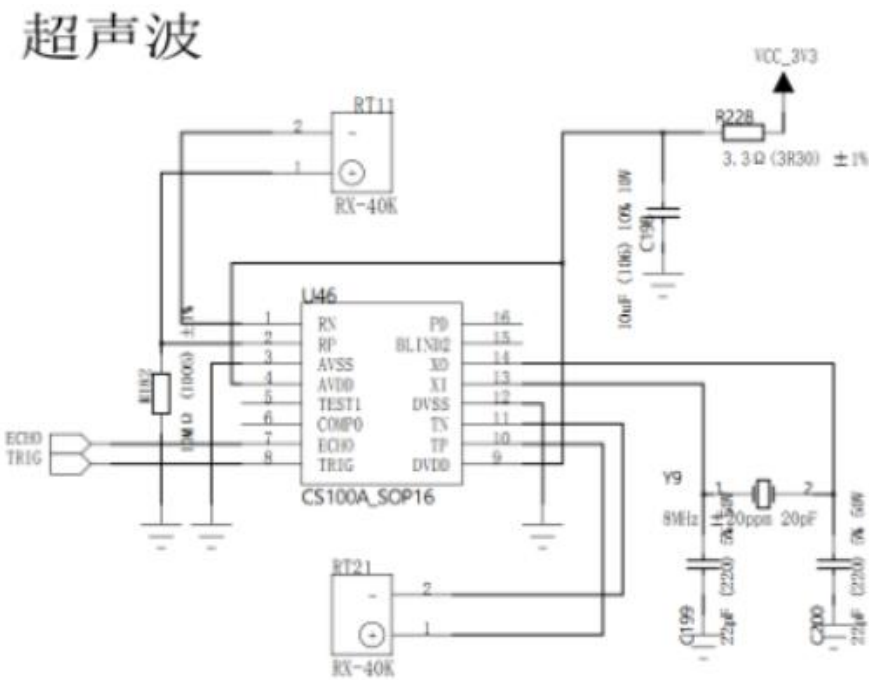


图 2 HC-SR04 超声波模块外围电路图

HC-SR04 具有两个控制端口 TRIG 和 ECHO，TRIG 口用于触发测距，给至少 10us 的高电平信号，接着模块自动发送 8 个 40khz 的方波，自动检测是否有信号返回。有信号返回，通过 IO 口 ECHO 输出一个高电平，高电平持续的时间就是超声波从发射到返回的时间。

测试距离=（高电平时间*声速（340M/S））/2；

超声波模块的 ECHO 和 TRIG 口与单片机的 P5.2P 和 5.3 相连接。此时需要

将核心板上拨码开关，选择场景为“智能小车”。

2.3 A/D 转换器原理

15 系列单片机都有其内部的 A/D 转换器，最高可以 8 路 A/D 采集，支持 8 位 最高 10 位的 A/D 分辨率。其转换速度可达到 300KHz (30 万次/秒)。8 路电压输入型 A/D,可做温度检测、电池电压检测、按键扫描、频谱检测、磁感应强电流检测等。

本实验中利用 STC 内部的 ADC 模块采集光敏传感器两端的电压值并通过 TFTLCD 屏幕进行显示。

电位器改变 AIN1 引脚上的电压值，AIN1 引脚与单片机引脚 P1.0 相连接。此时需要将核心板上拨码开关，选择场景为“智慧农业”。

3.实验步骤

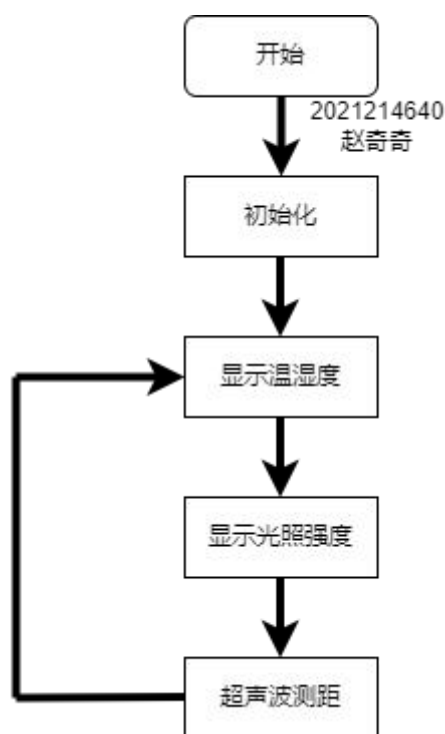


图 3 程序流程图

本实验程序流程图如图 3 所示，首先进行初始化，包括 IO 口初始化、定时器 0 初始化、温湿度传感器初始化、LCD 初始化、超声波测距初始化、光敏传感器初始化等。

接着在主函数循环中执行显示温湿度、显示光照强度、显示超声波测距距离等功能函数。

程序主函数如图 4 所示。

```
//主函数2021214640
void main(void)
{
    u32 i;
    Timer0Init(); //定时器0初始化
    IO_init();    //IO口初始化
    LCD_Init();   //液晶屏初始化
    delaylms(10);
    LCD_Clear(WHITE); //清屏白底

    while(1) {
        CH1and2();
        CH3();
        CH0();
    }
}
```

图 4 程序主函数

完成代码编写后生成 HEX 文件，并使用 STC-ISP 将该 HEX 文件烧录到单片机中。

观察实验结果，并拍照记录。

4.实验结果

LCD 屏显示结果如图 5 所示：

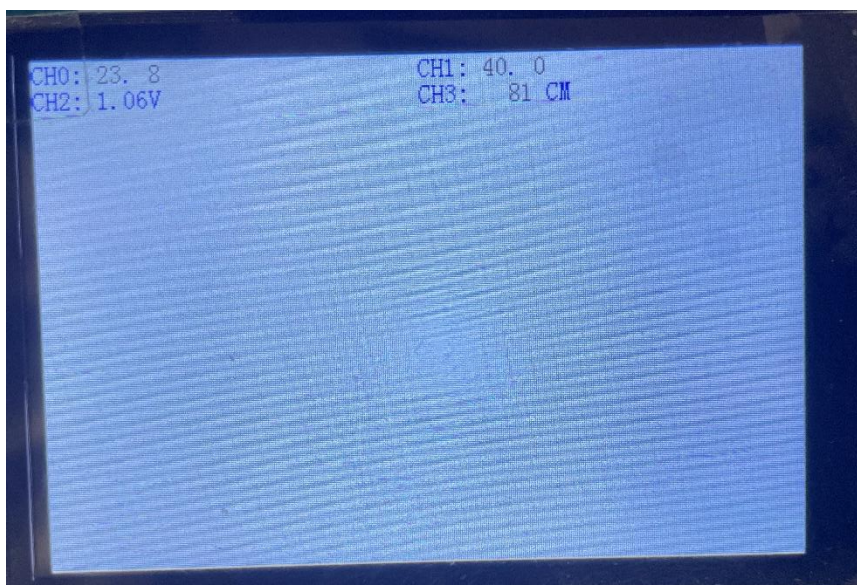


图 5 LCD 屏幕显示图

实验五 温湿度报警与键盘设置

1.实验目的

- (1) 掌握蜂鸣器的工作原理
- (2) 掌握 4*4 矩阵键盘的工作原理
- (3) 掌握单片机定时器中断的原理与使用方法
- (4) 单片机控制 DHT11 温湿度传感器
- (5) 掌握 TFT LCD 屏的显示

2.基本原理

除蜂鸣器的工作原理以外，其他部分均已在其他实验进行过详细介绍，因此此处不再赘述。

2.1 蜂鸣器工作原理

蜂鸣器

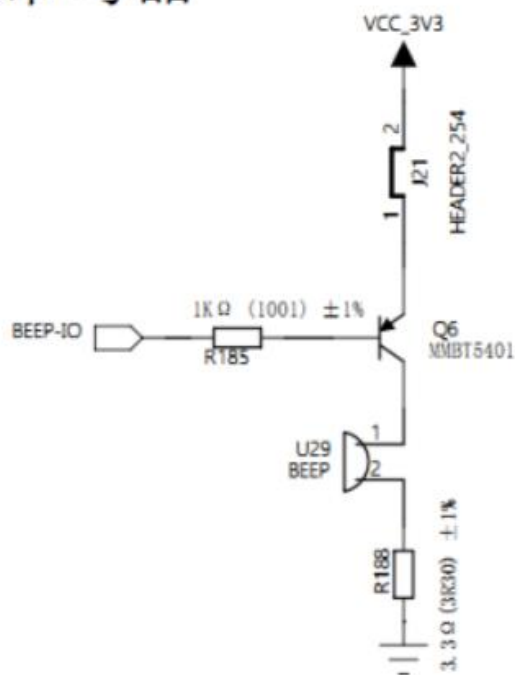


图 1 蜂鸣器工作电路图

蜂鸣器的工作原理如图 1 所示，场景功能板上的蜂鸣器由三极管控制，当

BEEP-IO 为低电平时蜂鸣器发声。BEEP-IO 与单片机上的 P5.5 相连接，注意要将蜂鸣器旁边的“蜂鸣器电源”跳线帽连接。

3.实验步骤

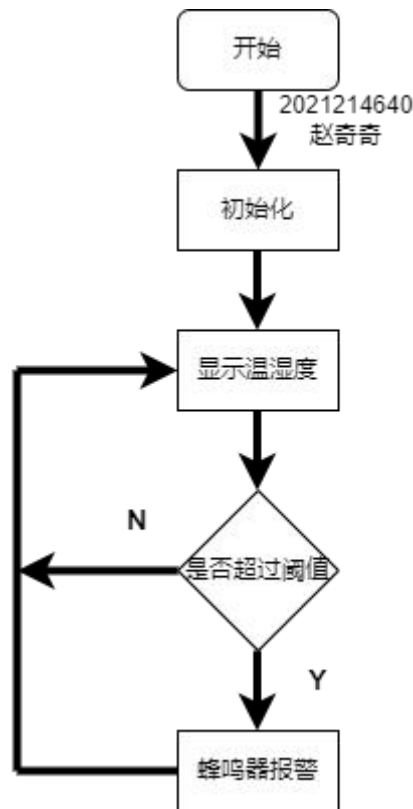


图 2 程序流程图

本实验程序流程图如图 2 所示，首先进行初始化，包括 IO 口初始化、定时器 0 初始化、定时器 1 初始化、温湿度传感器初始化、LCD 初始化等。

接着在主函数循环中执行显示温湿度、显示温湿度阈值等功能函数。
本实验中用到了两个定时器，在定时器 0 中进行矩阵键盘控制阈值加减的功能，按下按键一温度阈值加一，按下按键二温度阈值减一，按下按键三湿度阈值加一，按下按键四温度阈值减一，并且通过主函数显示在 LCD 屏上。

在定时器 1 中进行温湿度报警的判断，如果当前温湿度超过阈值，则蜂鸣器报警。

程序主函数如图 3 所示。

```

//主函数2021214640
void main(void)
{
    Timer0Init();
    Timer1Init();
    IO_init(); //IO口初始化
    LCD_Init(); //液晶屏初始化
    delayms(10);
    LCD_Clear(WHITE); //清屏白底
    P5M1 &= ~(1<<5), P5M0 |= (1<<5);
    BEEP_IO=1;

    KeyCode = 0; //给用户使用的键码, 1~16有效
    IO_KeyState = 0;
    IO_KeyState1 = 0;
    IO_KeyHoldCnt = 0;
    while(1) { //初始化温湿度传感器
        RH();

        Show_Str(0, 10, BLUE, WHITE, "temperature alarm:", 16, 0);
        LCD_ShowNum(150, 10, temperature, 4, 16);

        Show_Str(0, 50, BLUE, WHITE, "humidity alarm:", 16, 0);
        LCD_ShowNum(150, 50, humidity, 2, 16);

        Show_Str(0, 100, BLUE, WHITE, "temperature:", 16, 0);
        LCD_ShowNum(200, 100, U8T_data_H, 2, 16);
        Show_Str(216, 100, BLUE, WHITE, ".", 16, 0);
        LCD_ShowNum(224, 100, U8T_data_L, 2, 16);

        Show_Str(0, 150, BLUE, WHITE, "humidity:", 16, 0);
        LCD_ShowNum(200, 150, U8RH_data_H, 2, 16);
        Show_Str(216, 150, BLUE, WHITE, ".", 16, 0);
        LCD_ShowNum(224, 150, U8RH_data_L, 2, 16);
    }
}

```

图 3 程序主函数图

完成代码编写后生成 HEX 文件，并使用 STC-ISP 将该 HEX 文件烧录到单片机中。

观察实验结果，并拍照记录。

4.实验结果

显示当前温湿度与温湿度阈值如图 4 所示：

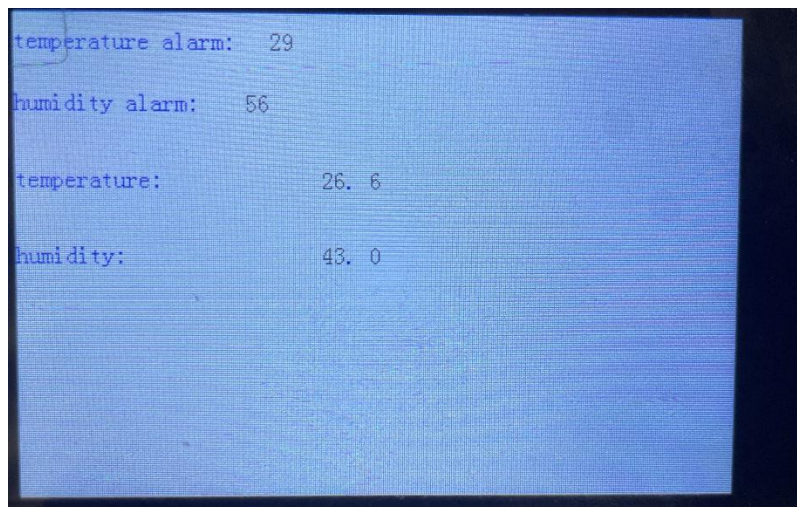


图 4 初始化主界面显示图

“哈”一口气后，温湿度发生变化，超过阈值，蜂鸣器报警，如图 5 所示。

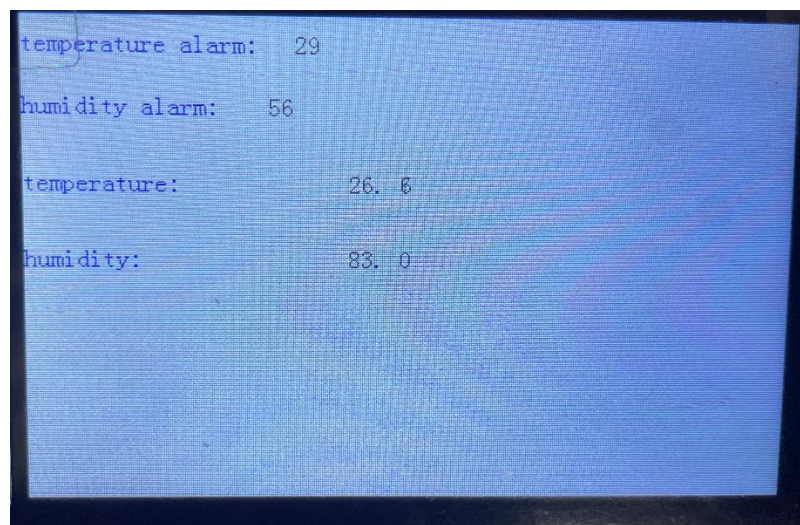


图 5 “哈”气后主界面显示图

本人能力有限，未成功完成实验六，因此没写实验六。

考核及成绩

附表3 《计算机与信息类创新实践》评分表

序号	课程目标	指标点	评价观察点	评价方式		得分 (百分制)
				依据/评价人	权重	
1	CO1: 掌握 KeilC 工具软件使用方法, 能用 C 语言在单片机系统上进行编程开发。	GR5.1	根据现场操作、演示及回答问题情况评判: 1) 学生利用单片机开发系统进行编程开发的能力状况。	实践验收/教师	0.2	
2	CO2: 通过分组合作方式完成综合性设计, 每位同学能完成自己负责的设计开发任务, 培养学生的团队协作能力。	GR9.1	根据现场验收和报告评判: 1) 课题任务整体完成情况; 2) 个人在团队里面发挥作用的大小。	设计验收/教师	0.1	
				设计报告/教师	0.2	
				设计答辩/答辩小组	0.1	
3	CO3: 通过课下自主学习完成相关的任务, 培养学生自主学习和终身学习的意识。	GR12.1	依据设计过程及报告评判: 1) 查阅资料、文献情况; 2) 整合已有软件和代码能力。	实践报告/教师	0.3	
				实践答辩/答辩小组	0.1	
<p>评分方法: 评价人根据观察点要求对各项课程目标的完成情况进行评估并给出得分 (百分制)。</p> <p>评价标准: 1) 观察点任务完成、能力达成, 且表现突出: 评为优秀, 得分范围为 85~100; 2) 观察点任务完成、能力达成, 表现良好: 评为良, 得分范围为 75~84.9; 3) 观察点任务完成、能力达成一般, 表现较好: 评为中, 得分范围为 66~74.9; 4) 观察点任务基本完成、能力基本达成, 表现一般: 评为及格, 得分范围为 60~65.9; 5) 观察点任务未完成、能力未达成, 表现差: 评为不及格, 得分范围为 0-59;</p>						
<p>总评成绩: _____ 导师签字: _____ 日期: _____</p>						