

课程编号: 0008190



计算机硬件类综合性课程设计 设计报告

(2020-2021-2)

题 目: 智能花盆

学部 (院): 信息学部

专业班级: 计算机科学与技术 180701

报 告 18074102 左帅

人:

考勤(10 分)	设计实现(70 分)	文档(20 分)	总评(100 分)
评语:			
教师签字		评阅日期	2021.07.

一、概述

1.1 选题意义

可以自动对植物进行浇水，省去了需要人工进行浇水的麻烦。可以自动保养植物，因此决定做了这个智能花盆。

1.2 作品功能概述

- 1.可以自动检测土壤中湿度，并以此为依据自动进行浇水。
- 2.同时利用键盘依据用户的设置进行植物类型的选择，从而达到控制湿度阈值等的目的。或者直接进行湿度光强阈值的调节
- 3.并且可以进行光感，若光线过强，可以拉起一个用于遮挡光线的帘子防止植物晒伤。
- 4.有一语音合成模块，当执行相关操作时，进行语音播报。在当前室温等数据出现严重异常时，进行警报的播报。
- 5.通过 OLED 屏显示各种内容。

二、总体设计

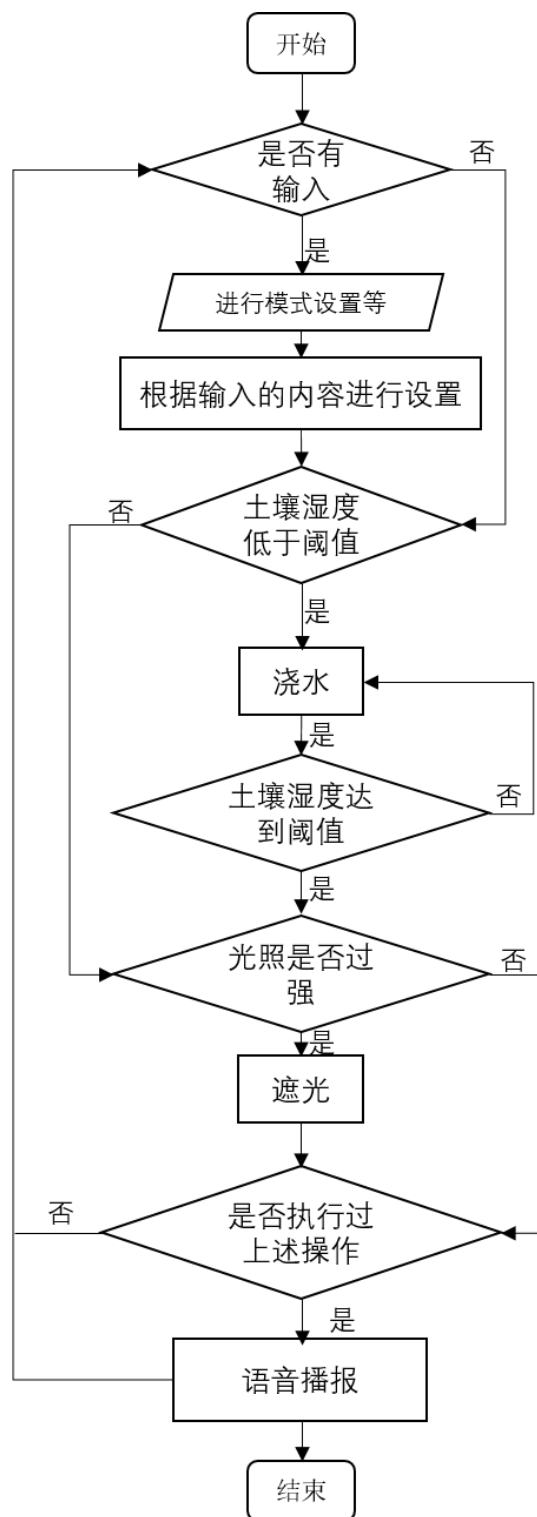


图 2-1 软件流程图

计算机硬件类综合性课程设计

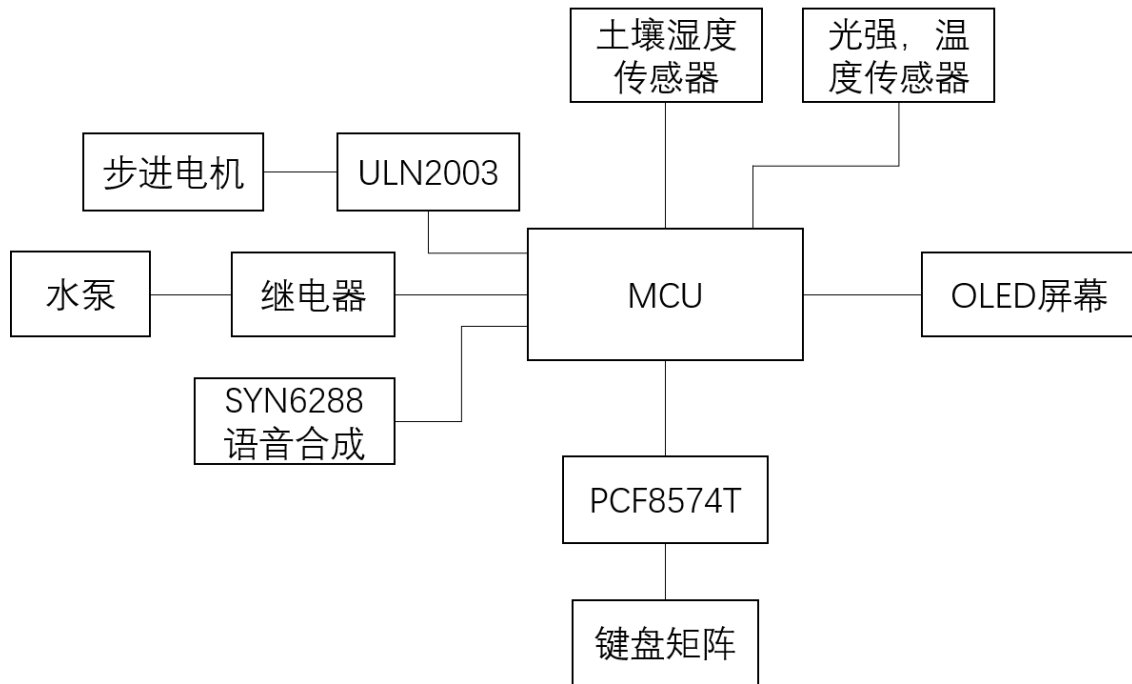


图 2-2 硬件框图

步进电机驱动通过数字输出方式进行连接；OLED 屏幕通过 I2C 方式与 MCU 连接，其地址为 0x78；温湿度传感器通过数字引脚输入方式与 MCU 连接；土壤湿度传感器通过模拟信号输入与 MCU 连接；SYN6288 通过软串口与 MCU 连接。矩阵键盘连接 PCF8574T 通过 I2C 与 MCU 连接，其首地址为 0x20。

三、硬件详细设计

3.1 硬件系统结构

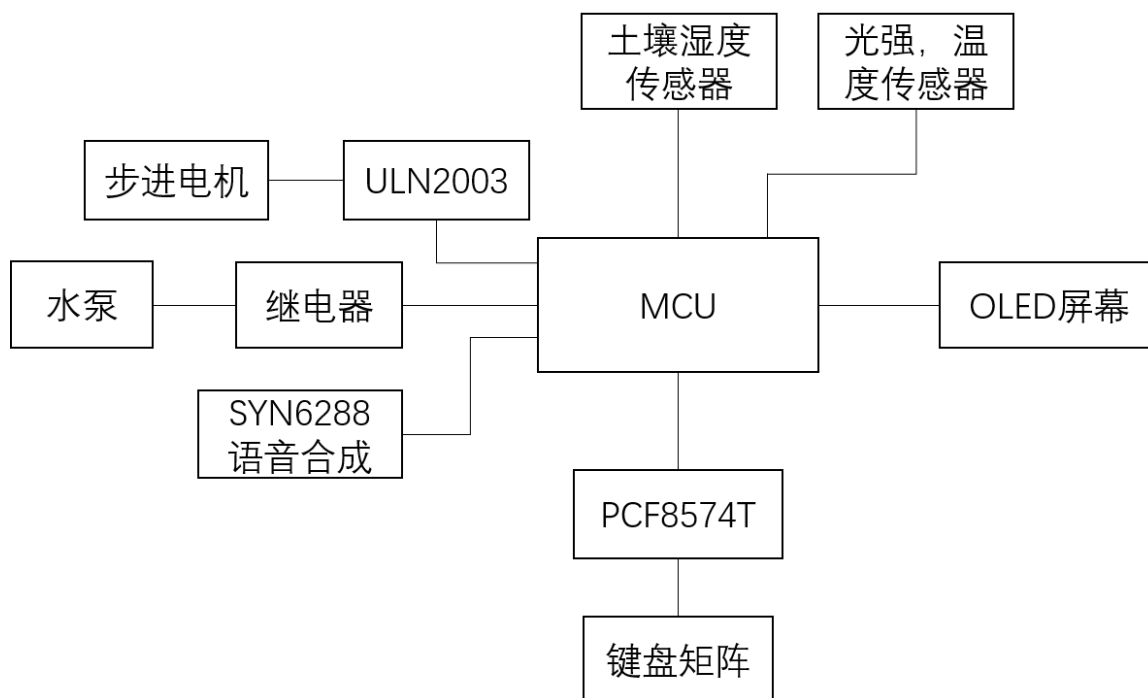


图 2-2 硬件框图

步进电机的直接接在控制板 ULN2003 上，ULN2003 的正极与 MCU 的 5V 相连，负极与 GND 相连。四相输出 IN1~IN4 与 MCU 的 8, 9, 10, 11 脚相连。

土壤湿度传感器模块的 VCC 与 GND 分别与 MCU 的 5V 与 GND 相连。OUT 接模拟输入 A1。

光线传感器，温度传感器，OLED 屏幕直接集成在 Grove Beginner Kit 上

矩阵键盘的八个引脚接入 PCF8574T 模块的 P0 到 P7，PCF8574T 模块的 SDA 与 SCL 分别接在 MCU 的 SDA 与 SCL 上，VCC 与 GND 分别与 MCU 的 5V 与 GND 相连。

SYN6288 语音合成模块的 VCC 与 GND 分别与 MCU 的 5V 与 GND 相连。RXD 与 7 脚相连，TXD 与 6 脚相连，使用软串口进行交互。

3.2 核心板

3.2.1 选型

核心板采用了 Grove Beginner Kit 上的 Seeeduino Lotus，其使用的处理器为 ATMEGA328P-MU，且与 Arduino UNO 完全兼容，其主频为 16MHz，有着 32KB 的 Flash，1KB 的 EPROM 以及 2KB 的 RAM，因此 ATMEGA328P-MU 是一款高性能，低功耗的 AVR 8 位微控制器。正因为其高性能，低功耗的特点，所以作为本次课设所采用的单片机。

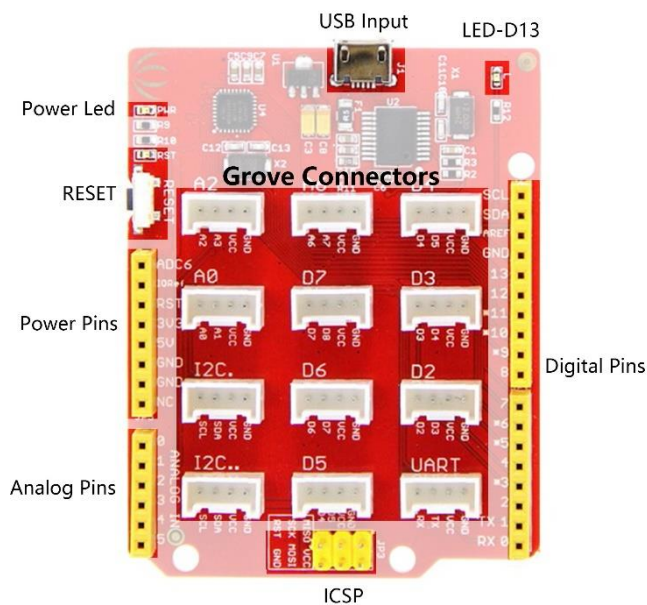


图 3-1 Seeeduino Lotus

3.2.2 模块原理图介绍

Seeeduino Lotus 的芯片采用了 ATMEGA328P-MU，同时将处理器的 IO 接口以 Grove 接口的形式接出。同时将 reset 接口与按钮连接。

ATMEGA328P-MU 的引脚分配如下图所示：

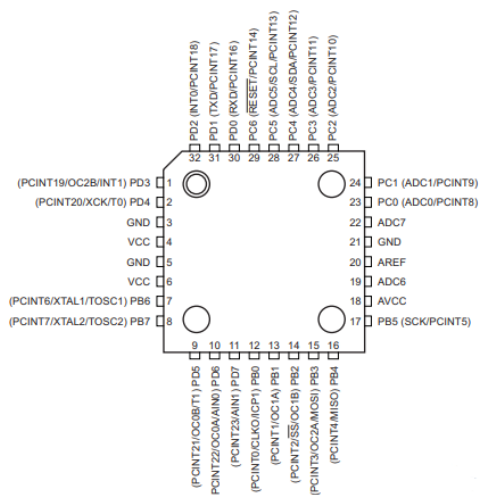


图 3-2 ATMEGA328P 引脚分配图

计算机硬件类综合性课程设计

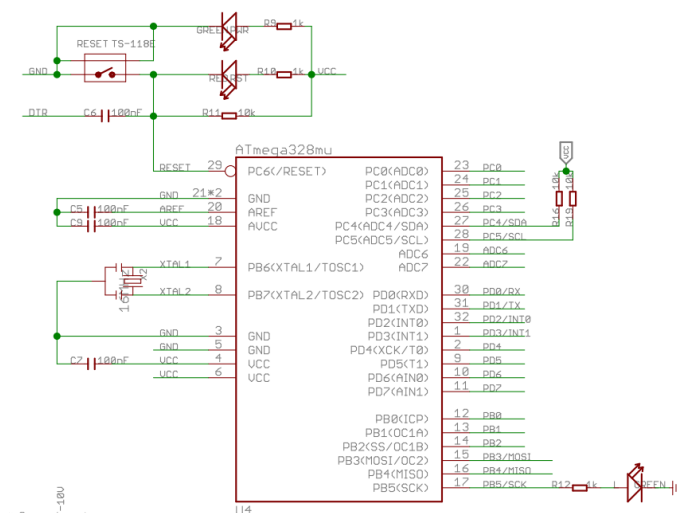


图 3-3 ATMEGA328P-MU 部分的原理图

Seeeduino Lotus 使用了 CP2102N 芯片来实现 USB 到串行接口的 USB 总线转换。

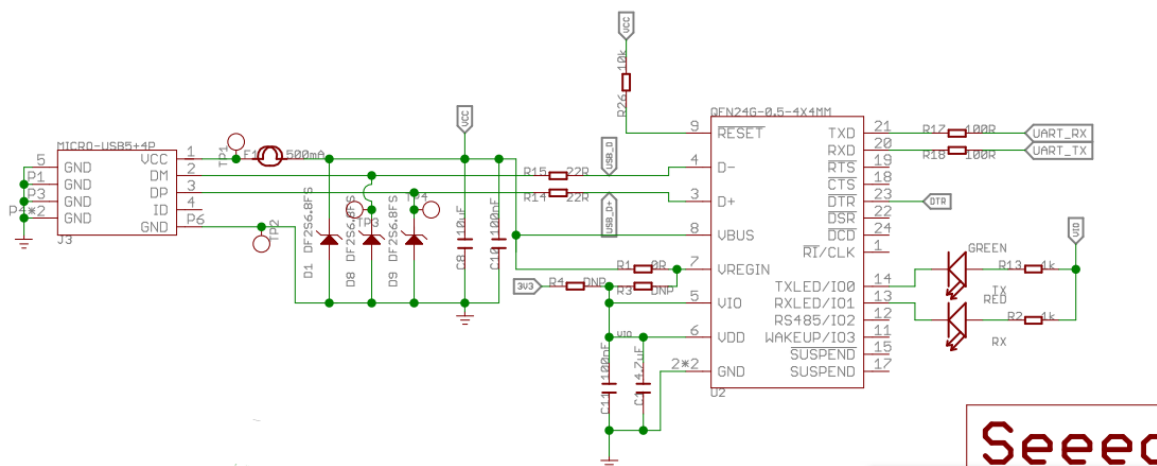


图 3-4 CP2102 部分原理图

Seeeduino Lotus 的接口定义如下表所示

表 3-1 Seeeduino Lotus 接口定义

引脚名称	功能
0	数字信号引脚或串口数据接收
1	数字信号引脚或串口数据发送
2~13	数字信号引脚
RESET	复位
A0~A5	模拟信号引脚
VIN	电源输入端
GND	地线
5V	输出 5V 电压
3V3	输出 3.3V 电压
AREF	模拟输入参考电压输入引脚

计算机硬件类综合性课程设计

IOREF	微控制器参考电压输入引脚
SDA	串行数据线
SCL	串行时钟线

3.2.3 模块 PCB 介绍

Seeeduino Lotus 有 14 个数字输入/输出和 7 个模拟输入/输出，一个 micro USB 接口，一个 ICSP 接口，12 个 Grove 接口和一个复位按钮。

3.3 土壤湿度传感器模块介绍

模块照片：



图 3-5 模块照片

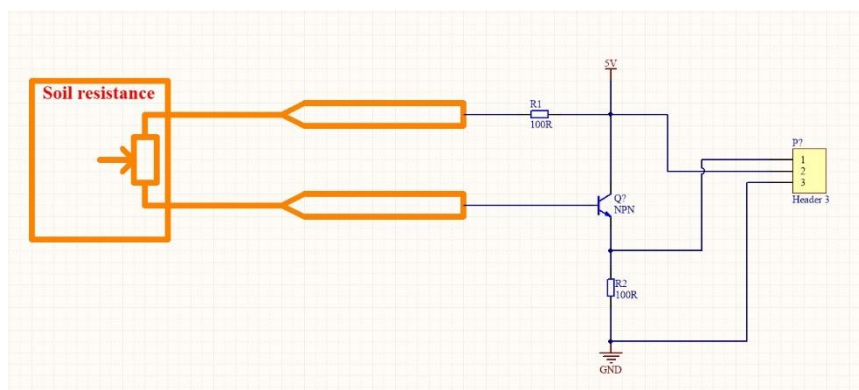


图 3-6 土壤湿度传感器原理图

土壤湿度传感器是判断土壤中水分含量的多少来判定土壤的湿度大小。如图所示，当土壤湿度传感器探头悬空时，三极管基极处于开路状态，三极管截止输出为 0；当插入土壤中时

计算机硬件类综合性课程设计

由于土壤中水分含量不同，土壤的电阻值就不同，三极管的基极就提供了大小变化的导通电流，三极管集电极到发射极的导通电流受到基极控制，经过发射极的下拉电阻后转换成电压。

电气参数:

电源电压: 3.3v or 5v

输出电压: 0~2.3v

工作电流: 最大 20mA

接口定义:

1 脚信号, 2 脚地, 3 脚电源正

3.4 步进电机驱动模块介绍

模块照片:

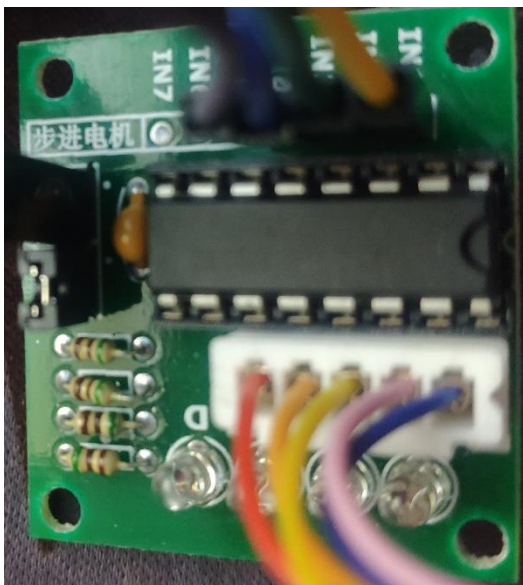


图 3-7 模块照片

关于步进电机的驱动采用的模块是 ULN2003 芯片。ULN2003 是高耐压、大电流复合晶体管阵列，由七个硅 NPN 复合晶体管组成，每一对达林顿都串联一个 2.7K 的基极电阻，在 5V 的工作电压下它能与 TTL 和 CMOS 电路直接相连，可以直接处理原先需要标准逻辑缓冲器来处理的数据。

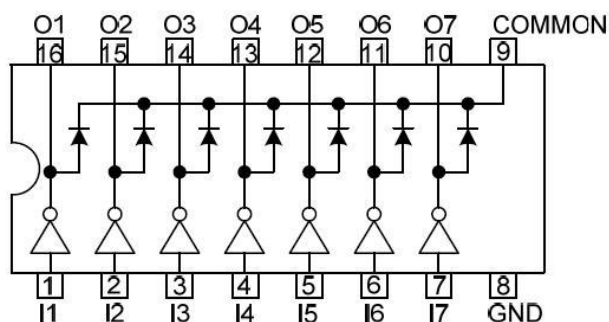


图 3-8 ULN2003

ULN2003 的引脚定义如下表所示:

表 3-2 ULN2003 的引脚定义

引脚序号	引脚名称	说明
1	I1	CPU 脉冲输入端

计算机硬件类综合性课程设计

2	I2	CPU 脉冲输入端
3	I3	CPU 脉冲输入端
4	I4	CPU 脉冲输入端
5	I5	CPU 脉冲输入端
6	I6	CPU 脉冲输入端
7	I7	CPU 脉冲输入端
8	GND	接地
9	COMMON	该脚接电源正极，实现续流作用。该脚接地，达林顿管的集电极对地接通。
10	O7	脉冲信号输出端，对应 7 脚信号输入端
11	O6	脉冲信号输出端，对应 6 脚信号输入端
12	O5	脉冲信号输出端，对应 5 脚信号输入端
13	O4	脉冲信号输出端，对应 4 脚信号输入端
14	O3	脉冲信号输出端，对应 3 脚信号输入端
15	O2	脉冲信号输出端，对应 2 脚信号输入端
16	O1	脉冲信号输出端，对应 1 脚信号输入端

3.4.1 步进电机驱动模块的原理图介绍

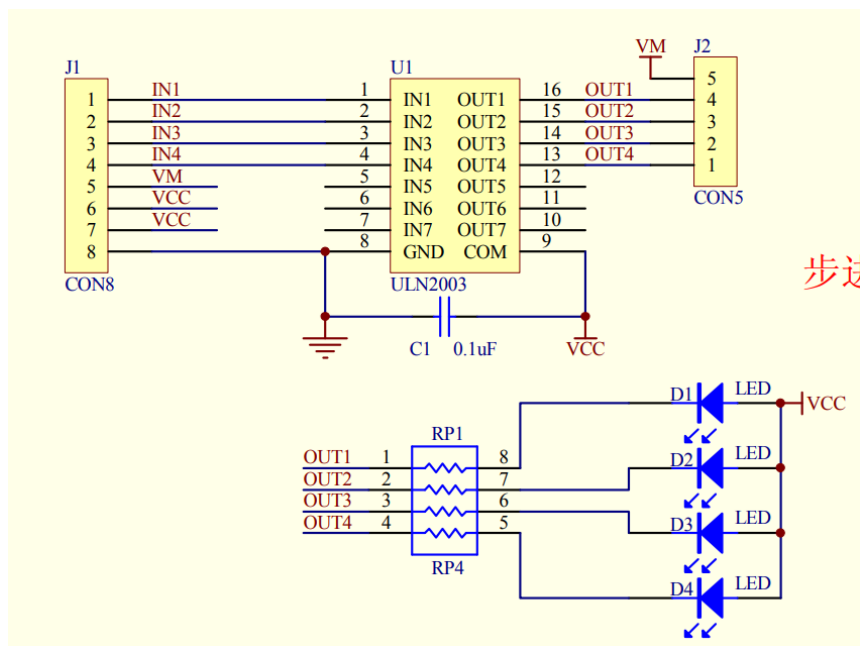


图 3-9 步进电机驱动模块原理图

当对步进电机按一定顺序施加一系列连续不断的控制脉冲时，它可以连续不断地转动。每一个脉冲信号使得步进电机的某一相或两相绕组的通电状态改变一次，也就对应转子转过一定的角度。

电气参数：

电源电压：5v

接口定义：

IN1~IN4 接单片机数字引脚

VCC 接电源

GND 接地

OUT1~OUT4 接步进电机

VM 接步进电机

3.5 PCF8574T IO 拓展模块介绍

模块照片:

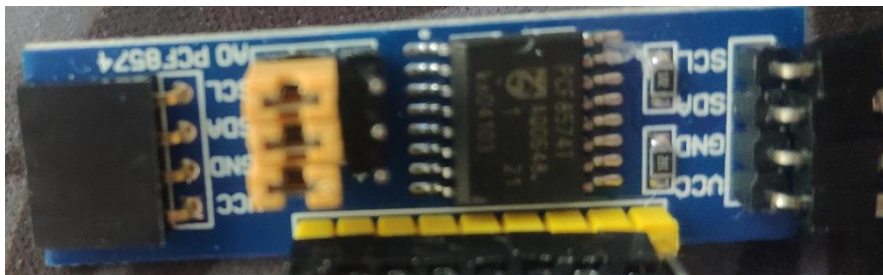


图 3-10 模块照片

PCF8574T 是 CMOS 电路,它通过两条双向总线可使大多数 MCU 实现远程 I/O 扩展。该器件包含一个 8 位准双向口和一个总线接口。pcf8574T 电流消耗很低,并且输出锁存,具有大电流驱动能力。

PCF8574T 的引脚定义如下表所示:

表 3-3 PCF8574 的引脚定义

引脚序号	引脚名称	说明
1	A0	地址输入 0
2	A1	地址输入 1
3	A2	地址输入 2
4	P0	准双向 I/O 口 0
5	P1	准双向 I/O 口 1
6	P2	准双向 I/O 口 2
7	P3	准双向 I/O 口 3
8	VSS	接地
9	P4	准双向 I/O 口 4
10	P5	准双向 I/O 口 5
11	P6	准双向 I/O 口 6
12	P7	准双向 I/O 口 7
13	INT	脉冲中断输入
14	SCL	串行时钟线
15	SDA	串行数据线
16	VDD	电源

电气参数:

电源电压:最大:6V 最小:2.5V

电源电流:40 μ A

接口定义:

VCC 接电源正

GND 接地

SDA 接 MCU 的 SDA

SCL 接 MCU 的 SCL

P1~P8 为拓展的 IO 接口

3.6 SYN6288 模块的设计

模块照片:

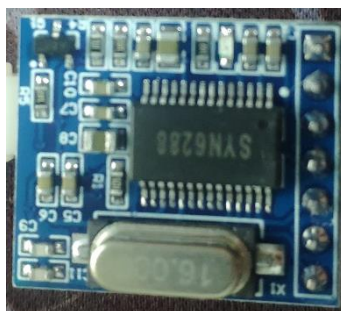


图 3-11 模块照片

SYN6288 是一个用于播报中文语音的模块。是北京宇音天下科技有限公司推出的一系列语音合成芯片模块之一，于 2010 年推出。该模块基于中文语音合成芯片 SYN6288，SSOP28L 封装。尺寸 10.2mm x 5.3mm。下图为 SYN6288 的基本工作流程与连接方式。

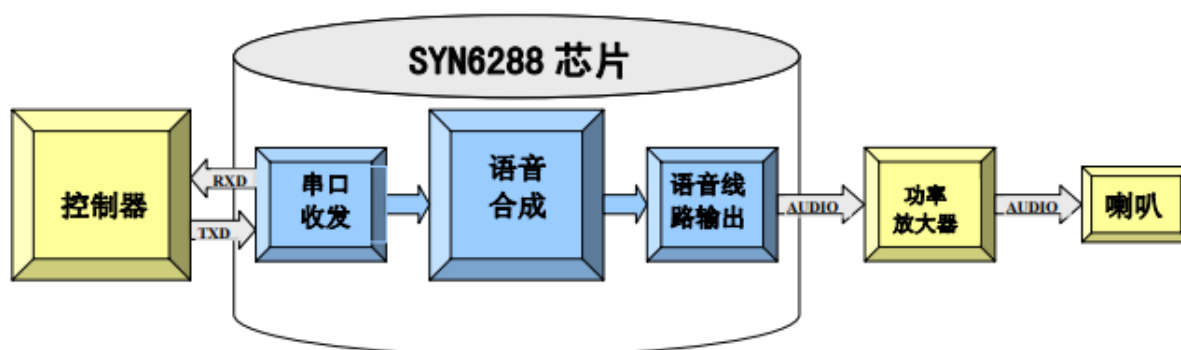


图 3-12 SYN6288 的基本工作流程

3.6.1 SYN6288 模块的原理图设计

根据 SYN6288 的数据手册可以得到该芯片的引脚定义如下：

表 3-4 SYN6288 引脚分配图

引脚序号	引脚名称	I/O	说明
1, 3	VSSIO0	I	总线模块 0 电源负极
2	VDDIO0	I	总线模块 0 电源正极
4	READY/BUSY	O	低电平表示 CHIP 空闲
5	Res.	-	Res 引脚
6	VDDIO1	I	总线模块 1 电源正极
7	VSSIO1	I	总线模块 1 电源负极
8, 12	VSSPP	I	语音输出模块电源负极
10	VDDPP	I	语音输出模块电源正极
9	BP0	O	推送 DAC 语音输出 1
11	BN0	O	推送 DAC 语音输出 2
28	RXD	I	串口数据接收
27	TXD	O	串口数据发送
26	VDDA	I	内部稳压电源正极
23	VSSA	I	内部稳压电源负极
25	XOUT	O	高速晶振输出

计算机硬件类综合性课程设计

24	XIN	I	高速晶振输入
22	REGOUT	O	电压自动调节输出
21	CVDD	I	处理器电源正极
18	CVSS	I	处理器电源负极
20	VDDIO2	I	总线模块 2 电源正极
17	VSSIO2	I	总线模块 2 电源负极
19	RST	I	芯片复位
16	VSS	I	电源负极

根据 SYN6288 的引脚分配情况，进行 SYN6288 芯片的原理图绘制。绘制后的芯片原理图如下图所示：

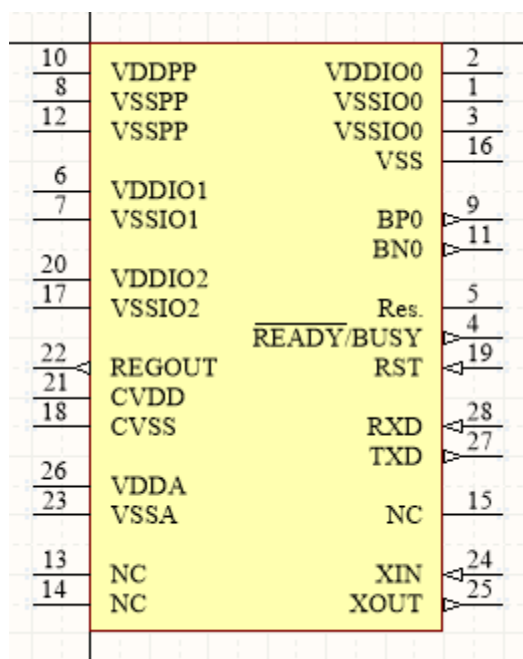


图 3-15 芯片原理图

SYN6288 共有六组外接电源，为保证电压稳定，则需要在每一组电源正极与负极之间使用一 47uF 和一 0.1uF 的电容器用于去耦。由于 SYN6288 的处理器的工作电压在 3V 左右，因此在使用 5V 供电时需要用到 REGOUT 引脚，将电压稳定在 $2.8V \pm 0.2V$ 。

在 RXD 输出时，需要用到反相器将 RXD 的相位进行反转，在此原理图中使用了以 NPN 三极管构成的简单反相器。

同时，将 BUSY 引脚与一 LED 灯连接，当 BUSY 输出高电平时，LED 灯亮，此时表示芯片正忙，即正在处理数据。

最后，将 VDD, RXD, TXD, GND, BUSY 这五个引脚与一个 5Pin 的插头相连，用于与单片机进行交互。将 BN0 与 BP0 与一个 2Pin 的插头相连，用于与扬声器相连接，以此进行声音的输出。

根据此规则，连接出的原理图如下：

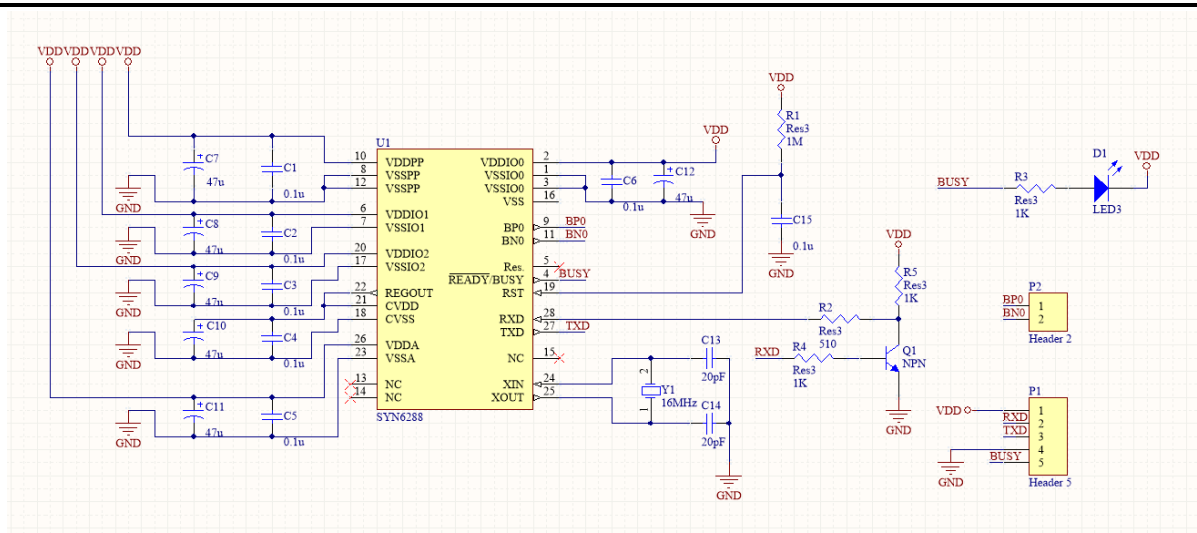


图 3-16 SYN6288 模块原理图

3.6.2 SYN6288 模块的 PCB 设计

通过原理图在 Altium Designer 进行 PCB 的绘制，由于 SYN6288 采用了 SSOP28L 封装芯片，故根据其封装，可以获得其 PCB 图。

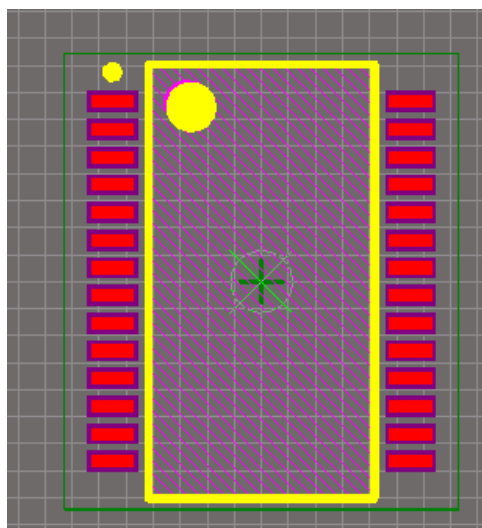


图 3-17 SYN6288 芯片 PCB 图

将 IC，电容，电阻等放入 PCB 图后进行连线，布局。

在布局时，我尽量将与对应接口相连接的元器件摆放在接口附近以方便连接，同时准备将地线在 Bottom Layer 环绕 PCB 板一圈。

在连线时，我遵守着地线>=电源线>其他线的原则。同时注意不拐直角，不走大斜线。遇到线路交叉时使用过孔将其中一条线从 Top Layer 转移到 Bottom Layer。在此基础上完成了 PCB 的绘制工作。

同时在设计过程中，密集且不失章法的摆放在尽可能小的板子上，尽量避免了大量使用过孔。为使版型美观流畅，我将所有的地线均放到了 Bottom Layer 上，而其他线除交汇时尽可能走 Top Layer

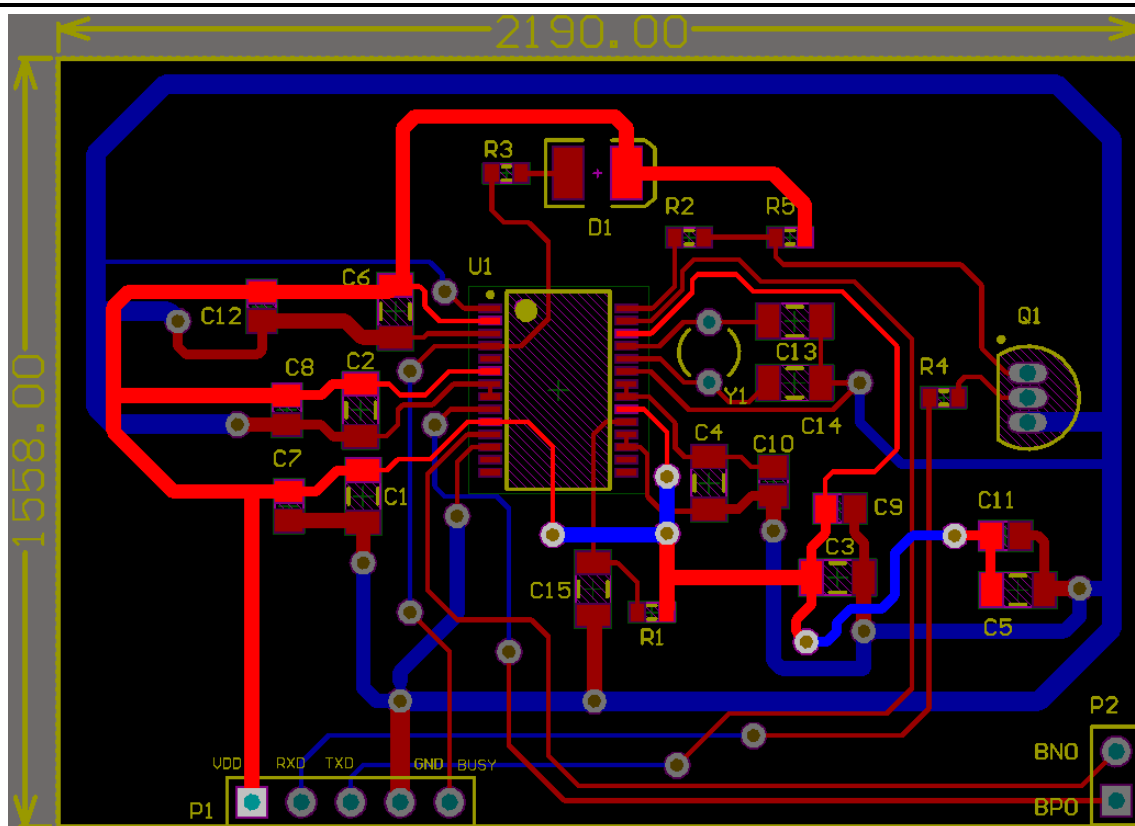


图 3-18 SYN6288PCB 模块

3.7 温湿度传感器模块介绍

3.7.1 温湿度传感器模块的原理图介绍

模块名称：温湿度传感器

模块图片：



图 3-19 模块图片

模块原理图：

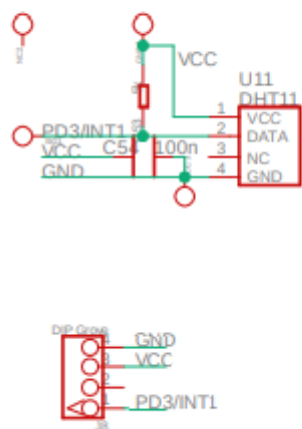


图 3-20 温湿度传感器模块原理图

模块简介：Grove-温度与湿度传感器提供一个提前校准的数字输出，相对湿度由独特的电容式传感器元件测量，温度由负温度系数（NTC）热敏传感器测量，不适用于低于 0 度的环境。

负温度系数温度传感器：温度越高阻值越低。

规格参数：

I/O 结构：GND、VCC、NC、SIG

GND 接板子上 GND，VCC 接板子上 5v，SIG 接数字信号 D2，

管脚	接口
GND	GND
VCC	5V
NC	--
SIG	D2 数字信号

表 3-5 接口定义

具体参数：

项目	条件	最小值	最大值	单位
VCC		3.3	5	V
测量时电流		1.3	2.1	mA
平均电流		0.5	1.1	mA
测量范围	湿度	20%	90%	RH
	温度	0	50	℃
精准度	湿度		±5%	RH
	温度		±2	℃

敏感度	湿度		1%	RH
	温度		1	℃

表 3-6 具体参数

3.8 光传感器模块介绍

3.8.1 光传感器模块的原理图介绍

模块名称：光传感器

模块图片：

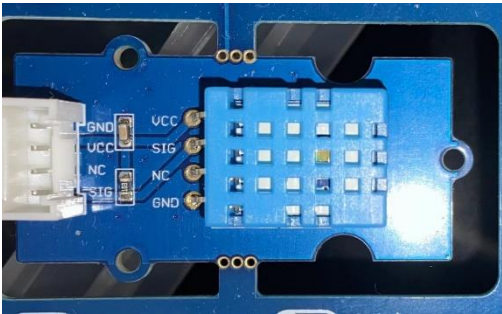


图 3-21 光传感器模块图片

模块原理图：

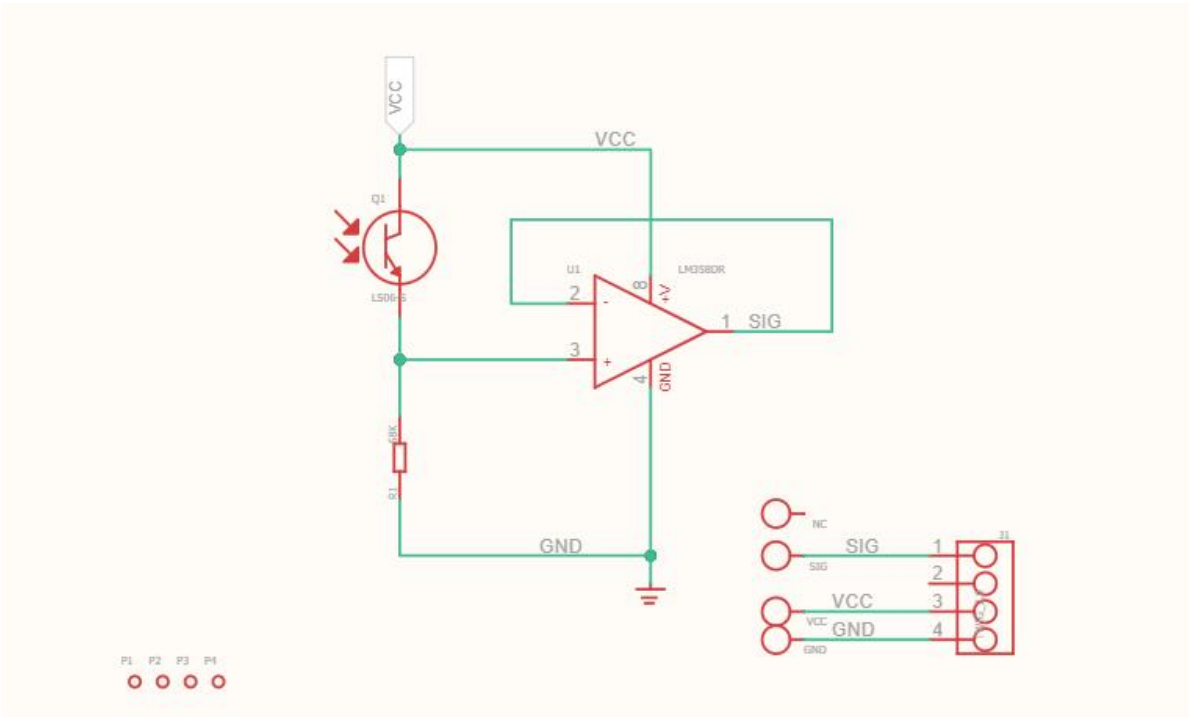


图 3-22 光传感器模块原理图

模块简介：

规格参数：光传感器集成了光电阻器（光依赖电阻器）来检测光线的强度。当光的强度增加时，光电阻器的阻力会降低。板子上的双 OpAmp 芯片 LM358 产生与光强度相对应的电压（即基于电阻值）。输出信号是模拟值，光越亮，值越大。

LM358：双运算放大器。内部包括有两个独立的、高增益、内部频率补偿的运算放大器，适合于电源电压范围很宽的单电源使用，也适用于双电源工作模式。使用范围包括传感放大器、直流增益模块和其他所有可用单电源供电的使用运算放大器的场合。

模块管角：

计算机硬件类综合性课程设计

I/O 结构: GND、VCC、NC、SIG

管脚	接口
GND	GND
VCC	5V
NC	--
SIG	A0 模拟信号

表 3-7 接口定义

具体参数:

项目	最小值	最大值	单位
工作电压	3	5	V
工作电流	0.5	3	mA
反应时间	20	30	毫秒

表 3-8 具体参数

3.9OLED 模块

3.9.1 OLED 模块的原理图介绍

模块图片:

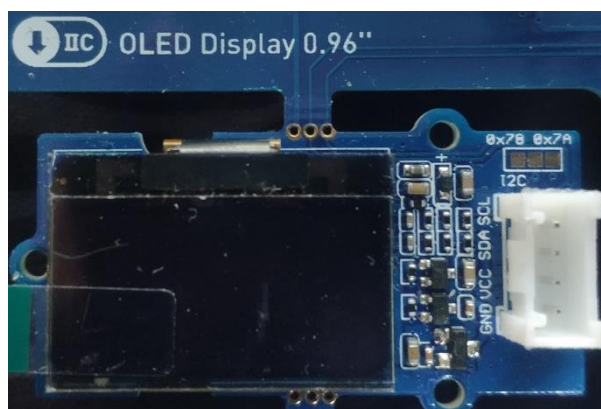


图 3-23 模块图片

Grove- OLED 显示器 0.96" (SSD1315) 是一个单色 (白色) 128×64 像素的显示矩阵模块其使用 I2C 接口。Grove - OLED 显示器 0.96" 可与 3.3V 和 5V 平台配合使用。其 I2C 地址为 0x78

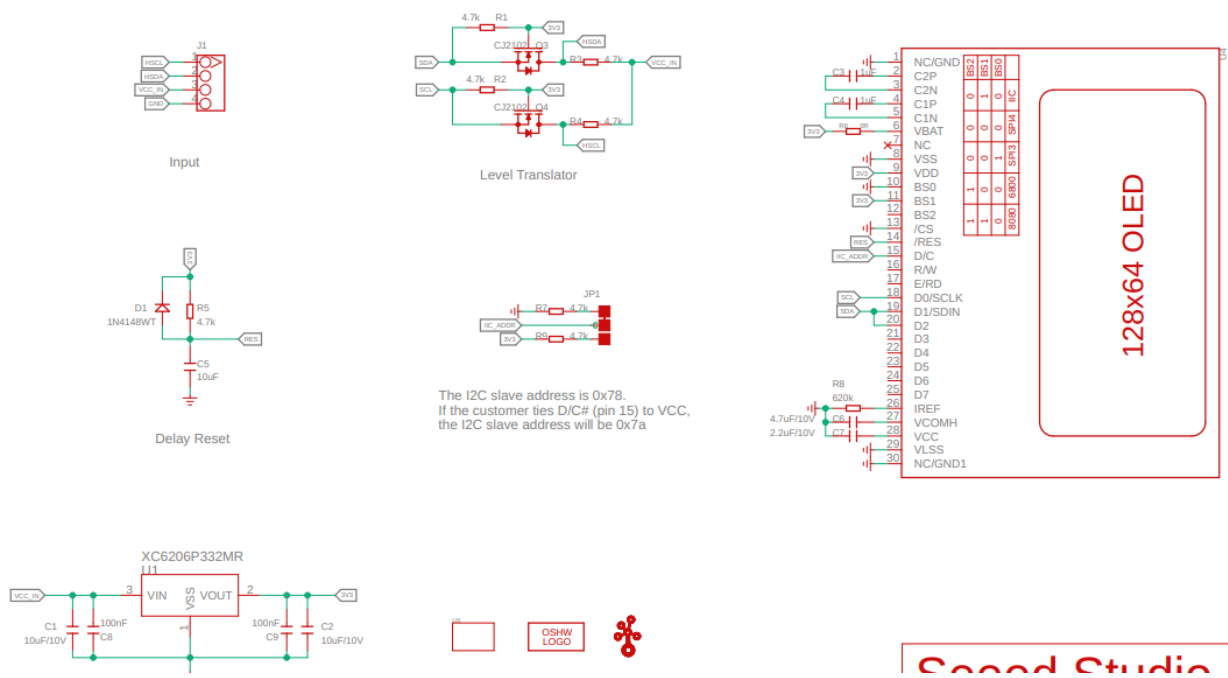


图 3-24 OLED 原理图

具体参数：

项目	值
输入电压	3.3V / 5V
输出电压	0 ~ 2.3V
像素	128 x 64
工作温度	-40℃ ~ +85 ℃
接口类型	I2C/Digital

表 3-9 具体参数

3.10 矩阵键盘模块
模块图片：

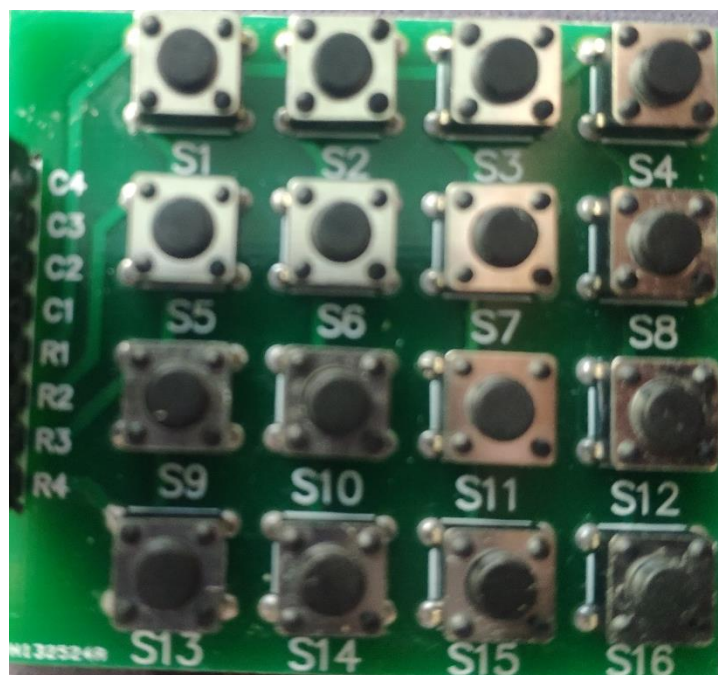


图 3-25 矩阵键盘照片

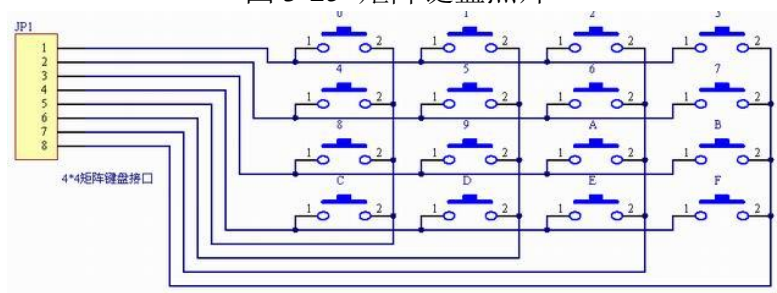


图 3-26 矩阵键盘原理图

矩阵键盘的工作原理如下：置第 1 行为低电平，其余行为高电平，读取列线数据，列线有低电平表示此行有按键按下。置第 2 行为低电平，其余行为高电平，读取列线数据，列线有低电平表示此行有按键按下。以此类推根据行线列线的电平不同可以识别是否有按键按下，哪一个按键按下，获取按键号。

四、软件详细设计

4.1 软件系统结构

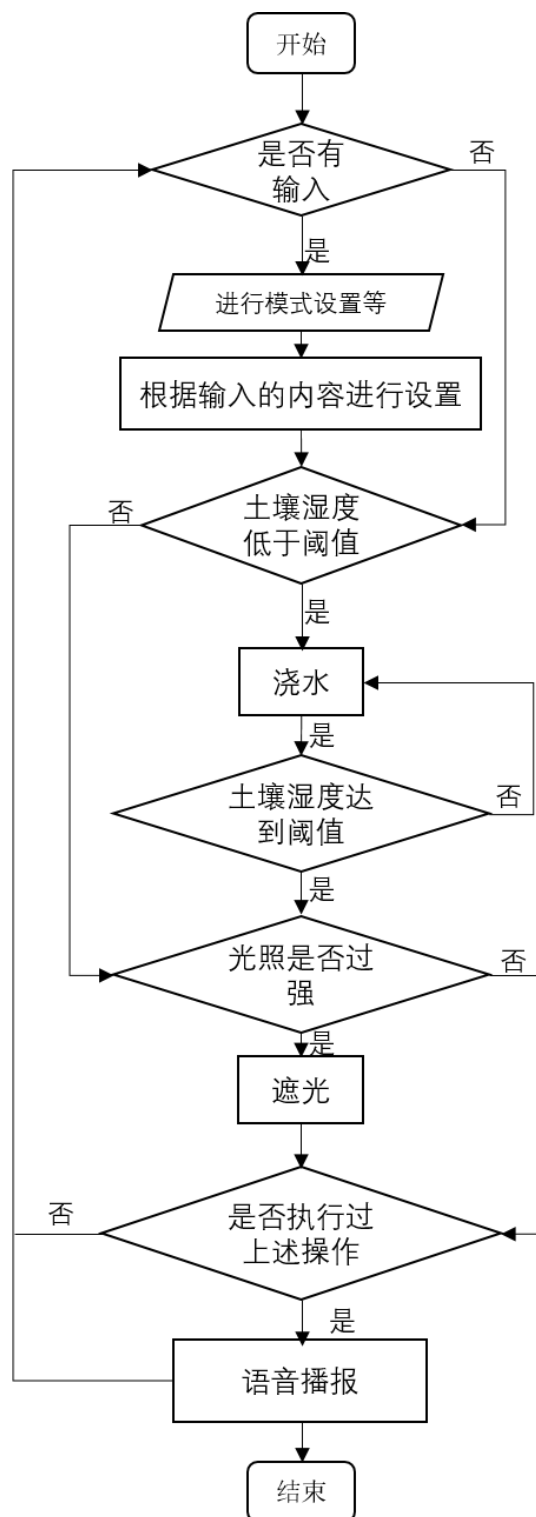


图 4-1 主程序流程图

4.2 步进电机模块的软件设计

```
void clockwise(int num)
```

```

{
    for (int count = 0; count < num; count++)
    {
        for (int i = 8; i < 12; i++)
        {
            digitalWrite(i, HIGH);
            delay(3);
            digitalWrite(i, LOW);
        }
    }
}
void anticlockwise(int num)
{
    for (int count = 0; count < num; count++)
    {
        for (int i = 11; i > 7; i--)
        {
            digitalWrite(i, HIGH);
            delay(3);
            digitalWrite(i, LOW);
        }
    }
}

```

采用了一相励磁的方式来驱动步进电机。

表 4-1 一相励磁顺序

STEP	A	B	\bar{A}	\bar{B}
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

4.3 SYN6288 模块的软件设计

程序段 1:

```

for (int i = 0; i < SIZE; i = i + 1) {
    character[i * 2] = ((buf[i * 3] & 0xF) << 4) + ((buf[i * 3 + 1] >> 2) & 0xF);
    character[i * 2 + 1] = ((buf[i * 3 + 1] & 0x3) << 6) + (buf[i * 3 + 2] & 0x3F);
}
output[0] = 0xFD;
output[1] = (SIZE * 2 + 3) >> 8;
output[2] = ((SIZE * 2 + 3) & 0xFF);
output[3] = 0x01;
output[4] = 0x03;

```

```

for (int i=0;i<SIZE*2;i++) {
    output[i+5]=character[i];
}
output[SIZE*2+5]=output[0];
for (int i=1;i<SIZE*2+5;i++) {
    output[SIZE*2+5]=output[SIZE*2+5] ^ output[i];
}

for (int i =0;i<SIZE*2+6;i++) {
    SynSerial.write(output[i]);
}

```

由于 Arduino 对 ino 文件使用的编码格式为 UTF-8, 而 SYN6288 接收的字符需要为 Unicode 编码, 因此对于在程序中直接定义的字符串, SYN6288 无法进行识别, 故需要对其进行转换, 将 UTF-8 转换为 Unicode 然后根据给定的帧格式写入串口, 从而才能进行语音播报。对于 Unicode 来说, 所有的字符均为两字节, 但是像英文字母, 只需要不到一个字节即可存储, 其前九位都是 0。而 UTF-8 为了节省空间, 制定了如下的编码规则。汉字共占 16 位, 因此在 UTF-8 中为三字节。

```

0xxxxxxx
110xxxxx 10xxxxxx
1110xxxx 10xxxxxx 10xxxxxx
11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx
1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx

```

SYN6288 所能接受的帧格式如下图所示:

		长度	命令字	命令参数	待发送文本	异或校验
数据	0xFD	0x00 0x0B	0x01	0x03	字音天下 0x8b 0xed 0x97 0xf3 0x59 0x29 0x4e 0x0b	0xc3
数据帧	0xFD 0x00 0x0B 0x01 0x03 0x8b 0xed 0x97 0xf3 0x59 0x29 0x4e 0x0b 0xC3					
说明	播放文本编码格式为“Unicode”的文本“字音天下”, 不带背景音乐					

图 4-2 SYN6288 所能接受 Unicode 的帧格式

因此在进行 Unicode 转码之后, 我们仍然需要对输出到芯片的内容进行计算, 将长度, 命令字, 命令参数添加到待输出的帧中, 并进行异或校验, 从而获得有关 SYN6288 的完整的帧, 以进行语音播报。在 Arduino 中因 UTF-8 占用了三个字节, 所以汉字字符串的每一个汉字, 对应了字符串数组的 3 个下标。

4.4 XX3 模块的软件设计

PCF8574 拓展键盘

```

#include <Key.h>
#include <Keypad.h>
#include <Keypad_I2C.h>
#define I2CADDR 0x20
const byte ROWS = 4;

```



```
const byte COLS = 4;

char keys [ROWS] [COLS] = {
    {'0', '1', '2', '3'},
    {'4', '5', '6', '7'},
    {'8', '9', 'A', 'B'},
    {'C', 'D', 'E', 'F'}
};
byte rowPins [ROWS] = {0, 1, 2, 3};
byte colPins [COLS] = {4, 5, 6, 7};
Keypad_I2C keypad (makeKeymap (keys), rowPins, colPins, ROWS, COLS, I2CADDR,
PCF8574);
```

可以通过 `char key = keypad.getKey ();`来获得当前的按键

4.5 OLED 模块的软件设计

U8g2 是嵌入式设备的单色图形库。U8g2 支持单色 OLED 和液晶显示器。

U8g2 还包括 U8x8 库，仅文本输出（字符）设备，只有字体允许适合到 8x8 像素网格。直接写到显示屏上。微控制器中不需要缓冲器。

```
#include <U8x8lib.h>
void setup(void) {
    u8x8.begin();
    u8x8.setPowerSave(0);
    u8x8.setFlipMode(1);
}
```

可以通过调用 U8x8lib.h 来使用 Grove Beginner Kit 上的 OLED 屏幕，本次课设使用 OLED 屏幕进行当前温湿度、光强的显示。同时，根据矩阵键盘输入的内容，进行相关设置阈值的调节。

由于相关代码贯穿与各处，故详细设计请见附件。

4.6 各个传感模块的软件设计

4.6.1 温度传感器

温度传感器可以使用 DHT.h 库

因此获得其相应数值的代码如下

```
temp = dht.readTemperature();
```

4.6.2 光强传感器和土壤湿度传感器

光强传感器和土壤湿度传感器的调用方法相似，用于都使用了 Analog 模拟信号接口，故获取这相应数值的代码为：

```
Lum = analogRead(A6);
```

```
Hum = analogRead(A1);
```

4.7 继电器模块的软件设计

继电器模块连接了共用端与常开端，因此在声明 `pinMode(2, OUTPUT);`之后直接使用 `digitalWrite(2,LOW);`与 `digitalWrite(2,HIGH);`即可。

五、系统测试

当光线强度大于阈值时，步进电机转动，并进行语音播报。
使用手机手电筒照射：



图 5-1



图 5-2

当土壤湿度低于阈值时，继电器导通，水泵运作

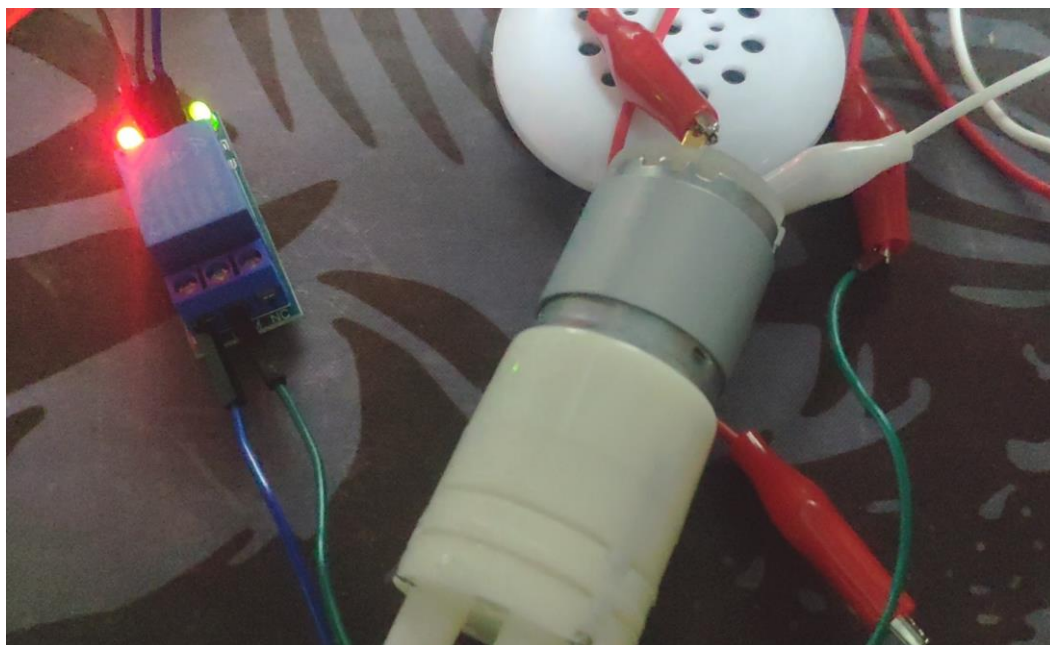


图 5-3

OLED 屏幕可以显示当前温度与光强，以及土壤湿度。



图 5-4



图 5-5

OLED 屏幕和键盘可以进行植物种类的选择



图 5-6



图 5-7



图 5-8

OLED 屏幕和键盘也可以直接进行阈值的设定



图 5-9



图 5-10

六、总结

通过本课程的学习，我对使用 AltiumDesigner 这一硬件设计工具的使用有所了解。懂得了 PCB 板在设计时所遵循的诸多原则。通过自己查阅数据手册设计 PCB 板的过程，让我感到受益匪浅。在另一方面，Arduino 这一软件开发工具，使我对于基于单片机的软件开发也有了更加深入的理解。这次课设令我感到收获颇丰。

附录

参考文献:

- [1] <https://www.makeriot2020.com/index.php/2020/10/05/using-i2c-with-a-4x4-matrix-keypad/> Using I2C with a 4×4 Matrix Keypad - Maker and IOT Ideas (makeriot2020.com)
- [2] Seeeduino Lotus 原理图和 PCB 图
- [3] ATMEGA328P 数据手册
- [4] SYN6288 数据手册
- [5] PCF8574 数据手册
- [6] ULN2003 数据手册
- [7] Grove Beginner Kit for Arduino Wiki 以及 Grove Beginner Kit 的原理图和 PCB 图。
- [8] <https://www.zhihu.com/question/23374078> Unicode 和 UTF-8 有什么区别?

计算机硬件类综合性课程设计

代码:

```
//Temperature and Humidity Sensor
#include "DHT.h"
#include <U8x8lib.h>
#include <SoftwareSerial.h>
#define DHTPIN 3 // what pin we're connected to
#define DHTTYPE DHT11 // DHT 11
DHT dht(DHTPIN, DHTTYPE);

U8X8_SSD1306_128X64_ALT0_HW_I2C u8x8(/* reset= */ U8X8_PIN_NONE);
volatile int timer1;
volatile int optionFlag = 0;
volatile int CoptionFlag = 0;

#include <Key.h>
#include <Keypad.h>
#include <Keypad_I2C.h>
#define I2CADDR 0x20
const byte ROWS = 4;
const byte COLS = 4;
//键盘对应的内容
char keys [ROWS] [COLS] = {
    {'0', '1', '2', '3'},
    {'4', '5', '6', '7'},
    {'8', '9', 'A', 'B'},
    {'C', 'D', 'E', 'F'}
};

SoftwareSerial SynSerial(6, 7);
char buffer[]="即将开始浇水当前室温二十六度";
char buffer2[]="当前光线强度过高即将进行遮光";
char buffer3[]="当前室内气温过高";
char buffer4[]="浇水结束";
int bufferflag = 0, buffer2flag = 0, buffer3flag = 0;
int flag = 0, counter = 0;
//语音合成，编码转换定义变量
#define SIZE (sizeof(buffer)/3)
#define SIZE2 (sizeof(buffer2)/3)
#define SIZE3 (sizeof(buffer3)/3)
#define SIZE4 (sizeof(buffer4)/3)
char character[SIZE * 2];
char character2[SIZE2 * 2];
char character3[SIZE3 * 2];
char character4[SIZE4 * 2];
char output[SIZE*2+6];
```

计算机硬件类综合性课程设计

```
char output2[SIZE2*2+6];
char output3[SIZE3*2+6];
char output4[SIZE3*2+6];

volatile int lum1 = 1500;
volatile int hum1 = 600;
char cOp[50];
int cOpi = 0;
volatile int waterflag = 0;
byte rowPins [ROWS] = {0, 1, 2, 3};
byte colPins [COLS] = {4, 5, 6, 7};
Keypad_I2C keypad (makeKeymap (keys), rowPins, colPins, ROWS, COLS, I2CADDR, PCF8574);

//进行温度计数时，修改 buffer 的后三个汉字
void numbers()
{
    int tem;
    char character[]="一";
    tem = (int)dht.readTemperature();
    //根据当前温度，修改汉字的内容
    if(tem%10==0){
        strcpy(character,"为");
        for(int i = 0;i<3;i++){
            buffer[sizeof(buffer)-11-i]=character[2-i];
        }
        strcpy(character,"十");
        for(int i = 0;i<3;i++){
            buffer[sizeof(buffer)-5-i]=character[2-i];
        }
        switch(tem/10){
            case 1:
                strcpy(character,"一");
                for(int i = 0;i<3;i++){
                    buffer[sizeof(buffer)-8-i]=character[2-i];
                }
                Serial.print(buffer);
                break;
            case 2:
                strcpy(character,"二");
                for(int i = 0;i<3;i++){
                    buffer[sizeof(buffer)-8-i]=character[2-i];
                }
                break;
            case 3:
                strcpy(character,"三");
```


计算机硬件类综合性课程设计

```
        for(int i = 0;i<3;i++){
            buffer[sizeof(buffer)-8-i]=charactor[2-i];
        }
        break;
    case 4:
        strcpy(charactor,"四");
        for(int i = 0;i<3;i++){
            buffer[sizeof(buffer)-8-i]=charactor[2-i];
        }
        break;
    }
}
else if(tem/10!=1){
    strcpy(charactor,"十");
    for(int i = 0;i<3;i++){
        buffer[sizeof(buffer)-8-i]=charactor[2-i];
    }
    switch(tem/10){
        case 1:
            strcpy(charactor,"一");
            for(int i = 0;i<3;i++){
                buffer[sizeof(buffer)-11-i]=charactor[2-i];
            }
            Serial.print(buffer);
            break;
        case 2:
            strcpy(charactor,"二");
            for(int i = 0;i<3;i++){
                buffer[sizeof(buffer)-11-i]=charactor[2-i];
            }
            break;
        case 3:
            strcpy(charactor,"三");
            for(int i = 0;i<3;i++){
                buffer[sizeof(buffer)-11-i]=charactor[2-i];
            }
            break;
        case 4:
            strcpy(charactor,"四");
            for(int i = 0;i<3;i++){
                buffer[sizeof(buffer)-11-i]=charactor[2-i];
            }
            break;
    }
}
switch(tem%10){
```

```
case 1:
strcpy(charactor,"一");
for(int i = 0;i<3;i++){
    buffer[sizeof(buffer)-5-i]=charactor[2-i];
}
Serial.print(buffer);
break;
case 2:
strcpy(charactor,"二");
for(int i = 0;i<3;i++){
    buffer[sizeof(buffer)-5-i]=charactor[2-i];
}
break;
case 3:
strcpy(charactor,"三");
for(int i = 0;i<3;i++){
    buffer[sizeof(buffer)-5-i]=charactor[2-i];
}
break;
case 4:
strcpy(charactor,"四");
for(int i = 0;i<3;i++){
    buffer[sizeof(buffer)-5-i]=charactor[2-i];
}
break;
case 5:
strcpy(charactor,"五");
for(int i = 0;i<3;i++){
    buffer[sizeof(buffer)-5-i]=charactor[2-i];
}
break;
case 6:
strcpy(charactor,"六");
for(int i = 0;i<3;i++){
    buffer[sizeof(buffer)-5-i]=charactor[2-i];
}
break;
case 7:
strcpy(charactor,"七");
for(int i = 0;i<3;i++){
    buffer[sizeof(buffer)-5-i]=charactor[2-i];
}
break;
case 8:
strcpy(charactor,"八");
```

```
for(int i = 0;i<3;i++){
    buffer[sizeof(buffer)-5-i]=charactor[2-i];
}
break;
case 9:
strcpy(charactor,"九");
for(int i = 0;i<3;i++){
    buffer[sizeof(buffer)-5-i]=charactor[2-i];
}
break;
}
}
else if(tem/10 == 1){
    strcpy(charactor,"为");
    for(int i = 0;i<3;i++){
        buffer[sizeof(buffer)-11-i]=charactor[2-i];
    }
    strcpy(charactor,"十");
    for(int i = 0;i<3;i++){
        buffer[sizeof(buffer)-5-i]=charactor[2-i];
    }
switch(tem%10){
    case 1:
        strcpy(charactor,"一");
        for(int i = 0;i<3;i++){
            buffer[sizeof(buffer)-5-i]=charactor[2-i];
        }
        Serial.print(buffer);
        break;
    case 2:
        strcpy(charactor,"二");
        for(int i = 0;i<3;i++){
            buffer[sizeof(buffer)-5-i]=charactor[2-i];
        }
        break;
    case 3:
        strcpy(charactor,"三");
        for(int i = 0;i<3;i++){
            buffer[sizeof(buffer)-5-i]=charactor[2-i];
        }
        break;
    case 4:
        strcpy(charactor,"四");
        for(int i = 0;i<3;i++){
            buffer[sizeof(buffer)-5-i]=charactor[2-i];
```

```
    }
    break;
case 5:
    strcpy(charactor,"五");
    for(int i = 0;i<3;i++){
        buffer[sizeof(buffer)-5-i]=charactor[2-i];
    }
    break;
case 6:
    strcpy(charactor,"六");
    for(int i = 0;i<3;i++){
        buffer[sizeof(buffer)-5-i]=charactor[2-i];
    }
    break;
case 7:
    strcpy(charactor,"七");
    for(int i = 0;i<3;i++){
        buffer[sizeof(buffer)-5-i]=charactor[2-i];
    }
    break;
case 8:
    strcpy(charactor,"八");
    for(int i = 0;i<3;i++){
        buffer[sizeof(buffer)-5-i]=charactor[2-i];
    }
    break;
case 9:
    strcpy(charactor,"九");
    for(int i = 0;i<3;i++){
        buffer[sizeof(buffer)-5-i]=charactor[2-i];
    }
    break;
}
}
//switch
```

//步进电机部分

void clockwise(int num)

```
{
    for (int count = 0; count < num; count++)
    {
        for (int i = 8; i < 12; i++)//一相励磁，正转
        {
```

计算机硬件类综合性课程设计

```
        digitalWrite(i, HIGH); // 逐个写入高电平
        delay(3);
        digitalWrite(i, LOW);
    }
}

void anticlockwise(int num)
{
    for (int count = 0; count < num; count++)
    {
        for (int i = 11; i > 7; i--) // 一相励磁，反转
        {
            digitalWrite(i, HIGH); // 逐个写入高电平
            delay(3);
            digitalWrite(i, LOW);
        }
    }
}

void goIntoOptions(char key) { // OLED 屏幕设置界面
    if(optionFlag == 0){
        u8x8.clearDisplay();
    }
    u8x8.setFont(u8x8_font_chroma48medium8_r);
    u8x8.setCursor(0, 33);
    u8x8.print("chooseplanttype:"); // 选择植物类型
    u8x8.setCursor(0, 50);
    if(optionFlag == 0){
        optionFlag = 1; // 此标志位用于表示当前设置界面所显示的内容，以便于对左右移进行响应

        u8x8.print("Desert Plant    >"); // 沙漠植物
    }
    if(key=='E') { // 右移
        if(optionFlag == 1){
            u8x8.print("<tropical plant>"); // 热带植物
            optionFlag = 2;
        }
        else if(optionFlag == 2){
            u8x8.print("<Temperate plant"); // 温带植物
            optionFlag = 3;
        }
    }
    if(key=='C') { // 左移
        if(optionFlag == 3){
```

```

        u8x8.print("<tropical plant>");
        optionFlag = 2;
    }
    else if(optionFlag == 2){
        u8x8.print("Desert Plant    >");
        optionFlag = 1;
    }
}
if(key=='D'){//确定
    valuesChange(optionFlag);
    u8x8.clearDisplay();
    optionFlag = 0;
}
u8x8.refreshDisplay();
}

void CustomizedOptions(char key){//OLED 屏幕设置界面，直接调节，光强与湿度阈值
    if(CoptionFlag == 0){
        u8x8.clearDisplay();
    }
    u8x8.setFont(u8x8_font_chroma48medium8_r);
    u8x8.setCursor(0, 33);
    u8x8.print("enter Lum:");
    u8x8.setCursor(0, 50);
    if(CoptionFlag == 0){
        CoptionFlag = 1;//表示当前设置的进度
    }
    if(CoptionFlag != 0){
        if(key>='0'&&key<='9'){
            u8x8.setCursor(0, 50);
            cOp[cOpi]=key;//将输入的内容显示到 OLED 屏幕上
            u8x8.print(cOp);
            cOpi++;//cOpi 为当前输入的字符在数值中应显示的位置
        }
    }
    if(key=='D'){//确定
        if(CoptionFlag==1){
            if(cOp[0]!='\0'){
                lum1 = atoi(cOp);//将输入的内容转换成数字进行存储
            }
            else{
                u8x8.print("illegal input");
                delay(3000);
            }
        }
        for(int i = 0 ; i<50 ; i++){
            cOp[i] = '\0';

```

计算机硬件类综合性课程设计

```
    }//清空字符串，方便再次设置
    cOpi = 0;//清空字符串的标志位
    CoptionFlag=2;//进入湿度设置
    u8x8.clearDisplay();
    u8x8.setCursor(0, 33);
    u8x8.print("enter Hum:");
    }
    else if(CoptionFlag==2){
        if(cOp[0]!='\0'){
            hum1 = atoi(cOp);//将输入的内容转换成数字进行存储
        }
        else{
            u8x8.print("illegal input");
            delay(3000); //如果输入有问题，则显示非法输入，设置结束
        }
        for(int i = 0 ; i<50 ; i++){
            cOp[i] = '\0';//清空字符串，方便再次设置
        }
        cOpi = 0;//清空字符串的标志位
        u8x8.clearDisplay();
        CoptionFlag = 0;
    }
}
}
u8x8.refreshDisplay();
}

void valuesChange(int optionFlag){
    switch(optionFlag){
        case 1:
            lum1 = 1700;
            hum1 = 300;
        case 2:
            lum1 = 1500;
            hum1 = 600;
        case 3:
            lum1 = 1500;
            hum1 = 500;
        default:
            break;
    }
}

void UTF8_To_Unicode(char*buf,int which){
```

计算机硬件类综合性课程设计

```
if(which == 1){
    for (int i =0;i<SIZE;i=i+1) {
        character[i*2]=((buf[i*3]&0xF)<<4)+((buf[i*3+1]>>2)&0xF);//将三个字节的 UTF-8，转换为两个字节的
Unicode
        character[i*2+1]=((buf[i*3+1]&0x3)<<6)+(buf[i*3+2]&0x3F);
    }
    output[0]=0xFD;//根据帧格式进行输出
    output[1]=(SIZE*2+3)>>8;
    output[2]=((SIZE*2+3)&0xFF);
    output[3]=0x01;
    output[4]=0x03;
    for (int i=0;i<SIZE*2;i++) {
        output[i+5]=character[i];//将 Unicode 写入
    }
    output[SIZE*2+5]=output[0];
    for (int i=1;i<SIZE*2+5;i++) {
        output[SIZE*2+5]=output[SIZE*2+5] ^ output[i];//计算异或校验和
    }

    for (int i =0;i<SIZE*2+6;i++) {
        SynSerial.write(output[i]);//输出
    }
}
else if(which == 2){
    for (int i =0;i<SIZE2;i=i+1) {
        character2[i*2]=((buf[i*3]&0xF)<<4)+((buf[i*3+1]>>2)&0xF);
        character2[i*2+1]=((buf[i*3+1]&0x3)<<6)+(buf[i*3+2]&0x3F);
    }
    output2[0]=0xFD;
    output2[1]=(SIZE2*2+3)>>8;
    output2[2]=((SIZE2*2+3)&0xFF);
    output2[3]=0x01;
    output2[4]=0x03;
    for (int i=0;i<SIZE2*2;i++) {
        output2[i+5]=character2[i];
    }
    output2[SIZE2*2+5]=output2[0];
    for (int i=1;i<SIZE2*2+5;i++) {
        output2[SIZE2*2+5]=output2[SIZE2*2+5] ^ output2[i];
    }

    for (int i =0;i<SIZE2*2+6;i++) {
        SynSerial.write(output2[i]);
    }
}
```


计算机硬件类综合性课程设计

```
else if(which == 3){
    for (int i =0;i<SIZE3;i=i+1) {
        character3[i*2]=((buf[i*3]&0xF)<<4)+((buf[i*3+1]>>2)&0xF);
        character3[i*2+1]=((buf[i*3+1]&0x3)<<6)+(buf[i*3+2]&0x3F);
    }
    output3[0]=0xFD;
    output3[1]=(SIZE3*2+3)>>8;
    output3[2]=((SIZE3*2+3)&0xFF);
    output3[3]=0x01;
    output3[4]=0x03;
    for (int i=0;i<SIZE3*2;i++) {
        output3[i+5]=character3[i];
    }
    output3[SIZE3*2+5]=output3[0];
    for (int i=1;i<SIZE3*2+5;i++) {
        output3[SIZE3*2+5]=output3[SIZE3*2+5] ^ output3[i];
    }

    for (int i =0;i<SIZE3*2+6;i++) {
        SynSerial.write(output3[i]);
    }
}
else if(which == 4){
    for (int i =0;i<SIZE4;i=i+1) {
        character4[i*2]=((buf[i*3]&0xF)<<4)+((buf[i*3+1]>>2)&0xF);
        character4[i*2+1]=((buf[i*3+1]&0x3)<<6)+(buf[i*3+2]&0x3F);
    }
    output4[0]=0xFD;
    output4[1]=(SIZE4*2+3)>>8;
    output4[2]=((SIZE4*2+3)&0xFF);
    output4[3]=0x01;
    output4[4]=0x03;
    for (int i=0;i<SIZE4*2;i++) {
        output4[i+5]=character4[i];
    }
    output4[SIZE4*2+5]=output4[0];
    for (int i=1;i<SIZE4*2+5;i++) {
        output4[SIZE4*2+5]=output4[SIZE4*2+5] ^ output4[i];
    }

    for (int i =0;i<SIZE4*2+6;i++) {
        SynSerial.write(output4[i]);
    }
}
}
```

```
void Watering(int hum){//浇水操作
    if(hum<hum1){
        if(bufferflag==0){
            numbers();//温度的输出
            UTF8_To_Unicode(buffer,1);//转码，语音合成
            bufferflag=1;
        }
        digitalWrite(2,LOW);//继电器开
        waterflag=1;
    }
    else if(hum>=hum1){//达到阈值
        if(bufferflag==1){
            UTF8_To_Unicode(buffer4,4);//转码，语音合成
            bufferflag=0;
        }
        digitalWrite(2,HIGH);//继电器关
        waterflag=0;
    }
}

void setup(void) {
    for (int i = 8; i < 12; i++) {
        pinMode(i, OUTPUT);
    }
    Wire.begin();//初始化 IIC
    pinMode(2, OUTPUT);
    keypad.begin (makeKeymap (keys));//初始化键盘
    SynSerial.begin(9600);//初始化软串口
    Serial.begin(9600); //初始化 debug 串口
    Serial.println(sizeof(buffer)-2);
    dht.begin(); //初始化传感器
    u8x8.begin(); //设置 OLED
    u8x8.setPowerSave(0);
    u8x8.setFlipMode(1);
}

void loop(void) {
    float temp;
    int Lum, Hum;
    char key = keypad.getKey ();
    Lum = analogRead(A6);//接收当前室内温湿度指标
    Hum = analogRead(A1);
    temp = dht.readTemperature();
    Watering(Hum);
}
```

计算机硬件类综合性课程设计

```
if(temp>=35&&buffer3flag==0){
    UTF8_To_Unicode(buffer3,3);
    buffer3flag=1;
}
else{
    buffer3flag=0;
}
if (key) {
    Serial.println (key);
    if(key == 'A' || CoptionFlag != 0)//进入设置界面
        CustomizedOptions(key);
    if(key == 'F' || optionFlag != 0){//进入设置界面
        Serial.println (key);
        goIntoOptions(key);
        Serial.println (optionFlag);
    }
}
if(optionFlag != 0){
    Serial.println(optionFlag);
    goIntoOptions('N');
}
if(timer1<25&&optionFlag == 0&&CoptionFlag == 0){//分别在 OLED 上显示温度与光强，湿度
    timer1++;//温度与光强的显示
    temp = dht.readTemperature();
    Lum = analogRead(A6);
    u8x8.setFont(u8x8_font_chroma48medium8_r);
    u8x8.setCursor(0, 33);
    u8x8.print("Temp:");
    u8x8.print(temp);
    u8x8.print("C");
    u8x8.setCursor(0,50);
    u8x8.print("Lum:");
    u8x8.print(Lum);
    u8x8.print(" ");
    //Serial.println(temp);
    //Serial.println(Lum);
    u8x8.refreshDisplay();
}
else if(optionFlag == 0&&CoptionFlag == 0){
    if(timer1 == 25){
        u8x8.clearDisplay();//清屏
    }
    timer1++;//土壤湿度的显示
    u8x8.setFont(u8x8_font_chroma48medium8_r);
    u8x8.setCursor(0,41);
```

```
u8x8.print("Hum:");
u8x8.print(Hum);
//Serial.println(temp);
//Serial.println(Hum);
u8x8.refreshDisplay();
if(timer1==50){
    timer1 = 0;
}
}
if(flag == 0&&Lum>=lum1){
    if(buffer2flag==0){
        UTF8_To_Unicode(buffer2,2);//转码
        buffer2flag=1;
    }
    clockwise(512);//正转
    flag = 1;
}
else if(flag == 1 && Lum<lum1){
    buffer2flag=0;
    anticlockwise(512);//反转
    flag = 0;
}
Serial.println(timer1);
delay(10);
}
```