

自动浇花系统

土壤湿度传感器模块介绍

土壤湿度传感器模块有两个铜条是传感器探头。 将它们插入土壤时,它们可以检测到水分。 土壤湿润,导电性越好,反映出它们之间的电阻越低。土壤干燥,导电性就相对差一点,因此他们之间的电阻越高。 它是模拟传感器,因此我们通过模拟输入获得电压值。也可以通过数字接口获取高低电平,一般我们使用的过程中用获取的模拟值。因为土壤的湿度可以分为几个等级,当我们使用土壤湿度传感器做一个自动浇花系统的时候,将方便的使用。



土壤湿度传感器模块实物图

- 该传感器适用干土壤湿度检测。
- 模块上的蓝色电位器用于调节土壤湿度阈值,顺时针调节时控制湿度较大,逆时针调节较小。
- 数字输出 D0 可以直接连接到微控制器,检测高低电平,从而检测土壤湿度。
- 模拟输出 AO(0~1023) 可以链接到 AD 模块,通过 AD 转换,可以更准确地获得土壤湿度值。

引脚说明

- VCC: 连接到 3.3 V-5 V。
- GND: 连接到 GND。
- DO: 数字输出接口



● AO: 模拟输出接口

继电器模块介绍

典型的继电器结构原理是继电器有一个输入回路,一般接低压电源,有一个输出回路,一般接高压电源。输入回路中有一个电磁铁线圈,当输入回路有电流通过,电磁铁产生磁力,吸力使输出回路的触点接通,则输出回路导电(通)。当输入回路无电流通过,电磁铁失去磁力,输出回路的触点弹回原位,断开,则输出回路断电(断)。



继电器模块实物图

实验目的

- 学习土壤湿度传感器模块工作原理:
- 学习继电器模块工作原理
- 使用 Arduino UNO 板控制继电器模块和土壤湿度传感器模块实现自动浇花的功能

实验原理

从上面的传感器,我们可以看到土壤湿度传感器模块两个铜条是传感器探头。 将它们插入土壤时,它们可以检测到水分。 土壤湿润,导电性越好,反映出它们之间的电阻越低。 它是模拟传感器,因此我们通过模拟输出获得电压值。 土壤越干燥,电压值越大,因为探针之间的电阻越来越高。主控板检测土壤的湿度,如果土壤的湿度不够,则控制继电器打开水泵给花浇水,当给花浇水后土壤湿度达到标准湿度值则关闭水泵,停止浇水,使用 LCD 实时显示土壤的湿度。

组件清单

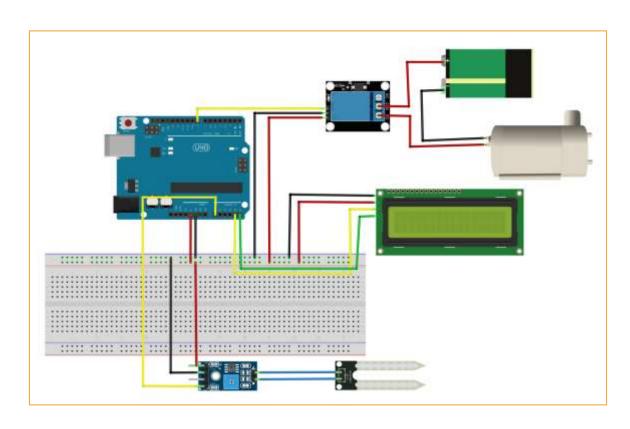
- Keywish Arduino UNO R3 主板
- 面包板



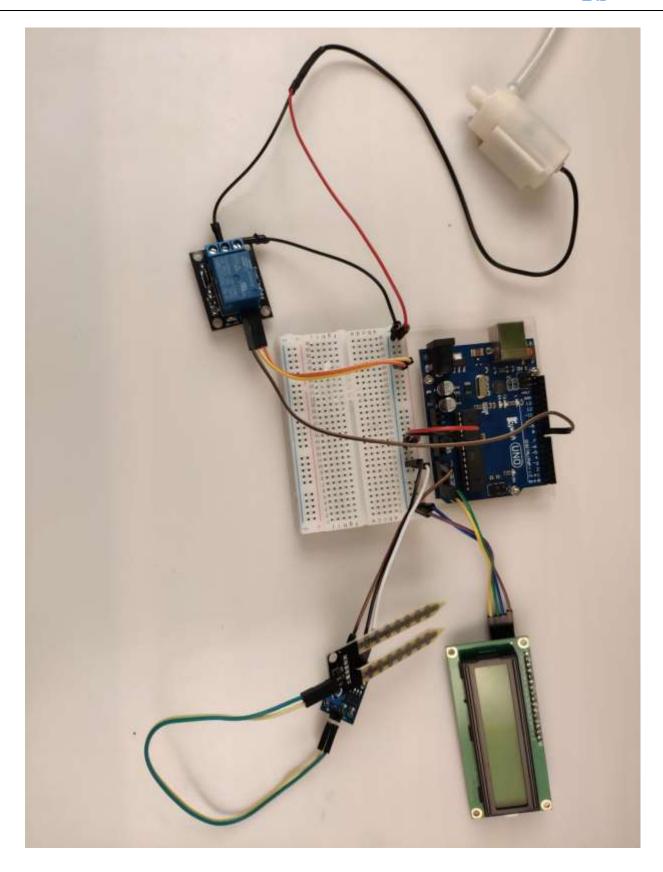
- USB 数据线
- LCD1602 模块 * 1
- 土壤湿度传感器 * 1
- 继电器模块*1
- 若干跳线

接线

LCD 显示屏 PCF8574T	Arduino UNO 板
GND	GND
VCC	5V
SDA	A4
SCL	A5
土壤湿度传感器转接板	Arduino UNO板
GND	GND
VCC	5V
DO	A0
继电器模块	Arduino UNO板
GND (-)	GND
VCC (+)	5V
S(IN)	8









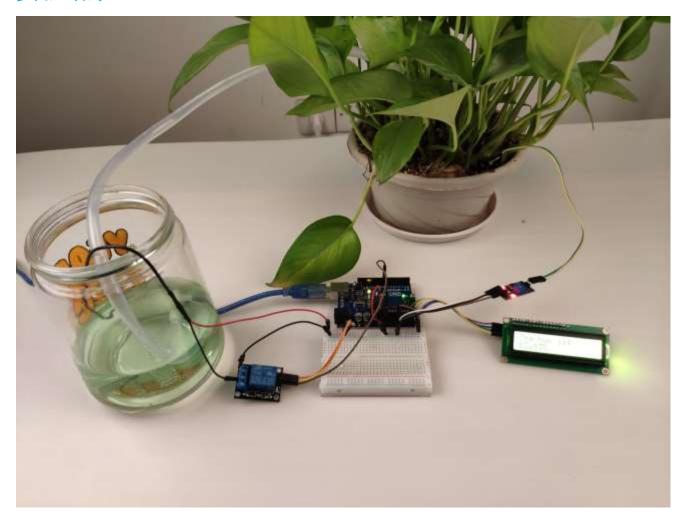
Arduino IDE 编程程序

```
#include <Wire.h>
#include "LCD.h"
#include "LiquidCrystal I2C.h"
int analogPin=A0;
int WaterPumpPin=8;
float i=0;
float j=0;
LiquidCrystal I2C lcd(0x27,2,1,0,4,5,6,7); // 0x27 is the I2C bus address for an unmodified
backpack
void setup()
{ // activate LCD module
 lcd.begin (16,2); // for 16 x 2 LCD module
 lcd.setBacklightPin(3,POSITIVE);
 lcd.setBacklight(HIGH);
 pinMode (analogPin, INPUT);
 pinMode (WaterPumpPin,OUTPUT);
 digitalWrite(WaterPumpPin,LOW);
 Serial.begin(9600);
}
void loop() {
   float data=analogRead(analogPin);
     Serial.println(data);
   i=data/1023;
   j = (1-i) *100;
   lcd.setCursor(0, 0);
   lcd.print("The hum is: ");
   // set the cursor to column 0, line 1
   // (note: line 1 is the second row, since counting begins with 0):
   lcd.setCursor(0, 1);
   // print the number of seconds since reset:
   lcd.print((float)j, 2);
   lcd.print("%");
   delay(200);
   if(data>800)
     digitalWrite(WaterPumpPin,HIGH);
    else
    {
```



```
digitalWrite(WaterPumpPin,LOW);
}
```

实验结果



mBlock 图形化编程程序

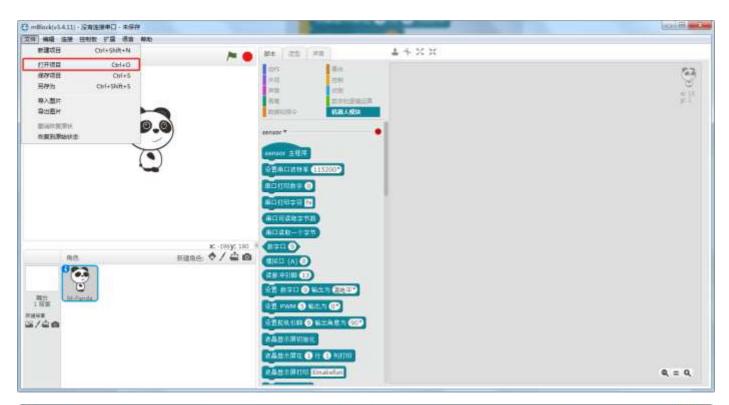
mBlock 编写自动浇花程序如下图所示:

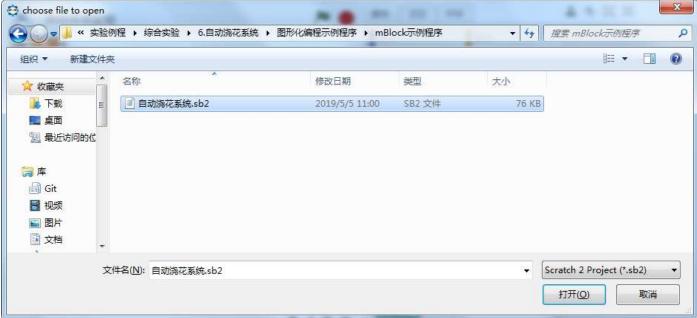


```
sensor 主程序
设置串口波特率 9600▼
液晶显示屏初始化
重复执行
 将 data ▼ 设定为 模拟口 (A) 0
 串口打印数字 data
 将 i▼ 设定为 (data )/ 1023
 将 j▼ 设定为 (1 - (data) * 100
 液晶显示屏在 1 行 1 列打印
 液晶显示屏打印 The hum is:
 液晶显示屏在 1) 行 2) 列打印
 液晶显示屏打印 ϳ
 液晶显示屏打印 %
 等待 0.2 秒
 如果 (data) > 800 ) 那么
  设置 数字口 8 输出为 高电平
  设置 数字口 8 输出为 低电平
```

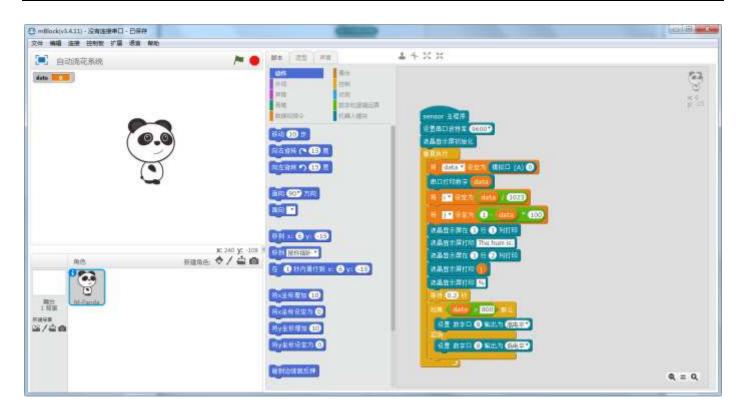
也可以直接打开已经写好的程序文件,它是一个.sb2 文件,以下是打开步骤:











Mixly 图形化编程程序

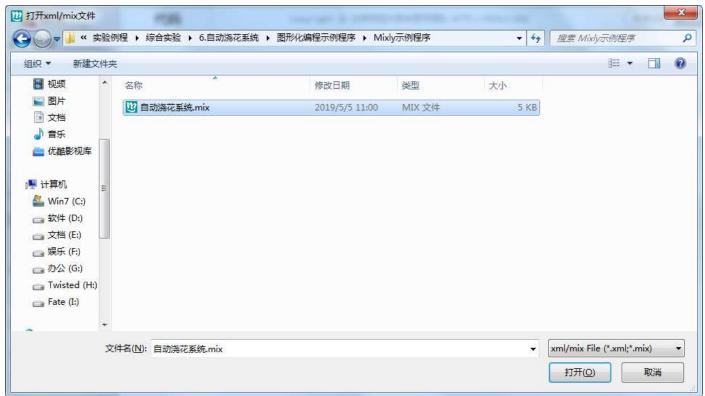
Mixly 编写自动浇花程序如下图所示:

```
初始化液晶显示屏
 引脚 (A0)
         设置 輸入 ▼
 设置 [8 高 ▼
 Serial v 波特率 9600
声明 data 为 小数 V 并赋值(
                     模拟输入 引脚 📗 🗚 🗸 🗸 🗸 🗸 🗸 🗘
Serial v 打印(自动换行)
                   data
声明 i 为 小数 🔻 并赋值
                    data ÷ 1 1023
声明 j 为 小数 🔻 并赋值
                                × * 100
                    1 -V (i
液晶显示屏 打印第1行
                "(the hum is: ")
               小数▼ j
延时 毫秒 🔻 200
🔯 如果
        data > * 800
执行
    设置 [8 高 ▼
否则
    设置 [8 低 ▼
```

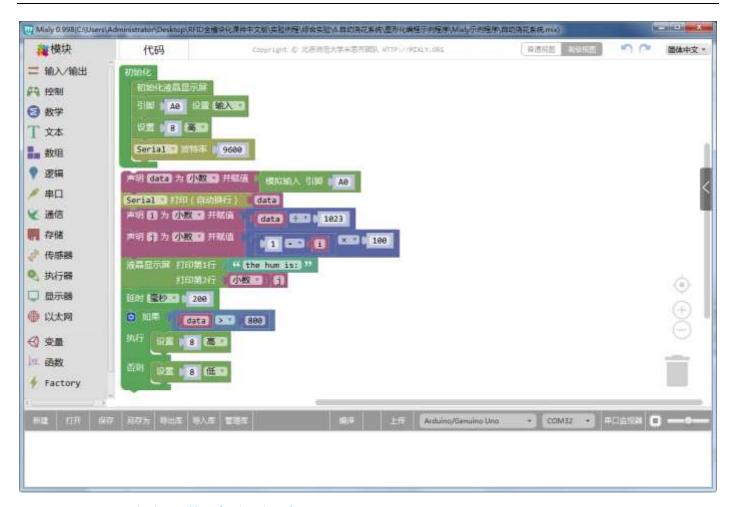


也可以直接打开已经写好的程序文件,它是一个.mix 文件,以下是打开步骤:









MagicBlock 图形化编程程序

MagicBlock 编写自动浇花系统实验程序如下图所示:



```
setup
 単行LCD1602初始化地址 0x27
 创建 全局 ▼ 变量 类型为 单箱度浮点值 ▼ 变量名为 i 变量
 创建 全局 ▼ 变量 类型为 单箱度浮点值 ▼ 变量名为 j 变量
 引脚 AD ▼ 模式 输入 ▼
 引脚 8 ▼ 模式 輸出 ▼
数字写 8 ▼ LOW ▼
 串目 Serial ▼ 波特率 9600 ▼
 创建 局部▼ 变量类型为 单糖度浮点值▼ 变量名为 (data) 变量
 设置变量 data 值为 模拟读 AO ▼
 串□ Serial ▼ 打印(自动换行) 获取变量 data 的值
 设置变量 i 值为 获取变量 data 的值 // 1023
 设置变量 j 值为 1 - 获取变量 i
 単行LCD1602光标 行 1 列 1
 串行LCD1682显示 the hum is:
 串行LCD1602光标 行 (2) 列 (1)
 事行LCD1602显示 %
 等等 2000 毫秒
 数字写 8 ▼
         HIGH -
  数字写 8 ▼
          LOW -
```



MagicBlock 打开步骤

也可以用 MagicBlock 直接打开已经做好的程序文件,它是一个.sb3 文件,以下是打开步骤:在 MagicBlock 界面菜单栏点击"文件","从电脑中上传",选择此程序打开

