Mushroom Grower V2

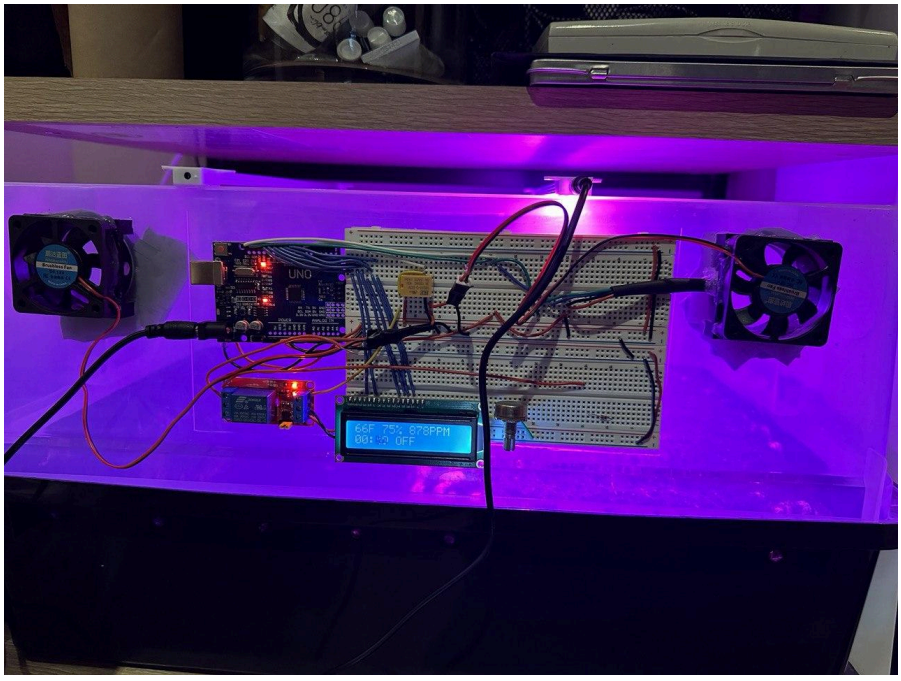
Electronic monitor and controller for a mushroom monotub

Background

A mushroom monotub is a large plastic tub that is modified to grow mushrooms in. Usually large tubs are bought from a store and modified; however there are also companies that make plastic tubs specifically for growing mushrooms. Mushrooms are typically very resilient, but different factors, such as the temperature, humidity, and carbon dioxide levels, can affect how quickly the mushroom grows and also the amount that grows.

Basics

The mushroom grower is a device used to monitor and control different aspects for a mushroom monotub. It measures and displays the temperature, humidity, and carbon dioxide levels inside of the tub. It also controls two outtake fans to ensure optimal carbon dioxide levels. It uses an Arduino Uno and is powered via a 12V DC power adapter. The fans are attached to the box via a custom designed mount that has space for a filter and space for the sensor's wires to fit through.



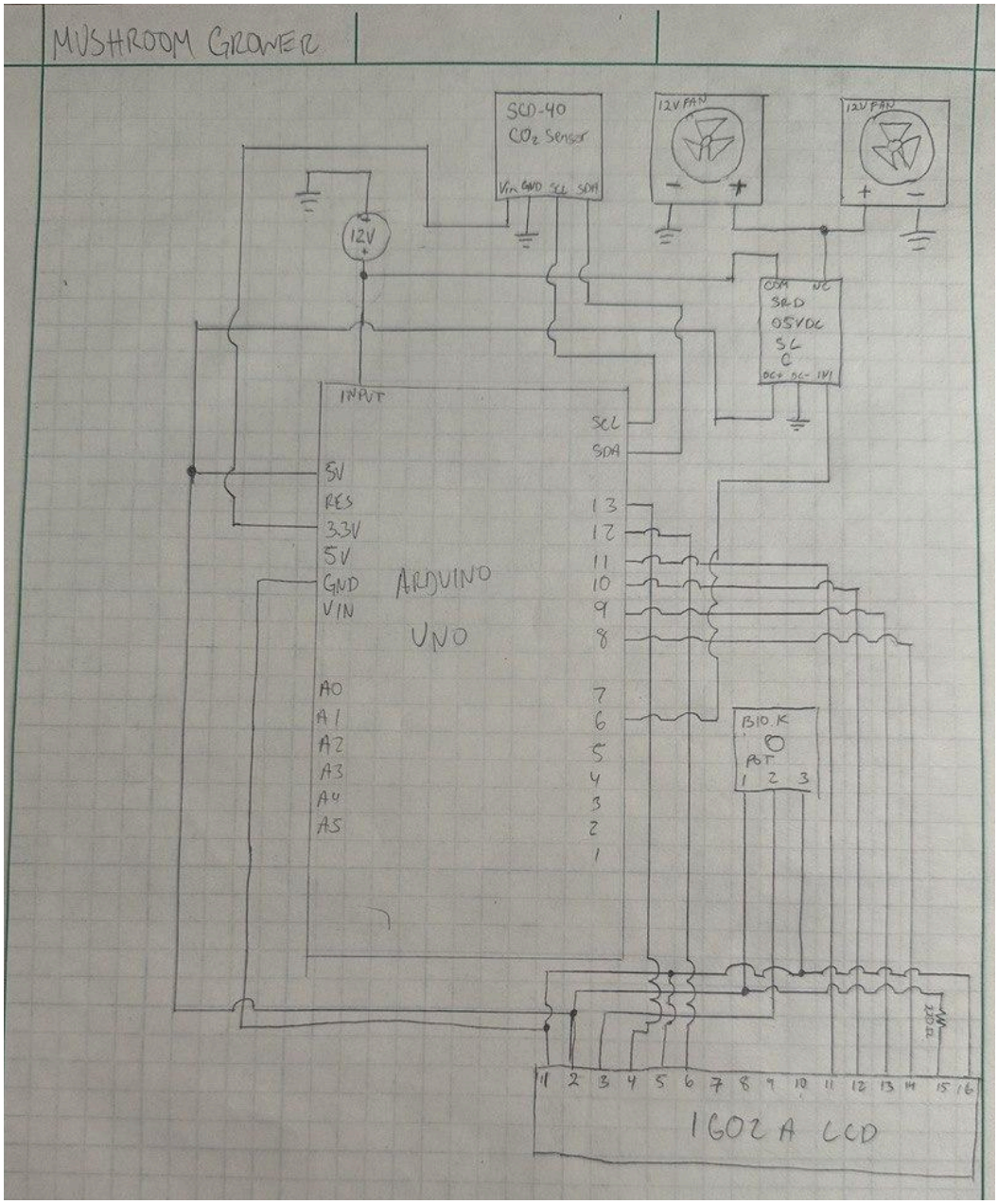
Operation

Once connected, the unit runs automatically. The fans can be turned on/off via a carbon dioxide threshold or they can be set on a timer. The LCD screen displays the temperature, humidity, and carbon dioxide levels. It also has a timer and an indicator for whether the fans are on or off. The sensor is mounted inside of the tub near the right fan.

Technical Details

The unit uses an Arduino Uno as the controller. The temperature, humidity, and carbon dioxide sensor is an SCD-40. The fans are 50mm x 50mm 12V DC brushless fans. The LCD is a 1602A LCD.

Circuit Diagram:



Arduino Code:

```
//Version 2.0. Using the CO2 sensor instead of the temp/humidity sensor
#include <LiquidCrystal.h>
#include <SparkFun_SCD4x_Arduino_Library.h>
#include <stdio.h>
#include <string.h>
#include <Wire.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 13, en = 12, d4 = 11, d5 = 10, d6 = 9, d7 = 8;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

//Set up the sensor
SCD4x CO2Sensor;

//Set up extra variables
int timeOff = 60;  //The number of seconds the fan should be off for. 540 seconds = 9
minutes off
int timeOn = 60;   //The number of seconds the fan should be off for. 60 seconds = 1
minute on
bool fanOff = true;
int fanPin = 6;
int numSecondsRemaining = timeOn;
int minutes, seconds;
int temp, carbon, humidity;
float tempF;

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);

  // Set up the pin to output signal to fan.
  pinMode(fanPin, OUTPUT);
  digitalWrite(fanPin, HIGH);

  //Setting up the CO2 sensor
  Wire.begin();
  if (CO2Sensor.begin() == false) {
    lcd.print("CO2 NOT INIT");
    while (1)
      ;
```

```
  }
}

void loop() {
  print_CO2();
  // print_lux();
  set_fan();
  print_time();
  delay(1000);
}

void print_CO2() {
  temp = CO2Sensor.getTemperature();

  tempF = (float)temp * 1.8 + 32;
  carbon = CO2Sensor.getCO2();
  humidity = CO2Sensor.getHumidity();
  // write the strings to the top line
  lcd.setCursor(0, 0);
  lcd.print("                ");  //Clear the line in case of artifacts
  lcd.setCursor(0, 0);

  //print temp
  lcd.print(tempF, 0);
  lcd.print("F ");
  //print humidity
  lcd.print(humidity);
  lcd.print("% ");
  //print CO2 ppm
  lcd.print(carbon);
  lcd.print("PPM");
}

void print_lux() {
}

void set_fan() {
  // Calculate number of minutes and seconds remaining
  numSecondsRemaining = numSecondsRemaining - 1;
  if (numSecondsRemaining == 0) {
    if (fanOff) {
      numSecondsRemaining = timeOff;
      digitalWrite(fanPin, LOW);
```

```
    } else {
      numSecondsRemaining = timeOn;
      digitalWrite(fanPin, HIGH);
    }
    fanOff = !fanOff;
    //Re-initializing lcd to prevent jumbled letters that randomly occur when fan is
switched on/off. Need to add a capacitor to the power and ground lines to help prevent
this
    //Caused by temporary disconnect which resets the encoding for the screen.
    delay(1000);
    lcd.begin(16, 2);
  }
}

void print_time() {
  //prints the time and then status of the fan
  //Write the strings to the bottom line
  lcd.setCursor(0, 1);
  lcd.print("                ");
  lcd.setCursor(0, 1);

  //Format the time into mm:ss
  seconds = numSecondsRemaining % 60;
  minutes = (numSecondsRemaining - seconds) / 60;
  if (minutes < 10) {
    lcd.print("0");
  }
  lcd.print(minutes);
  lcd.print(":");

  if (seconds < 10) {
    lcd.print("0");
  }
  lcd.print(seconds);

  // print the status of the fan
  if (fanOff) { lcd.print(" OFF"); }

  if (!fanOff) { lcd.print(" ON");  }
}
```

Future Plans

Although humidity is monitored, there is currently no way to increase humidity if needed aside from manually spraying the mushrooms with water. Adding in a humidifier will greatly increase the automation of the unit. It is currently being designed. Related, adding in a soil humidity sensor will give more accurate readings of the overall state of the mushroom.

As different types of mushrooms have different optimal growing conditions, implementing presets would be useful when switching the type of mushroom grown in the monotub. A settings menu would also be useful for changing parameters (such as the fan on/off timing, the humidifier threshold, etc.) without having to manually reprogram the Arduino each time.

Miscellaneous Pictures

Pink oyster mushrooms and lion's mane mushrooms grown using the Mushroom Grower in a monotub. Partially harvested.