Project Report

Introduction

Controlling the humidity level is necessary for a wide variety of settings, such as residential homes, commercial establishments, and conservatories. Humidity controllers are electronic devices that can be used to maintain a certain amount of humidity in space by relegating a humidifier or dehumidifier to a secondary role. For the purpose of this project, a DHT sensor will send its readings to an Arduino Uno, which will then use those readings to operate a humidifier.

Objective

The creation of a humidity device that, when combined with the readings from a differentialhygrometer-thermostat (DHT) sensor, will allow for the effective regulation of a humidifier is one of the primary objectives of this project. The amounts of humidity that are present in the system should be maintained within a specific limit.

Equipment Used

- Arduino Uno: A microcontroller that is equipped with sets of digital and analog I/O pins that can be interfaced with others boards or circuits.

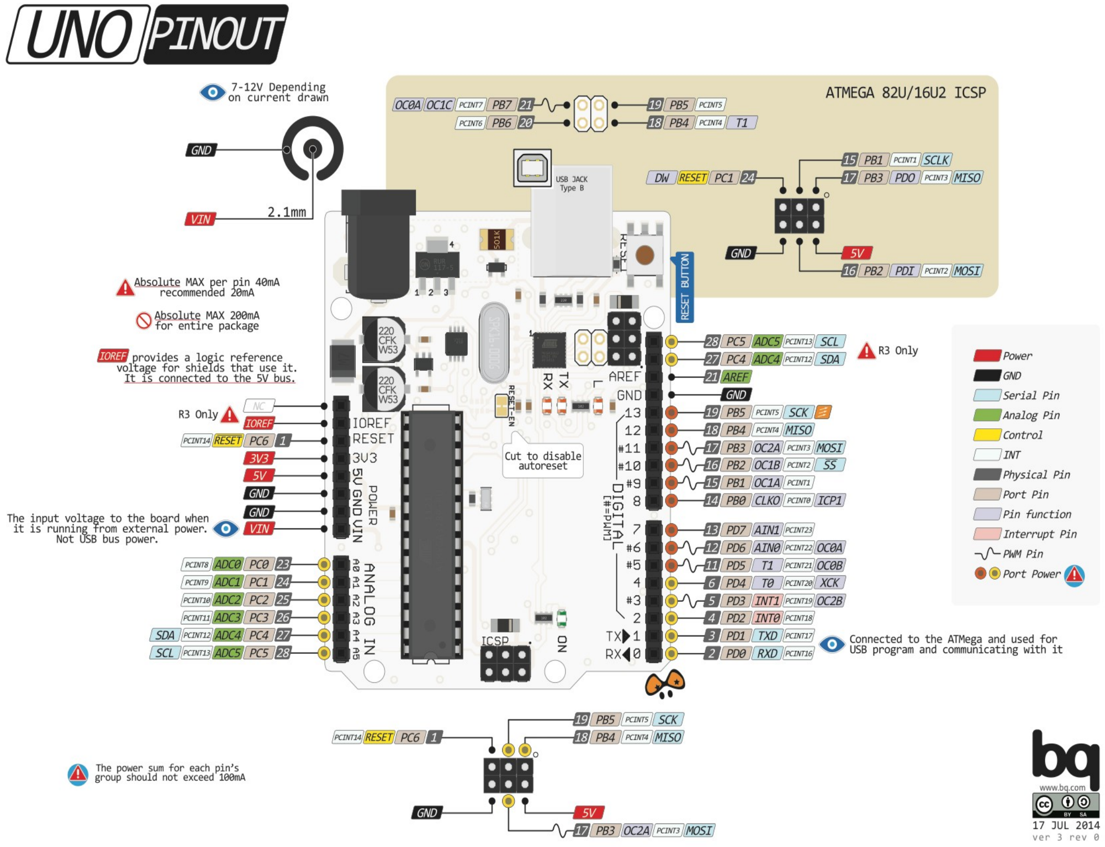


Figure 1.1: Arduino Uno's pin functions. (source: Wikipedia)

- DHT11 sensor: Device that uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin.
- 4 Relay module: An electrical switch that is operated by an electromagnet. It is triggered on a separate low-power signal from the microcontroller.
- Humidifier.
- Breadboard.
- Jumper wires.
- Potentiometer: A three-terminal resistor that is used to control electrical devices such as volume, display, etc..

- LCD display.

Procedure

1. Set up the DHT11 and connect with the Arduino Uno using jumper wires. The sensor has three pins: VCC, GND and DATA (from left to right). Connect the VCC and GND pins to the 5V and GND, corresponding. Thus, connect DATA pin to D2 on the Arduino Uno.

2. Connect the LCD display to the existed setup and using the potentiometer to adjust resolution.

LCD pins

+) Vss -> Gr

+) Vdd -> 5V

+) V0 -> Potentiometer

+) RS -> D7

+) RW -> Gr

+) E -> D8

+) D4 -> D9

+) D5 -> D10

+) D6 -> D11

+) D7 -> D12

+) A -> 5V

+) K -> Gr

Potentiometer

+) Left-most pin -> 5V

+) Mid pin -> V0
+) Right-most pin -> Gr

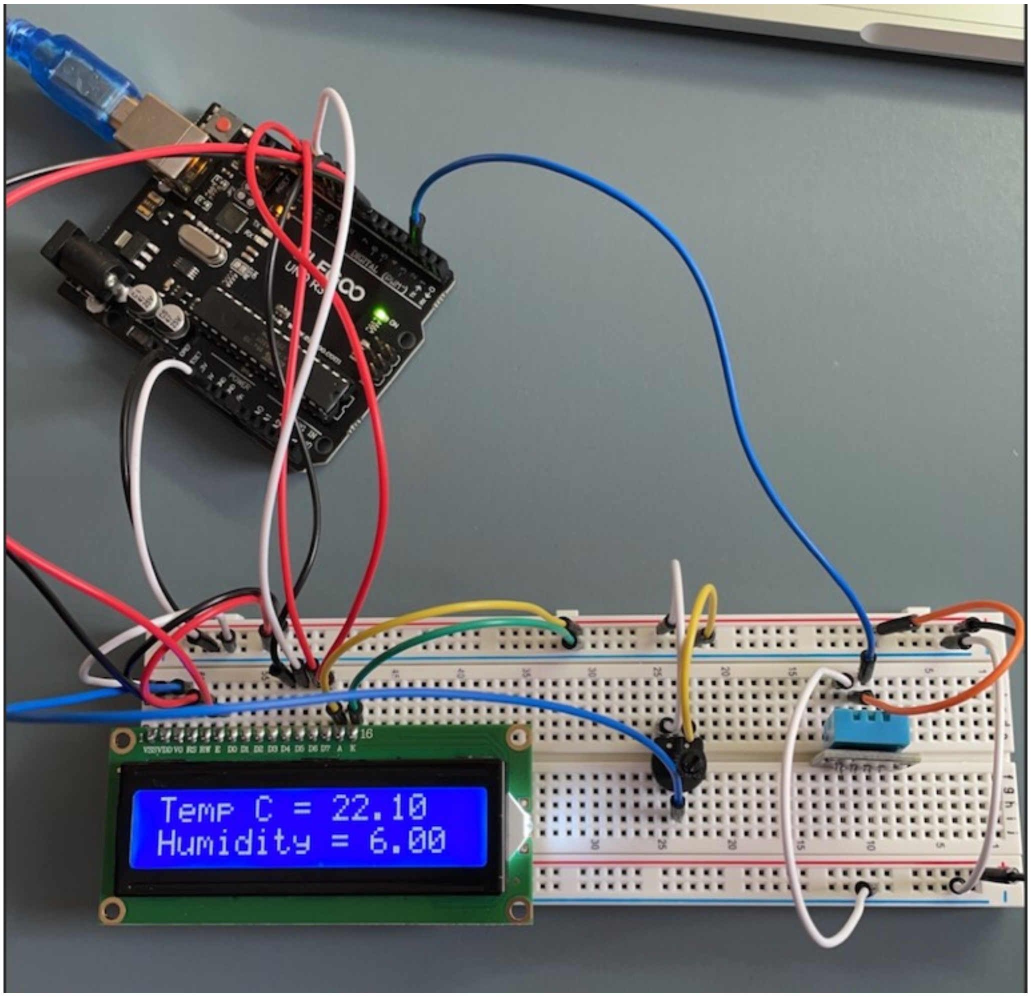


Figure 1.2: LCD and sensor connection with the Arduino

3. Connect the relay module to the microcontroller: I use D3 (digital pin 3) on the Arduino Uno as output which send an input signal to the pin IN1 on the relay module that indicates the first relay on the module will be used. Then, I connected the Vcc pin on the relay module to the 5V on the breadboard alongside with GND pin to the GND on the breadboard. Now, the relay will

be turned on if the result from the captured from the sensor. From this point, the relay will act like a switch it will stop the current when it is turned OFF and vice versa.

4. Next, in order to connect the humidifier to the existing setup, I have to look up how the relay connect to the high voltage channel and the electrical device which is a humidifier in this case. From the research I came up with a simple diagram of how my set up will look like.

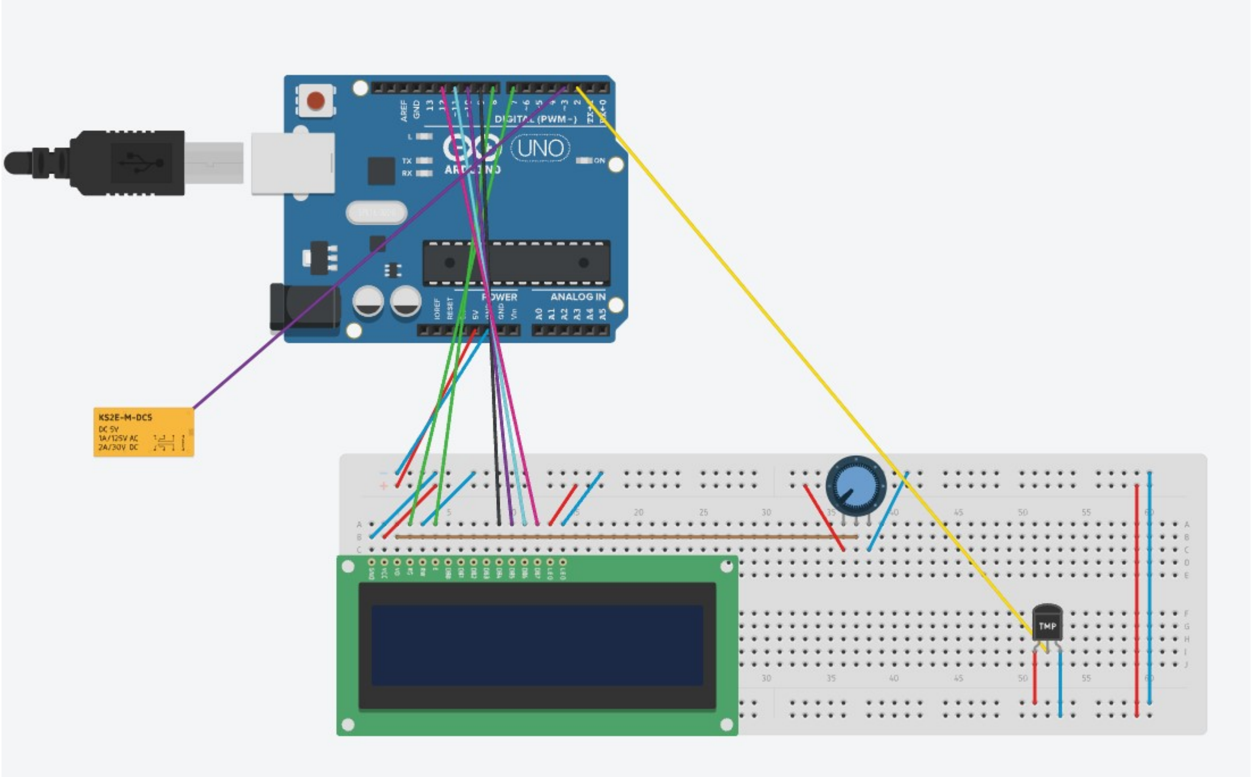


Figure 1.3: Simple circuit diagram connecting all components

I cut the USB wire into 2 parts, then I tied the black wires (GND) together while two red wires (Vcc) are plugged into the middle port and left port, correspondingly.

5. Final setup

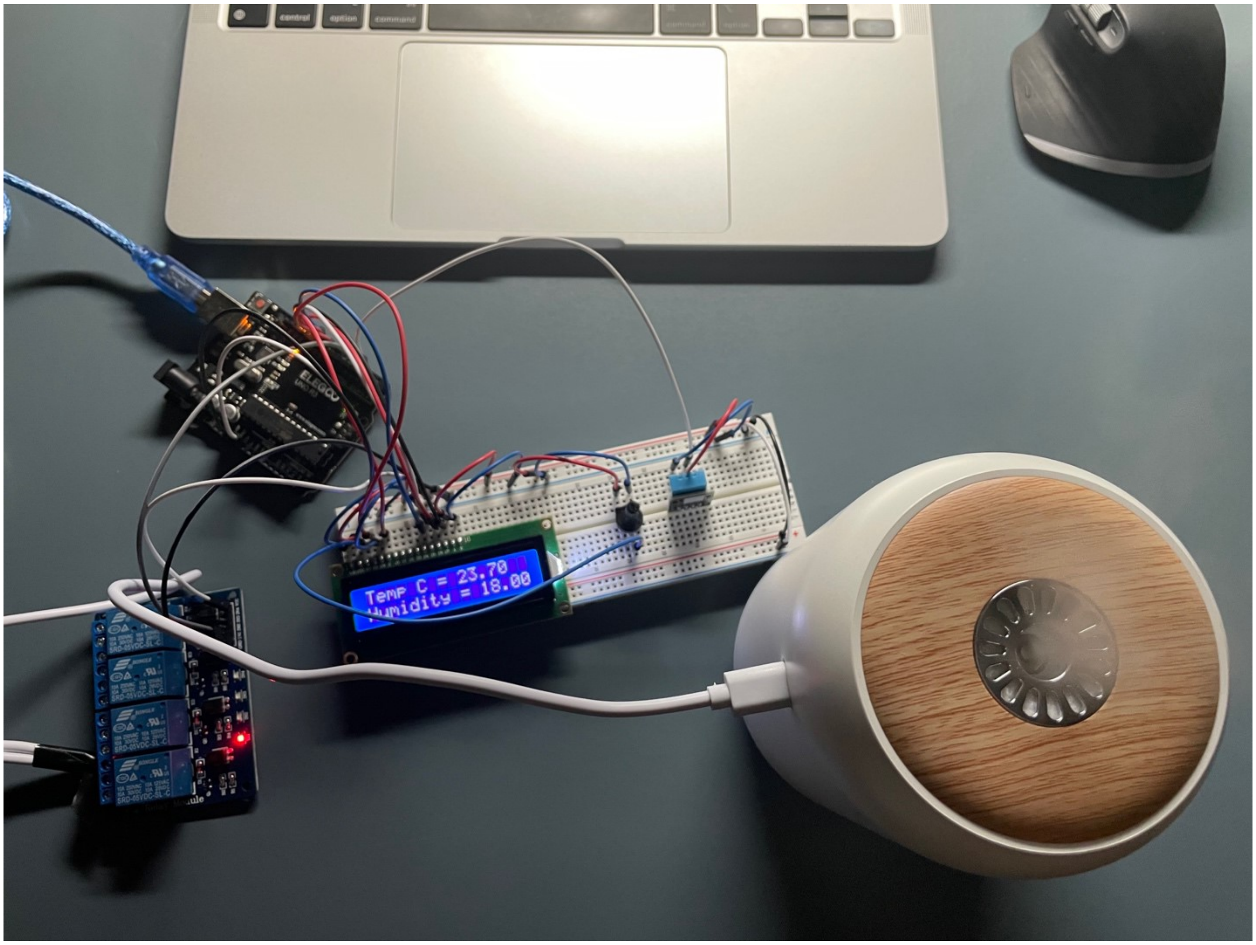
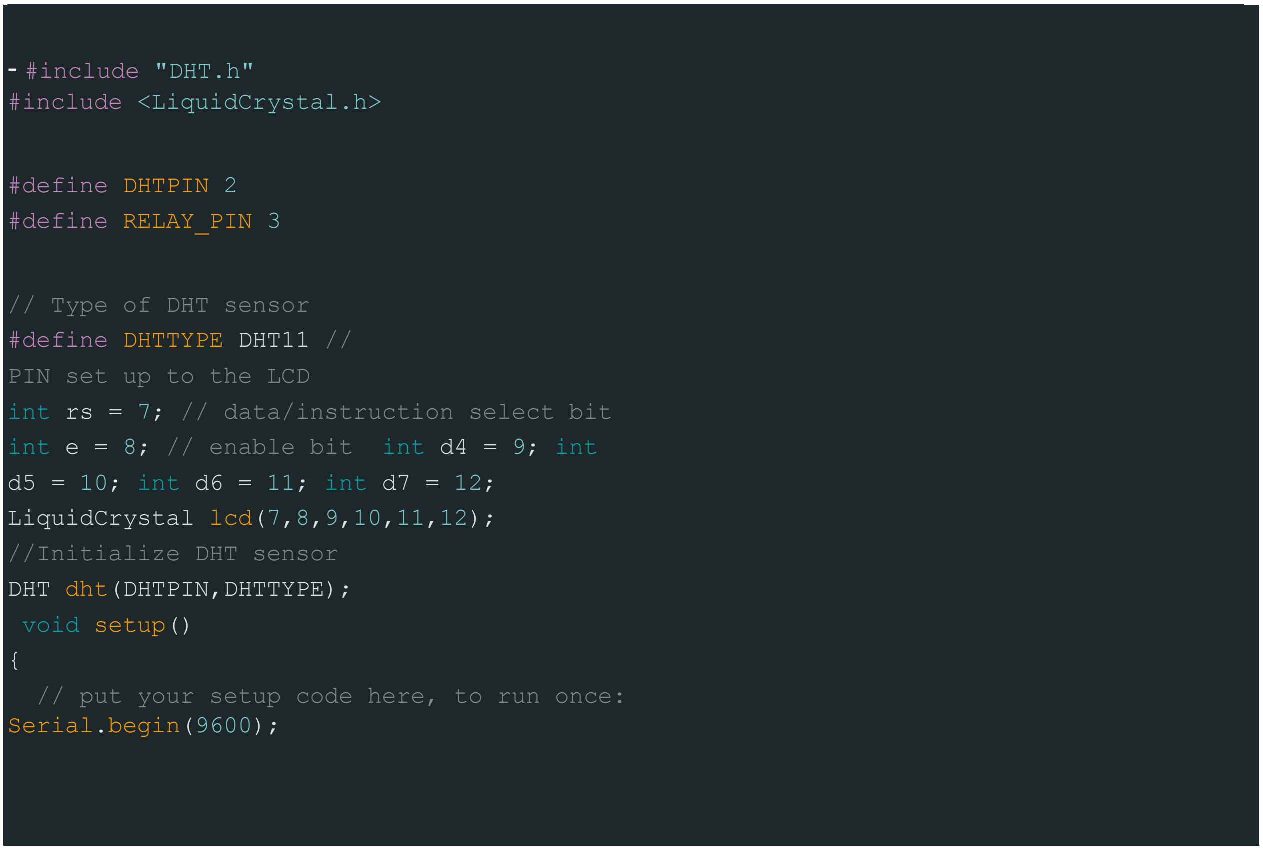


Figure 1.4: Final setup

Project Analysis

There are a few problems during the progress of the project. Firstly, the captured information from the sensor is unreliable which made me change some features in the code like the standard humidity level. Furthermore, the mobility of the setup is also a concern. In the future, this can be improved with a few modifications, such as I will use the flat jumper wire instead of the wires that I used initially and I would use a different version of the microcontroller which is smaller and can be planted onto the breadboard. However, the desired goal is achieved since the humidifier will be turned off whenever the reasonable humidity level is reached. This prevents "too much humid" event which can also cause problems as much as "under humid". Secondly, I would have used another humidifier since I cannot get access to the switch of the device that doesn't allow me to automatically turn on and turn off the device. Nevertheless, thanks to this project, I have learnt more about how software and hardware work together. The connection between electrical components and hardware controller that are controlled by a software code.

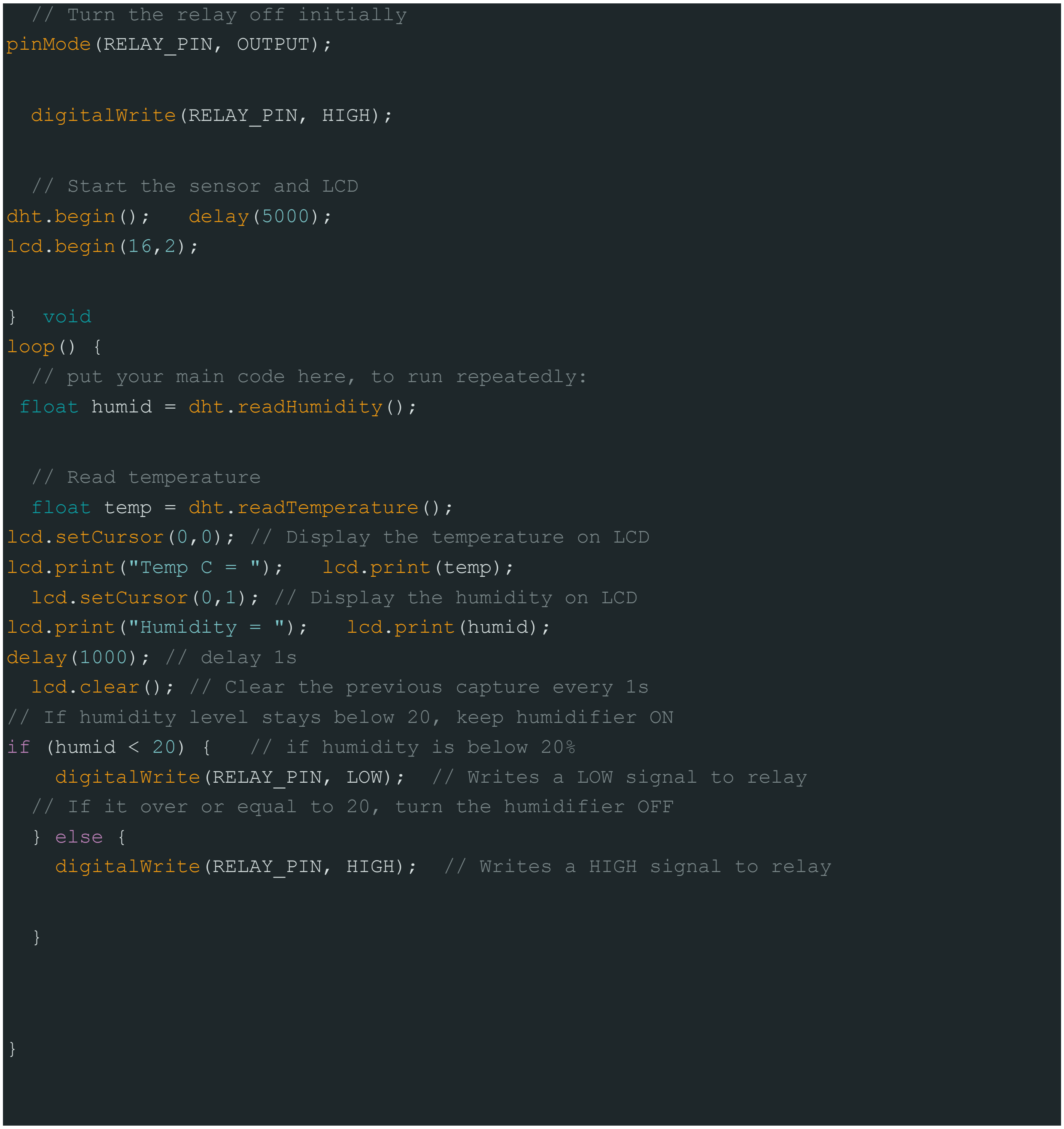
Appendix



```
- #include "DHT.h"
#include <LiquidCrystal.h>

#define DHTPIN 2
#define RELAY_PIN 3

// Type of DHT sensor
#define DHTTYPE DHT11 //
PIN set up to the LCD
int rs = 7; // data/instruction select bit
int e = 8; // enable bit  int d4 = 9; int
d5 = 10; int d6 = 11; int d7 = 12;
LiquidCrystal lcd(7,8,9,10,11,12);
//Initialize DHT sensor
DHT dht(DHTPIN,DHTTYPE);
 void setup()
{
  // put your setup code here, to run once:
Serial.begin(9600);
```



```
  // Turn the relay off initially
pinMode(RELAY_PIN, OUTPUT);


  digitalWrite(RELAY_PIN, HIGH);


  // Start the sensor and LCD
dht.begin();   delay(5000);
lcd.begin(16,2);


}  void
loop() {
  // put your main code here, to run repeatedly:
 float humid = dht.readHumidity();


  // Read temperature
  float temp = dht.readTemperature();
lcd.setCursor(0,0); // Display the temperature on LCD
lcd.print("Temp C = ");   lcd.print(temp);
  lcd.setCursor(0,1); // Display the humidity on LCD
lcd.print("Humidity = ");   lcd.print(humid);
delay(1000); // delay 1s
  lcd.clear(); // Clear the previous capture every 1s
// If humidity level stays below 20, keep humidifier ON
if (humid < 20) {   // if humidity is below 20%
    digitalWrite(RELAY_PIN, LOW);  // Writes a LOW signal to relay
  // If it over or equal to 20, turn the humidifier OFF
  } else {
    digitalWrite(RELAY_PIN, HIGH);  // Writes a HIGH signal to relay


  }




}
```

References

[1] https://en.wikipedia.org/wiki/Arduino_Uno.

[2] https://www.arduino.cc/en/uploads/Main/arduino-uno-schematic.pdf. [3]“How does relay module work”https://www.geya.net/5v-relay-module-how-it-works-

and-application/.

[4] “How to control high voltage using relay module”

https://www.allaboutcircuits.com/projects/use-relays-to-control-high-voltagecircuitswwith-an-arduino/

[5] https://www.seeedstudio.com/blog/2020/01/03/arduino-tutorial-control-high-voltagedevices-with-relay-modules/

[1][2][3][4][5].