Coral Chamber System
Electronic Manufacturing Guide

By: John Reko

Introduction

This tutorial will detail how to assemble the electronic components of the Coral Chamber project. The Coral Chamber project is an ENGI 200 project at Rice University. The goal of this project is to create a chamber with salt water flowing through it that can be mounted on a microscope. This will allow researchers to image coral and then genetically engineer algae to lessen coral bleaching. The water must be temperature controlled and there must be many ways to ensure that water doesn't get onto the microscope. This electronics tutorial will detail many of the control elements of the system that uses an Arduino Mega to run. This tutorial assumes no electronics experience, but this may be hard to complete without electronics experience, especially soldering experience. All of this can be done on a solderless breadboard if that is possible for the application. The schematic for this project is attached at the end of the document.

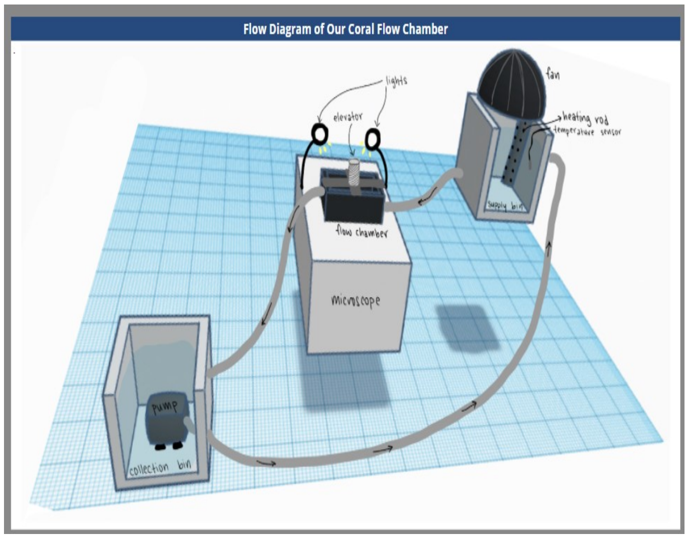

Figure 1. Coral Chamber Flow Diagram

Materials

- 1 x Arduino Mega 2560 Rev3
- 1 x 16x2 LCD
- 1 x 12v solenoid valve
- 2 x Adafruit NeoPixel Ring - 24 x 5050 RGB LED
- Waterproof temperature sensor
- 1 x TIP120 Darlington Transistor
- 1 x 1N4001 Diode
- 2 x 470Ω Resistor
- 1 x 1 kΩ Resistor
- 1 x 5.1 kΩ Resistor
- 2 x 220 Ω Resistor
- 8 x 10 kΩ Potentiometers

- 1 x Micro push button
- 1 x Slide switch
- 3 x 5V DC 120V AC solid state relay
- 1000W water heater
- Desk fan
- Aquarium pump
- Green light emitting diode (LED)
- Red light emitting diode (LED)
- Soldered breadboard
- Jumper wires
- Breadboard wires
- Ideally a digital multimeter (DMM)
- Soldering Iron
- Solder
- Wire cutters
- Solder Pump
- Heat gun or lighter
- Heat shrinks
- Hot glue gun with glue
- 7V DC power supply
- Laptop
- USB 2.0 cable

Assembly

Powering and Programming Arduino

To program Arduino, one must download the Arduino IDE from the website. Then they can plug the USB 2.0 cable into the appropriate slot and upload code by selecting the correct code and port. In order for this system to work, a 7V DC power supply must be used to power the board. Just plug the 7V DC power supply into a standard wall outlet and the barrel connector on the Arduino to power the board. A light will light up when the board has power.

Powering Breadboard

To power the breadboard, just solder one jumper wire to the positive bus (usually has red line going alongside) and one wire to the negative bus (the bus right next to the positive bus). Plug these wires into the $5V$ and GND pins on the Arduino.

Setting up LCD Display

The schematic to hookup the LCD 16x2 display shows that the connections are easy. The only part that is not just plugging into $GND, 5V$, or the Arduino is the potentiometer. Let this output be referred to as L1. Digital pins on the Arduino will be labeled Dxx, where xx is the pin number.

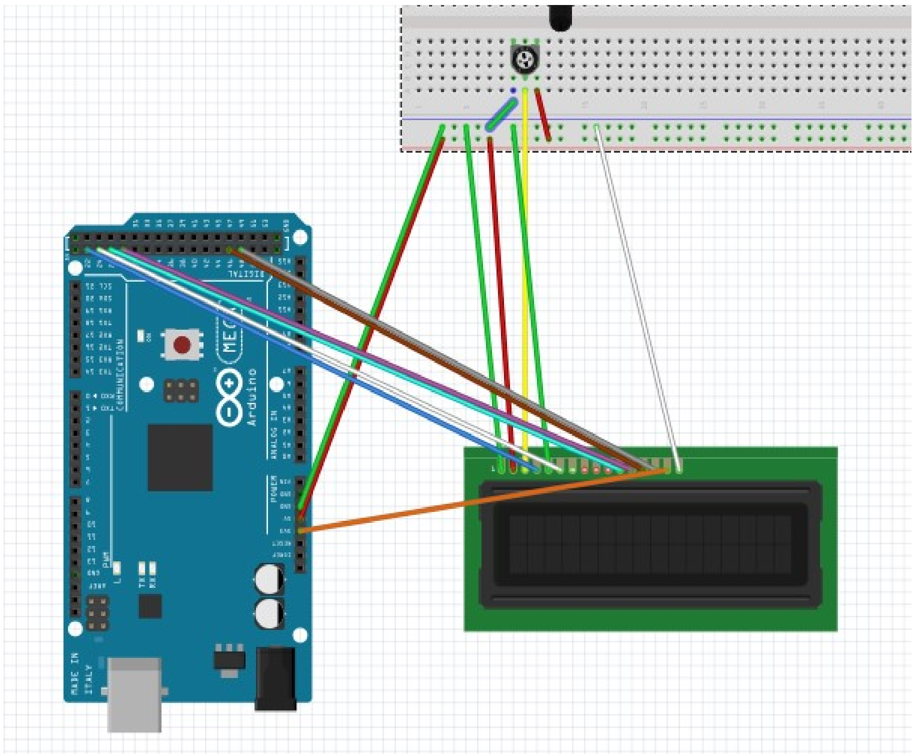

Figure 2. LCD Schematic

To set up the LCD 16x2 display, the following connections need to be made. If a pin on the display is not listed, then it doesn't need to be connected:

Pin LCD	Connection
VSS	GND
VDD	+5V
V0	L1
RS	D22
RW	GND
E	D24
D4	D26
D5	D28
D6	D46
D7	D48
A	+3.3V
K	GND

Table 1: LCD display pins

Solenoid Valve

To connect the solenoid valve, the 1kΩ resistor and diode are needed. The diode acts as the "snubber" diode that prevents transient voltage across the inductive load. The NPN Darlington transistor is also needed for this part to act as a switch for the valve.

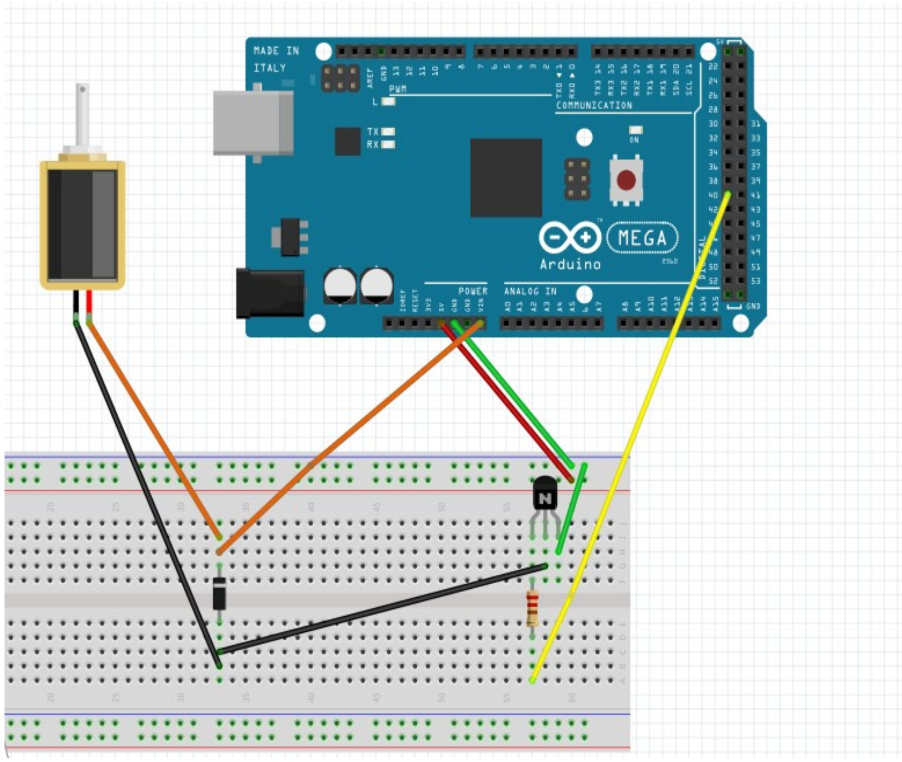

Figure 3. Solenoid Valve Schematic

The connections involve the Vin pin on the Arduino which supplies a 7V potential that comes from the 7V power supply that is powering the Arduino. The connections to the prongs of the Darlington transistor can be known as $S1$, $S2$, and $S3$, the order from left to right in the schematic. The way in which the wires are connected to the solenoid doesn't matter, the order can be reversed.

Component	Connection
S1	D40
S2	Valve
S3	GND
Valve	Vin

Table 2: Valve Pin Designation

Note: The transistor needs to be facing the correct way or it won't be in the correct "mode". Check the data sheet to ensure that the order is correct. The previous pin designations apply like this S1 $\implies$ Base, S2 $\implies$ Collector, S3 $\implies$ Emitter. Also note that the diode needs to be placed in the correct direction that can be seen in the schematic.

Water Level Sensors

In the schematic soil moisture sensors are used, but in reality use water level sensors like this:

Figure 4. Water Level Sensor

This is a simple connection, just connect $5v$ and GND to the buses set earlier and the signal pin to the Arduino. The pins on the Arduino are analog pins A6 and A7. The sensor connected to A6 will set the output of pin D30 to low if the sensor gets wet. The sensor connected to A7 will shut down the system when the sensor gets wet. Make sure there is enough wire to reach around 2 feet away from the breadboard.

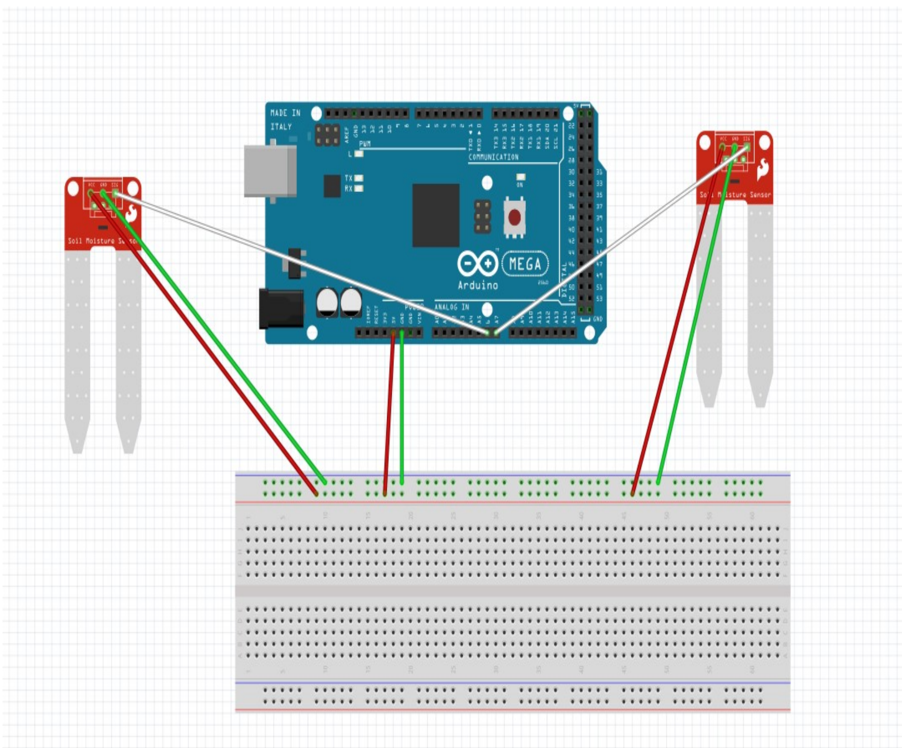

Figure 5. Water Level Sensor Schematic

Neopixels

The wiring of the two Neopixel RGB LED strips is simple. Just connect the $5v$ and GND pins on the lights to the positive and negative buses with a jumper wire. Then connect the LED output pins on the Arduino to a 470 Ω resistor on the breadboard that then connects to the data pin on the Neopixel. This can be done with a jumper. The LED pins on the Arduino are pins 37 and 38.

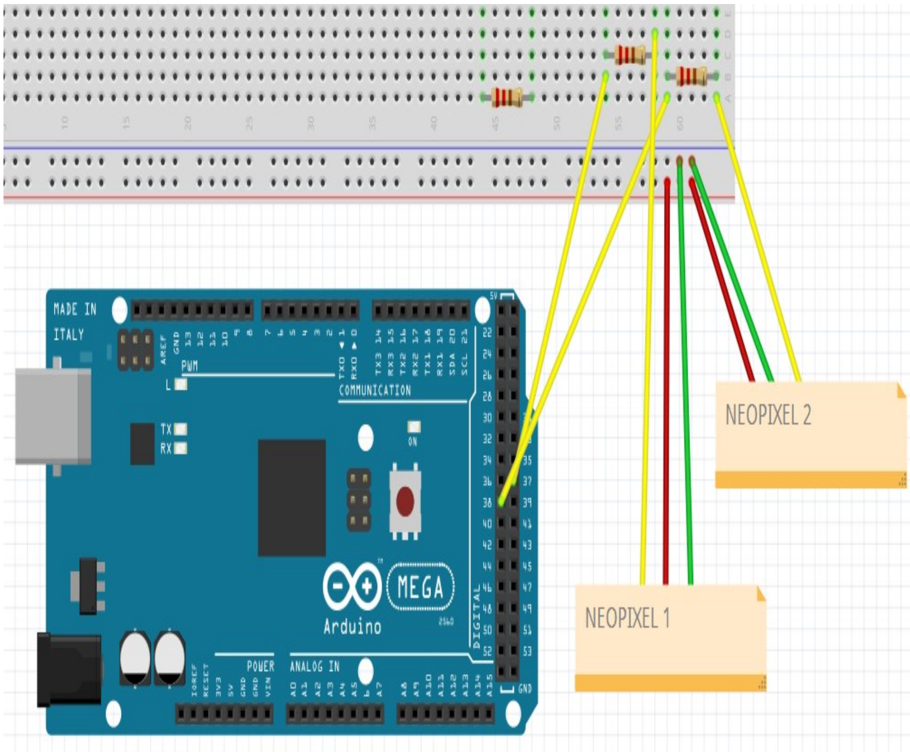

Figure 6. Neopixel Schematic

Potentiometers

At this point there are still seven potentiometers left to attach to the board. Individually, these will be attached to $5V$, GND, and one of the analog input pins.

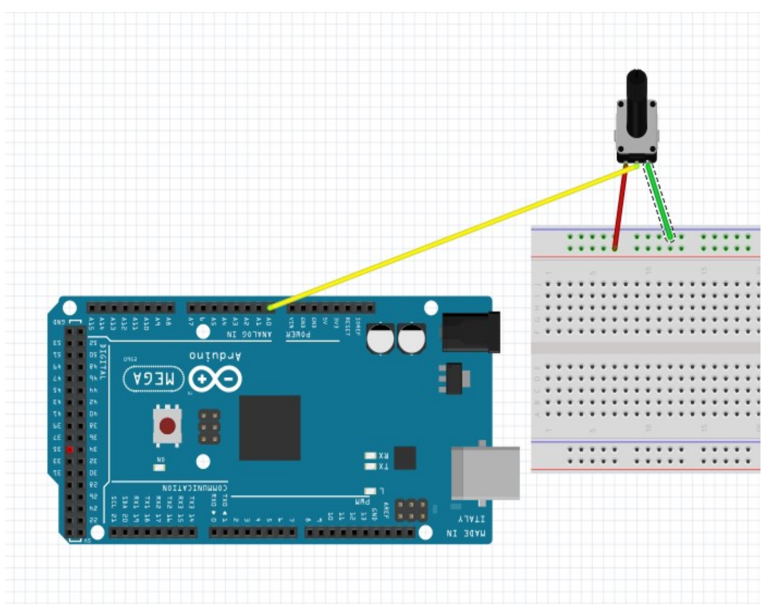

Figure 7. Individual Potentiometer Schematic

Since there are seven potentiometers, use pins A0 - A5 for the first six. These will control the lights. The final potentiometer will be connected to A8. This controls the temperature that the system will operate at.

Temperature Sensor

To begin, make sure the temperature sensor is a three wire, waterproof temperature sensor. For the code in this project to run, the temperature sensor must use the Dallas One Wire digital communication protocol. To start, connect the blue wire to the GND bus. Then connect a breadboard wire from the $5v$ bus to a row on the board. After this, connect the red wire of the sensor to this row. Put one leg of a 5.1 kΩ resistor in this row and the other in a separate row. Connect the yellow wire and a jumper wire to that that row. Finally, connect the jumper to pin D33 of the Arduino.

Indicator LEDs

For both LEDs, twist one end of a 220Ω resistor around the long leg of the LED. Solder this connection, put a heat shrink over it, and then use the heat gun or the lighter to tighten the heat shrink. Solder a jumper wire to the other leg of the resistor and repeat the heat shrinking. Then solder a longer wire to the other leg of the LEDs and repeat the heat shrinking. Connect the leg without the resistor to the GND bus on the board. Connect the other pin of the green and red LEDs to pin D34 and D39 respectively.

Reset Button and Power Switch

Solder two wires to two terminals of the micro push button. Then connect one of these wires to GND and the other to the $RESET$ pin on the Arduino. The power slide switch is three terminals. Connect the two outer terminals to GND and $5V$ buses and then solder a jumper wire onto the middle terminal. This can then connect to pin D53 on the Arduino.

Relay Wires

This project involves three relays for a pump, fan, and a water heater. To connect the relays, strip a section of the wires that go to the wall outlet on all three devices. Find the wire the corresponds to "hot". For a two pronged plug, this is the wire that is in line with the thinner prong. For a three pronged wire, the "hot" wire is usually black. Cut this wire into two pieces. Then connect the side closest to the plug into the common terminal of the relay. The other end will go into the normally open terminal. The relays should also rely on 5V, GND, and a signal to run. Connect the 5V and GND pins on the relay to the matching buses. Finally, connect the signal pins on the relays to the Arduino. The relays correspond to the pins: D30 $\implies$ pump, D31 $\implies$ fan, D32 $\implies$ heater. Make sure the wires that connect to the relays and the board/Arduino are sufficiently long to reach to a wall outlet.

Optional Connections

If wanted, can connect a speaker to the breadboard. It is in the code and the Arduino will send out a 5V square wave to play a sound. Amplifying the signal is optimal as the Arduino doesn't supply a ton of power. This can be done many different ways, but won't be done in this manual.

Power Considerations

If more components are added on this board, then it is recommended that an external power supply is used to power the breadboard. Each of the 48 Neopixels draws up to 60 milliamps of current, making the current draw quite high. It works with the 5V pin as the Neopixels are rarely using the full 60 milliamps. Use a 5V DC, 2A power supply to power if power consumption is an issue. Just plug into wall, add a 1000μF capacitor between the terminals of the power supply, connect the negative wire to the GND pin on the Arduino, and connect the positive wire to the $5v$ bus on the breadboard. This alternate powering method is highly recommended as it will protect the Arduino Mega.

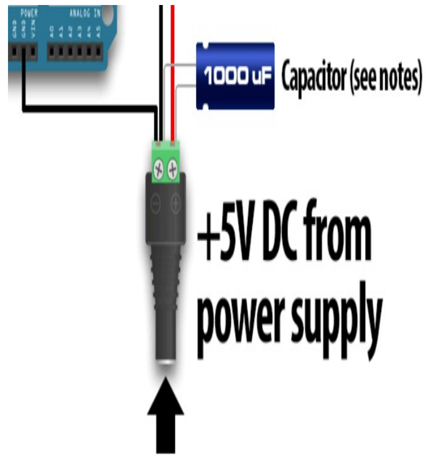

Figure 8. Alternate power configuration. Red wire goes to 5V. From Adafruit Neopixel Uberguide.

Code

The code for this project can be found on GitHub. If you are looking at this document, then you have access to the GitHub repository, so just copy the code, put it into the Arduino IDE, and upload to the Arduino Mega.

Troubleshooting

Biggest issue with soldered breadboard is that the $5v$ and GND rails get connected. Make sure to use the DMM with the diode test mode. If there is a beeping sound between the two rails, go back, remove the solder, and try again. There is ambiguity in the materials list intentionally. There are many different components one can use to complete this project. Make sure that the components are compatible. One example is to make sure that the valve works with the 7V power supply.

Safety

When testing if the relays work, do NOT test when the plug is plugged into a live wall socket. The recommended way to test is to use a power strip, turn off the power strip, and then plug everything, then plug in the power strip. The exposed wire going into the relay has potentially lethal voltage, so make sure to cover with plenty of electrical tape.

Final Steps

To finish this project, put the breadboard and the Arduino into a box that allows all the components that have been connected to be easily accessed during use. To ensure jumper wires plugged into the Arduino do not come out, use a hot glue gun to keep them in place. Then plug the Arduino into a 7V DC power supply and let the system run. The files to laser cut a box as well as an instruction guide can be found in the GitHub repository. There is also a Youtube channel that has video explanations of how to use the system. The channel's name is Coral Chamber Engi 200. There are many non-electrical components that need to be added, which will be in another manufacturing guide. If there are any questions about the process, talk to the OEDK at Rice University and they may be able to give the contact information of the team.

Schematic

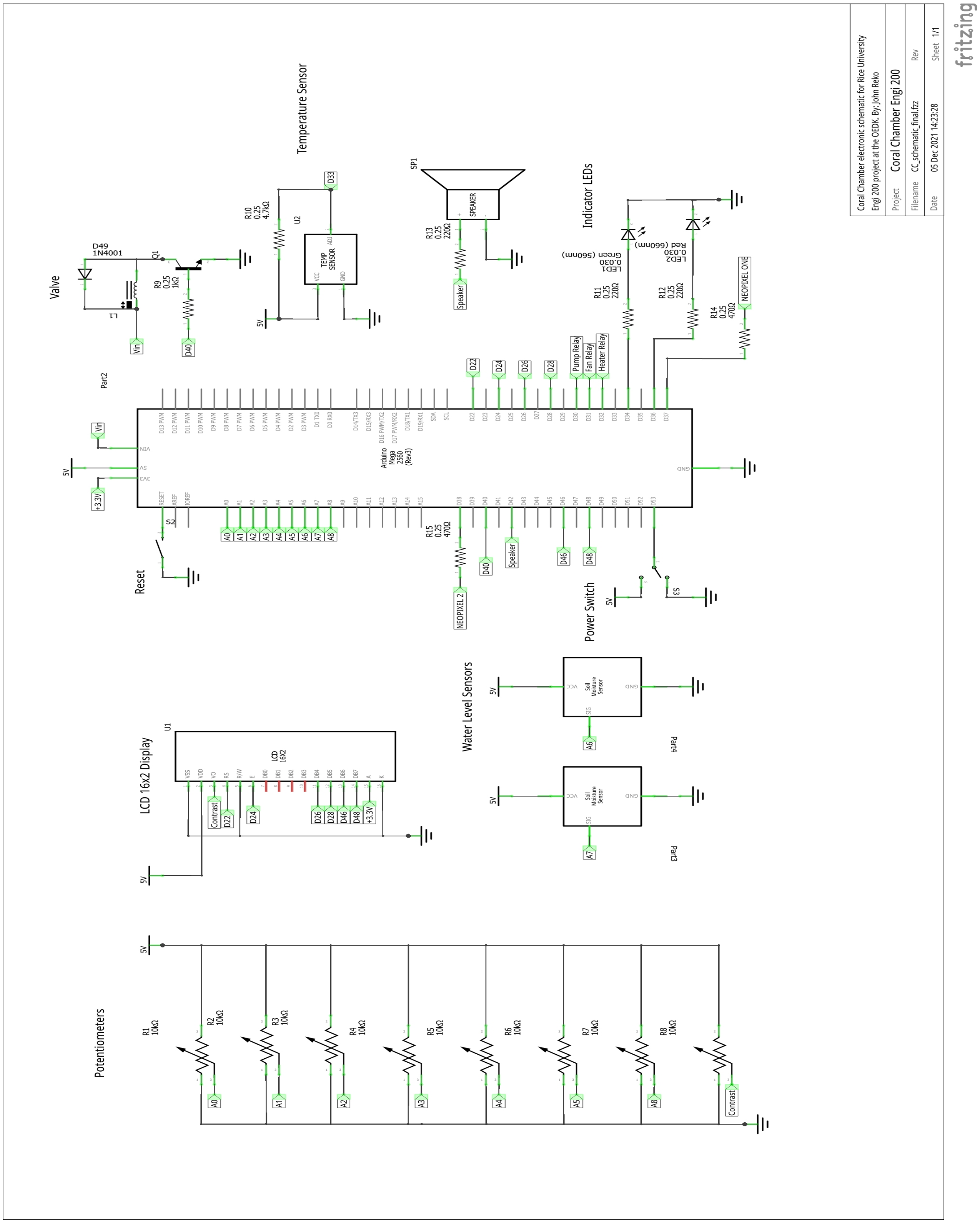

Figure 9. Electronics Schematic