Project Report

Team Members:
Gauri Krishnan, Sudeepti Polnati

December 3, 2024

Contents

1 Problem Statement **2**

2 Aim of the Project **2**

3 Components Used **3**

4 Circuit Schematics and Process Diagrams **4**

5 Flowchart **6**

6 Description of the Project **7**
6.1 Temperature Sensor 7
6.2 Fan and Water Pump 8
6.3 Liquid Crystal Display 9

7 Results **10**

8 Simulation and Short Video Demonstration **13**

9 Photos **14**

10 Bibliography **18**

1 Problem Statement

When certain machines are in use for a long period of time, excessive heating occurs and this negatively affects the efficiency of the system. In certain cases, the system might even shut down, thereby causing huge production loses.

To solve this problem, we would like to design a system that maintains the temperature within an ideal range. Once the temperature crosses a certain threshold, the cooling mechanism will start working. We will also make the system fully automated, so that overheating will be monitored and the machine can automatically be cooled down.

This ensures that the machines do not overheat, efficiency of production will not be affected and thus, huge losses can be prevented.

2 Aim of the Project

Our aim is to create a prototype of a cooling system that continuously monitors the temperature of the machine and automatically starts cooling it down once the temperature crosses a certain threshold value.

3 Components Used

- Adafruit TMP117 High Accuracy I2C Temperature Sensor
- 3-6V DC Water Pump (with inlet and outlet) and pipe
- 12V DC Fan
- 5V 2-Relay Module
- Aluminium Sheet
- Jumper wires
- Arduino UNO
- Arduino Cable
- Plastic Box
- Soldering iron(to heat the plate)
- Liquid Crystal Display Module
- 5kΩ Potentiometer

4 Circuit Schematics and Process Diagrams

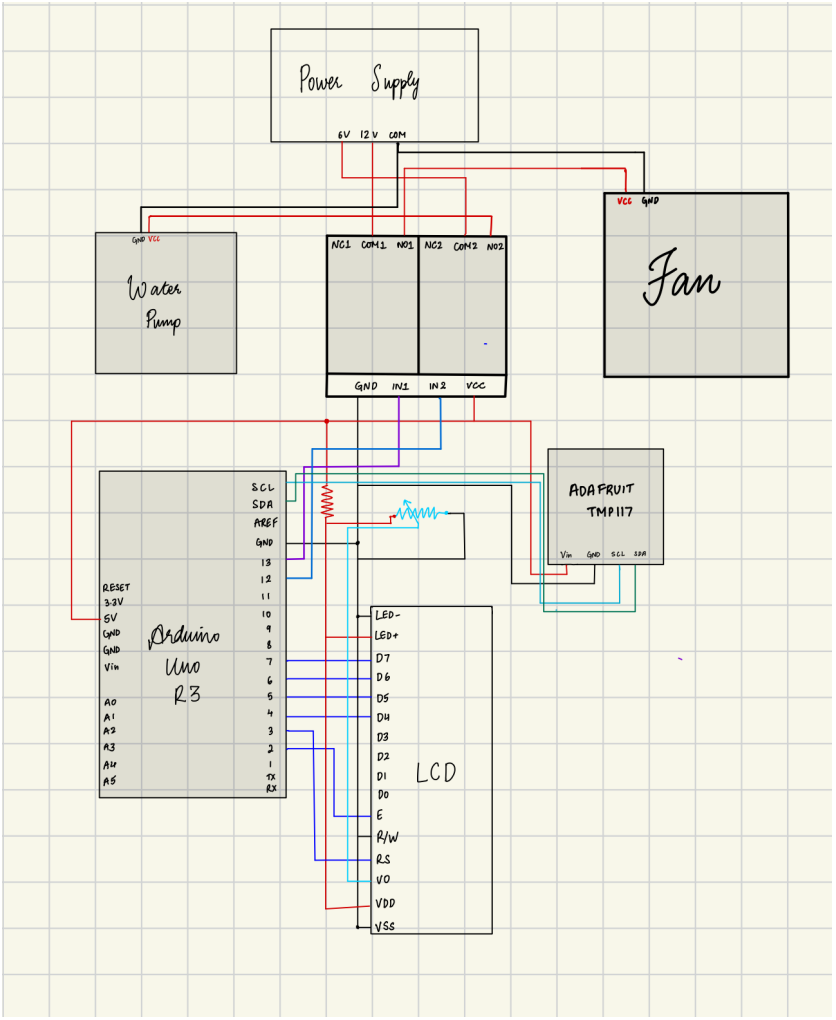

Figure 1: Circuit Schematic

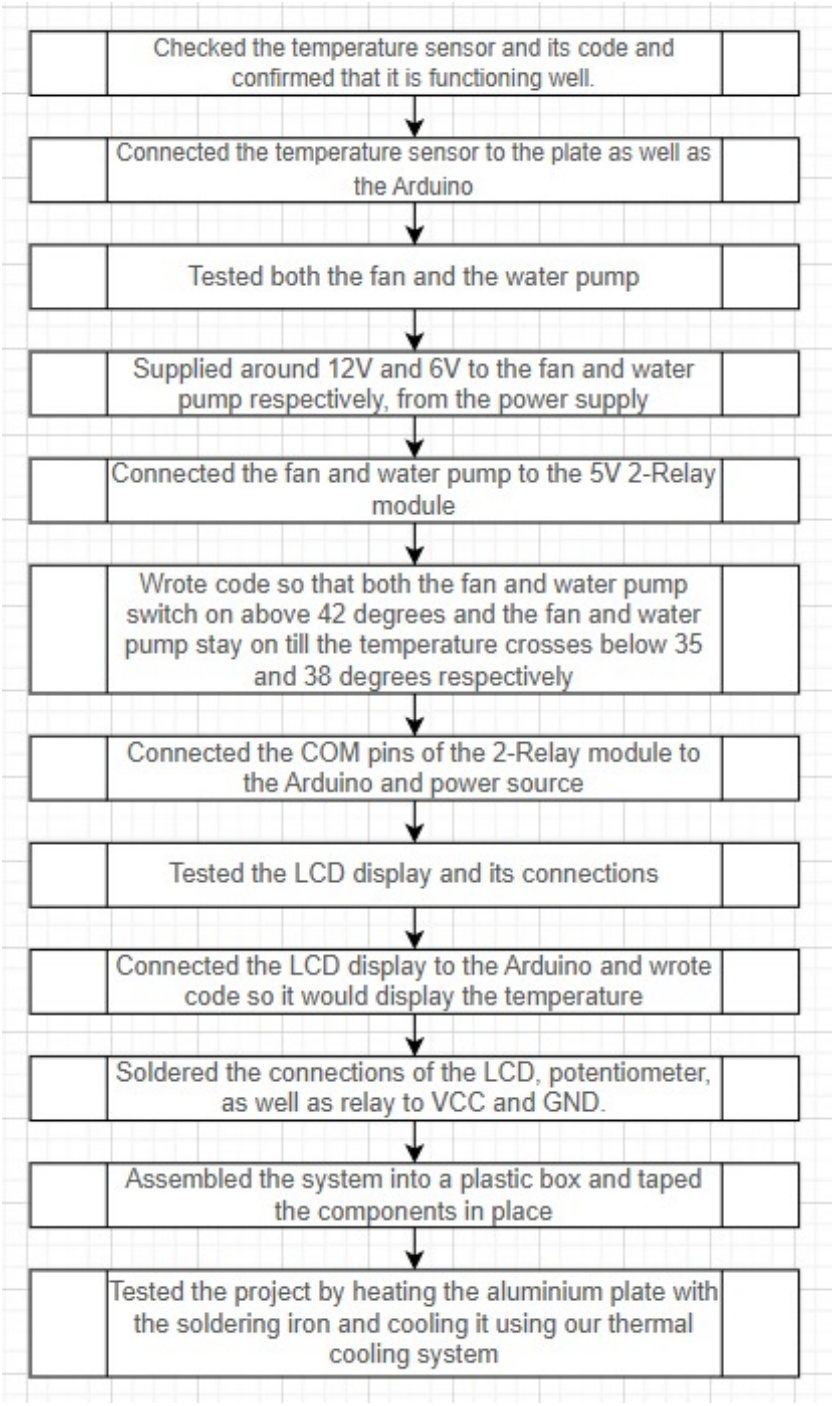

Figure 2: Process Diagram

5 Flowchart

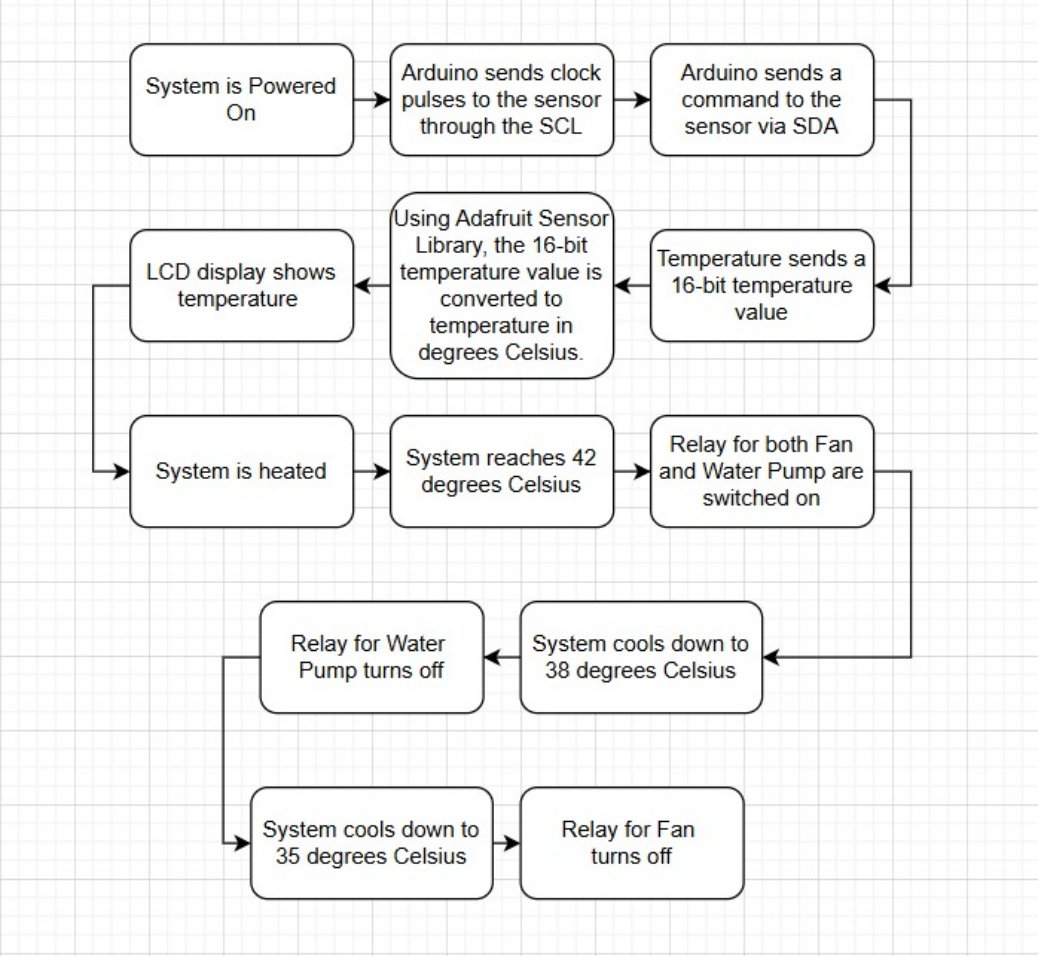

Figure 3: Flowchart

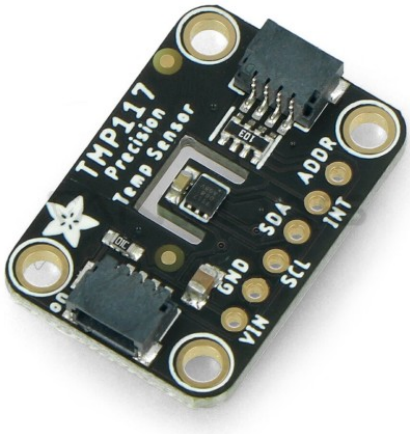

Figure 4: Adafruit TMP117 High Accuracy I2C Temperature Sensor

6 Description of the Project

- In this project, we will use an aluminium plate that is heated up to mimic the heat build up in the machine, sense this heat build-up, and then cool it down when it crosses the high threshold.
- Conversely, when the temperature is below the lower threshold value, the cooling system should switch off.
- To achieve this, we need to use a temperature sensor to detect when the heat is beyond the higher threshold value, and when it is below the lower threshold, so that we can turn the cooling system on and off accordingly.
- Our cooling system incorporates a fan and a water pump, and their operation is controlled using a relay to determine when to switch them on.
- To display the temperature, we used an LCD. This ensures that the user is able to see the temperature and thus, provides a means for manual monitoring as well.

6.1 Temperature Sensor

- The Adafruit TMP117 is a high-precision digital temperature sensor that operates using the I^2C communication protocol.

- The temperature sensor is ideal for battery-operated systems due to its low power requirements.
- It outputs temperature data as a 16-bit digital value in degrees Celsius.
- It requires two wires:
  - SCL (Serial Clock Line): Synchronizes data transfer.
  - SDA (Serial Data Line): Transmits temperature data and receives configuration commands.
- The sensor's internal temperature value is stored in a temperature register.
- The Arduino sends a command to the TMP117 via the SDA line to access this register. The TMP117 responds by sending the 16-bit temperature value.
- Then, we used the Adafruit Unified Sensor Library to interact with the TMP117 sensor.
- The sensors_event_t structure is part of the Adafruit Unified Sensor Library and is designed to hold sensor data in a standardized format. The *temp.temperature* field holds the temperature value retrieved from the TMP117.
- The tmp117.getEvent(&temp) method reads the 16-bit raw temperature data from the TMP117's temperature register and converts the raw value into degrees Celsius using the TMP117's resolution (0.0078125°C per LSB) .
- Then the converted value is stored in the temperature field of the sensors_event_t structure.

6.2 Fan and Water Pump

- For our cooling system, we are using a 12V DC Fan and a 3-6V DC Water Pump.
- When the aluminium plate reaches a certain threshold temperature, say 42, the Arduino will switch on the fan and water pump by using the relay.
- When the plate is cooled down enough, and the temperature sensor reaches a lower threshold value, 38, the water pump is switched off. When the temperature is 35, the fan switches off.

- By directing the fan towards the plate and also by spraying water onto it, we are using conduction as well as evaporation to cool down the object.

- We are using a 5V 2-Relay Module so that we can control the fan and the water pump seperately.

- The +ve terminal of the fan is connected to the NO1 pin from the relay. COM1 from the relay is connected to the power supply which provides 12V DC for the fan. The -ve terminal of the fan is connected directly to the power supply.

- The +ve terminal of the water pump is connected to the NO2 pin from the relay. COM2 from the relay is connected to the power supply which provides 6V DC for the water pump. The -ve terminal of the water pump is connected directly to the power supply.

6.3 Liquid Crystal Display

- The process of controlling the display involves putting the data that forms the image of what should be displayed into the data registers, then putting instructions in the instruction register. The LiquidCrystal Library simplifies this so that we do not have to write the low-level instructions.

- The LCD has 8 data pins (D0-D7). We are using the pins D4-D7. The states of these pins (high or low) are the bits that we are writing to a register when we write.

- The LCD also has a register select (RS) pin that controls where in the LCD's memory we are writing data to. When it is in the high state, we can select the data register which holds what goes on the screen. Conversely, when it is in the low state, the instruction register can be selected, which is where the LCD's controller looks for instructions on what to do next.

- The Read/Write (R/W) pin that selects reading mode or writing mode. If it is set to low, the data is being written. When it is high, the data is being read.

- The Enable pin enables writing to the registers.

- There is also the display contrast pin (Vo), power supply pins (+5V and GND) and LED Backlight (LED+ and LED-) pins that can be used to power the LCD, control the display contrast, and turn on and off the LED backlight, respectively.

7 Results

- The LCD displays the temperature detected by the sensor.
- When the temperature of the plate goes above 42°C, the cooling system begins working.
- The fan and the water pump both get switched on by the relay.
- The water from the pump is sprayed directly onto the plate and the wind from the fan is directed towards the plate.
- Once the plate gets cooled down to 38°C, the water pump turns off. However, the fan stays on till the plate is cooled down to 35°C.

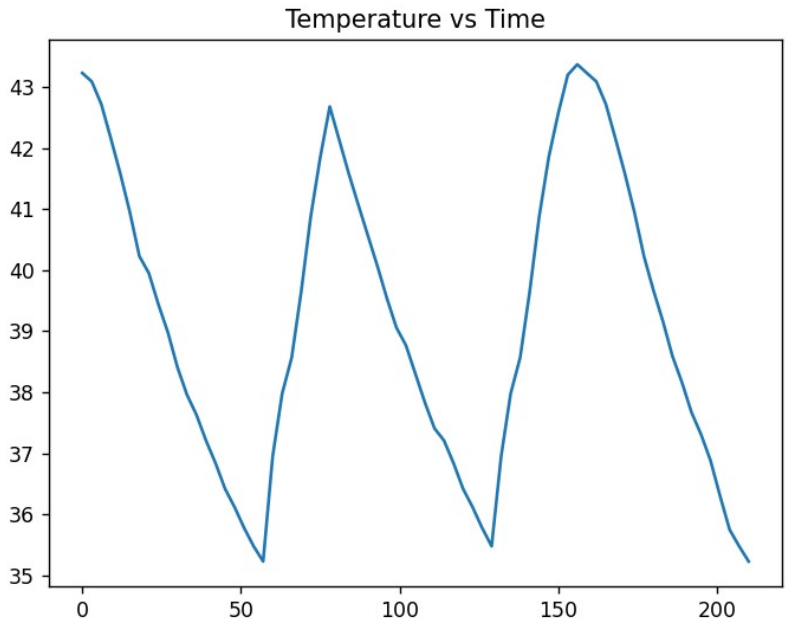

Figure 5: Plot of temperature with time.
This graph represents the run-time characteristics of our cooling system given constant heating.

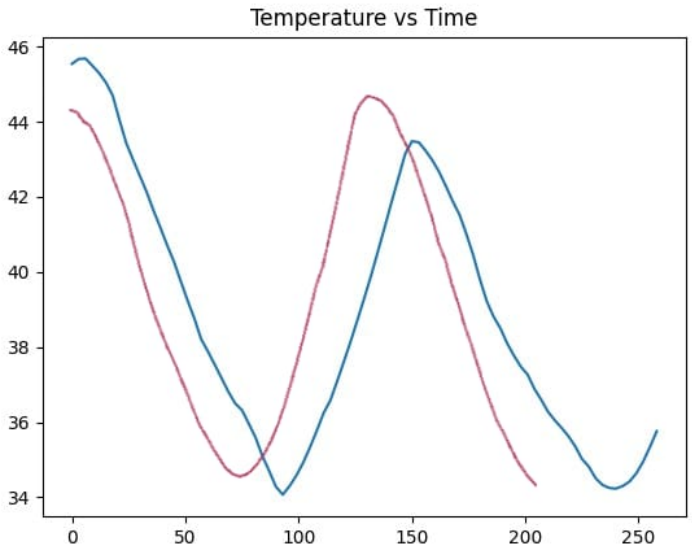

Figure 6: Plot of temperature with time for fan and water pump seperately.

- This red curve represents the run-time characteristics if only the fan is used, whereas the blue curve represents the cooling when only the water pump is used.
- As can be observed from the graph, the fan is more effective in cooling in our hardware setup.

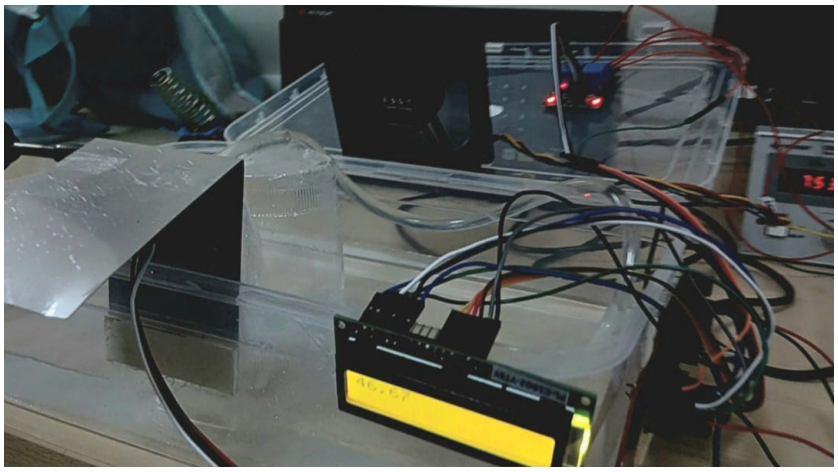

Figure 7: Temperature at 46.67°C.

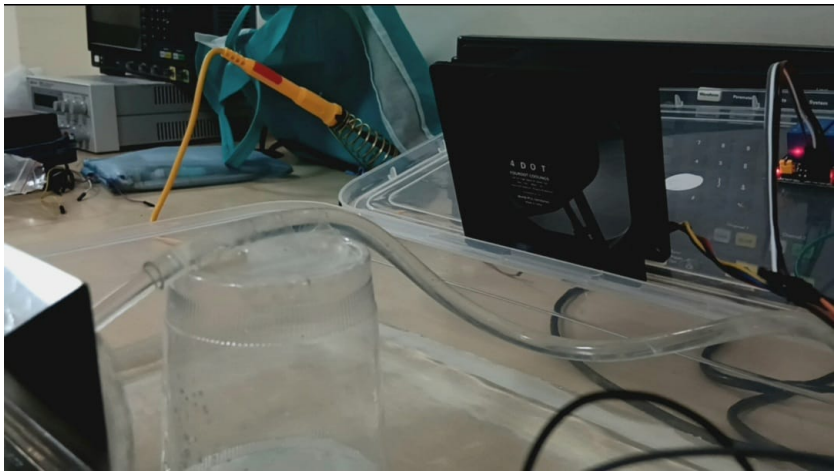

Figure 8: Close up of water pump and fan in action.

8 Simulation and Short Video Demonstration

- Simulation Software: We used TinkerCad to test out the LCD connections before making the hardware connections.
  `https://www.tinkercad.com/things/f3ENMcSZqvN-lcd?sharecode=pfXGgBAMBYzql1EuV5jJ32cFFYbez2s6pcNQdVDW8ZQ`
- Demonstration Video: `https://photos.app.goo.gl/xzVNmXDe73BoFsD36`

9 Photos

Include photos of the setup or key components.

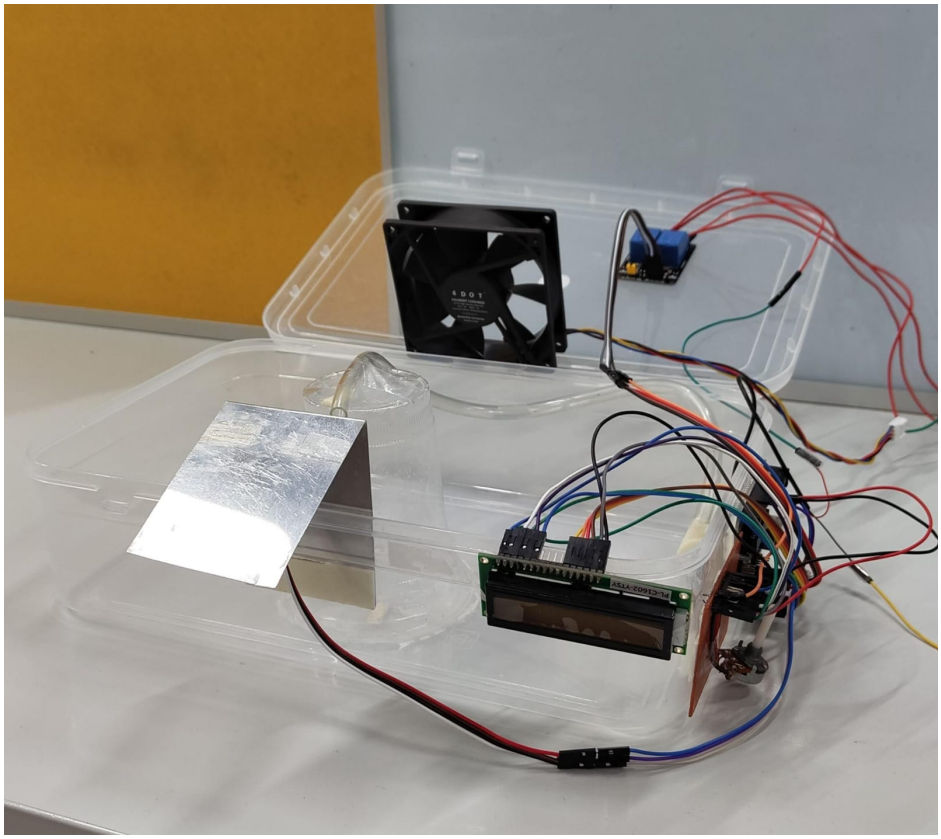

Figure 9: Hardware Setup

Key Components:

Figure 10: 12V DC Fan

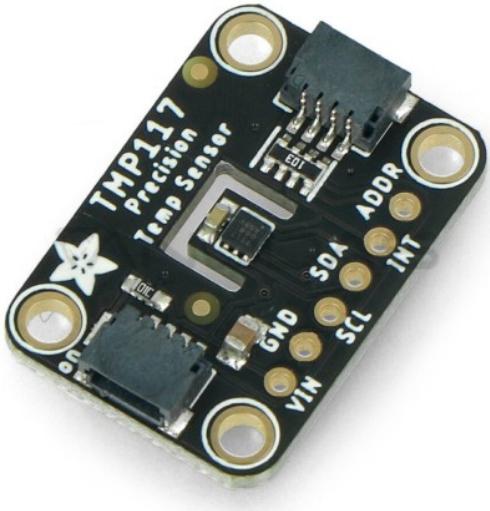

Figure 11: Temperature Sensor

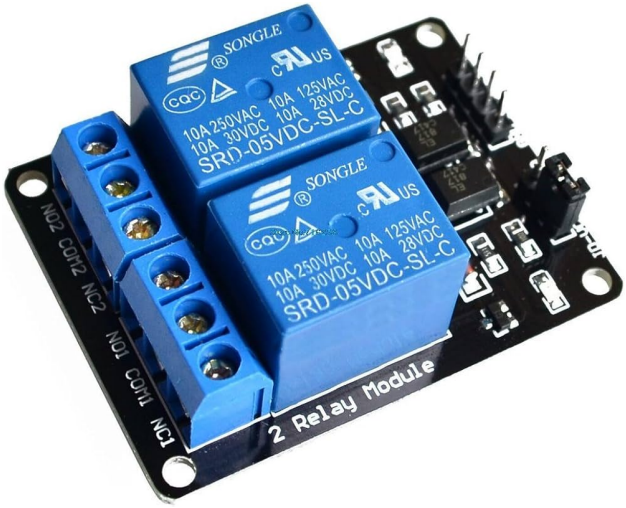

Figure 12: 5V 2-Relay Module

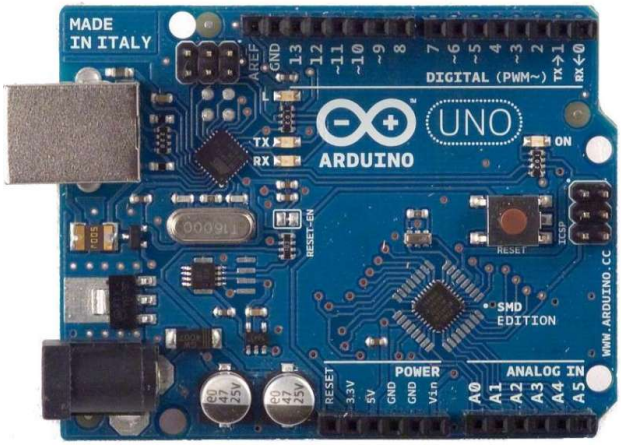

Figure 13: Arduino Uno

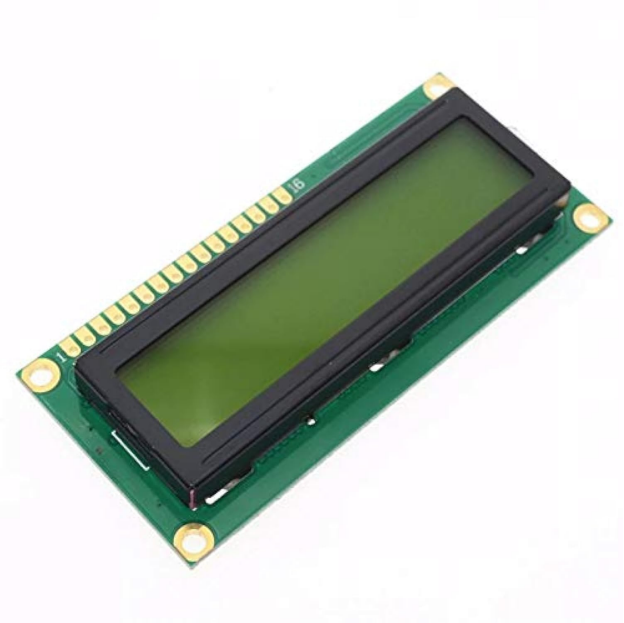

Figure 14: Liquid Crystal Display

10 Bibliography

References

[1] https://learn.adafruit.com/adafruit-tmp117-high-accuracy-i2c-temperature-moni
arduino

[2] https://www.youtube.com/watch?v=fimVrT7NFFU

[3] https://www.youtube.com/watch?v=xNIedHi9PL8

[4] https://docs.arduino.cc/learn/electronics/lcd-displays/