**Name: JAMIE GONZALES ESGUERRA (9) Class: PA12**

# Problem statement

Temperature is an important factor to the growth of plants. Temperature can affect the photosynthesis of plants whereby the plants use sunlight, water and carbon dioxide to create oxygen and energy in the form of sugar. Different plants require different temperature. For example, lettuce cannot exceed a temperature above 28°C. Hence, it is important to regulate and maintain the temperature in a greenhouse to maximise crop yield. I aim to regulate the temperature based on the requirements of a specific plant to produce consistent and predictable higher crop yield using sensors and actuators. The farmers would be given a choice whether they want it to work automatically or manually by using a switch. As an example, I would follow the temperature requirement of a lettuce (28°C). When the program is set manually, the farmers are obliged to operate the DC Motor Fan when needed. When it is automatic, the Dc Motor fan would turn on itself when needed. For farmers to know whether the temperature has exceeded the limit, the buzzer would turn on and the Tri\_LED colour red would be lighted up. Furthermore, to reduce mistakes, the farmers would not be able to turn on the DC Motor Fan when the temperature is lower than the threshold instead, the orange led would light up to indicate that the DC Motor Fan should not be turned on. Lastly, Ubidots is used to show the temperature of the temperature sensor over time.

# System Requirement

This system design is built to constantly monitor the temperature in the greenhouse by using a temperature sensor which has an ability to measure the surrounding temperature. By converting the binary to voltage and voltage to degree Celsius, we would have the data of the surrounding temperature in degree Celsius. I programmed the Arduino to have a temperature threshold of 28°C which is the maximum temperature that a lettuce can take.

I used MQTT to send the temperature data to Ubidots for the farmers to have a visualization of how the temperature changes over time. This is useful as it can show the specific temperature at a specific date and time.

When the temperature sensor detects that the temperature above the threshold, a **buzzer** turns on.

The **LED\_TRI** has three colours that represents the following:

**Green light** – Temperature is below the threshold

**Red light** – Temperature is too high

**Blue light** – Fan is being activated by the farmer (will only show when temp. is higher than the threshold and the program is set to manual.)

The farmers can either choose to operate the DC Motor Fan **manually or automatically** by using a **switch**.

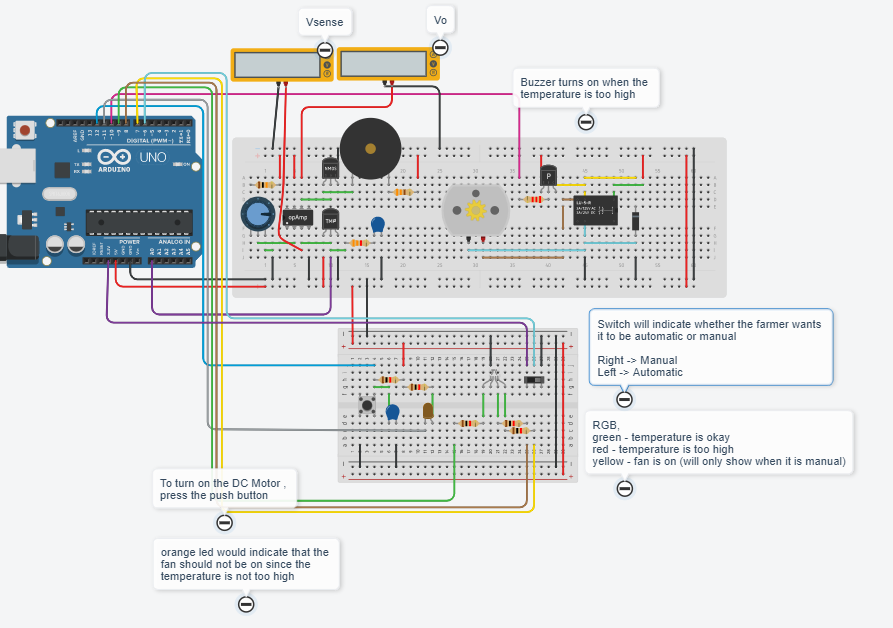
## Automatic

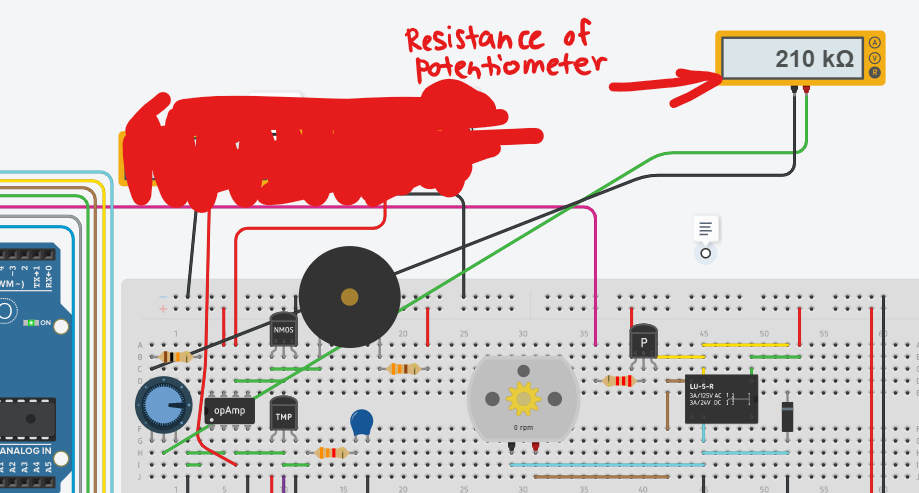
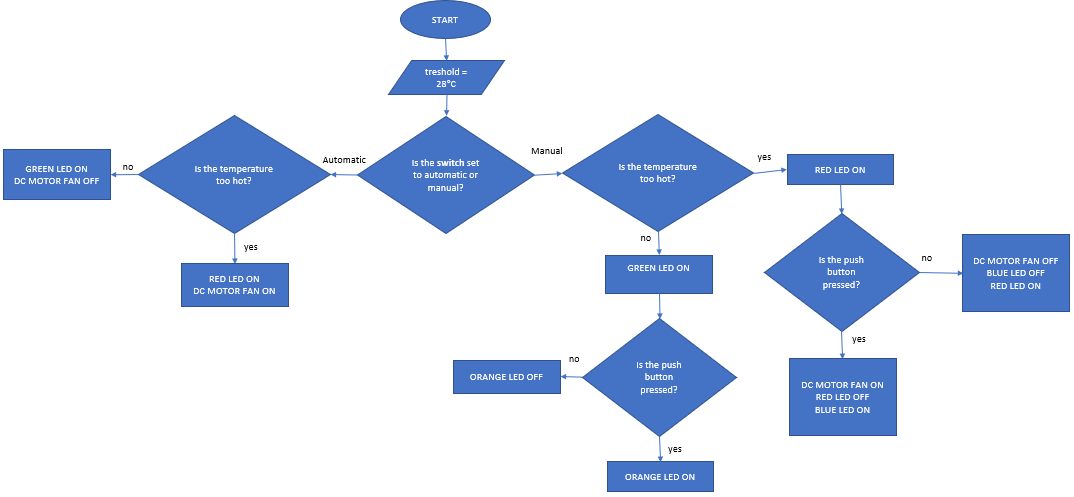
When the farmer set the **switch** to the automatic operation, the farmers do not need to manually turn on the DC Motor fan when the temperature is too hot by using the **push button**. The Arduino will turn on the DC Motor fan by itself to lower the temperature. Hence, this allows the farmers to have flexibility on their time as they may not be always around to maintain the temperature of the greenhouse.

### Manual

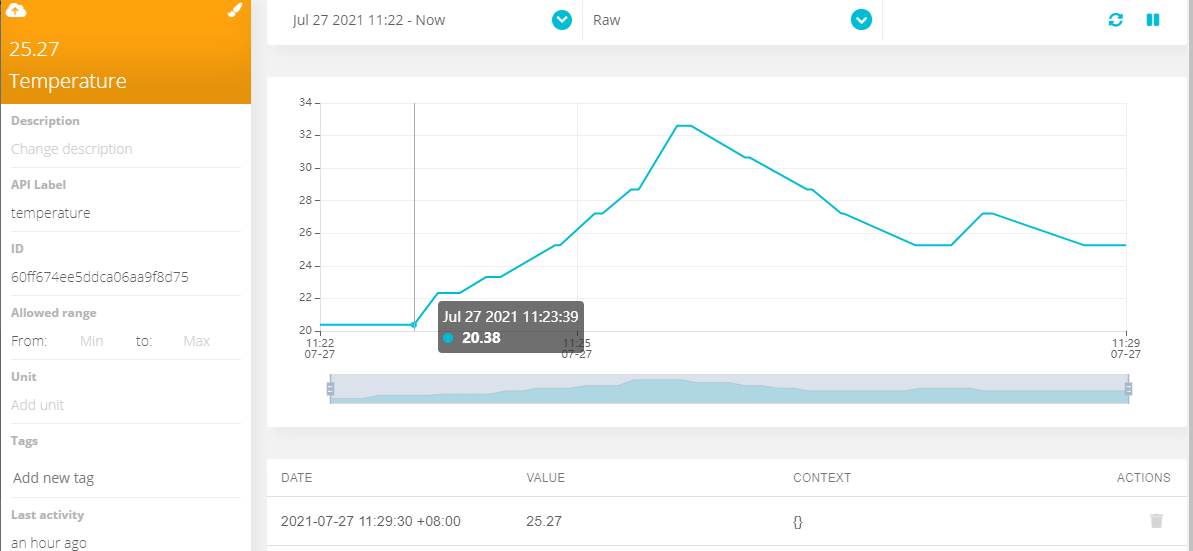
When the temperature is too high and it is operating **manually,** the farmer must switch on the fan by himself/herself. However, the farmer would not be able to switch on the DC motor fan when the temperature is below the limit. Instead, it will show a warning in which the **orange led** will turn on, indicating that the fan does not need to be turned on.

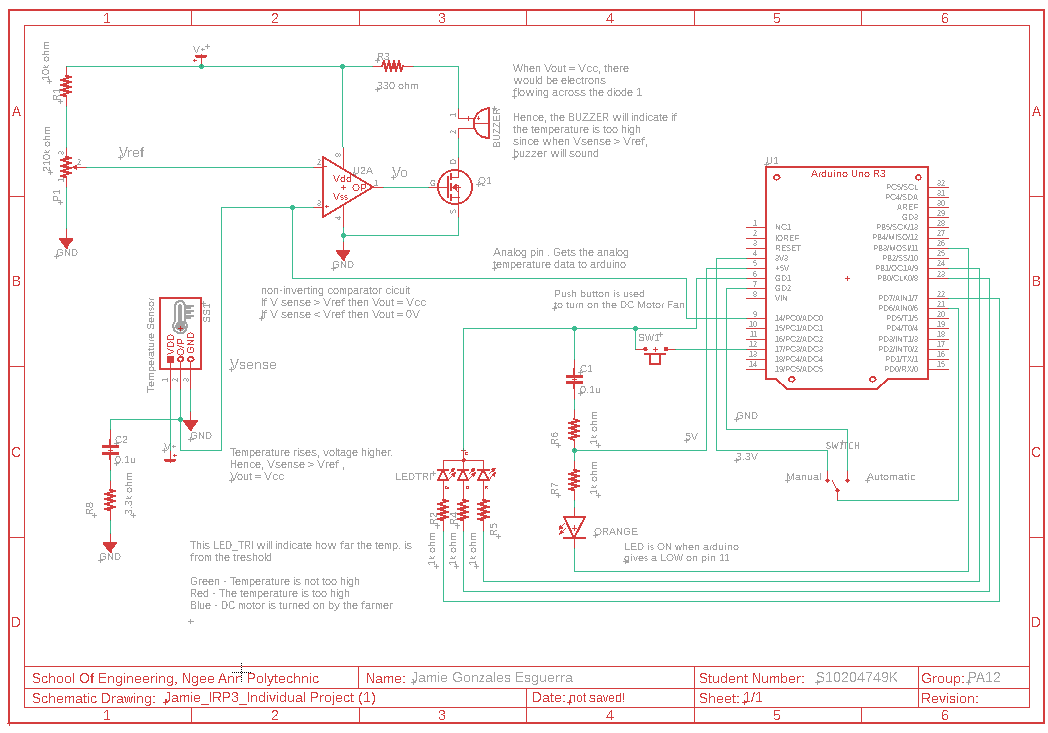
# System Architecture



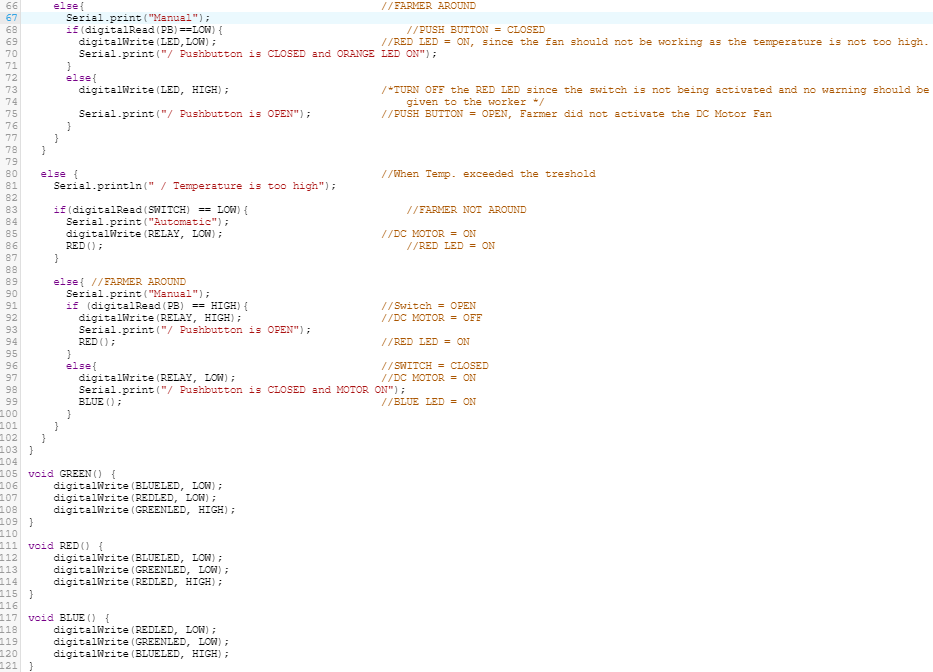
* Temperature threshold of 28°C
* An electromagnetic relay to turn on the DC Motor Fan when needed
* Switch used to either change the program to automatic or manual.
* After adjusting the potentiometer, the should be **210k ohm** for the buzzer to turn on when the temperature exceeds the **threshold of 28°C**. 
* TRI\_LED to indicate whether the temperature is too hot (red light), below the treshold (green) or when the fan is being manually activated.
* 

Temperature of greenhouse over time (above 28°C, buzzer on)



Temperature Sensing Sub-System Schematic & Prototypes 





# SL Reflections

After visiting the Citysprout, I gained a better understanding of how a greenhouse works. Two things that sparked my interest was when they used the fishes’ waste as a fertilizer and utilize vertical farming for a better plant growth.

Greenhouse is crucial as it can bring a significant benefit to produce higher crop yield and achieve the Singapore’s 3030 goal by constantly monitoring the important environmental factors such as the temperature and moisture. This relates to the aim of my project: To regulate the temperature based on the requirements of a specific plant to produce consistent and predictable higher crop yield using sensors and actuators. The IRP3 lessons have helped me developed relevant skills to design a prototype that helped me achieve my aim. For example, I learnt to use an operational amplifier and relay to control my buzzer and dc motor fan respectively. Not only I have learnt the theory knowledge of these components but I have also picked up the skills to create this in a breadboard. Furthermore, I have learnt how to create a schematic diagram using eagle which is is very meaningful and useful for me as I can potentially use this knowledge for another project in the future.After the webinar with Citysprouts on MsTeams, they have mentioned that they would prefer their equipment to operate automatically for convenience. I included this feedback as an additional feature to my project by using a switch to operate the DC Motor Fan automatically or manually.

Another important skill that I have gained is the use of Ubidots. Even though we weren’t able to implement the Bluetooth to an actual Arduino due to covid restrictions, I believe that understanding the use of Ubidots can have a significant benefit for engineers as we can constantly monitor our prototype without physically around it.

One of the challenges I faced was programming my tinkercad. My program LED and relay turns on when I send it a “LOW” instead of a “HIGH”. This has caused confusion to me as I originally use “HIGH” to turn an LED. To solve this, I comment on my program every time I turn on or off an actuator. Additionally, the serial monitor would also indicate whether the relay or led is on.

With all the knowledge I gained through IRP3, I am not more confident when designing/programming a circuit and how I can implement op amp/relay/schematic diagram/Ubidots to my prototype.

**Urban Agricultural Risk Assessment Form**  
