Project Prototype Manual

Section 1: Introduction

The smart plant monitoring system prototype is an IoT-based project designed to monitor environmental parameters such as light, temperature, humidity, and soil moisture in real-time. The system collects data from sensors and sends it to the cloud (via IFTTT) for logging in Google Sheets

The system allows users to maintain optimal conditions for plant growth by automating environmental monitoring, which reduces the need for manual observation.

This manual will guide you through the setup, operation, and maintenance of the Smart Plant Monitoring System. It covers aspects such as:

- The required components
- How to connect and configure the system
- How to operate the system and monitor sensor readings
- Troubleshooting common issues

Section 2: System overview

The system collects sensor data every 60 seconds to monitor:

- Light Intensity (via a GY-30 light sensor)
- Temperature and Humidity (via a DHT11 sensor)
- Soil Moisture (via an analog soil moisture sensor)

The data is transmitted to a Google Sheet via IFTTT, allowing users to track environmental conditions over time. The system also provides visual feedback via LED indicators based on threshold values (e.g., light intensity, temperature, humidity, and soil moisture).

The system operates as follows:

- The system checks the environmental conditions every minute.
- If any sensor reading exceeds its threshold (low light, high/low temperature, low soil moisture), an LED turns on to alert the user.
- The data is sent to IFTTT for logging to Google Sheets.
- If Wi-Fi connection drops, the system attempts to reconnect and displays an alert message on the LCD.

Section 3: Hardware components

Below we include a table for hardware components and their functions:

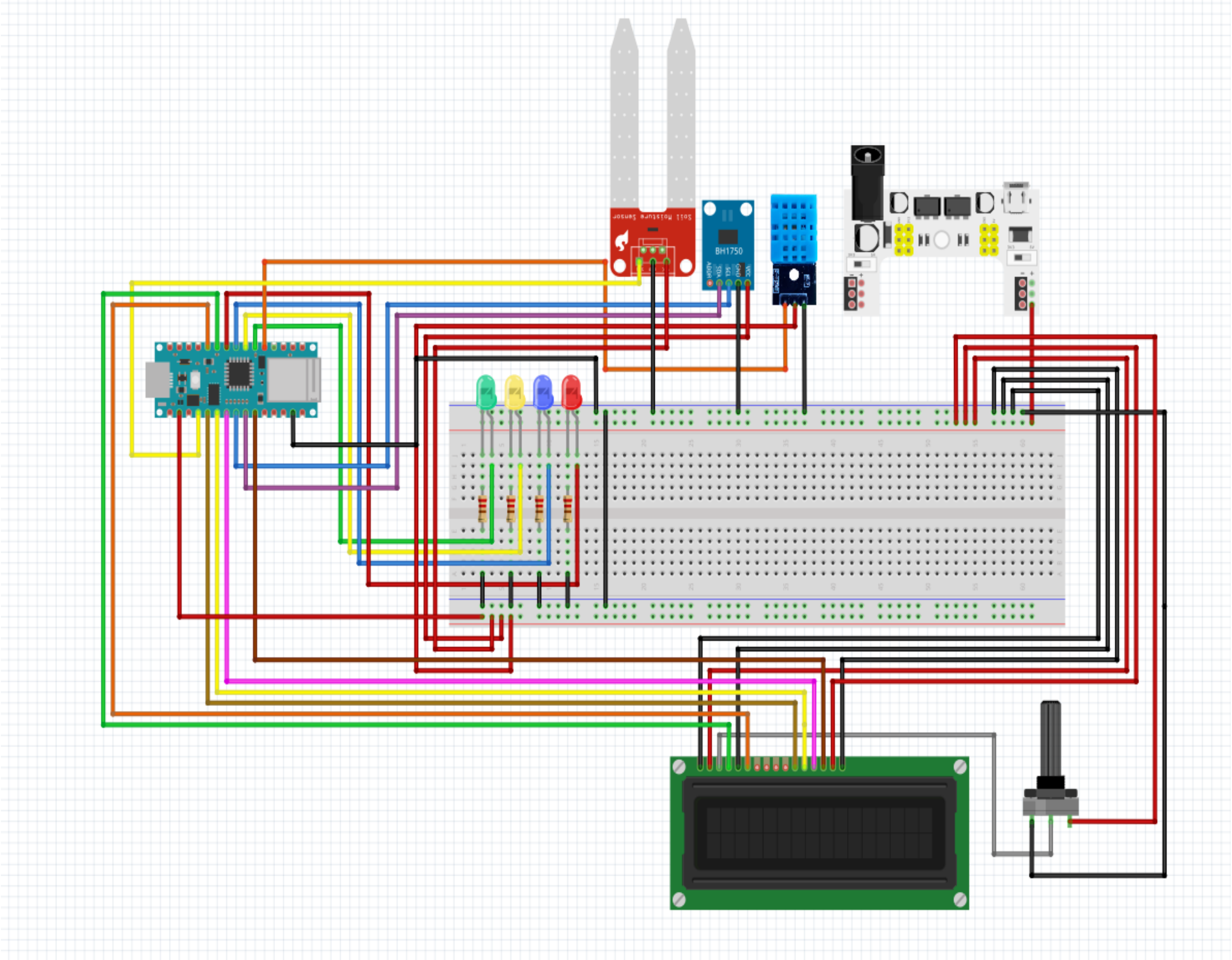
Component	**Function**
Arduino Nano 33 IoT	Microcontroller with WiFi Support
DHT11	Temperature and Humidity Sensor
GY-30 (BH1750FVI)	Light Intensity Sensor
SparkFun Soil Moisture Sensor	Soil Moisture Sensor
LCD1602 Module	LCD
Potentiometer (10k)	LCD screen adjustment
LEDs (4 LEDs used)	LEDs for user feedback
Breadboard	Wiring the system together
Breadboard Power Supply Module (MB102) **NOTE**: DC Barrel Jack Input from battery	Supplies 5V power to the LCD Acts as an ON/OFF switch for the system Connections to the Battery and the Microcontroller
Battery Pack (6x AA batteries) **NOTE**: DC Barrel Jack	Power for the system
Jumper Cables	Wiring the components

Section 4: Wiring and Pinout Diagram

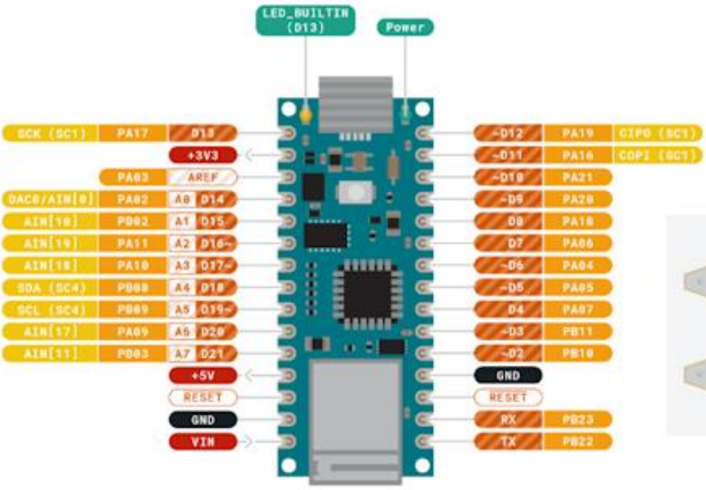
Wiring Diagram Table:

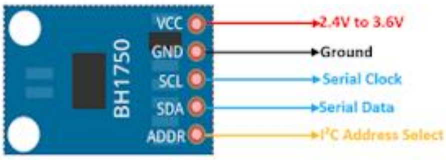
LCD Display	(connects to)		**DHT11 Sensor**	(connects to)		**Soil Moisture Sensor**	(connects to)
VSS	**Breadboard GND Rail**		Signal	**Breadboard 3.3V Rail**		Signal	A0
VDD	**Breadboard 5V Rail**		VIN/DATA	D2		GND	**Breadboard GND Rail**
V0	Potentiometer		GND	**Breadboard GND Rail**		VCC	**Breadboard 3.3V Rail**
RS	D7						
RW	**Breadboard GND Rail**		**GY-30 Sensor**	(connects to)		**LEDs**	(connects to)
E	D8		ADDR	**NOT USED**		Green	D3
D4	A1		SDA	A4		Yellow	D4
D5	A2		SCL	A5		Blue	D5
D6	A3		GND	**Breadboard GND Rail**		Red	D6
D7	A6		VCC	**Breadboard 3.3V Rail**		**ALL GND WIRES**	**Breadboard GND Rail**
A	**Breadboard 5V Rail**						
K	**Breadboard GND Rail**						
Arduino Nano 33 IoT			**Power Supply Module**				
3.3V	**Breadboard 3.3V Rail**		Right Rail Voltage Select Jumper (+ pin)	**Breadboard 5V Rail**			
GND	**GND Rail**						

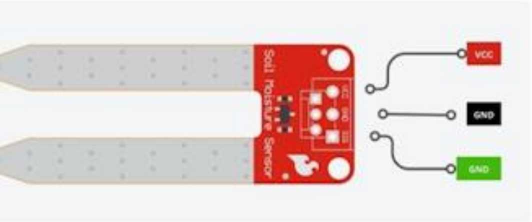
Wiring Diagram:

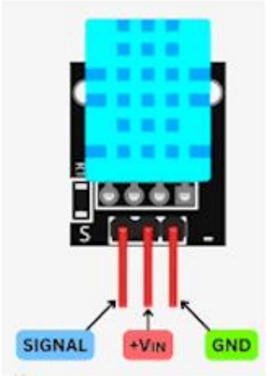


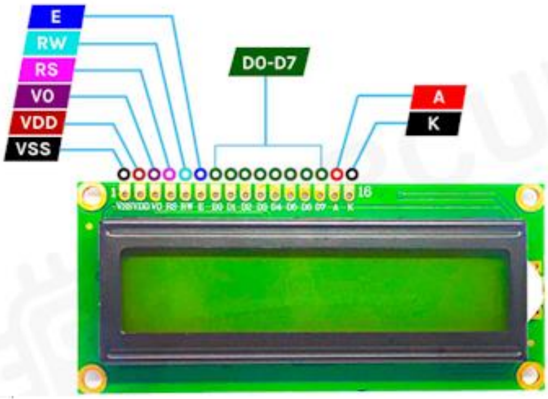
Pinout Diagram:

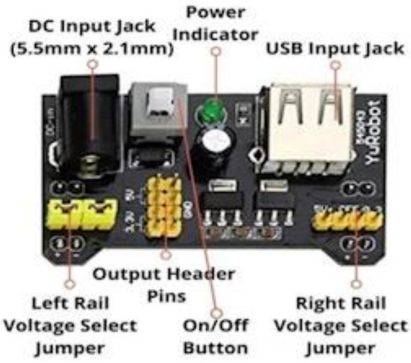











DC Input Jack (5.5mm x 2.1mm)
Power Indicator
USB Input Jack
Output Header Pins
Left Rail Voltage Select Jumper
On/Off Button
Right Rail Voltage Select Jumper
YwRobot
3.3V
5V
OFF

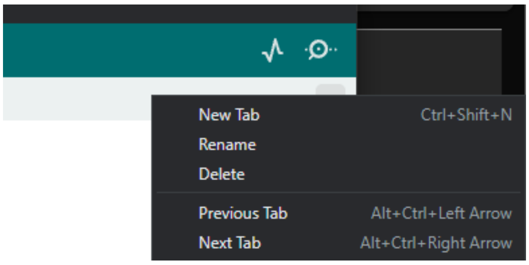
Section 5: Software Setup Steps

Arduino IDE Setup

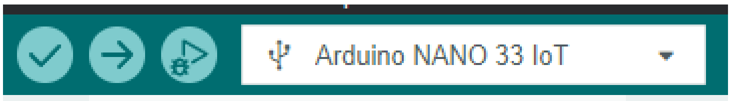
1. Install Arduino IDE: Download and install the latest version of the Arduino IDE from Arduino's website.
2. Install Libraries: Install the following libraries via the Library Manager:

LiquidCrystal	by Arduino, Adafruit
BH1750FVI	by PeterEmbedded
DHT sensor library	by Adafruit
Wire	**NOTE:** Installed by default
WiFiNINA	by Arduino

- secrets.h **NOTE:** This is created by clicking the three dots on the right side of the screen, selecting new tab and renaming the new tab to 'secrets.h'



3. Select Board: Choose Arduino Nano 33 IoT in the drop-down bar at the top of the page (see screenshot in step 4 where the 'Arduino NANO 33 IoT is)
4. Upload Code: Open the provided Arduino code downloaded from GitHub in the Arduino IDE and upload it to the Arduino board by using the **'Right arrow button'** at the top of the page



NOTE: GitHub link can be found in **Section 10:**

Wi-Fi Credentials

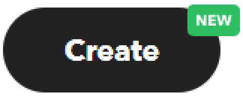
1. In the code, the Wi-Fi credentials are stored in a secrets.h file, which includes the SSID and password for connecting the Arduino to the Wi-Fi network.

- Edit these values with your own:
- Network name in WIFI_SSID[] = "#####";
- Network password in WIFI_PASSWORD[] = "#####";
- **NOTE**: Make sure when you replace the ##### you include the **" "** and **;** in the code

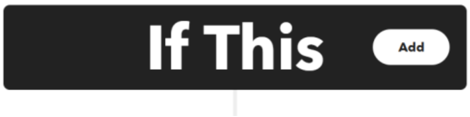
Section 6: IFTTT Setup

IFTTT Setup

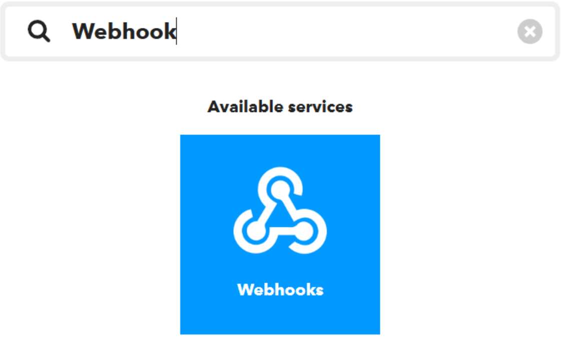
1. Create an account at IFTTT
2. Create a new Applet with Webhooks as the service to trigger. (Top Right Corner)



3. Click the **'Add'** in the **If this** banner

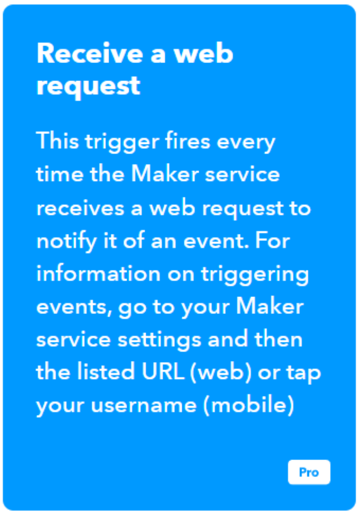


4. In the search bar type in **'Webhook'** and select the service



Webhook

5. Pick **'Receive a web request'**

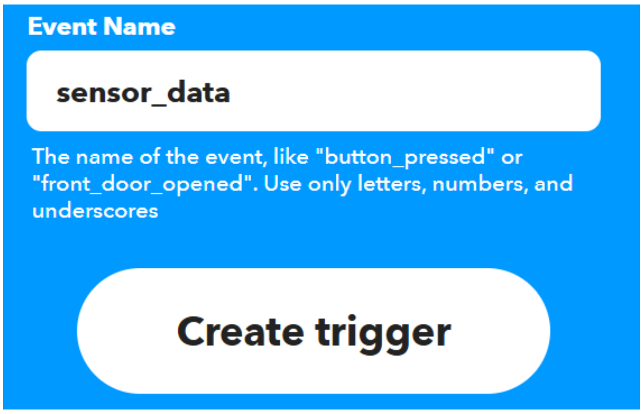


Receive a web request

This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap your username (mobile)

NOTE: This will require a premium account (Free for the first 7 days)

6. Create your event name such as **‘sensor_data’** and press **‘create trigger’.**

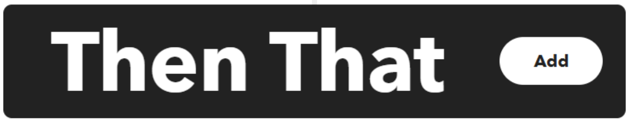


Event Name

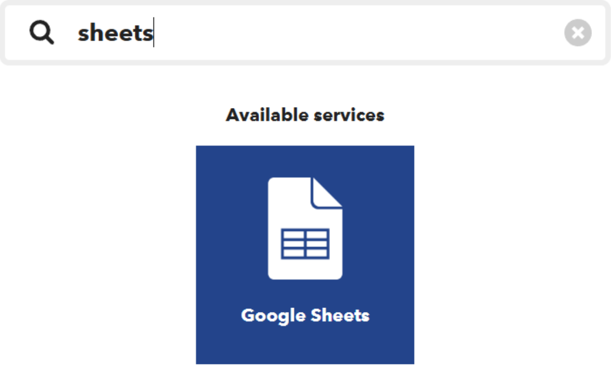
sensor_data

The name of the event, like "button_pressed" or "front_door_opened". Use only letters, numbers, and underscores

7. Click the ‘Add’ on the **Then that** banner

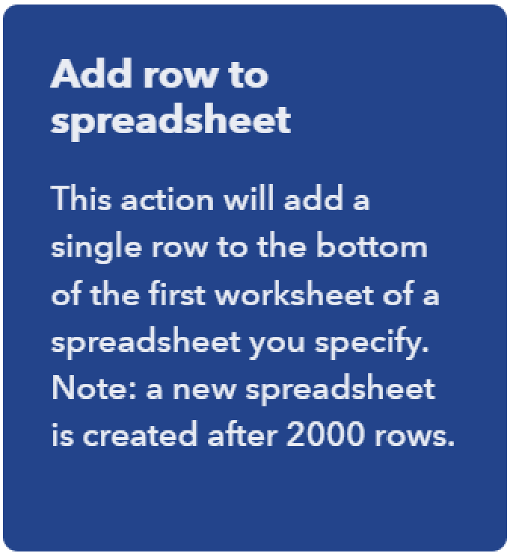


8. In the search bar, type in ‘**sheets’** and select the service



sheets

9. Pick **‘Add row to spreadsheet’**

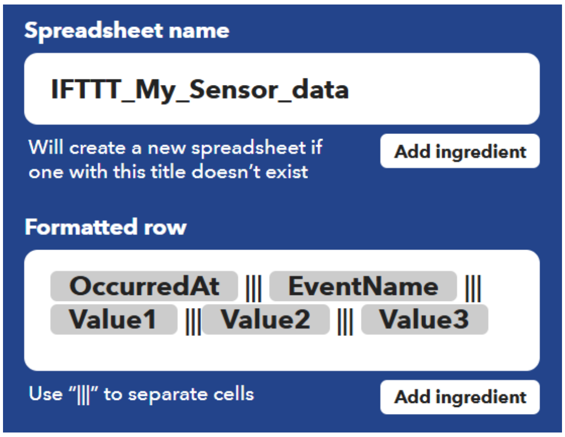


Add row to spreadsheet

This action will add a single row to the bottom of the first worksheet of a spreadsheet you specify. Note: a new spreadsheet is created after 2000 rows.

10. For this next step:

- Link together your Google Sheets account
- Create your spreadsheet name
- And you can leave the rest as default



Spreadsheet name

IFTTT_My_Sensor_data

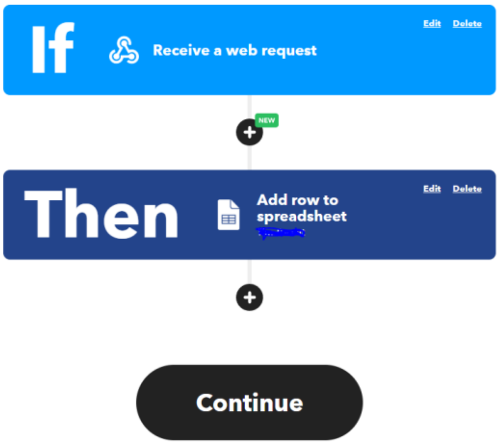
Will create a new spreadsheet if one with this title doesn't exist

Formatted row

OccurredAt ||| EventName ||| Value1 ||| Value2 ||| Value3

Use "|||" to separate cells

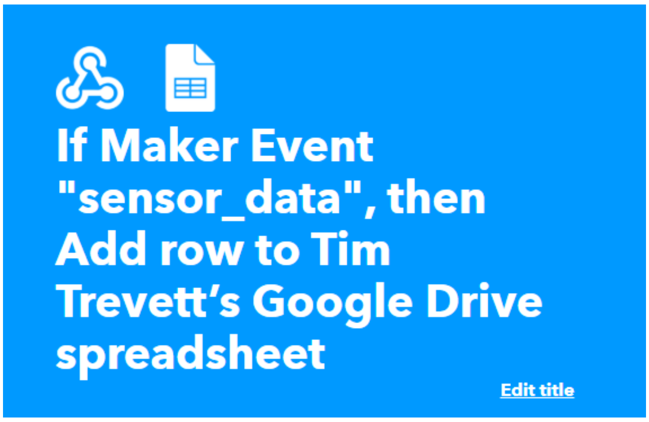
11. Press the **'Create Action'** at the bottom of the page
12. Press **'Continue'** when you see:



If Receive a web request

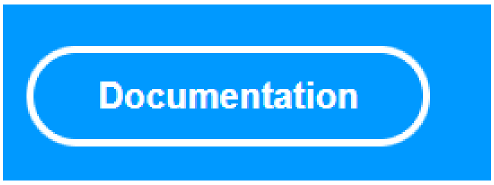
Then Add row to spreadsheet [redacted]

13. Press **'Finish'**
14. When you get to the next screen and see this page, click the '**webhook icon'** in the top left of this screenshot.

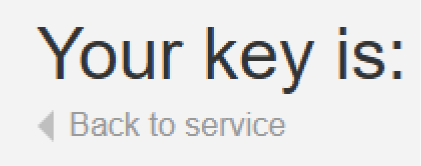


If Maker Event "sensor_data", then Add row to Tim Trevett's Google Drive spreadsheet

15. Select ‘Documentation on the page



16. You can see at the top of the page your **‘Key’**. You can copy this key for use in your code in the ‘secrets.h’ file.



Your key is:

For example it will look like:

```
#define IFTTT_EVENT "#####"

#define IFTTT_KEY "#####"
```

Replace the Event ##### with your event name from step 6.

NOTE: You can find this by pressing ‘**Back to service’** on that same page and then pressing ‘**My Applets**’ under the documentation button

Replace the ###### with your Key from that page.

17. Lastly, when your code activates its first webhook IFTTT event. Your spreadsheet will automatically create itself in Google Sheets and you can open that file to view your data.
    NOTE: You will need to make your own graphs to visualise the data.

Section 7: Operating instructions

Once the code has been uploaded and the circuit has been created. Proceed to Step 2.

Turning on the system:

- Plug in the Arduino Nano 33 IoT into the Power Supply module
- Plug the battery into the Power Supply module (DC Barrel Jack input)
- Press the ‘Toggle switch button’ in the middle of the Power Supply module.

1. The system will automatically start and display "Initialising..." on the LCD.
2. The Wi-Fi connection attempt will be shown on the LCD. Once connected, the system will display "Connected to Wi-Fi!".

Monitoring Data:

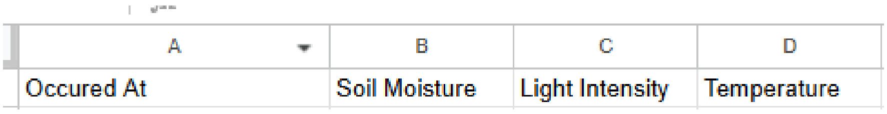
1. The system continuously reads sensor data every 60 seconds and updates the LCD with the current temperature, humidity, soil moisture, and light intensity.
2. Each sensor's threshold is checked, and the corresponding LED lights up if the value goes beyond the set limits.

Viewing Data on Google Sheets:

1. Data is sent to a Google Sheet via IFTTT for real-time monitoring. Each set of readings (light, temperature, soil moisture) is logged with a timestamp.

2. Open the Google Sheet to view the logged data and analyse trends.

   NOTE: Not a necessary step. Setting headers for the data before running the system allows for easy understanding of which data column relates to which sensor. Example below:



A	B	C	D
Occured At	Soil Moisture	Light Intensity	Temperature

Section 8: Troubleshooting

1. Sensor Failures

- DHT11 Sensor:
  - If the temperature or humidity values are being displayed as NaN (Not-a-Number) or as an absurdly high number, check the sensor jumper cable connections or try replacing the sensor
- GY-30 Sensor:
  - If light readings are inaccurate or fail, check the wiring and ensure the sensor is powered correctly
- Soil Moisture Sensor:
  - If the soil moisture readings appear to be stuck or incorrect, clean the sensor contacts and check if the sensor is submerged properly in the soil

    NOTE: Ensure the **'metal'** contact on the sensor is the only part in the soil, not the whole sensor itself.

2. Wi-Fi Connection Issues

- Failed Wi-Fi Connection
  - If the system is unable to connect to Wi-Fi by displaying ". . . . . " on the LCD, verify the Wi-Fi SSID and password in the secrets.h file.
  - If that is not the issue, turning on and off the system can help resolve the Wi-Fi Connectivity issue

    NOTE: The system does not automatically retry failed Wi-Fi connections in the current build iteration.

Section 9: FAQ

Q) How can I change the threshold values for the sensors?

A) Edit the threshold constants in the Arduino code (e.g., LIGHT_THRESHOLD, TEMP_LOW_THRESHOLD).

Q) How do I reset the system if it stops working?

A) Press the reset button on the Arduino or disconnect and reconnect the power supply.

Q) Why isn't the Wi-Fi reconnecting?

A) Make sure the Wi-Fi SSID and password are correct in the secrets.h file. Also, check the Wi-Fi signal strength.

NOTE: System may need to be turned ON/OFF to reconnect

Q) Can I add more sensors to the system?

A) Yes, you can add additional sensors (e.g., CO2 sensor) by connecting them to the available pins and modifying the code to handle their data

NOTE: Due to limited pin availability, check the wiring and pinout diagram to confirm whether more sensors can be added.

Section 10: Code Listing

GitHub Link:

https://github.com/TimTrev/SIT210_Task11.1_Prototype

Video Demo Link:

https://youtu.be/dZKQn37gzRU