Report

Detecting and controlling temperature with humidity of planting trees in the shade

Prepared By

6404062636528 Samanya Daengdej

Present to

Assoc. Prof. Kobkiat Saraubon, Ph.D

This report is part of the Internet of Things subject.

Department of Computer and Information Science

Term 2 Year of education 2023

King Mongkut's University of Technology North Bangkok

Preface

This report is part of the Internet of Things course, aiming to explore the application of temperature and humidity control in shaded tree planting. It is intended to aid understanding and serve as a resource for the class.

The author hopes this report proves beneficial to readers and students interested in the topic. Any suggestions or corrections are welcomed and appreciated.

Samanya Daengdej

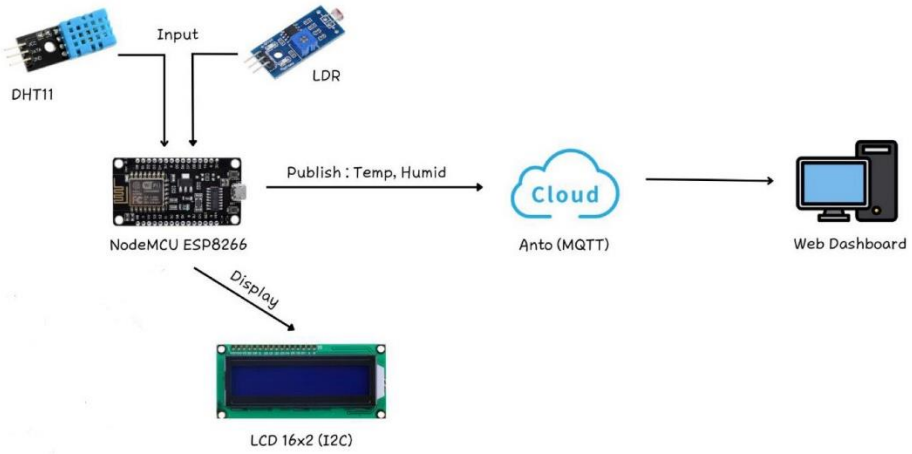
11 March 2024

Table of Contents

Description

Detection and control system for temperature and humidity for plants are highly beneficial tools in efficiently caring for and promoting the growth of plants. They enable cultivators to continuously monitor the environment, particularly temperature and humidity, which are critical factors in plant growth. Therefore, controlling and maintaining the air conditions in the area where plants grow is especially crucial.

Architecture



The greenhouse plant monitoring system is designed to optimize the greenhouse environment for plant growth, leading to higher yields and improved quality of plants for export or local markets. The system enables efficient control and adjustment of the climate in the monitored greenhouse according to the specific needs of the plants.

Light Level Monitoring :

The system utilizes light sensors to detect the light level in the greenhouse, determining whether it is day or night. This information is used to adjust the lighting conditions in the greenhouse accordingly, such as turning on lights during the day and dimming them at night to create a suitable environment for plant sleep.

Temperature and Humidity Monitoring :

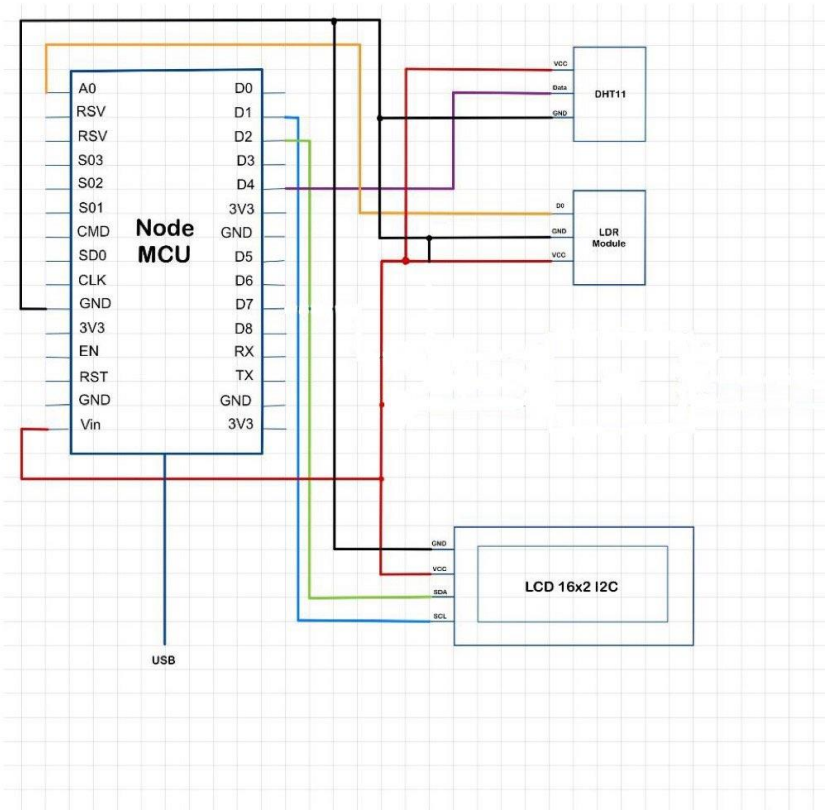
The system employs DHT11 sensors to measure temperature and humidity levels inside the greenhouse. This data is used to adjust ventilation and humidity control to provide optimal conditions for plant growth. Generally, the ideal temperature for plant growth ranges between 25-30 degrees Celsius, and the system regulates humidity levels to prevent plant diseases and issues related to excessively high or low humidity.

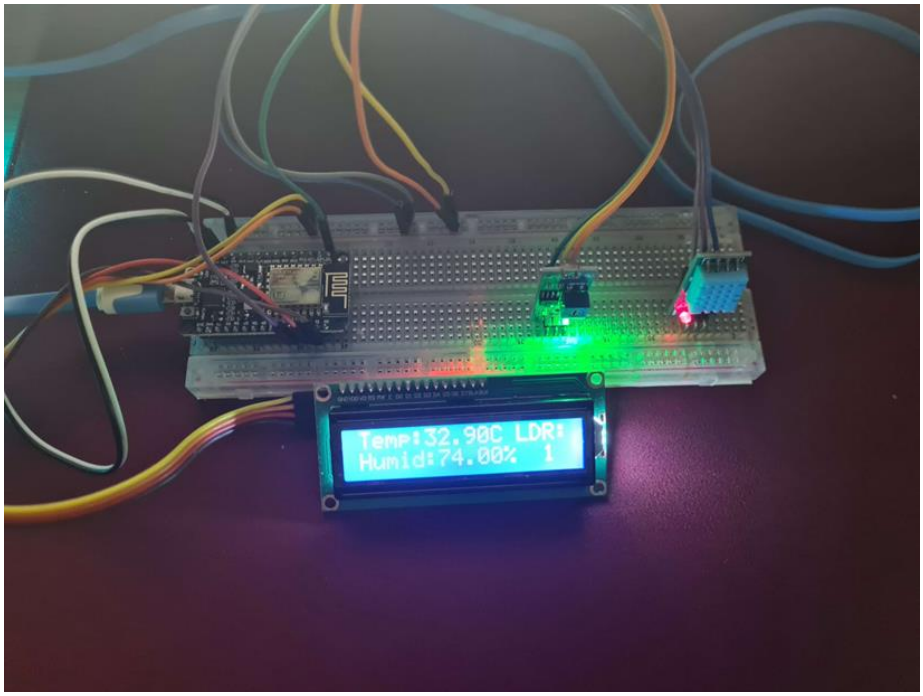
Dashboard Display Cloud System :

Data collected from the sensor system is transmitted to a dashboard through a cloud system. Users can monitor the status and customize the monitoring parameters as needed. Real-time data on the dashboard allows users to be aware of the greenhouse conditions and make timely adjustments based on the plants' requirements.

The greenhouse plant monitoring system is a powerful tool for greenhouse environment management. It empowers growers to enhance plant yield and quality by maintaining precise climate control tailored to the specific needs of their plants.

Diagram


A0
RSV
RSV
S03
S02
S01
CMD
SD0
CLK
GND
3V3
EN
RST
GND
Vin
D0
D1
D2
D3
D4
3V3
GND
D5
D6
D7
D8
RX
TX
GND
3V3
Node
MCU
USB
VCC
Data
GND
DHT11
D0
GND
VCC
LDR
Module
GND
VCC
SDA
SCL
LCD 16x2 I2C


Temp:32.90C LDR:
Humid:74.00% 1

Code

18_Samanya_Daengdej_Project.ino

```
#include <AntoIO.h>
#include <DHT.h>
#include <LiquidCrystal_I2C.h>

#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

int sensorPIN = A0;
int sensorValue = 0;

LiquidCrystal_I2C lcd(0x27, 16, 2); // Change the I2C address and dimensions if different

const char* ssid = "Galaxy888"; //ชื่อwifi
const char* pass = "ylzz3835"; //รหัสwifi
const char* user = "Protile"; //cloudของ Anto
const char* token = "Zx19FXn9Xk8PpYUUEBX1XuyPgAfebCfXD31DvrM7"; //key จาก Anto
const char* thing = "DHT11"; //อุปกรณ์ที่ใช้
AntoIO anto(user, token, thing); //กำหนดตัวแปรให้ anto

void setup() {
  Serial.begin(115200);

  lcd.begin(); // Initialize the LCD with 16 columns and 2 rows
  lcd.backlight();
  lcd.home();
  lcd.print("Hello, world!");
  lcd.clear();

  delay(10);

  Serial.println();
  Serial.println();
  Serial.println("Anto library version: ");
  Serial.println(anto.getVersion());
```

18_Samanya_Daengdej_Project.ino

```
  Serial.print("\nTrying to connect ");
  Serial.print(ssid);
  Serial.println("...");

  anto.begin(ssid, pass, messageReceived);
  Serial.println("\nConnected Anto done");

  pinMode(D7, OUTPUT);
  dht.begin();
}

void loop() {
  anto.mqtt.loop();
  if (!anto.mqtt.isConnected()) {
    Serial.println("Disconnected"); //เช็คถ้าไม่ได้Connect ให้แจ้งเตือน
  }

  sensorValue = digitalRead(analogRead(sensorPIN));

  float humid = dht.readHumidity();
  float temp = dht.readTemperature();

  if (isnan(humid) || isnan(temp)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  //LCD display
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Temp:");
  lcd.print(temp);
  lcd.print("C ");
  lcd.print("LDR: ");
  lcd.setCursor(0, 1);
```

18_Samanya_Daengdej_Project.ino

```
72    lcd.print("Humid:");
73    lcd.print(humid);
74    lcd.print("%  ");
75
76    lcd.print(sensorValue);
77
78
79    //serial
80    Serial.print(F("Humidity: "));
81    Serial.print(humid);
82    Serial.print(F("% Temperature: "));
83    Serial.print(temp);
84    Serial.print(F("C "));
85    Serial.print("LDR: ");
86    Serial.print(sensorValue);
87    Serial.println();
88
89    //Publish to Cloud
90    if (temp >= 0) {
91      anto.pub("Temp", temp);
92    }
93    if (humid >= 0) {
94      anto.pub("Humid", humid);
95    }
96    if (sensorValue == 0) {
97      anto.pub("LDR_1", "Light");
98    }
99    if (sensorValue == 1) {
100     anto.pub("LDR_1", "No Light");
101   }
102
103   delay(500);
104 }
```

```
105
106 void messageReceived(String thing, String channel, String payload) {
107   Serial.print("Recieved: ");
108   Serial.print(thing);
109   Serial.print("/");
110   Serial.print(channel);
111   Serial.print("-> ");
112   Serial.println(payload);
113 }
```

Data Structure

Use the MQTT Protocol to send data MQTT Topic: anto/key/channel

Anto/ xd5yrm4uDvGZI9CFv9FaqrbyEYjFpjG4ZuoyETtK/Temp

Anto / xd5yrm4uDvGZI9CFv9FaqrbyEYjFpjG4ZuoyETtK/Humid

IoT Platform / Service

Features:

1. Dashboard:

Highlights / Strength :

- User-friendly device management and control interface.
- Easy setup and customization options.
- Strong data transmission security measures.
- Efficient data creation and management capabilities.

Weaknesses :

1. Complexity may pose challenges for inexperienced users.
2. Risk management required due to potential cyber threats.

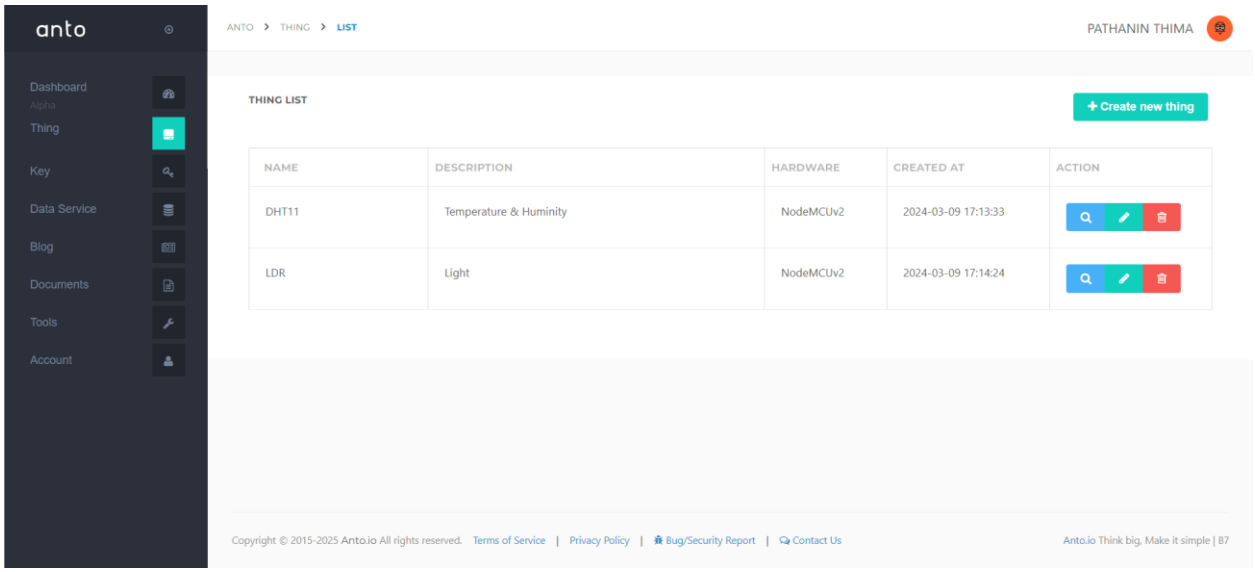
Package Specifications:

1. Connectivity: Supports Wi-Fi, Ethernet, and 3G/4G(Hotspot).
2. Programming Language: Compatible with Python, C++, Arduino, and Node.js.
3. Data Collection: Capable of handling up to 10 million data points daily.
4. Device Management: Supports management of up to 100,000 devices.
5. Communications: Offers various communication protocols like MQTT, HTTP, HTTPS, WebSocket, and CoAP.
6. Security: Provides encryption and threat detection mechanisms.

7. Free Service: Offers a free tier with some limitations.

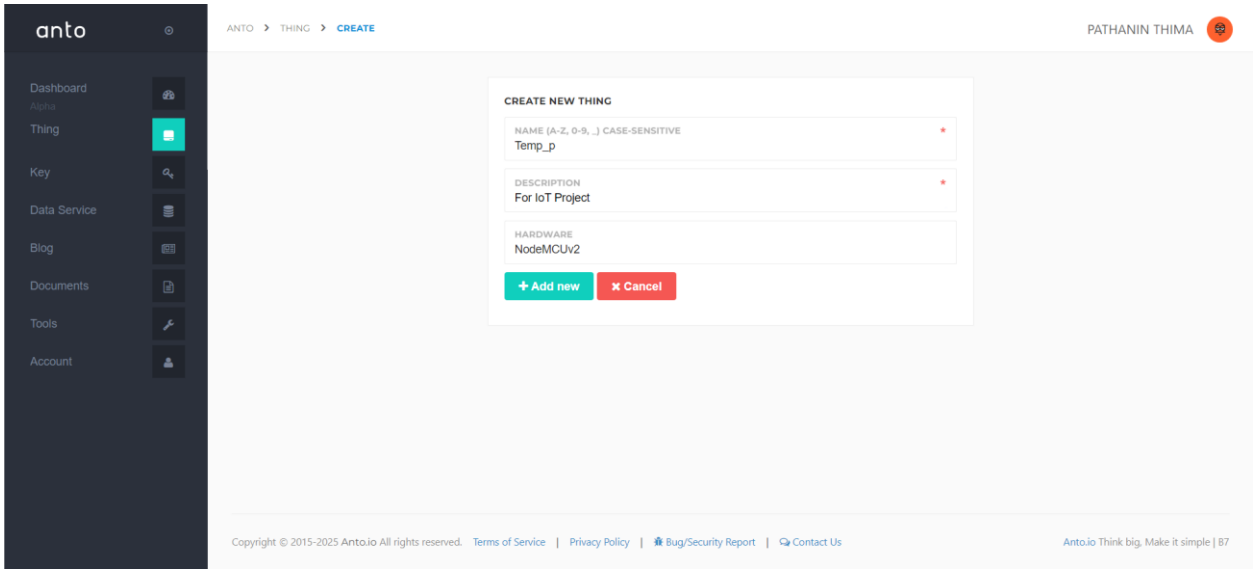
Setup Steps

Step 1: Come to the home page of https://www.anto.io/login and log in successfully.

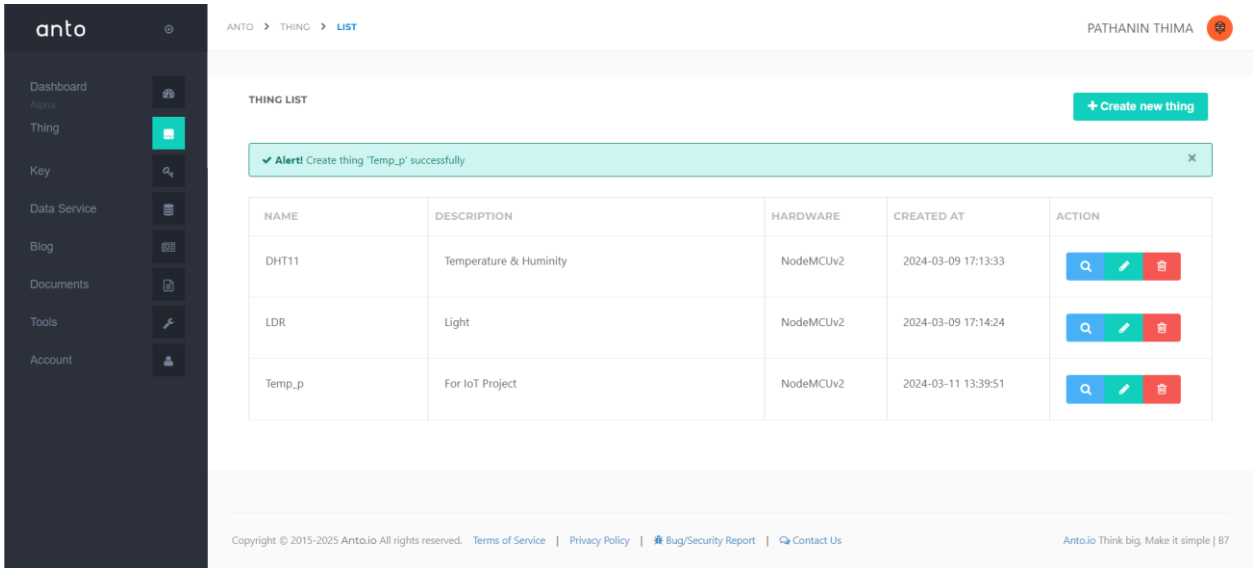


Step 2: Press the Create new thing button.

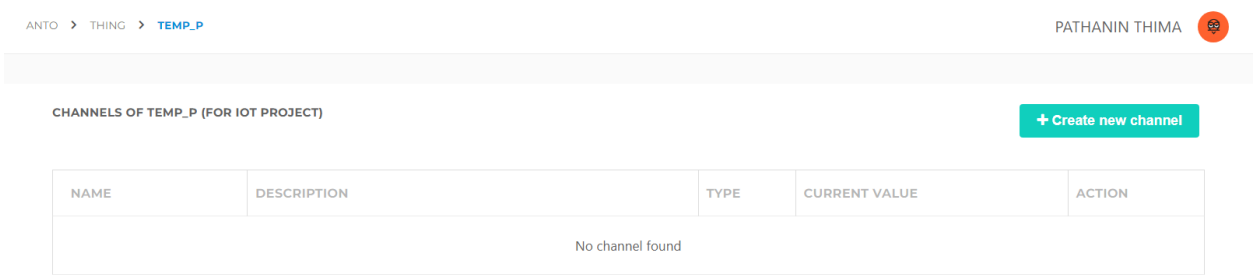
Step 3: Enter a name, fill in complete information, and then press Add new.



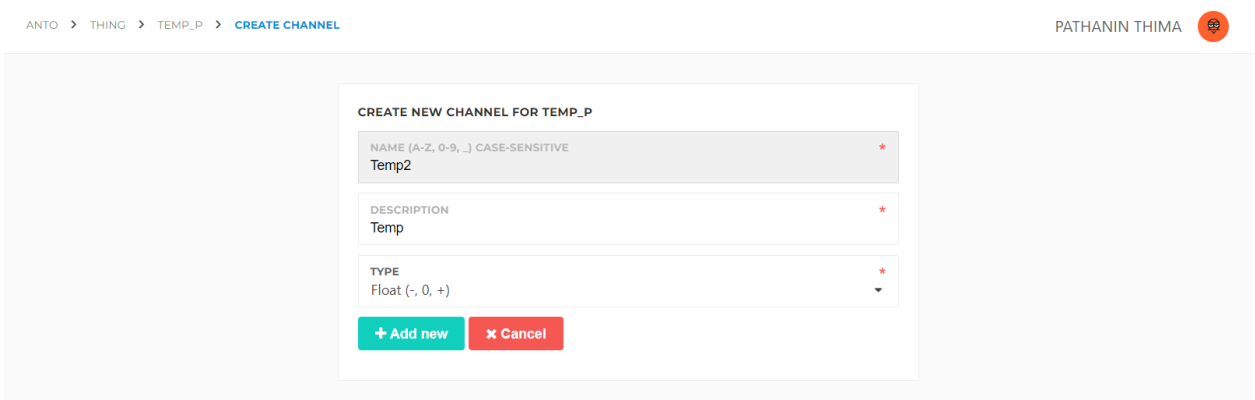
Step 4: Press Temp_p's magnifying glass.



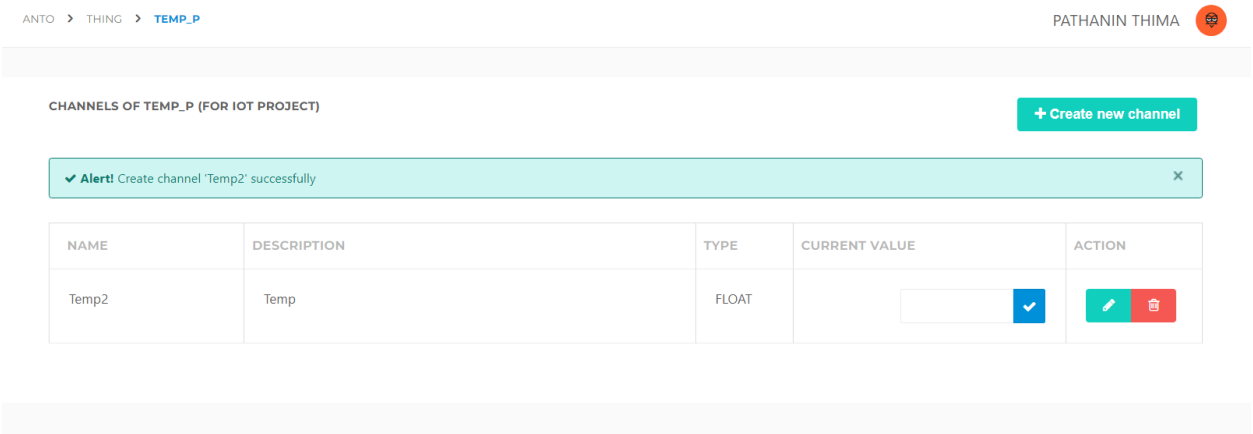
Step 5: Press the Create new channel button.



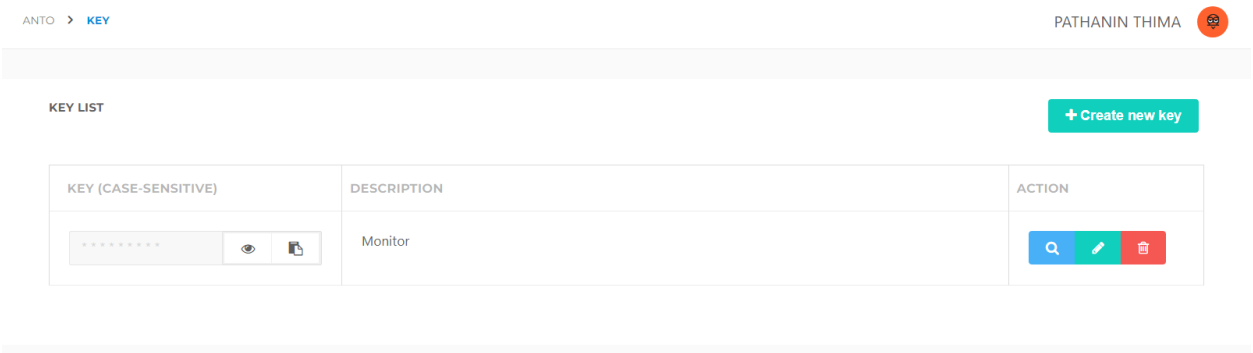
Step 6: Enter all the information as required and press the Add new button.



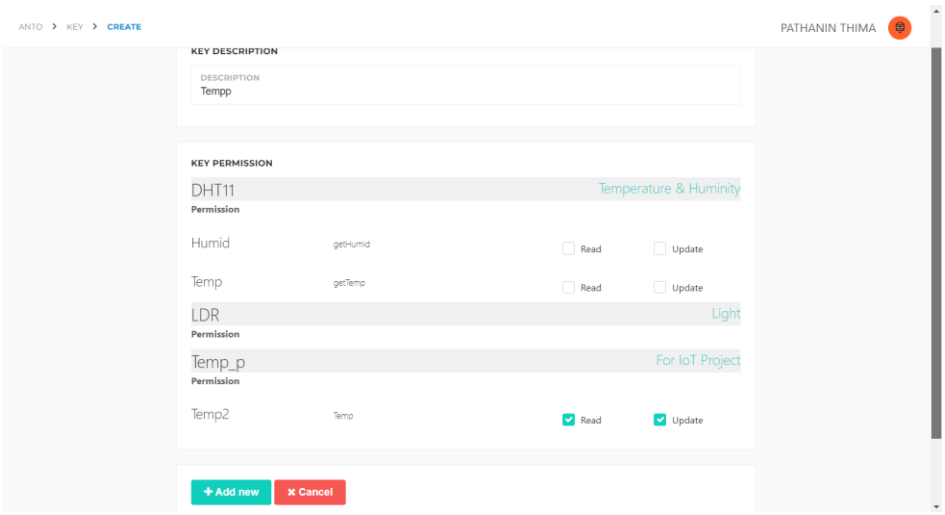
as follows:



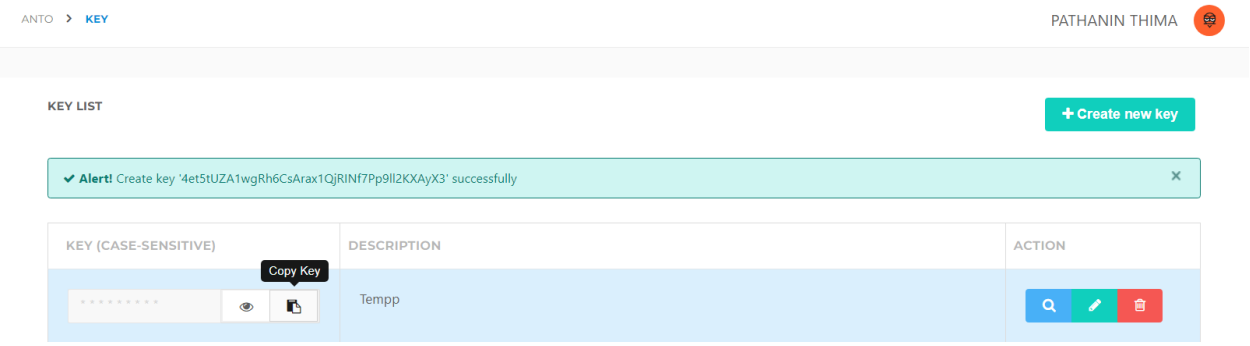
Step 7: When finished, click on the key bar on the left and press the Create new key button.



Step 8: Set the key. Select read and update of Temp_p and press the Add new button.

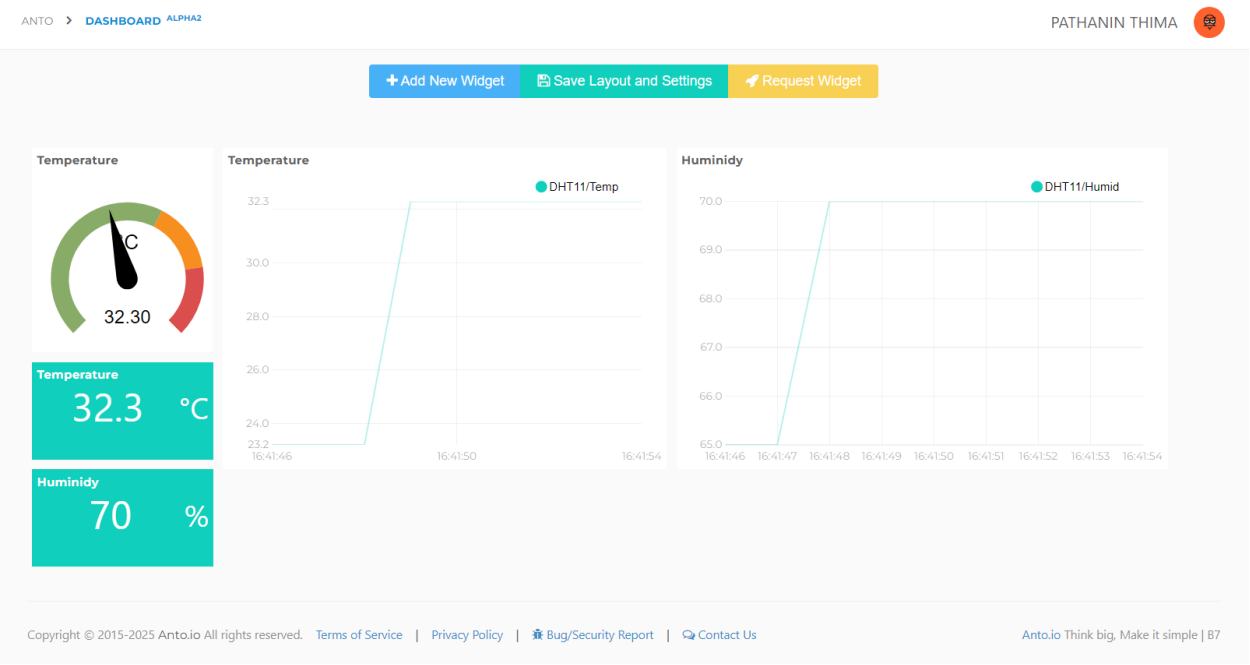


Step 9: Copy the key and put it in the code and run it.



Step 10: Click on the Dashboard bar on the left, then press the Add New Widget button, then set the correct and complete settings, then press the Save Layout and Settings button.

as follows:



Data transmission between devices and Anto's IoT Platform :

Anto IoT can seamlessly connect and communicate with various IoT platforms, supporting communication protocols such as MQTT, HTTP, HTTPS, Websocket, and CoAP, making it easy to connect Anto IoT with different IoT platforms without worrying about communication between devices and different IoT platforms. This flexibility allows users to choose an IoT platform that meets their needs and project requirements freely.