****

**IOT102 PROJECT**

**LoRa-Based Transmission**

**Group member: Trần Bảo Toàn SE173140**

**Ngô Huỳnh Tấn Lộc SE173126**

**Đoàn Ngọc Tuân SE173227**

**HO CHI MINH CITY, 2023**

**I. PROJECT DESCRIPTIONS**

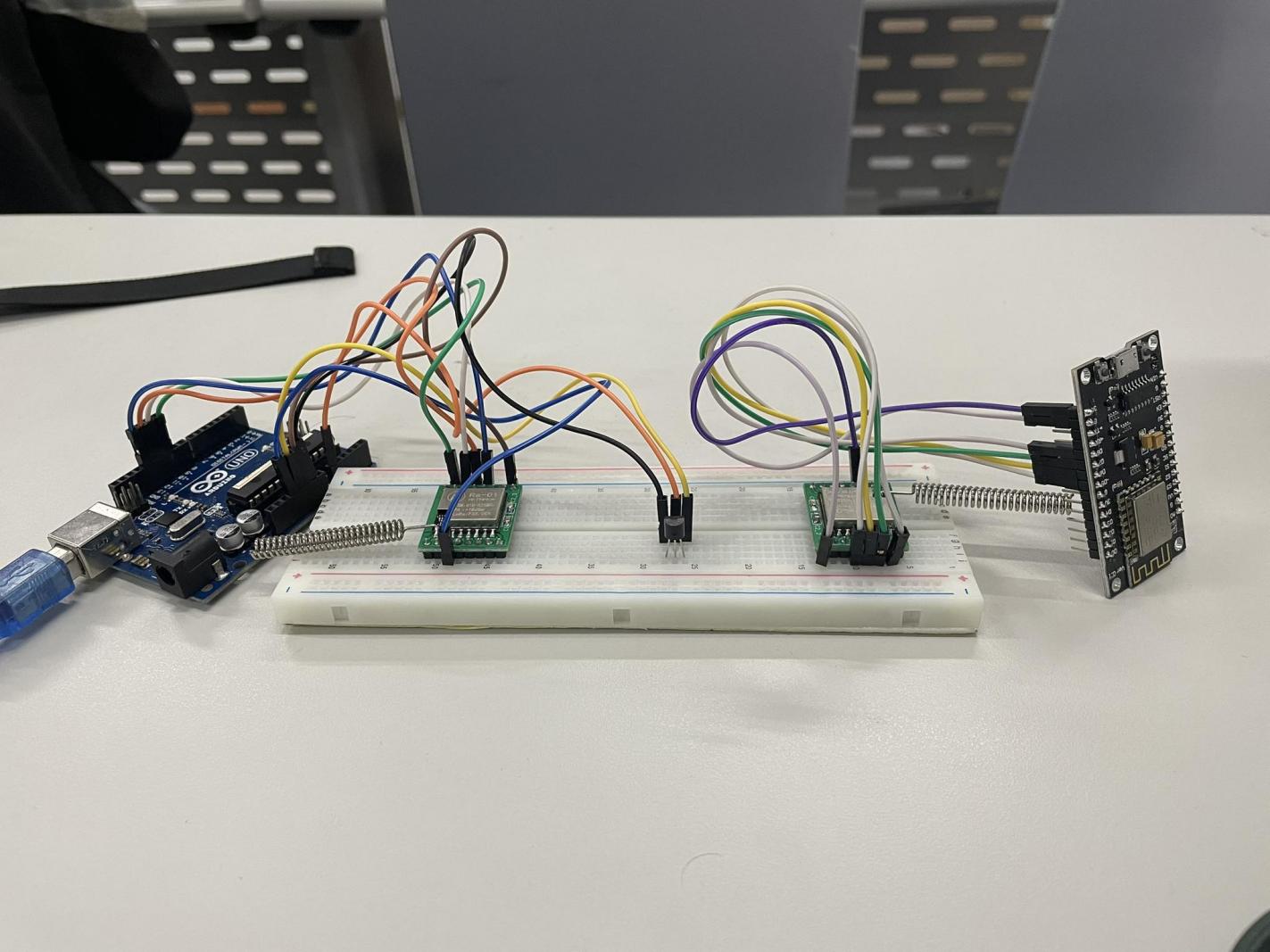
- ESP8266 LoRa Gateway for Arduino and LM35 Temperature Sensor- In this project, we will see how to make ESP8266 LoRa Gateway for Arduino and LM35 temperature sensor. Since this is an IoT or internet of things based project; so the temperature can be monitored from any part of the world provided if the internet connection is available.

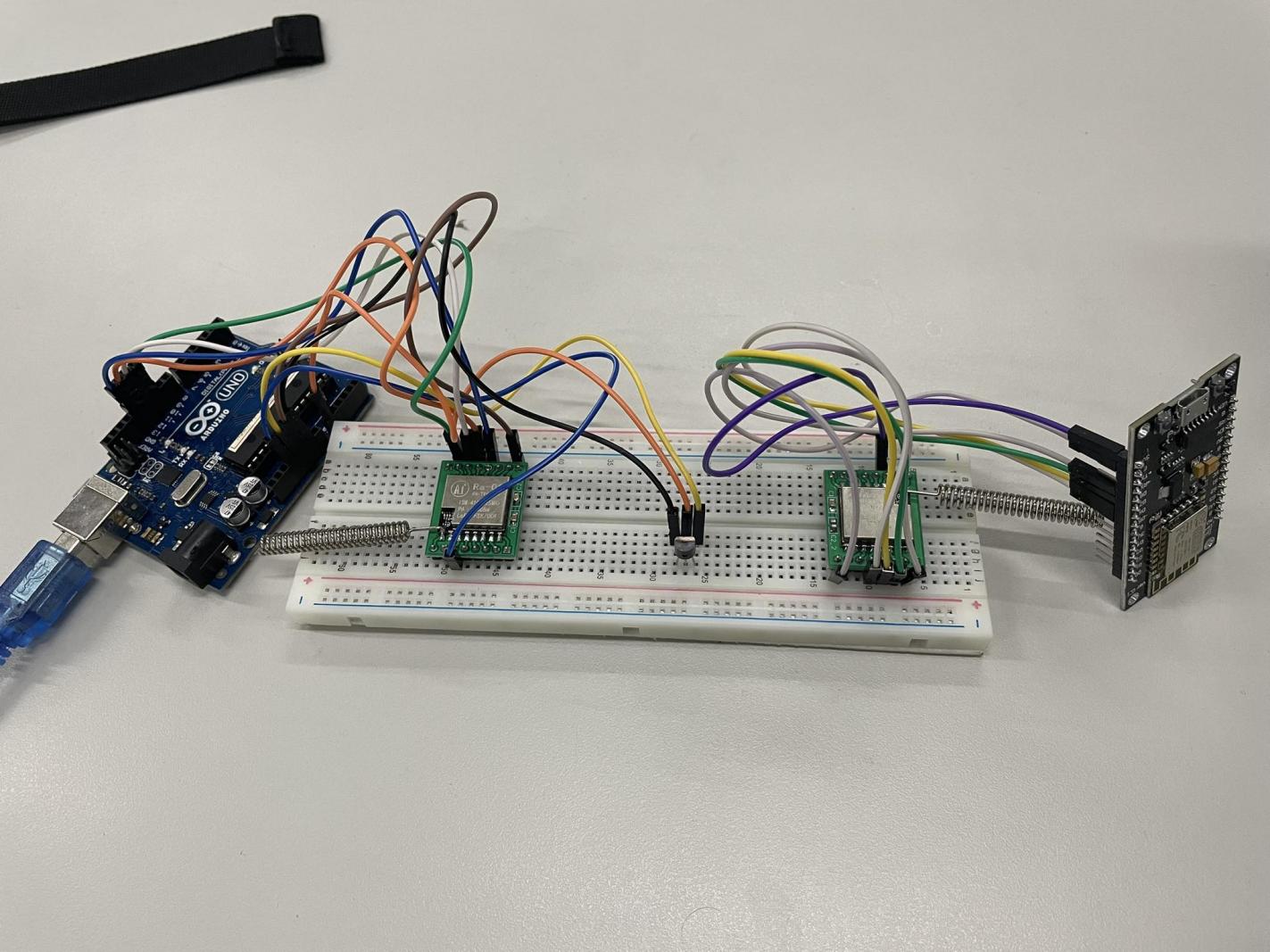
- Use Blynk App to control LEDs online from any where.

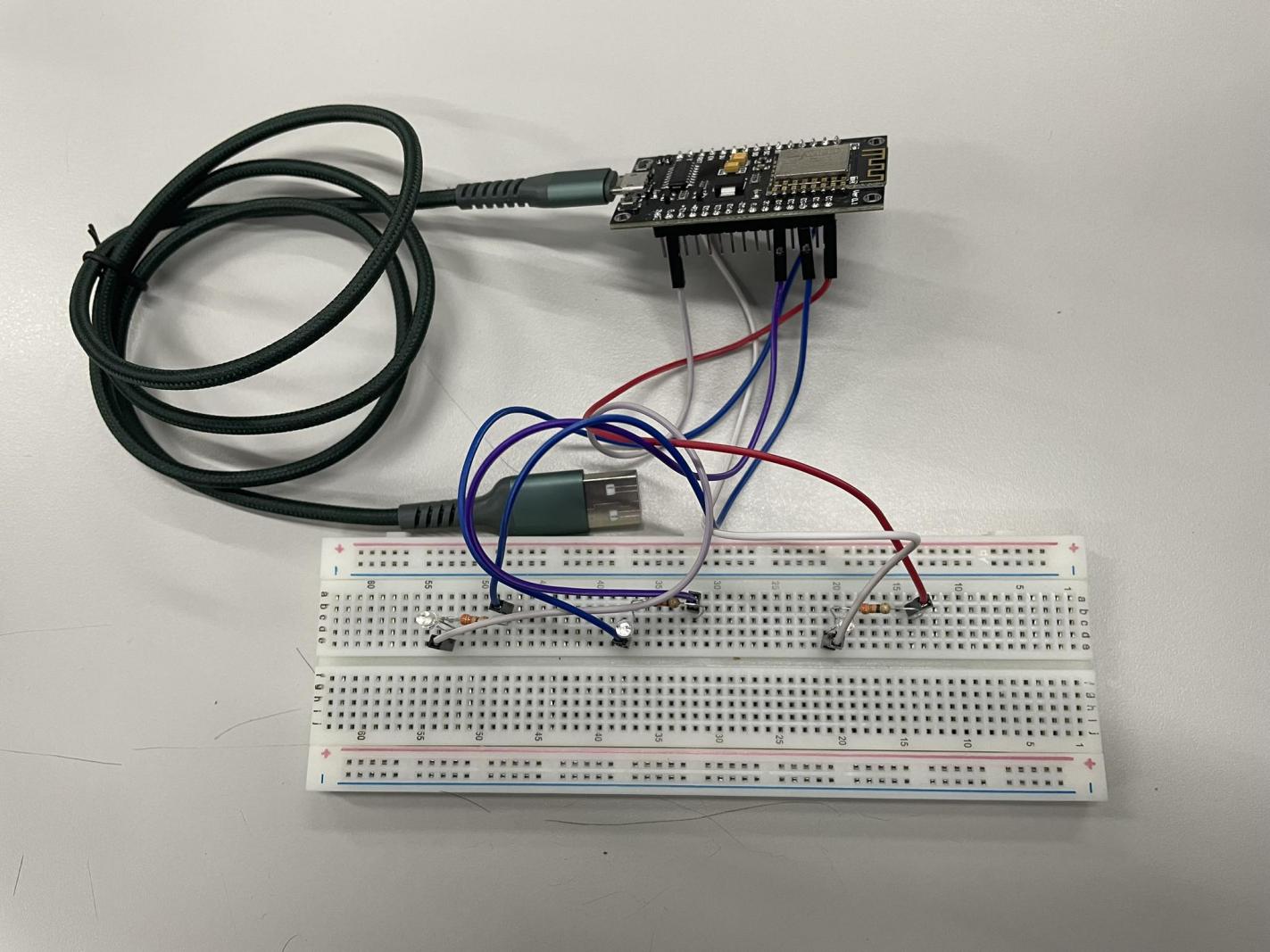
**II. REQUIRED HARWARE**

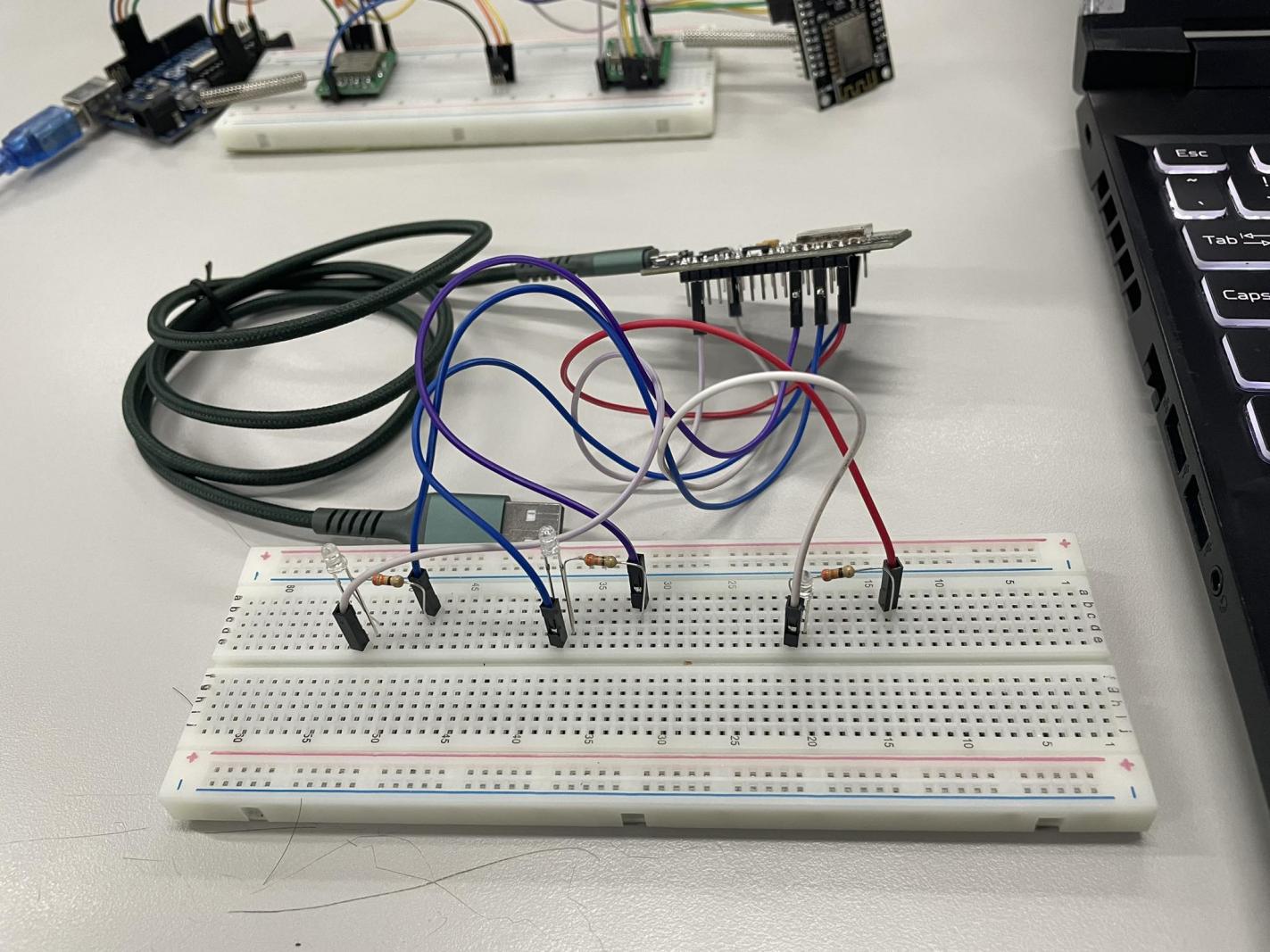
* Arduino Uno
* 2 Bread board
* 2 Module LoRa
* 2 ESP8266 Wifi Module
* 3 LEDs
* LM35 Temperature Sensor

**III. CIRCUIT**



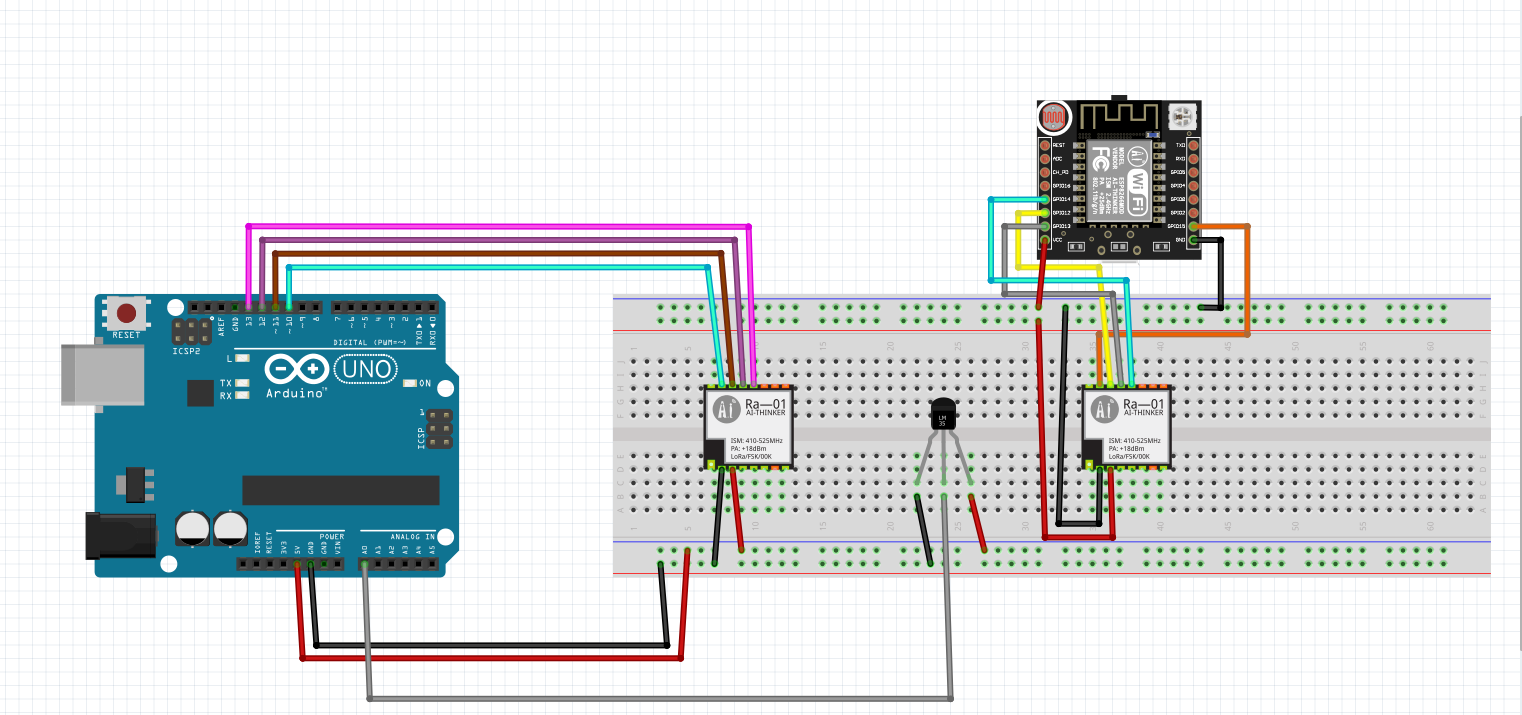




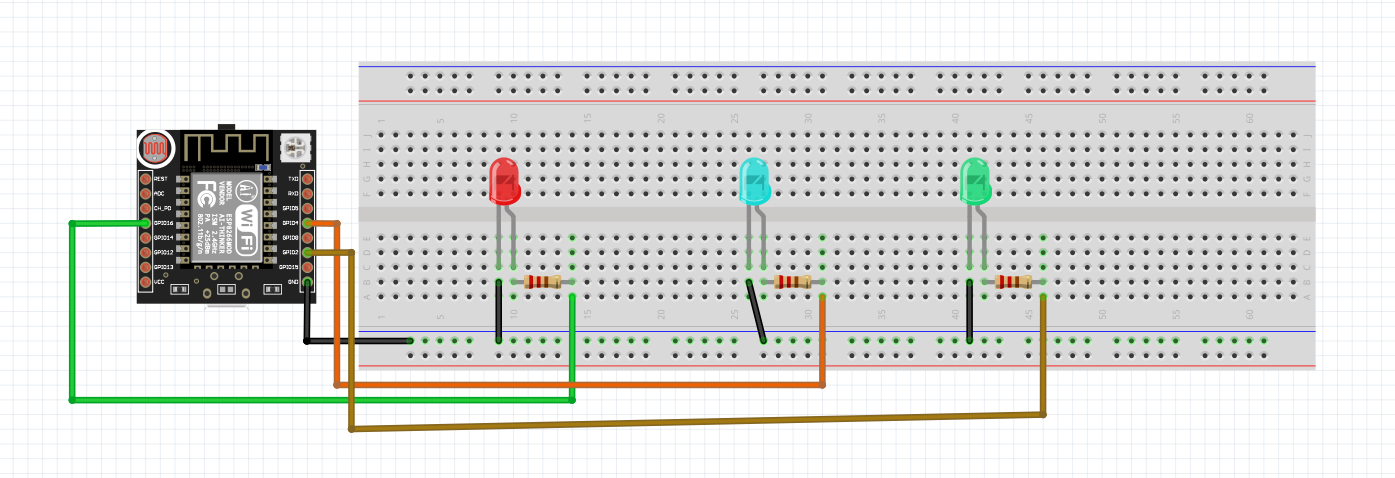


**III. SCHEMATIC**

**-** LoRa to Aruino & LoRa to ESP8266

****

- Control LEDs using ESP8266 and Blynk App



**IV. CODE**

- Code for LoRa to Arduino

|  |
| --- |
| // include libraries  #include <SPI.h>  #include <LoRa.h>  #include <Wire.h>  String outgoing;  // outgoing message  byte msgCount = 0;  // count of outgoing messages  byte MasterNode = 0xFF;  byte Node1 = 0xBB;  //Declare Temp Pin  int temperature\_pin = 0;  String Mymessage = "";  void setup() {    Serial.begin(9600);  // initialize serial    Wire.begin();    pinMode(temperature\_pin, INPUT);    if (!LoRa.begin(433E6)) {      Serial.println("LoRa init failed. Check your connections.");      while (true)        ;  // check if start LoRa failed, do nothing    }    LoRa.setTxPower(20);    delay(2000);  }  void loop() {    //Read temp    int rawvoltage = analogRead(temperature\_pin);    float millivolts = (rawvoltage / 1024.0) \* 5000;    float celsius = millivolts / 10;    float ftemp = (celsius \* 9) / 5 + 32;  //Print temp to Serial Monitor    Serial.print(millivolts);    Serial.println(" millivolts");    Serial.print(celsius);    Serial.println(" degrees Celsius");    Serial.print((celsius \* 9) / 5 + 32);    Serial.println(" degrees Fahrenheit");    delay(1000);      Mymessage = Mymessage + celsius + "," + ftemp;    sendMessage(Mymessage, MasterNode, Node1);    delay(100);    Mymessage = "";    // }  }  void sendMessage(String outgoing, byte MasterNode, byte otherNode) {    LoRa.beginPacket();             // start packet    LoRa.write(MasterNode);         // add destination address    LoRa.write(Node1);              // add sender address    LoRa.write(msgCount);           // add message ID    LoRa.write(outgoing.length());  // add payload length    LoRa.print(outgoing);           // add payload    LoRa.endPacket();               // finish packet and send it    msgCount++;                     // increment message ID  } |

- Code for LoRa to ESP8266

|  |
| --- |
| #include <SPI.h>  // include libraries  #include <LoRa.h>  #include <Wire.h>  #include <BlynkSimpleEsp8266.h>  #define Nss 15  // D8 pin CS  #define rst 16  // D0  #define dio0 2  // D4  int buzzer = D3;  char auth[] = "f8ZJiDonWir78en\_TYx1BlWbkGNCWjrj"; //token from template on Blynk  ///Connect SCL to D1 and Connect SDA to D2  /\* WiFi credentials \*/  char ssid[] = "FPTU\_Student"; //Your Wifi’s name  char pass[] = "12345678"; //Your Wifi’s password  byte MasterNode = 0xFF;  byte Node1 = 0xBB;  String SenderNode = "";  String outgoing;    // outgoing message  byte msgCount = 0;  // count of outgoing messages  String incoming = "";  float ctemp;  float ftemp;  void setup() {    Serial.begin(9600);  // initialize serial    LoRa.setPins(Nss, rst, dio0);    if (!LoRa.begin(433E6)) {  // initialize ratio at 915 MHz      Serial.println("LoRa init failed. Check your connections.");      while (true)        ;  // if failed, do nothing    }    Serial.println("LoRa init succeeded.");    Blynk.begin(auth, ssid, pass);  }  void loop() {    Blynk.run();    // parse for a packet, and call onReceive with the result:    onReceive(LoRa.parsePacket());  }  void onReceive(int packetSize) {    if (packetSize == 0) return;  // if there's no packet, return    // read packet header bytes:    int recipient = LoRa.read();  // recipient address    byte sender = LoRa.read();    // sender address    if (sender == 0XBB)      SenderNode = "Node1:";    byte incomingMsgId = LoRa.read();   // incoming msg ID    byte incomingLength = LoRa.read();  // incoming msg length    while (LoRa.available()) {      incoming += (char)LoRa.read();    }    if (incomingLength != incoming.length()) {  // check length for error      //Serial.println("error: message length does not match length");      ;      return;  // skip rest of function    }    // if the recipient isn't this device or broadcast,    if (recipient != Node1 && recipient != MasterNode) {      // Serial.println("This message is not for me.");      ;      return;  // skip rest of function    }    // if message is for this device, or broadcast, print details:    //Serial.println("Received from: 0x" + String(sender, HEX));    //Serial.println("Sent to: 0x" + String(recipient, HEX));    //Serial.println("Message ID: " + String(incomingMsgId));    // Serial.println("Message length: " + String(incomingLength));    // Serial.println("Message: " + incoming);    //Serial.println("RSSI: " + String(LoRa.packetRssi()));    // Serial.println("Snr: " + String(LoRa.packetSnr()));    // Serial.println();    String q = getValue(incoming, ',', 0);    String r = getValue(incoming, ',', 1);    ctemp = q.toFloat();    Serial.print(ctemp);    Serial.println(" C");    ftemp = r.toFloat();    Serial.print(ftemp);    Serial.println(" F");    Blynk.virtualWrite(V1, ctemp);    Blynk.virtualWrite(V2, ftemp);    if (ctemp >= 45) {      Serial.println("To hot");    }    if (ctemp < 45) {      Serial.println("Normal");    }    //delay(1000);    incoming = "";  }  String getValue(String data, char separator, int index) {    int found = 0;    int strIndex[] = { 0, -1 };    int maxIndex = data.length() - 1;    for (int i = 0; i <= maxIndex && found <= index; i++) {      if (data.charAt(i) == separator || i == maxIndex) {        found++;        strIndex[0] = strIndex[1] + 1;        strIndex[1] = (i == maxIndex) ? i + 1 : i;      }    }    return found > index ? data.substring(strIndex[0], strIndex[1]) : "";  } |

- Code for Control 3 LEDs

|  |
| --- |
| #define BLYNK\_PRINT Serial  #include <ESP8266WiFi.h>  #include <BlynkSimpleEsp8266.h>  #define BLYNK\_TEMPLATE\_ID "TMPL6BLUdPPg3"  #define BLYNK\_TEMPLATE\_NAME "Control LEDs"  #define BLYNK\_AUTH\_TOKEN "4bUNTi5EKwLjf5FF\_VRIWEwCOM-zi2B8"  char auth[] = BLYNK\_AUTH\_TOKEN;  char ssid[] = "FPTU\_Student";//Enter your WIFI name  char pass[] = "12345678";//Enter your WIFI password  //Get the button value  BLYNK\_WRITE(V0) {    digitalWrite(D0, param.asInt());  }  BLYNK\_WRITE(V2) {    digitalWrite(D2, param.asInt());  }  BLYNK\_WRITE(V4) {    digitalWrite(D4, param.asInt());  }  void setup() {    Serial.begin(115200);    //Set the LED pin as an output pin    pinMode(D0, OUTPUT);    pinMode(D2, OUTPUT);    pinMode(D4, OUTPUT);    //Initialize the Blynk library    Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);  }  void loop() {    //Run the Blynk library    Blynk.run();  } |

**V. DEMONSTRATING RESULTS**

* **Tóm tắt**

- Để nhận thông tin từ LoRa gửi sang LoRa nhận. Chúng ta cần lấy thông tin nhiệt độ môi trường đo được từ thiết bị cảm biến nhiệt độ LM35.

- Upload Code LoRa to Arduino ở trên để chúng ta có thể theo dõi được nhiệt độ mà LM35 đo được trong Serial Monitor.

- Upload Code LoRa to ESP8266 tại một sketch khác để kết nối được với Wifi. Lưu ý: Đổi tên wifi và Mật khẩu tại code này theo đúng wifi mà bạn đang sử dụng và sử dụng mã Token lấy từ Template trên ứng dụng Blynk.

- Ta có thể theo dõi nhiệt độ mà LoRa nhận nhận được tại Serial Monitor của code LoRa to ESP8266 hoặc ngay trên website Blynk.

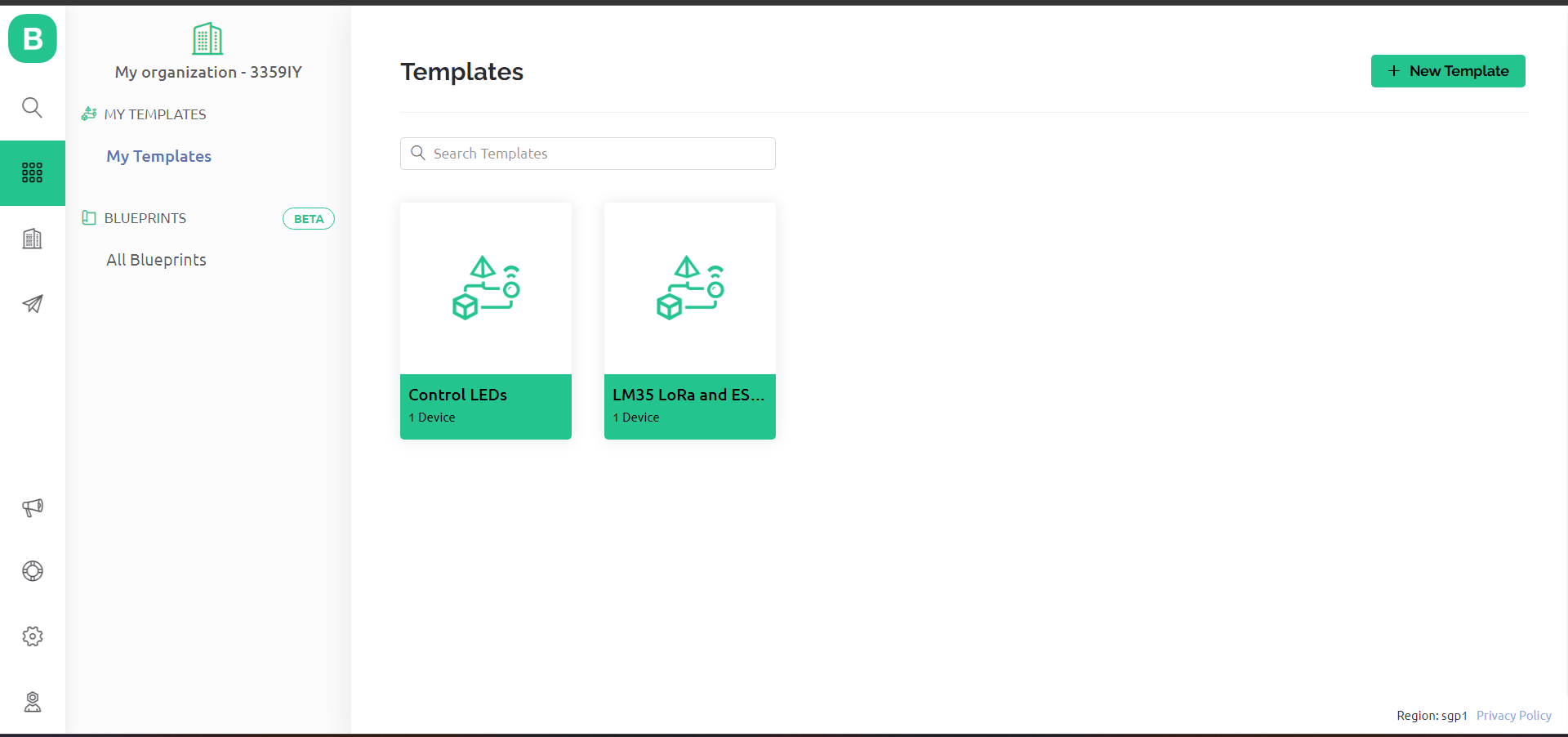
- Upload Code Control 3 LEDs rồi lên Blynk để có thể bật tắt đèn.

* **Hướng dẫn chi tiết về LoRa, ESP8266 và LM35**

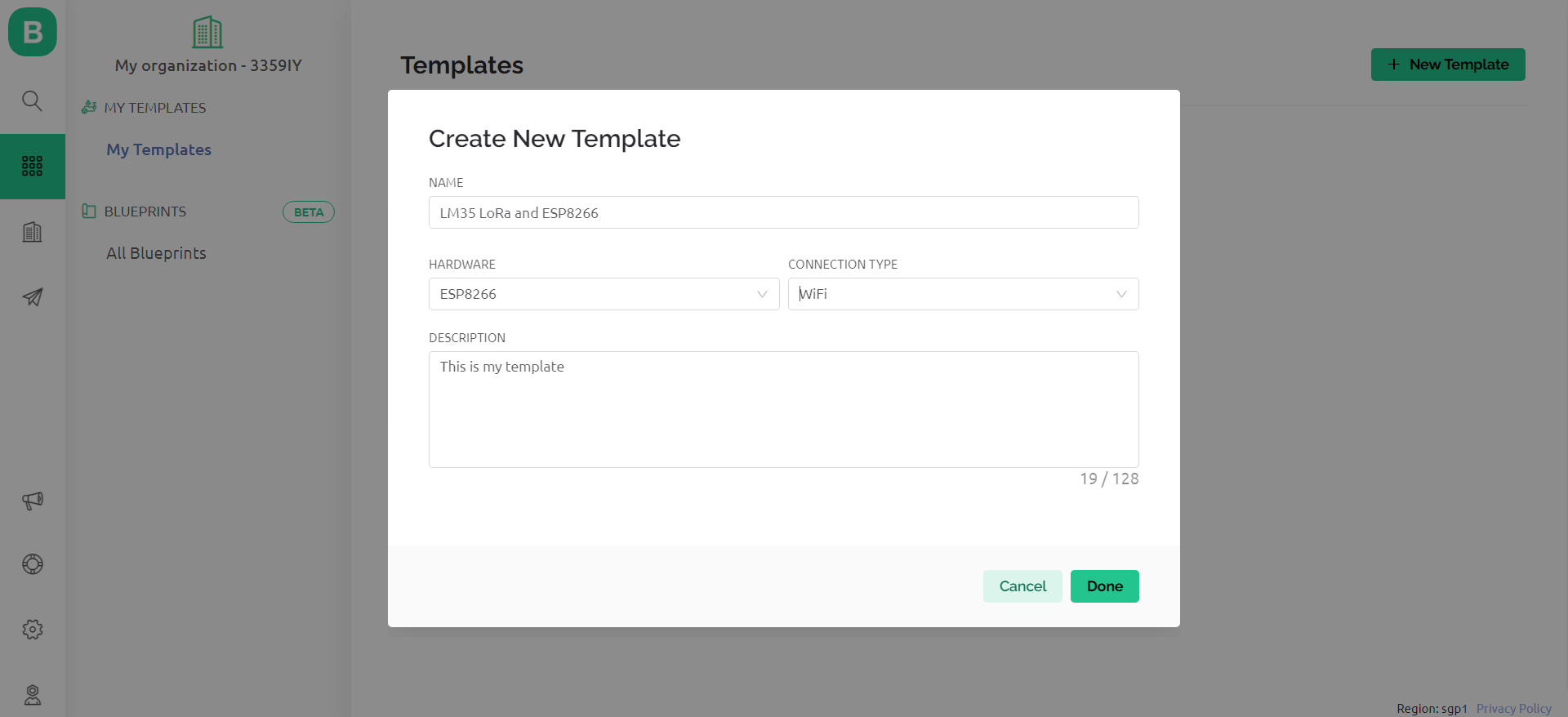
- Copy code LoRa (1) to Arduino vào sketch (1) và upload để gửi đi thông tin nhiệt độ được đo từ LM35.

- Mở App Blynk hoặc website <https://blynk.cloud/> và tạo một tài khoản cho riêng mình.

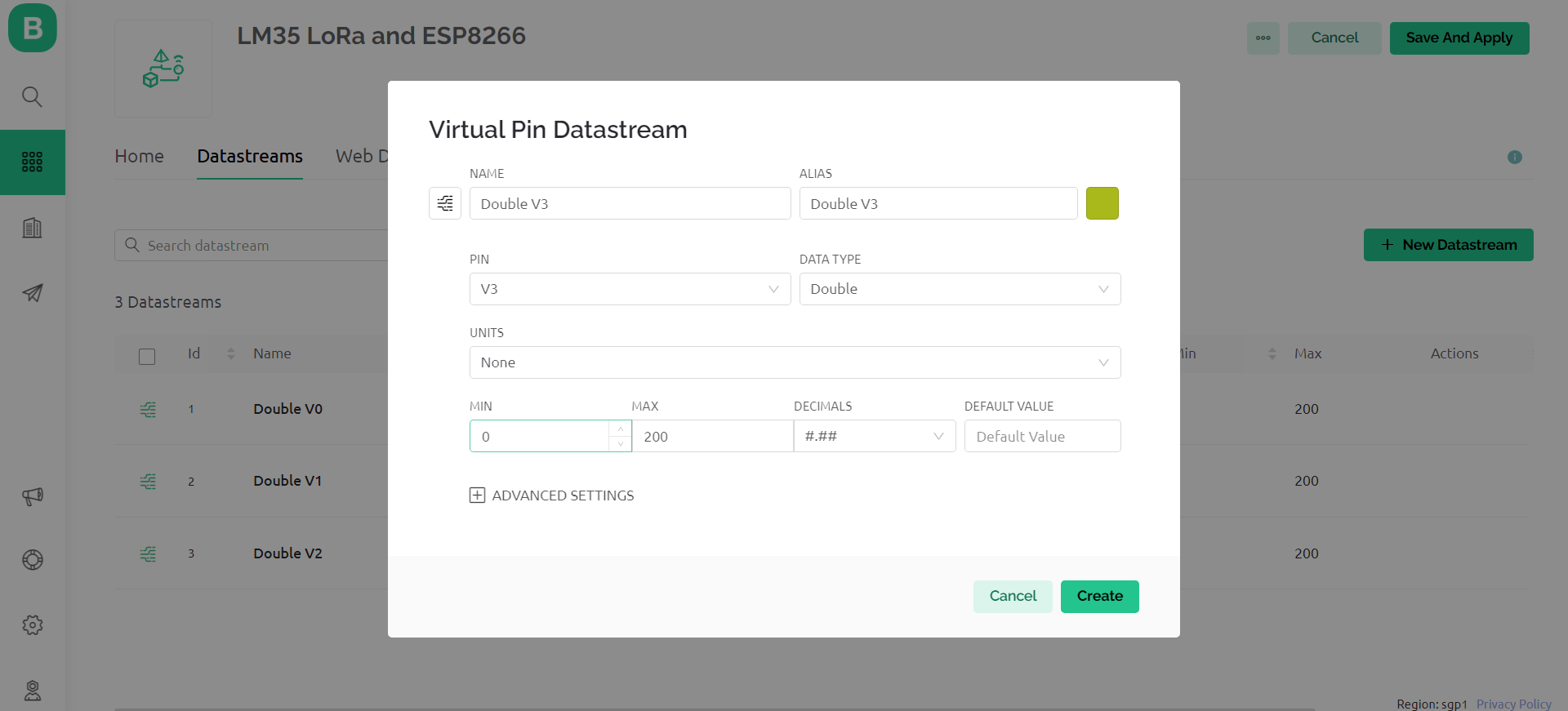
+ Tại trang Template bấm vào (+ New Template)



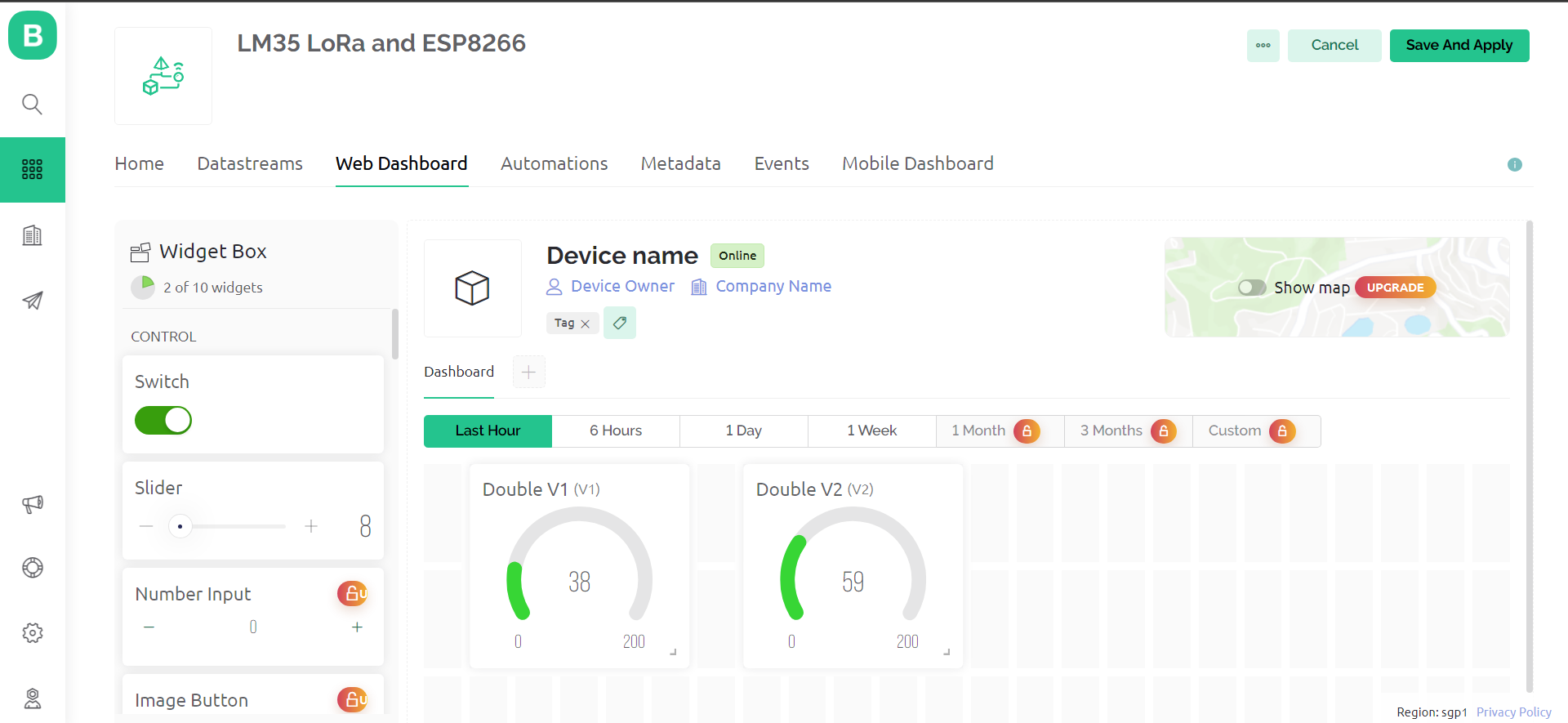
+ Sau đó điền thông tin của template (Lưu ý là chọn board ESP8266)



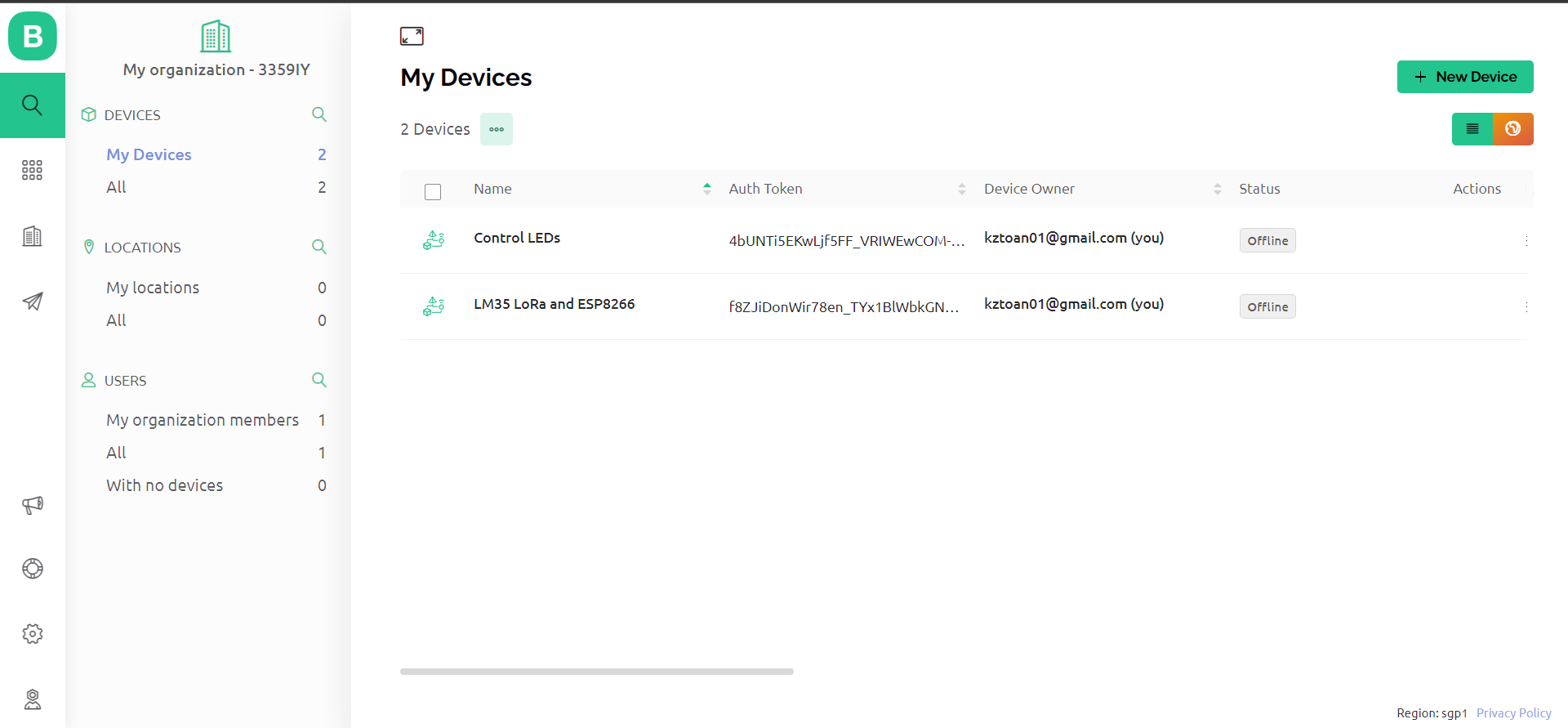
+ Tại tab DataStream ta tạo mới 2 Virtual DataStream mới và kiểu dữ liệu là double và MAX là 200.



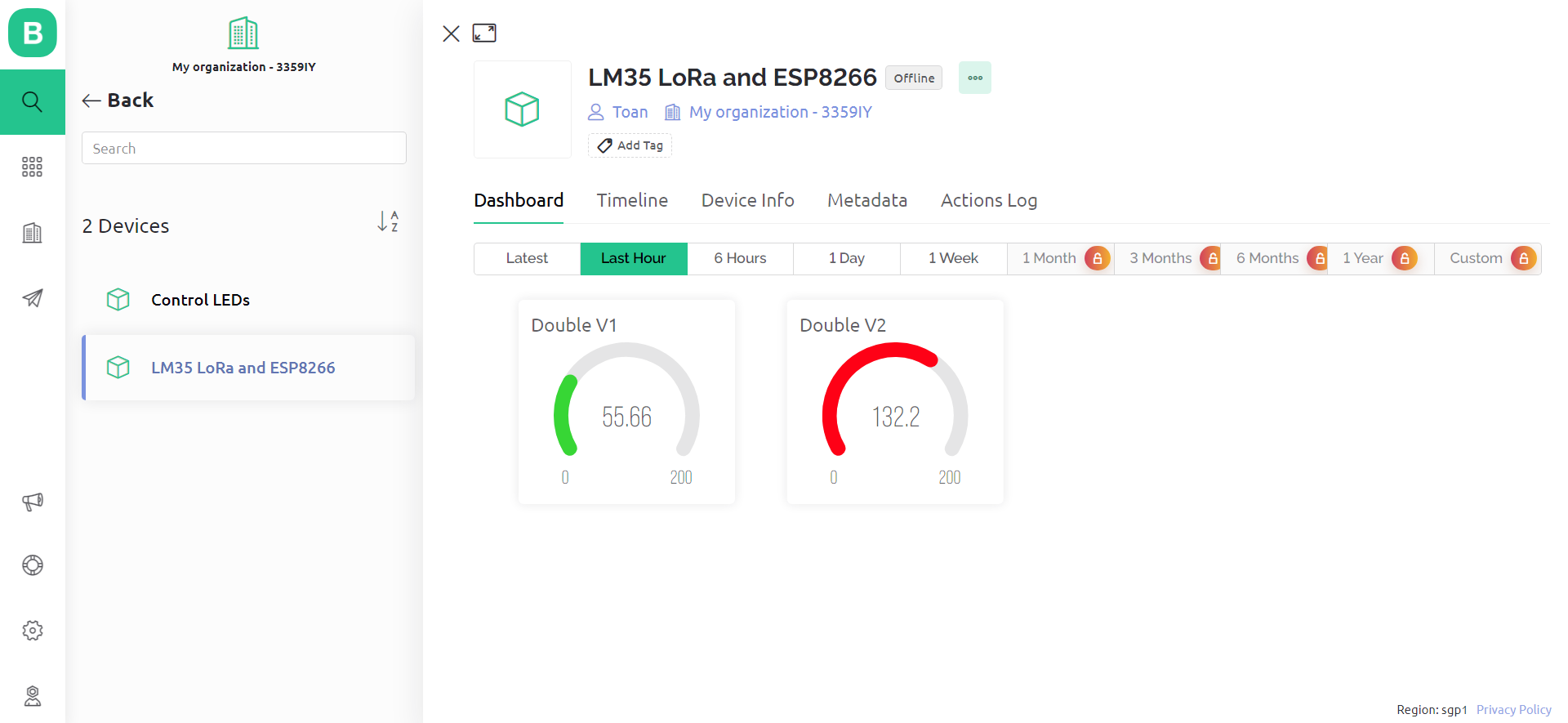
+ Sau khi tạo xong ta vào Web Dashboard để thêm các thiết bị theo dõi nhiệt độ là Gauge ứng với 2 DataStream đã tạo ở trên. Cuối cùng bấm vào Save and Apply ở góc trên bên phải màn hình.



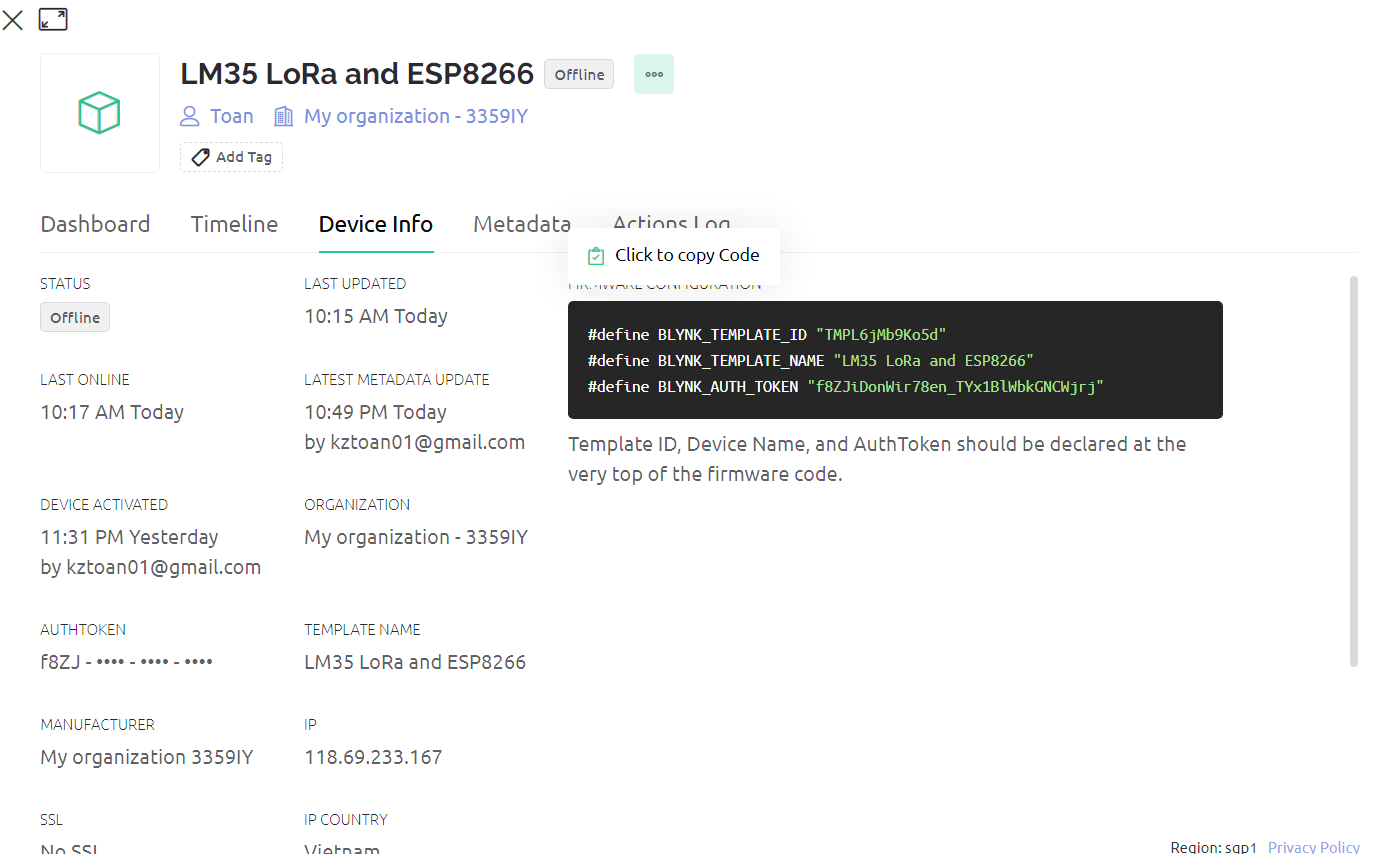
+ Ta trở lại trang Devices chính mà bấm thêm device để thêm Template mà ta vừa tạo ở trên.



+ Sau khi tạo xong ta có thể thấy 2 thiết bị Gauge mà ta đã thêm vào lúc trước ở đây.



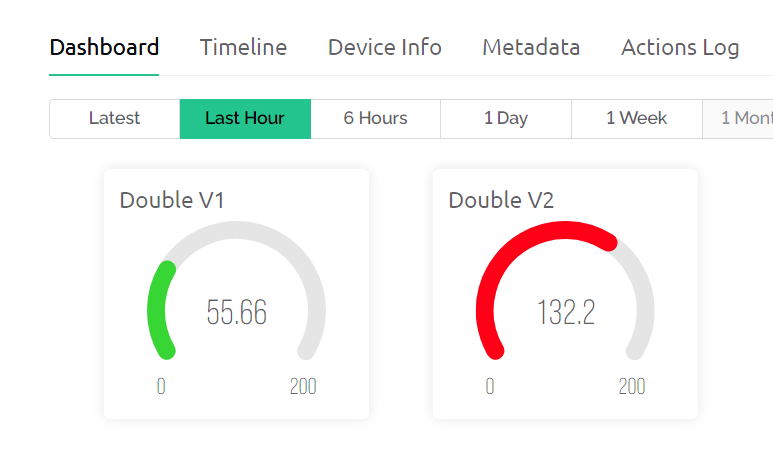
+ Bấm vào Device Info để lấy mã Token mà Blynk cung cấp để có thể kết nối được từ ESP8266 sang Blynk



+ Tại code LoRa (2) to ESP8266 chúng ta có thể sửa lại mã Token, tên Wifi và password khớp với dữ liệu chúng ta đang có.

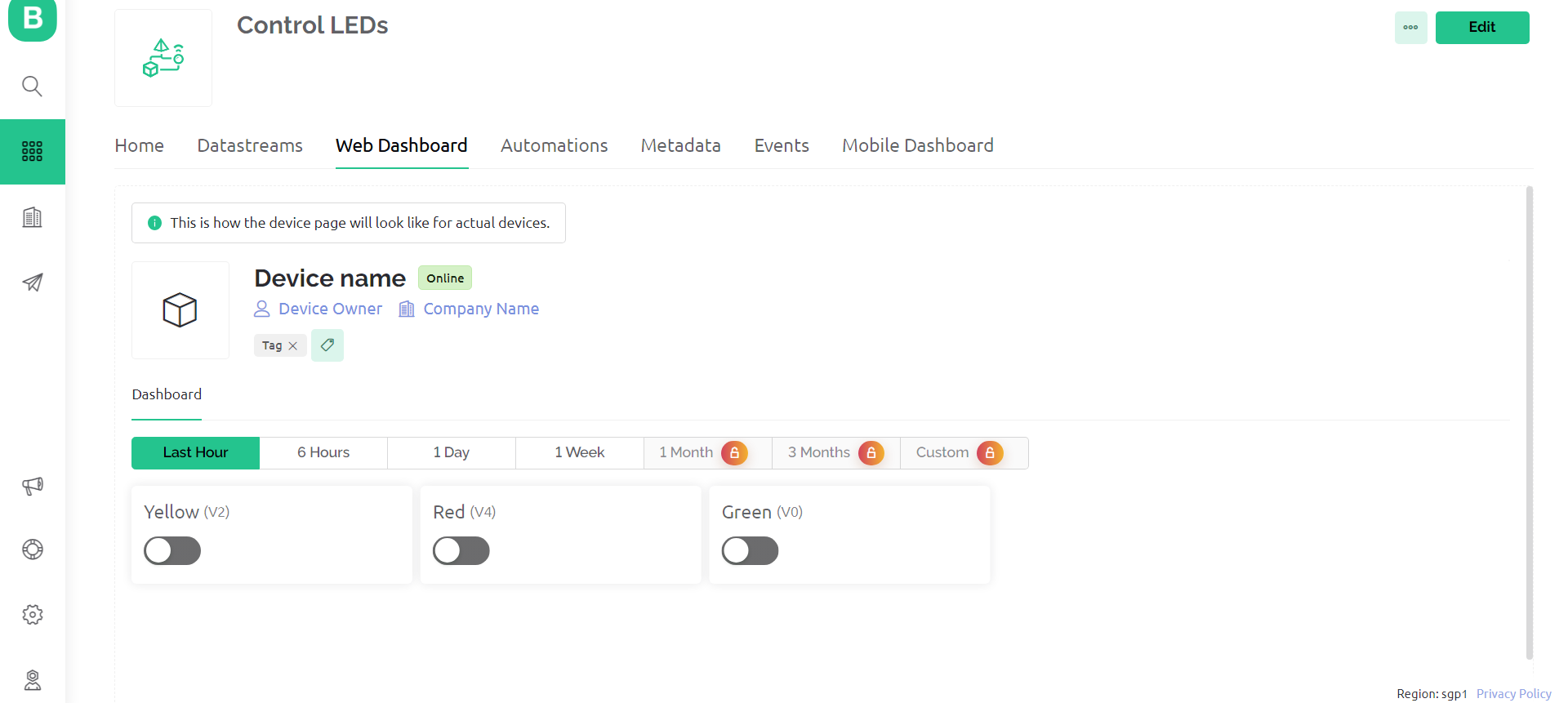
- Copy code LoRa (2) to ESP8266 vào sketch (2) và upload để nhận thông tin từ LoRa (1) về nhiệt độ mà LoRa (1) đo được.

- Sau đó mở Blynk lên và lúc này ta có thể thấy Status của thiết bị mà chúng ta vừa thêm đã Online và 2 cái Gauge cũng đã nhận được thông tin chính xác về 2 bảng nhiệt độ đó là độ C và độ F.

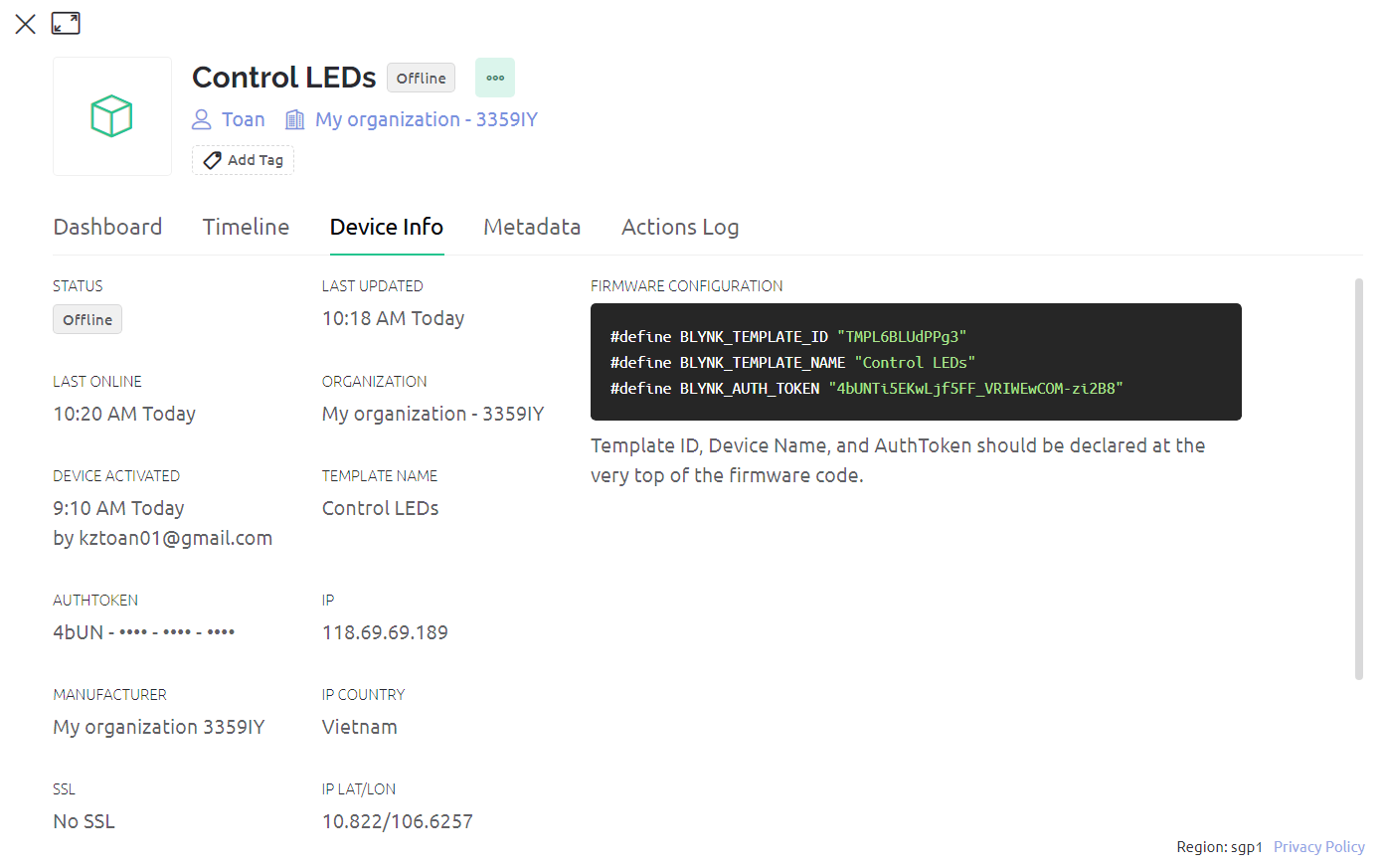


* **Hướng dẫn chi tiết về điều khiển 3 đèn LEDs nhờ Blynk**

- Tạo thêm 1 template mới giống như hưỡng dẫn ở trên để điều khiển 3 bóng đèn LEDs. Lưu ý chỗ Web Dashboard nên để 3 nút Switch để có thể dễ dàng điều khiển. Sau đó Save and Apply

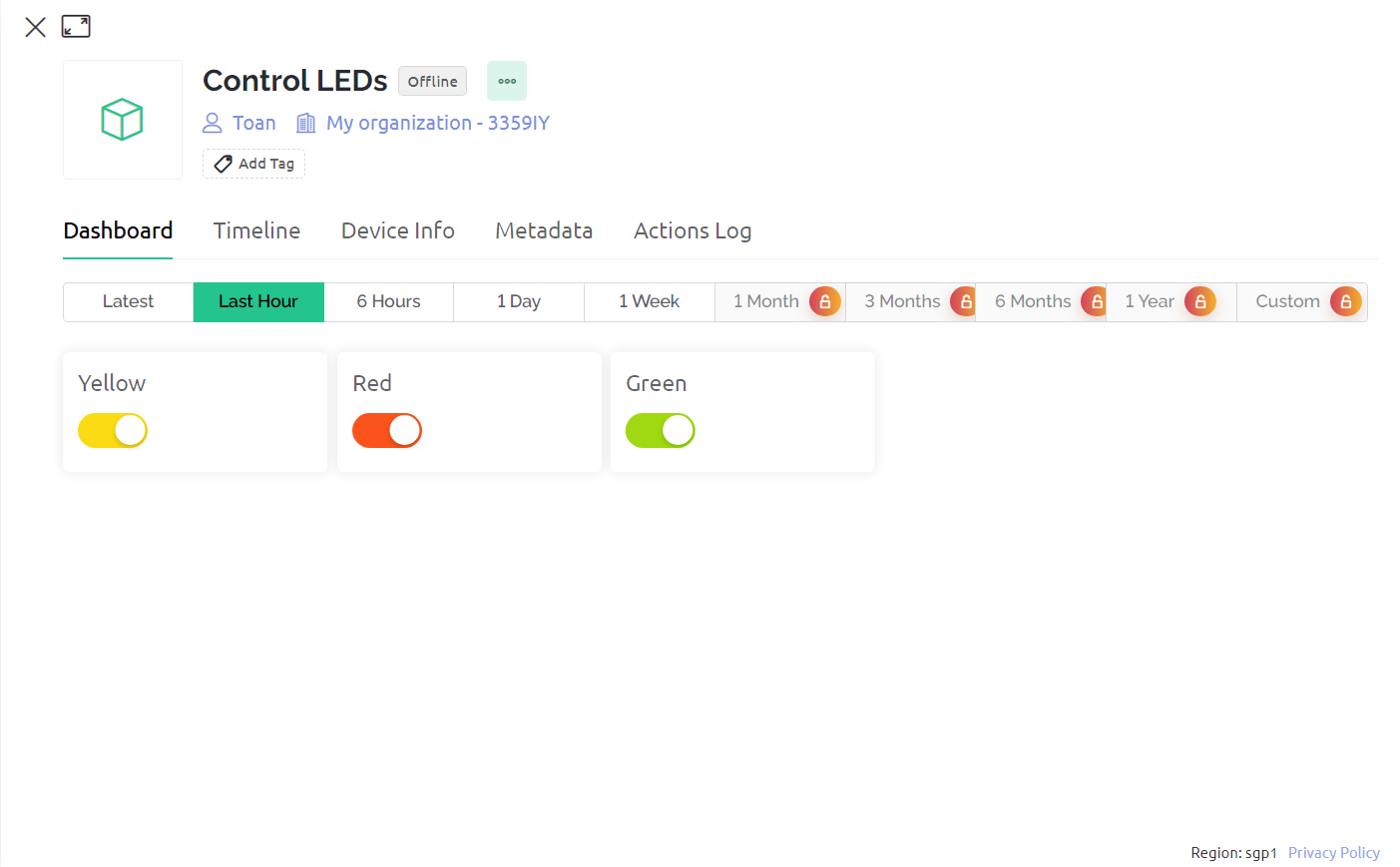


+ Tương tự với ở trên chúng ta tạo mới thiết bị và copy mã Token từ thiết bị này để dán vào code Control LEDs.



- Copy code Control LEDs vào sketch (3) vào bấm Upload.

- Lúc này trên website Blynk ở mục thiết bị Control LEDs chúng ta đã có thể điều khiển các bóng đèn bật tắt tùy ý ta muốn mọi lúc mọi nơi trên thế giới.



-----------------------------------***KẾT THÚC BÁO CÁO***--------------------------------