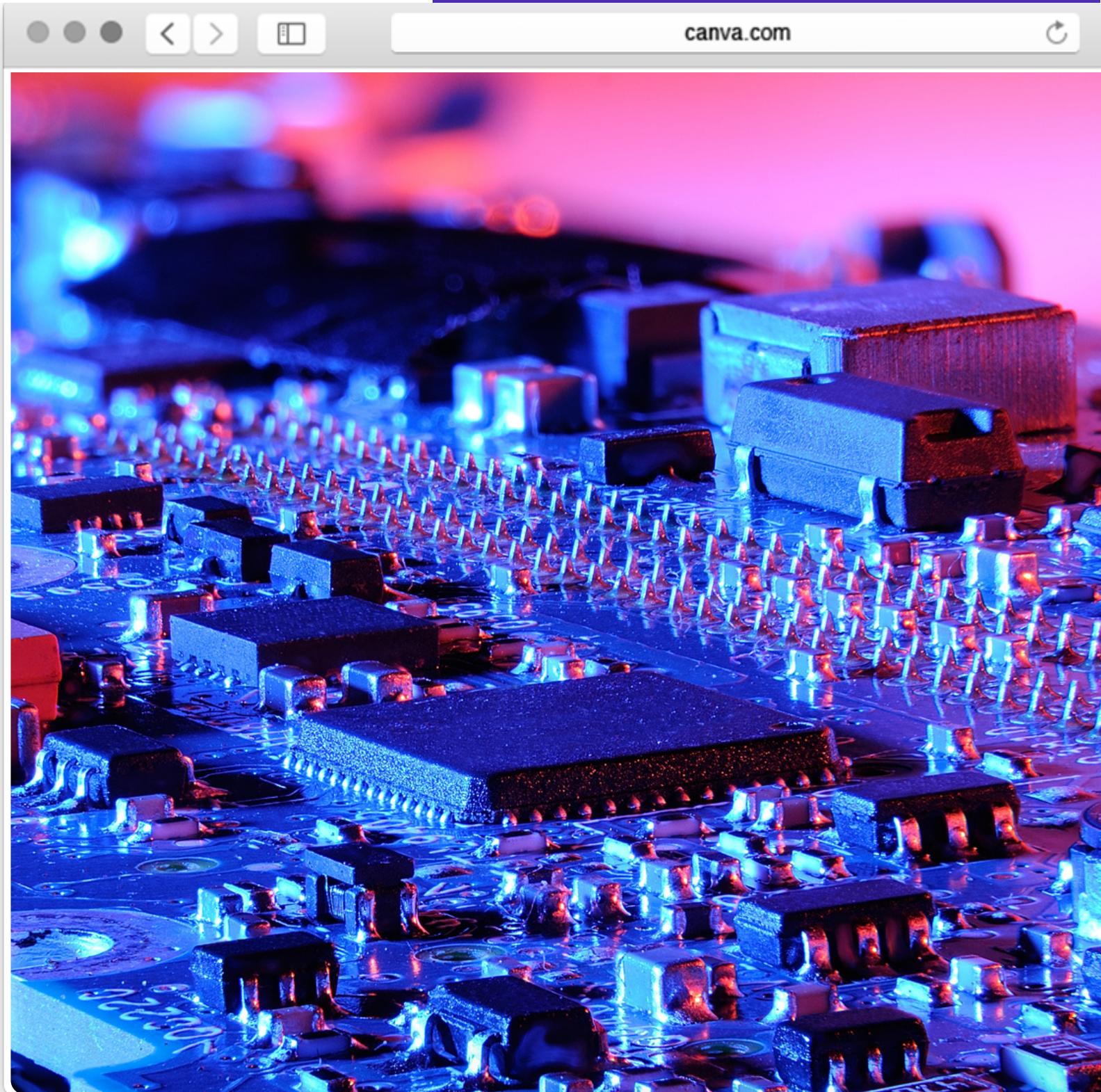


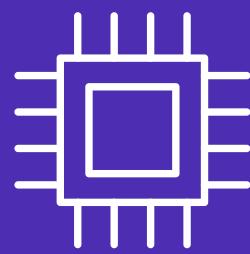
Microcontroller

Sensor & Measurement Final Project

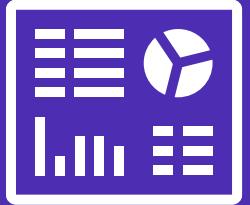
Dave Danadiva Agusta Peerera || BE-4 || 12006002



List of Contents



**Microntroller & Circuit
Connections**

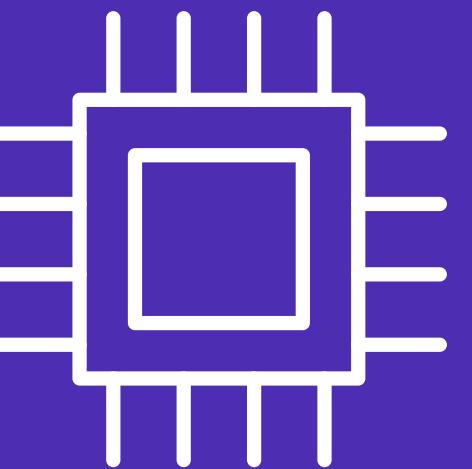
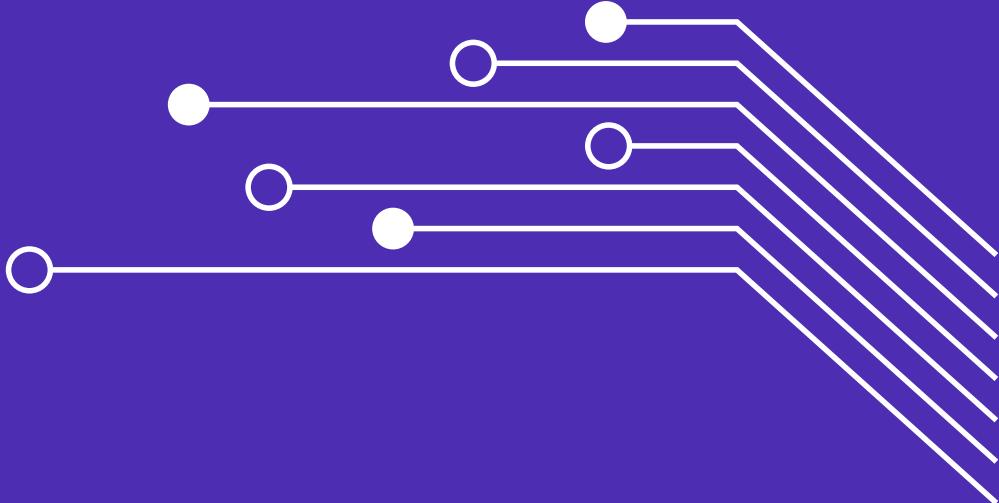


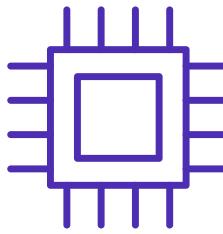
Dashboarding



Low Power Mode

Microntroller & Circuit Connections

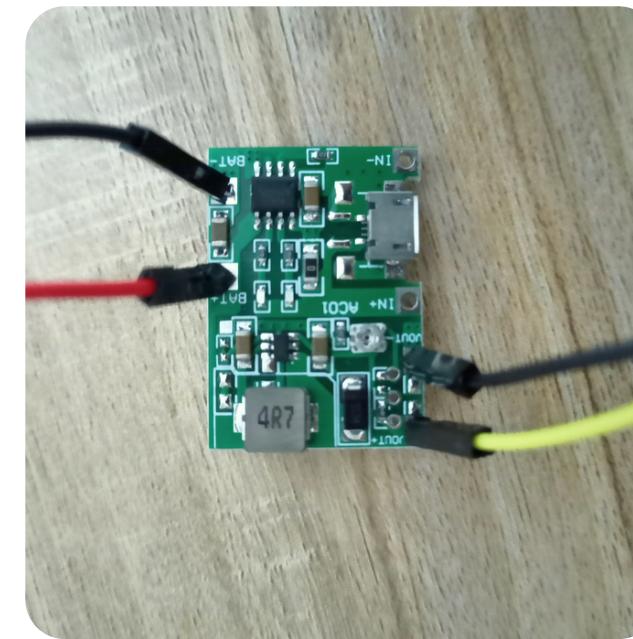




Materials



Lipo Battery 3.7 V
& Battery Holder



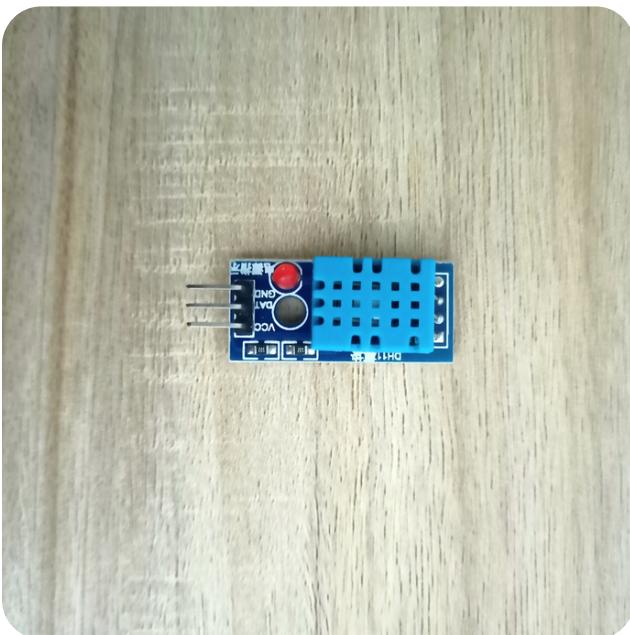
TP4056 With Step Up



ESP32 Dev Board



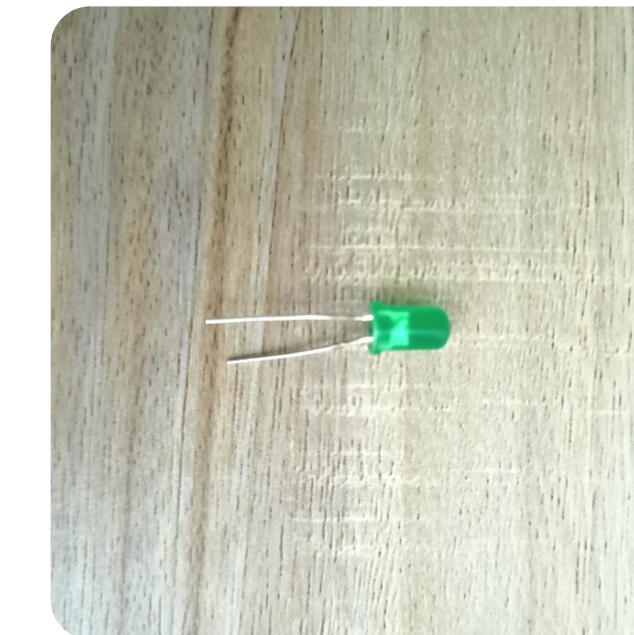
Jumper Wires



DHT 11 Sensor



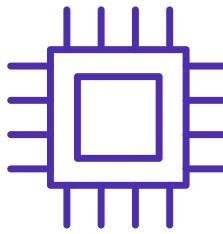
Breadboard



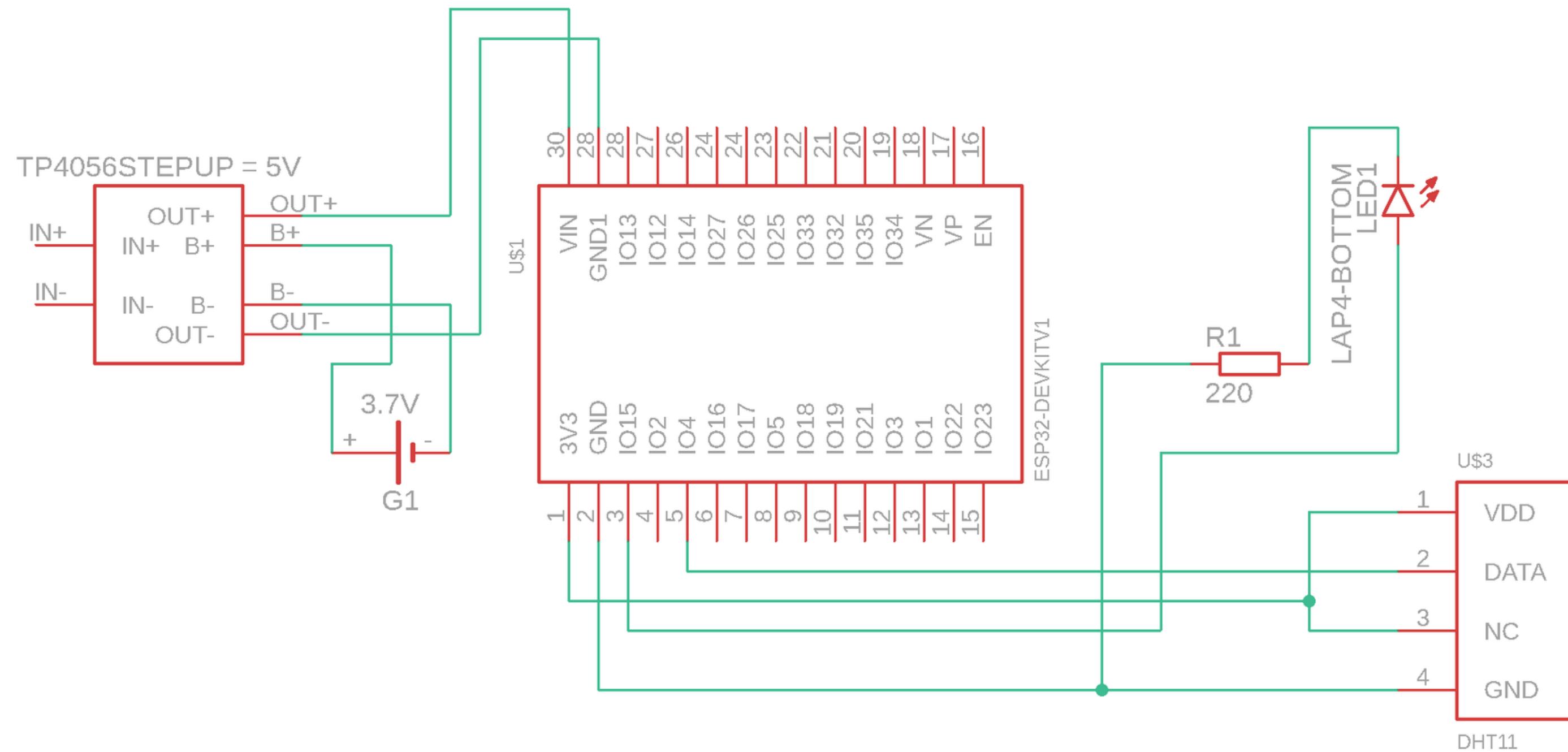
LED

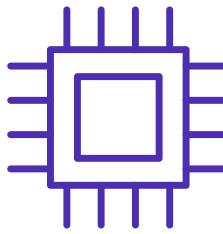


Resistor 220 Ohm

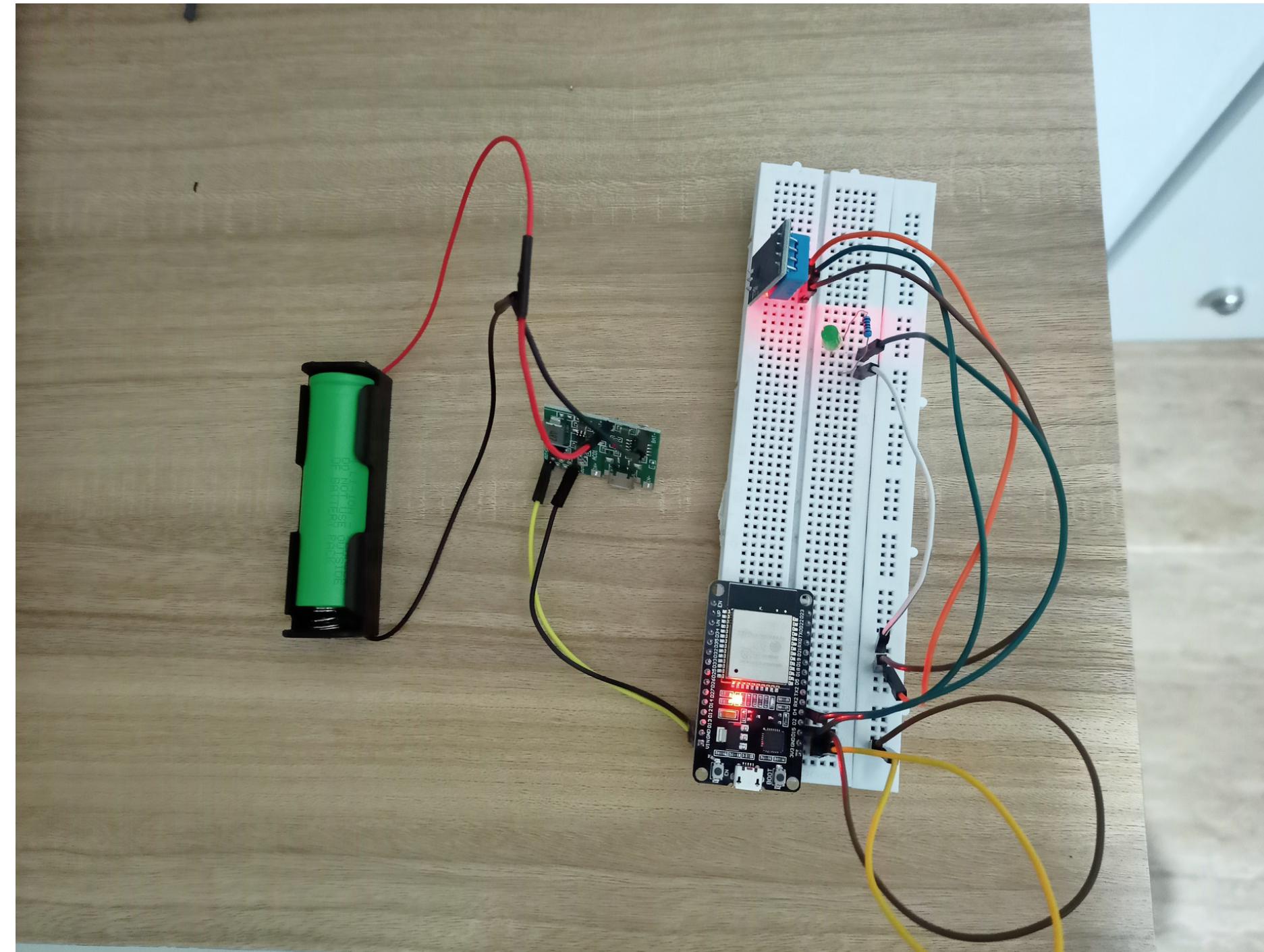


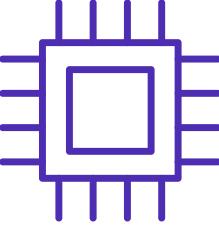
Circuit Schematics



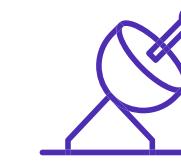


Circuit Connection Preview





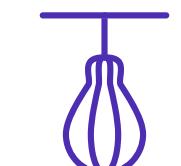
Problems with ESP01



ESP01 not found



Tried to update firmware but failed



**Can be detected by serial port but
cant send meesage to it**

COM5

```

`DCOP;`S9,????p[9]
/ \
/ / \
/ / / \
/ / \ , / / / \
/ / v1.1.0 on Arduino Uno

#StandWithUkraine https://bit.ly/swua

[521] Connecting to Rukita_Laterra_2B
[1532] ESP is not responding

```

Autoscroll Show timestamp Newline 115200 baud Clear output

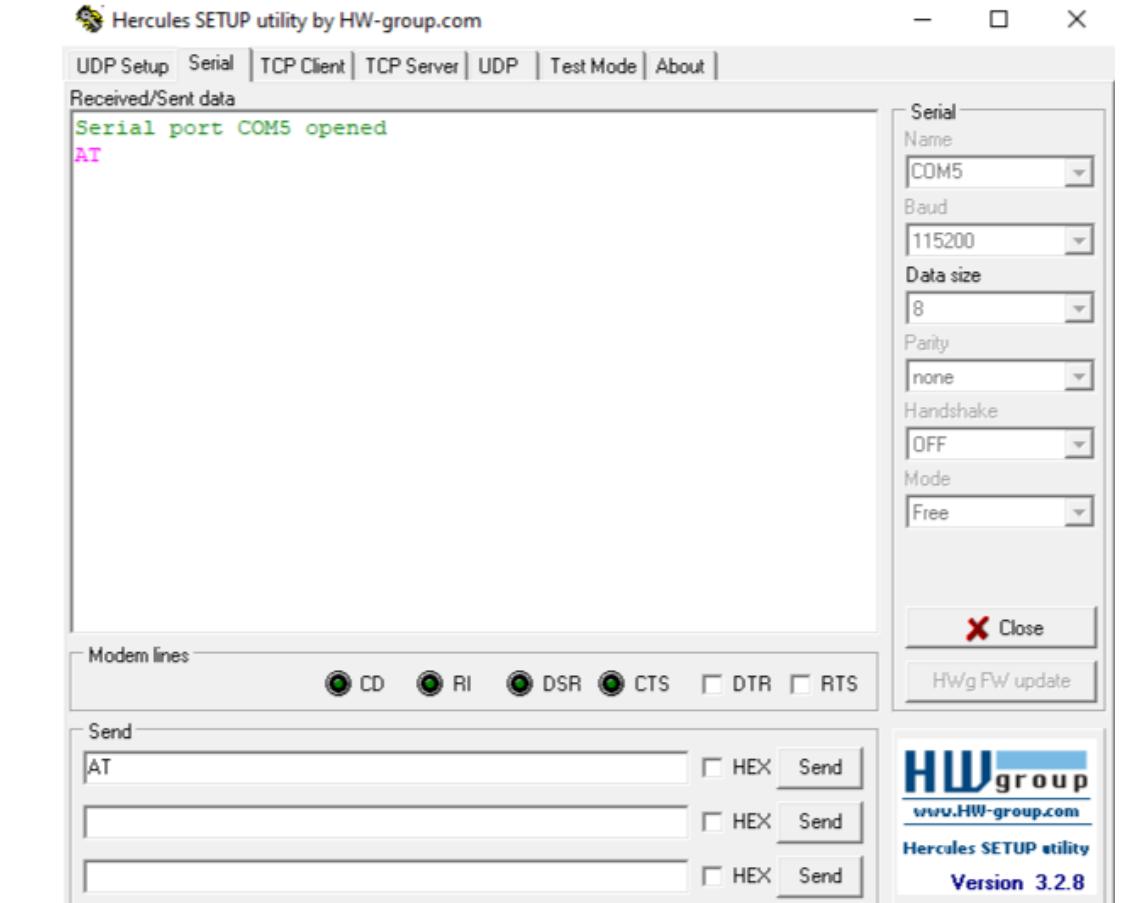
COM5

```

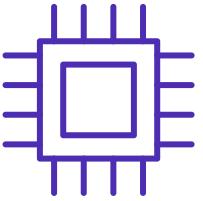
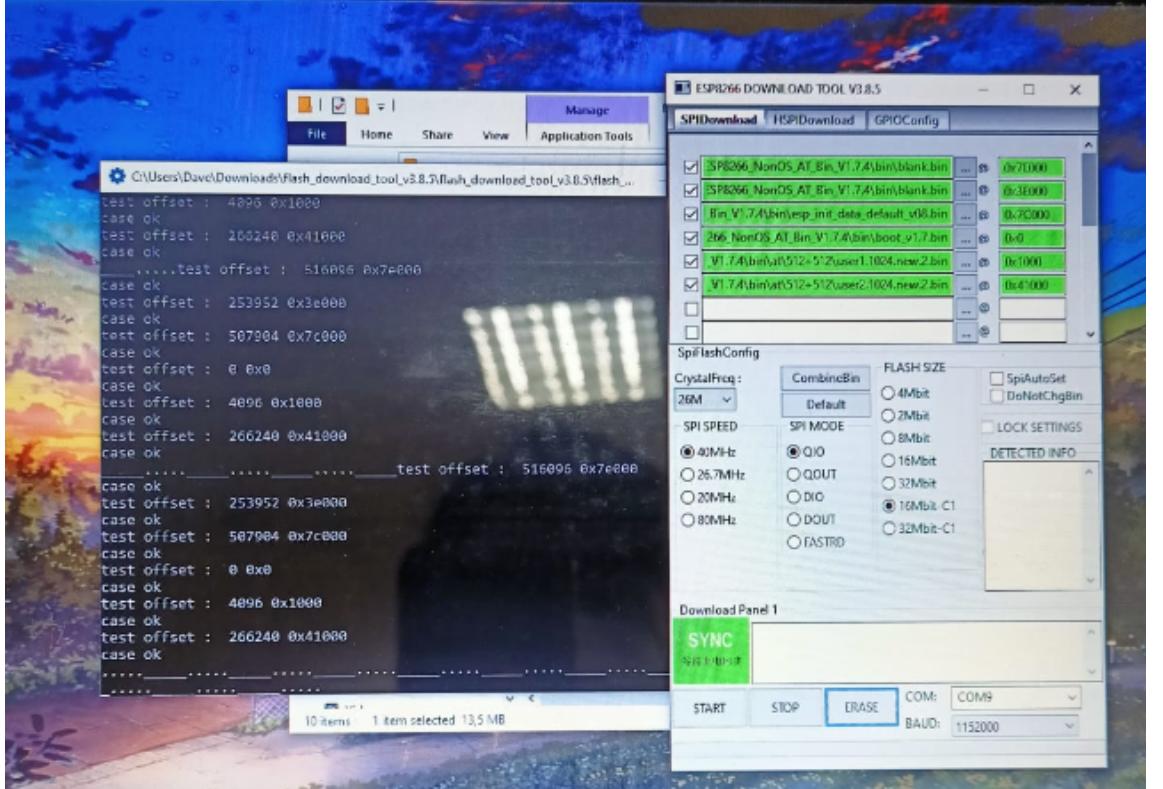
AT

```

Autoscroll Show timestamp Newline 115200 baud Clear output



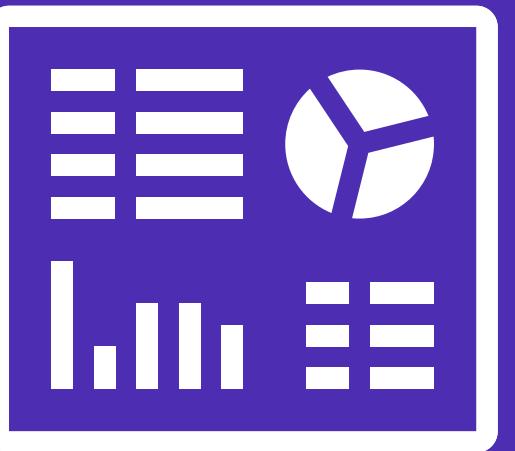
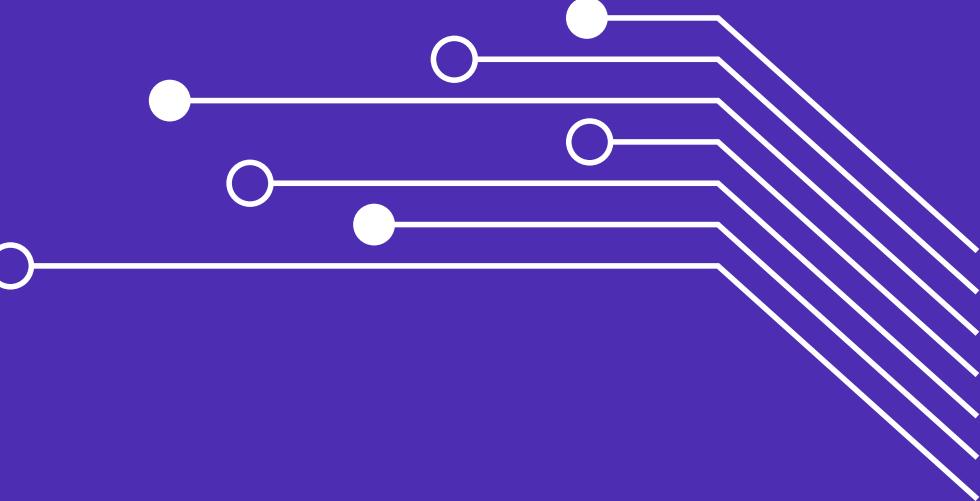
Error Screenshots

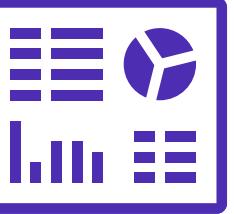
More Info:

<https://docs.google.com/document/d/1wKXw9ZIje-pyQeQzrR9S0YbLccKFPUa0T6bcKFAtw3U/edit?usp=sharing>

Dashboarding



Dashboard Preview



B X

IOT Device Offline

Dave My organization - 5373ZQ

Dashboard Timeline Device Info Metadata Actions Log

Latest Last Hour 6 Hours 1 Day 1 Week 1 Month 3 Months Custom

Humidity: 93 % Temperature (F): 84.2 °F Temperature (C): 29 °C

LED: OFF Uptime: 181 s Boot: 3

Average Temperature Report

Humidity Stats: Too Humid Temperature Stats: Temperate

Region: sgp1 [Privacy Policy](#)

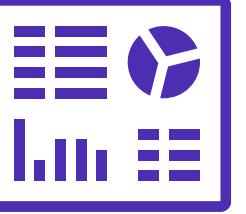
GAUGE

SWITCH

COUNTER

STRING

GRAPH



Gauge Settings

TITLE (OPTIONAL)

Datastream

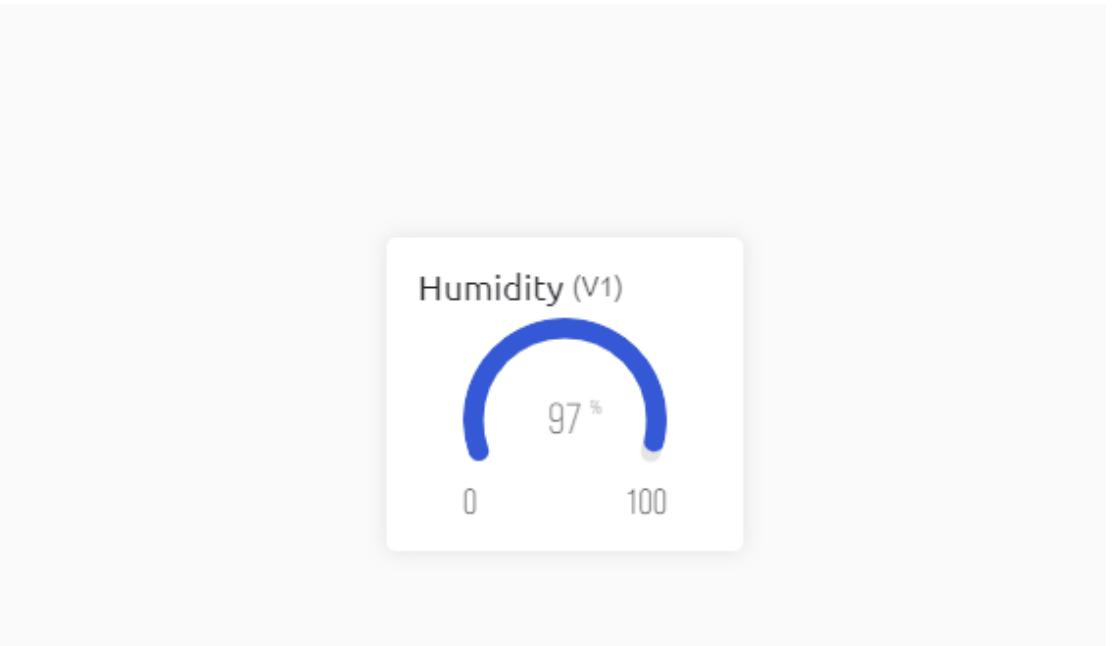
Humidity (V1) ▼ ✖

Override Datastream's Min/Max fields

LEVEL COLOR

Change color based on value

 ▼ ↶ ← ↶



Color Coding

Gauge Settings

TITLE (OPTIONAL)

Datastream

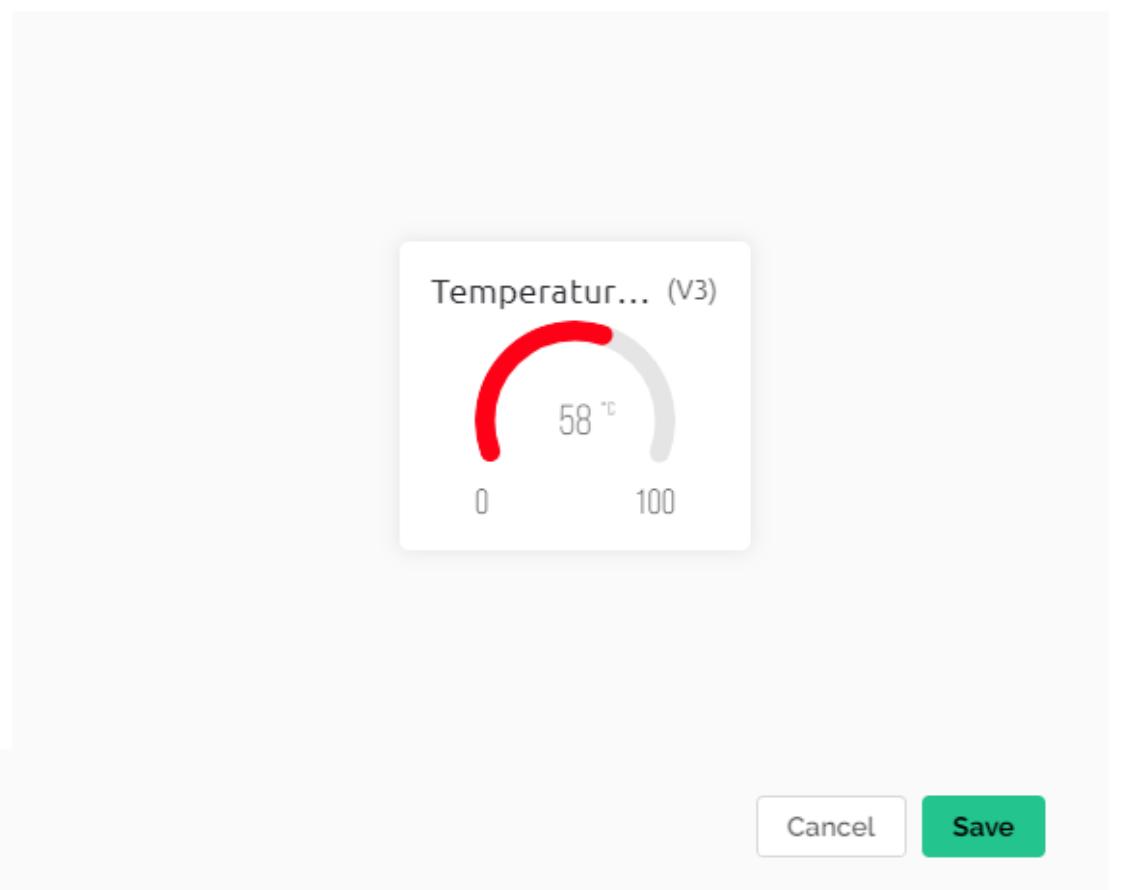
Temperature C (V3) ▼ ✖

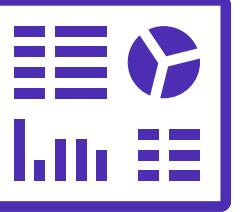
Override Datastream's Min/Max fields

LEVEL COLOR

Change color based on value

 ▼ ↶ → ↶



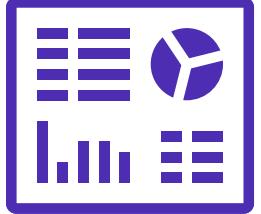


Code Parts

**Library &
Variable**

**Virtual Pin
Setting**

**Setup &
Loop**



Library & Variable

```
// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID "TMPLpKXTy9M0"
#define BLYNK_DEVICE_NAME "LED Switch"
#define BLYNK_AUTH_TOKEN "toISQm1I-u0Sgbvv10g2xmeV1rYwjqir"

// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

#include <DHT.h>

char auth[] = BLYNK_AUTH_TOKEN;

// WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Rukita_LaTerra_1B";
char pw[] = "hasslefree66";

// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
// See the Device Info tab, or Template settings
#define ledPin 15          // What digital pin is the LED
#define DHTPIN 4           // What digital pin is the DHT

// Type of DHT used
#define DHTTYPE DHT11      // DHT 11

DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer;

// Configuration for Deep Sleep
#define uS_TO_S_FACTOR 1000000ULL /* Conversion factor for micro seconds to seconds */
#define TIME_TO_SLEEP 60          /* Time ESP32 will go to sleep (in seconds) */

RTC_DATA_ATTR int bootCount = 0;
```

Virtual Pin Setting

```
// Control Virtual Pin
BLYNK_WRITE(V0)
{
    int pinValue = param.asInt(); // assigning incoming value from pin V0 to a variable

    // process received value

    if (pinValue > 0){
        digitalWrite(ledPin, HIGH);
    }
    else {
        digitalWrite(ledPin, LOW);
    }
}

void myTimerEvent()
{
    if (bootCount == 1){
        Blynk.virtualWrite(V7, millis() / 1000);
    }

    else {
        Blynk.virtualWrite(V7, (60 * (bootCount-1) + millis() / 1000));
    }
    Blynk.virtualWrite(V8, bootCount);
}
```

```
void sendSensor()
{
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    float f = dht.readTemperature(true);

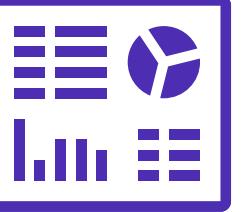
    if (isnan(h) || isnan(t) || isnan(f)) {
        Serial.println("Failed to read from DHT sensor!");
        return;
    }
    // You can send any value at any time.
    // Please don't send more than 10 values per second.
    Blynk.virtualWrite(V1, h);
    Blynk.virtualWrite(V3, t);
    Blynk.virtualWrite(V2, f);

    //----- Binning Humidity-----
    if (h < 40){
        Blynk.virtualWrite(V4, "Too Dry");
    }
    else if (h > 60) {
        Blynk.virtualWrite(V4, "Too Humid");
    }
    else {
        Blynk.virtualWrite(V4, "Comfort Zone");
    }

    //----- Binning Temp-----
    if (t < 25 ){
        Blynk.virtualWrite(V5, "Cold Temperature");
    }

    else if (t >= 25 & t <= 35){
        Blynk.virtualWrite(V5, "Temperate");
    }

    else {
        Blynk.virtualWrite(V5, "Hot Temperature");
    }
}
```



Setup & Loop

```
void setup()
{
    // Debug console
    Serial.begin(115200);
    pinMode(ledPin, OUTPUT);

    ++bootCount;

    Blynk.begin(auth, ssid, pw);

    dht.begin();

    // Condition in which esp will on and sleep
    while(millis()/1000 < 60){

        // Setup a function to be called every second
        timer.setInterval(2000L, sendSensor);
        timer.setInterval(1000L, myTimerEvent);

        Blynk.run();
        timer.run();

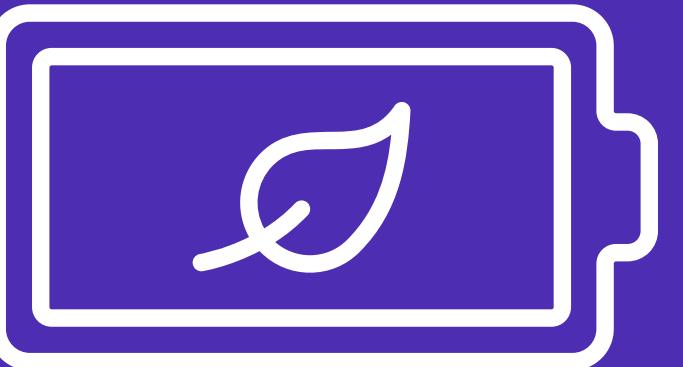
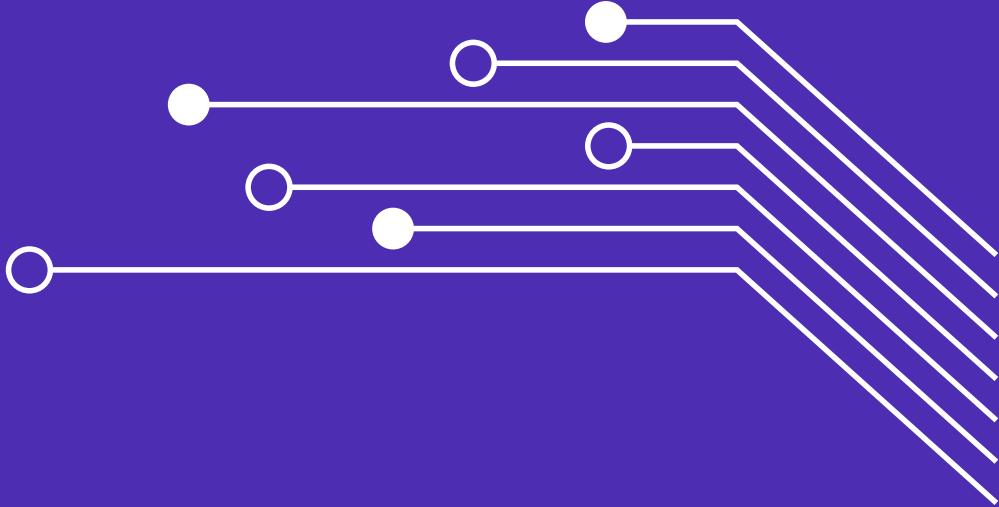
    }

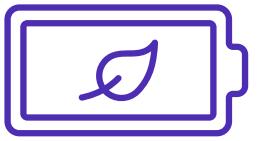
    // Start Deep Sleep
    esp_sleep_enable_timer_wakeup(TIME_TO_SLEEP * uS_TO_S_FACTOR);
    esp_deep_sleep_start();

}
```

```
void loop()
{
    //No Code
}
```

Low Power Mode





Different Types of Low Power Mode

Power Mode	Active	Modern Sleep	Light Sleep	Deep Sleep	Hibernation
CPU	ON	ON	PAUSE	OFF	OFF
Wi-Fi/BT baseband and radio	ON	OFF	OFF	OFF	OFF
RTC memory and RTC peripherals	ON	ON	ON	ON	OFF
ULP co- processor	ON	ON	ON	ON/OFF	OFF



Deep Sleep Code

01

Library & Variable

```
// Configuration for Deep Sleep
#define uS_TO_S_FACTOR 1000000ULL
#define TIME_TO_SLEEP 60

RTC_DATA_ATTR int bootCount = 0;
```

02

Virtual Pin Setting

```
void myTimerEvent()
{
    if (bootCount == 1){
        Blynk.virtualWrite(V7, millis() / 1000);
    }

    else {
        Blynk.virtualWrite(V7, (60 * (bootCount-1) + millis() / 1000));
    }
    Blynk.virtualWrite(V8, bootCount);
}
```

03

Setup & Loop

```
++bootCount;

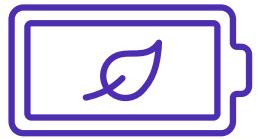
// Condition in which esp will on and sleep
while(millis()/1000 < 60){

    // Setup a function to be called every second
    timer.setInterval(2000L, sendSensor);
    timer.setInterval(1000L, myTimerEvent);

    Blynk.run();
    timer.run();

}

// Start Deep Sleep
esp_sleep_enable_timer_wakeup(TIME_TO_SLEEP * uS_TO_S_FACTOR);
esp_deep_sleep_start();
```



Power Analysis

01

Wake UP Condition

$$V = 5 \text{ V}$$

$$I = 145.5 \text{ mA}$$

$$P = 5 * 145.5 \text{ m} = 0.728 \text{ Watt}$$



02

Deep Sleep

$$V = 5 \text{ v}$$

$$I = 11.45 \text{ mA}$$

$$P = 5 * 11.45 \text{ m} = 0.057 \text{ Watt}$$



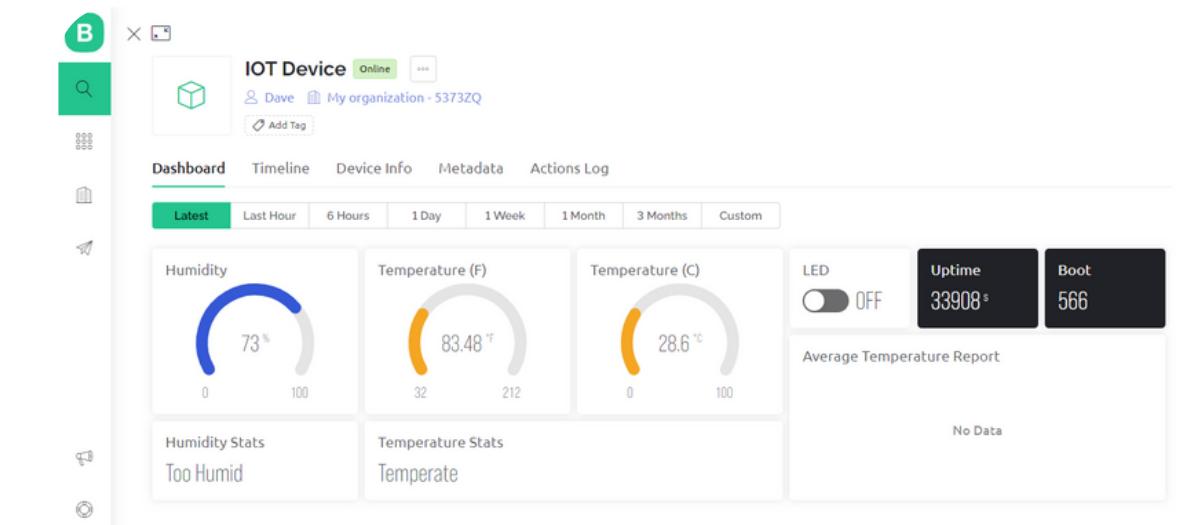
03

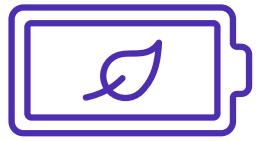
Battery Life

566 Boot

33908 seconds uptime

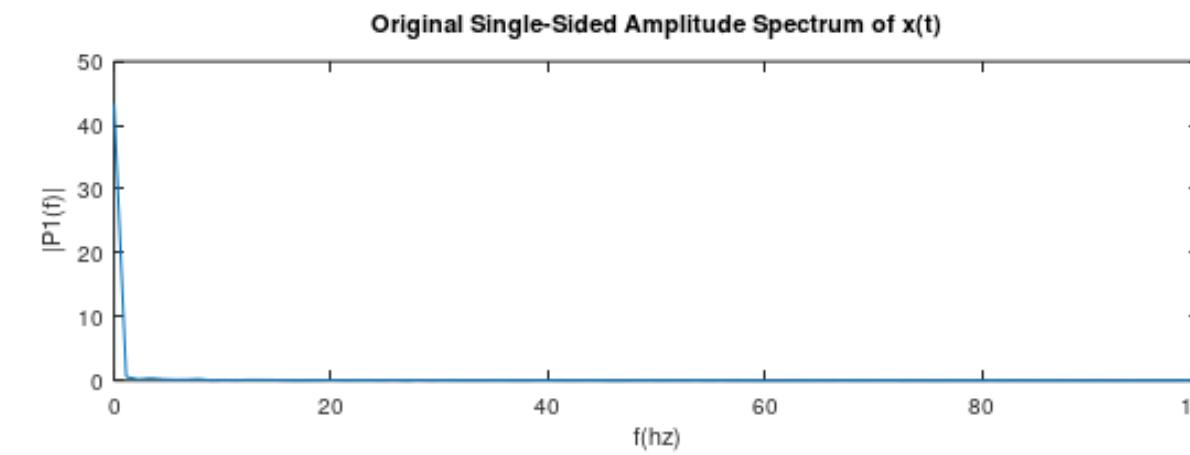
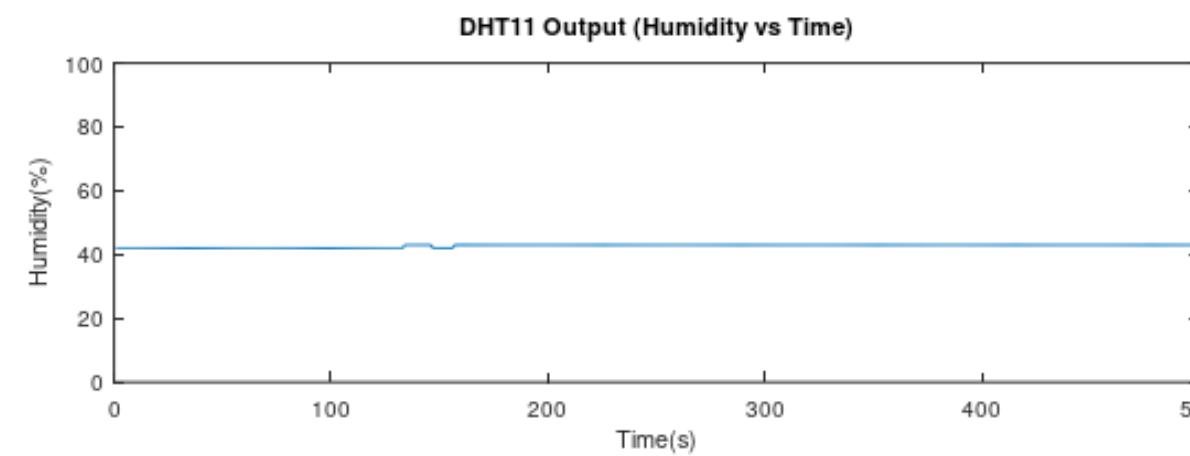
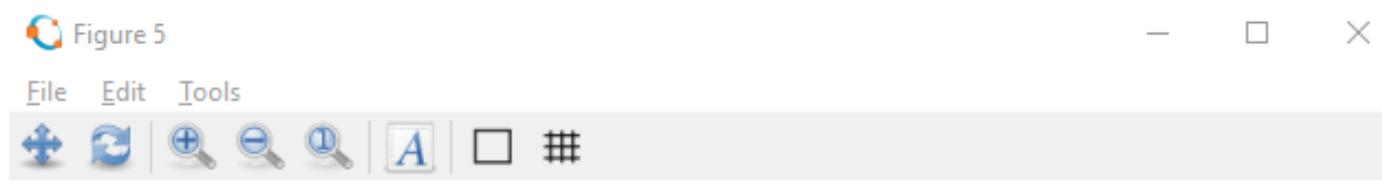
or 9 hours & 41 minutes uptime



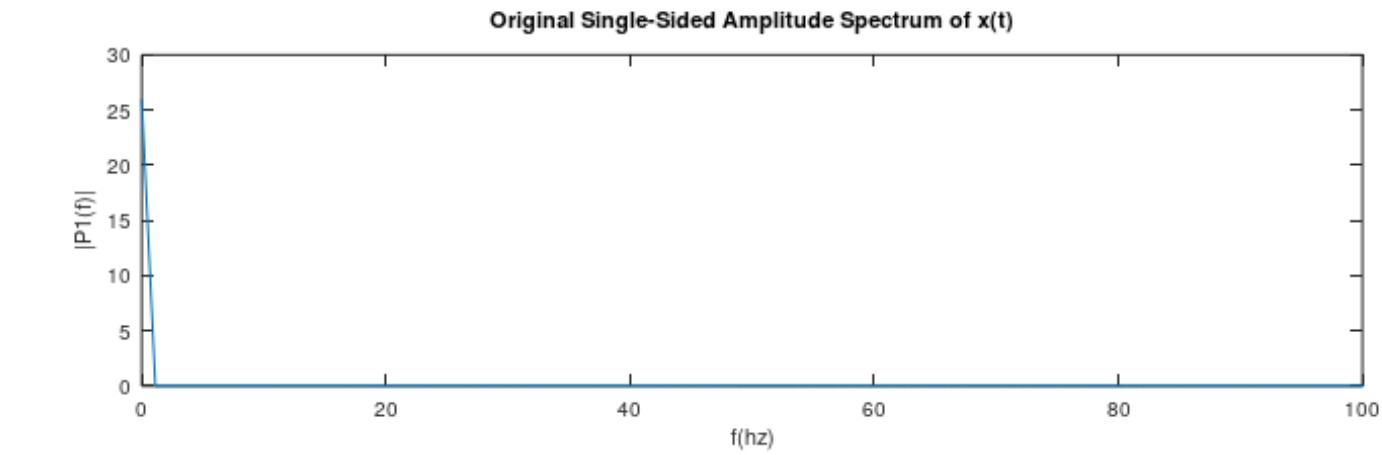
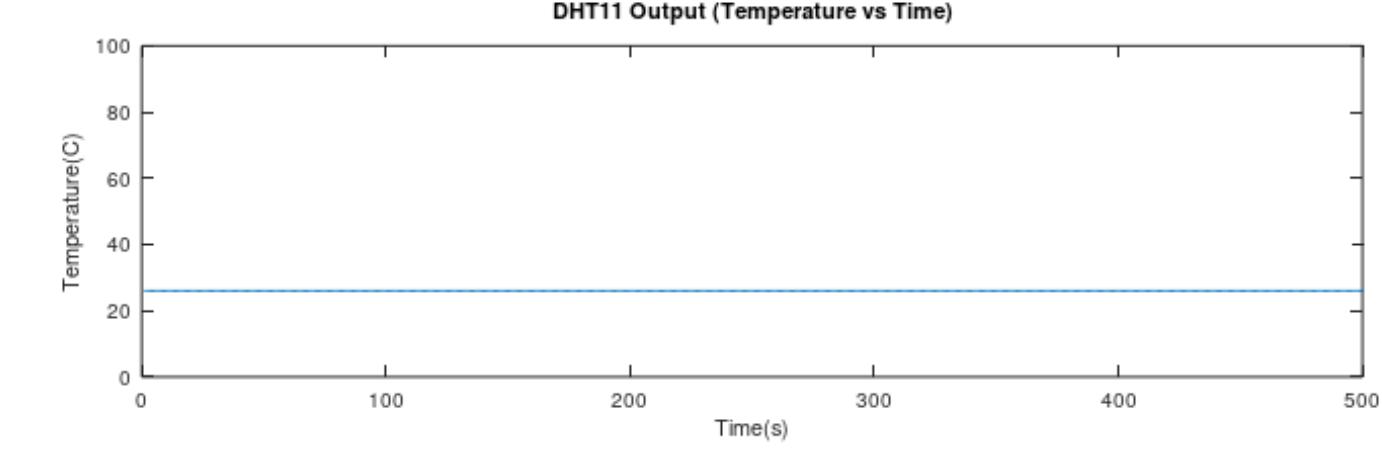
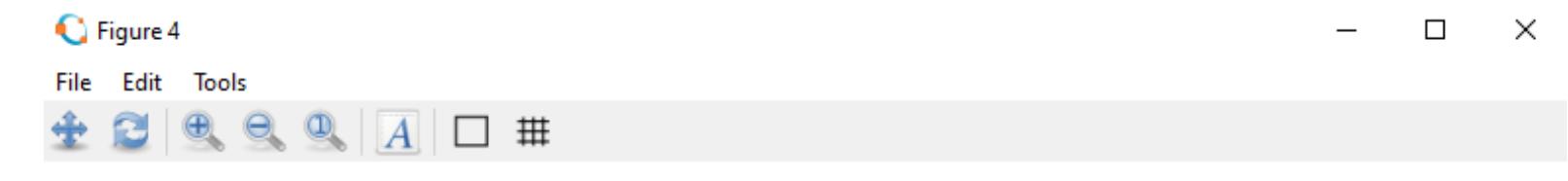


Fast Fourier Transform

Humidity

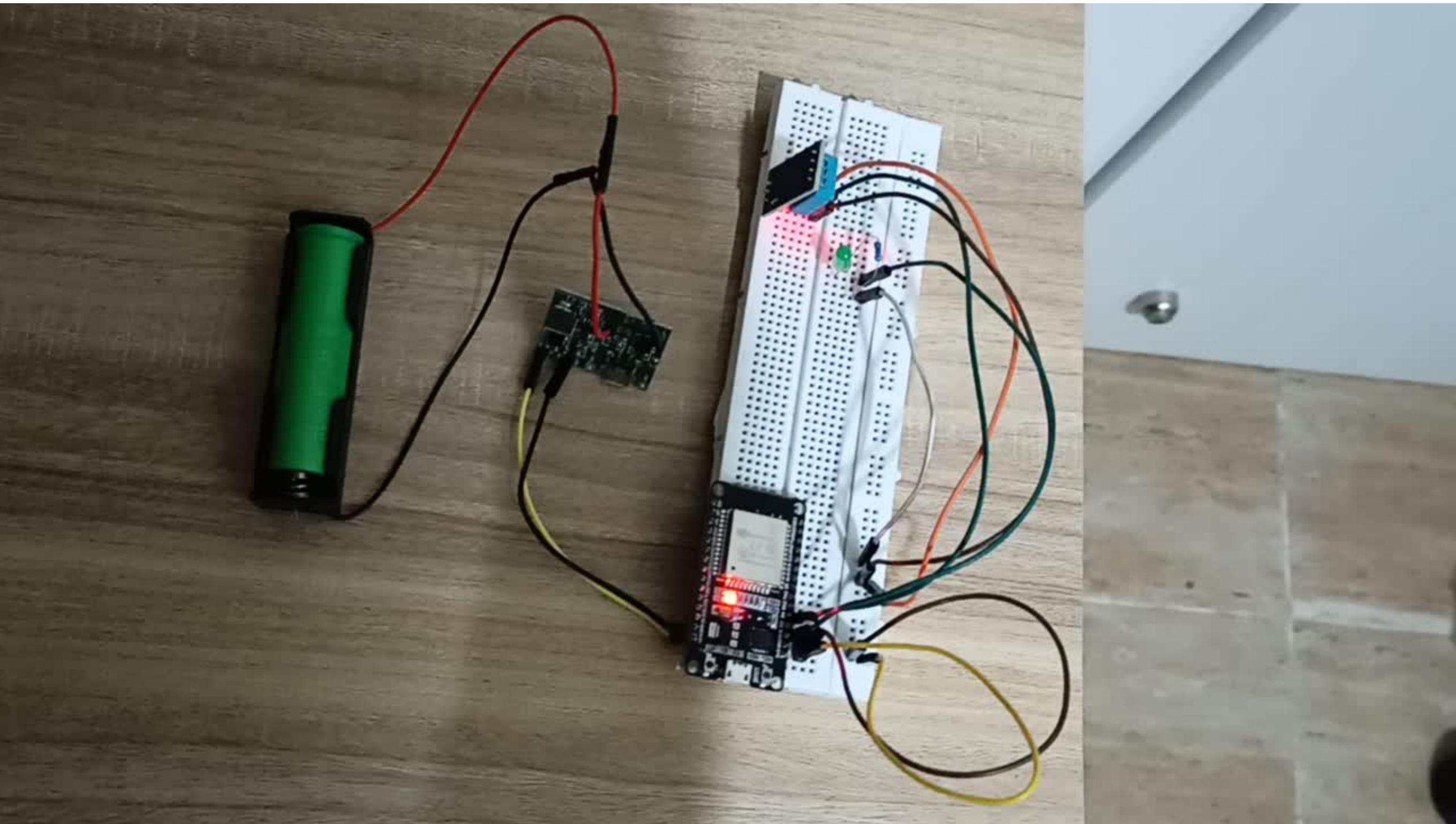
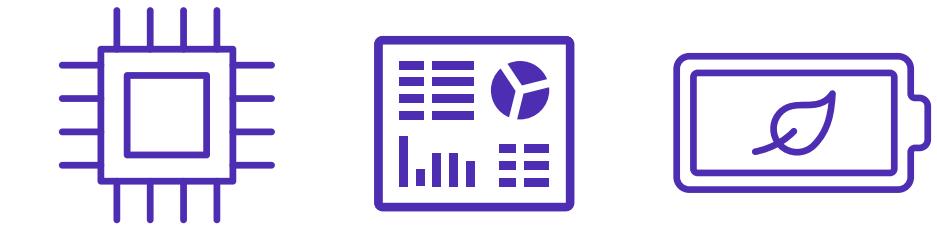


Temperature

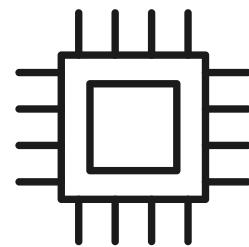


(10.923, 28.967)

Demo Video

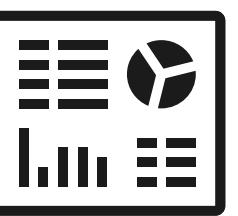


Conclusion



Microcontroller & Circuit Connection

For this project, ESP32 was used because the esp01 does not respond; hence the teensy board cannot be used. Several efforts have been made but still can't solve the problem.



Dashboarding

Dashboard was made using Blynk in which it connected with the device through wifi connection



Low Power Mode

ESP32 was programmed with deep sleep feature to save power in which it turns off wifi connection but still have the RTC memory turned on

Thank you

Contact:

dave.peerera@student.sgu.ac.id

