



Notifications

Your account is now verified. We are looking forward to your awesome projects. Visit our [Help Center](#) to learn how to get the most from Maker Pro.

3 months ago

Welcome! Thank you for joining Maker Pro. We are looking forward to your awesome projects. Visit our [Help Center](#) to learn how to get the most from Maker Pro.

3 months ago

Please check your email to verify your account.

3 months ago

[View All Notifications](#) [Close](#)

Mwangi Denis

...

[Start Project](#)

[Home](#)

Explore

Categories

- [3D Printing](#)
- [Amateur Radio](#)
- [Audio](#)
- [Augmented Reality](#)
- [Automation](#)
- [Automotive](#)
- [Cloud Computing](#)
- [Computers & Peripherals](#)
- [Consumer Electronics](#)
- [Cyber Security](#)
- [Displays](#)
- [Drones](#)
- [Health & Fitness](#)
- [Home Automation](#)
- [Industrial](#)
- [Industrial IoT](#)
- [IoT](#)
- [Lighting](#)
- [Machine Learning](#)
- [Mobile](#)
- [Motor Control](#)
- [Power](#)
- [Robotics](#)
- [Security / Identification](#)
- [Sensors](#)
- [Smart Grid/Energy](#)
- [Telecom](#)
- [Virtual Reality](#)
- [Wearables](#)
- [Weather](#)

[View All](#)

Platforms

[Linux](#)

[Raspberry Pi](#)

[Arduino](#)

[ESP8266](#)

[Custom](#)

[PIC](#)

[PCB](#)

[MicroPython](#)

[View All](#)[Projects](#) [Education](#) [Profile](#) [Profile Settings](#) [Logout](#) Close Menu[Start Project](#)[Import Project](#)

Notifications

Your account is now verified. We are looking forward to your awesome projects. Visit our [Help Center](#) to learn how to get the most from Maker Pro.

3 months ago

Welcome! Thank you for joining Maker Pro. We are looking forward to your awesome projects. Visit our [Help Center](#) to learn how to get the most from Maker Pro.

3 months ago

Please check your email to verify your account.

3 months ago

[View All Notifications](#) [Close](#)[Profile](#) [Settings](#) [Bookmarks](#) [Logout](#)

- [Home](#)
- [Blynk](#)
- [Projects](#)
- [How to Create an IoT Smart Garden Using ESP32 and Blynk](#)
- Blynk

How to Create an IoT Smart Garden Using ESP32 and Blynk

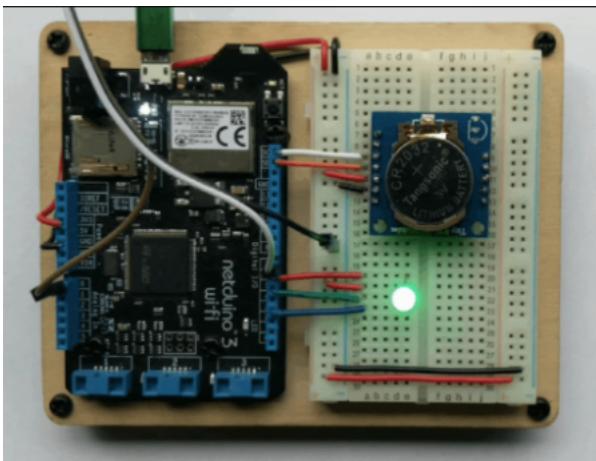
[Jinani Wijesinghe](#)

-
- 8
- 2
- 0
- 5793

May 19, 2020

In this tutorial, we are going to create an IoT-based garden you can care for from your phone using Blynk!

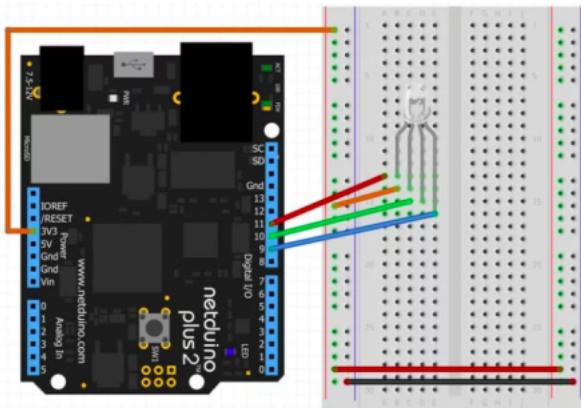
- 8
-
-
-
-
-
- [Fork Project](#)
- [Materials](#)
- [Project](#)
- [Trending](#)



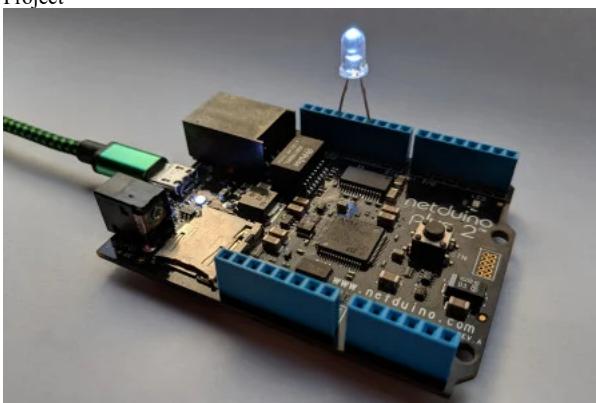
[Monitor a Plant's Soil Moisture Using Netduino and Xamarin Project](#)



[MedUino - Smart Medicine Reminder with Arduino Project](#)



[Create Rainbow Colors with an RGB LED and Netduino Project](#)



[Netduino Pulse-Width-Modulation LED Project](#)

Hardware

1	DFR0478	\$ 19
1	101020011	\$ 5.9

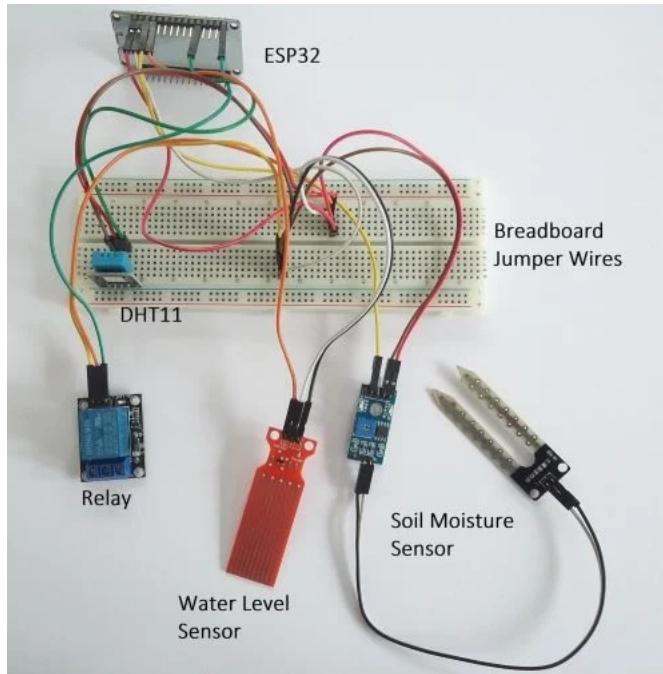
1	1298	\$ 49.95
1	MO750	\$ 289.99
1	2-1415037-1	\$ 6.24
1	153	\$ 4.95

Gardening is a very gratifying hobby. However, because of our busy lifestyles, plants in our gardens might not always get the attention they deserve. What if we could remotely monitor their conditions and give them the care they need? There is a way! In this tutorial, we are going to create an IoT-based smart gardening system using the Blynk App.

This smart garden project is a plant environment monitoring system that monitors soil moisture level, air temperature, humidity, and water level. Then, depending on the conditions, the system allows users to deliver what is needed remotely through a mobile app. Moreover, this system can be scaled and expanded to automate the entire gardening process!

Required Hardware

- ESP32
- DHT11 Temperature and Humidity Sensor
- Water Level Sensor
- Soil Moisture Sensor
- Relay Module
- Breadboard and Jumper Wires
- Smartphone with the Blynk app installed

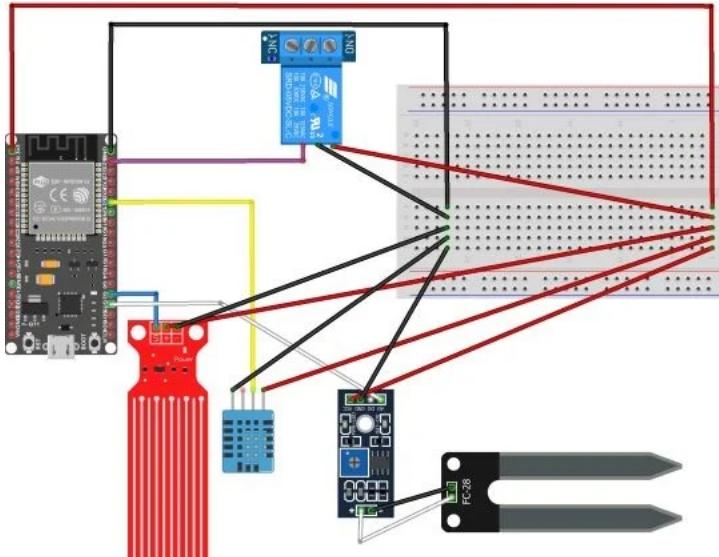


Required Hardware

Wiring the Hardware

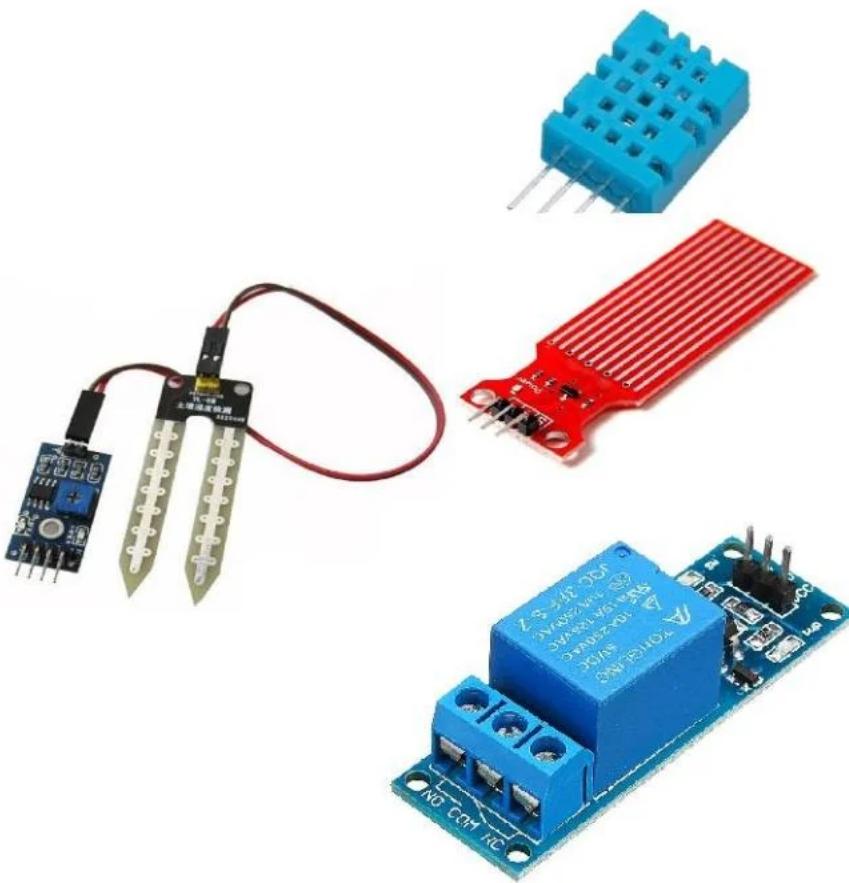
In this project, we will use a soil moisture sensor to measure the volumetric water content in the soil. The DHT11 sensor will measure temperature and humidity levels in the plant's surroundings, and the water level sensor will measure the water level in a small tank.

The Fritzing diagram showing the hardware connections is shown below.



fritzing

The following diagram shows how each sensor is connected to the analog pins of the ESP32 development board.



Components	ESP32 Pin
DHT11	GPIO 21
Water Level Sensor	GPIO 2
Moisture Sensor	GPIO 15
Relay Module	GPIO 23

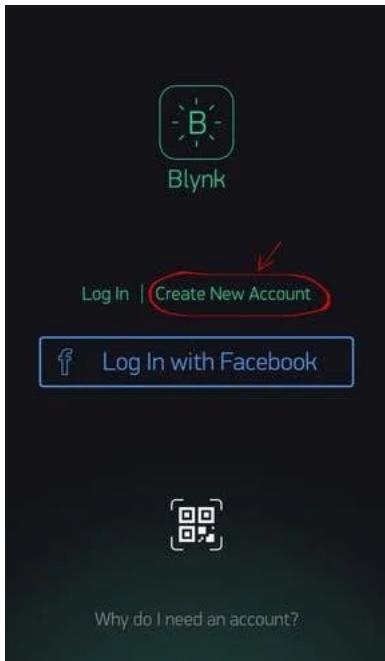
In this setup, we are monitoring the sensor data and interfacing a relay module to control high voltage devices such as motors and lighting devices. Through the Blynk app, we can remotely monitor various parameters in our garden and activate devices as needed.

If the water level in the tank is reduced to the minimum value, we can switch ON a motor remotely to fill the tank. If the temperature is reduced to the minimum requirement, we can switch ON the lighting to increase the temperature to a certain value. If your plants need more water, you can open a valve remotely to water the plants.

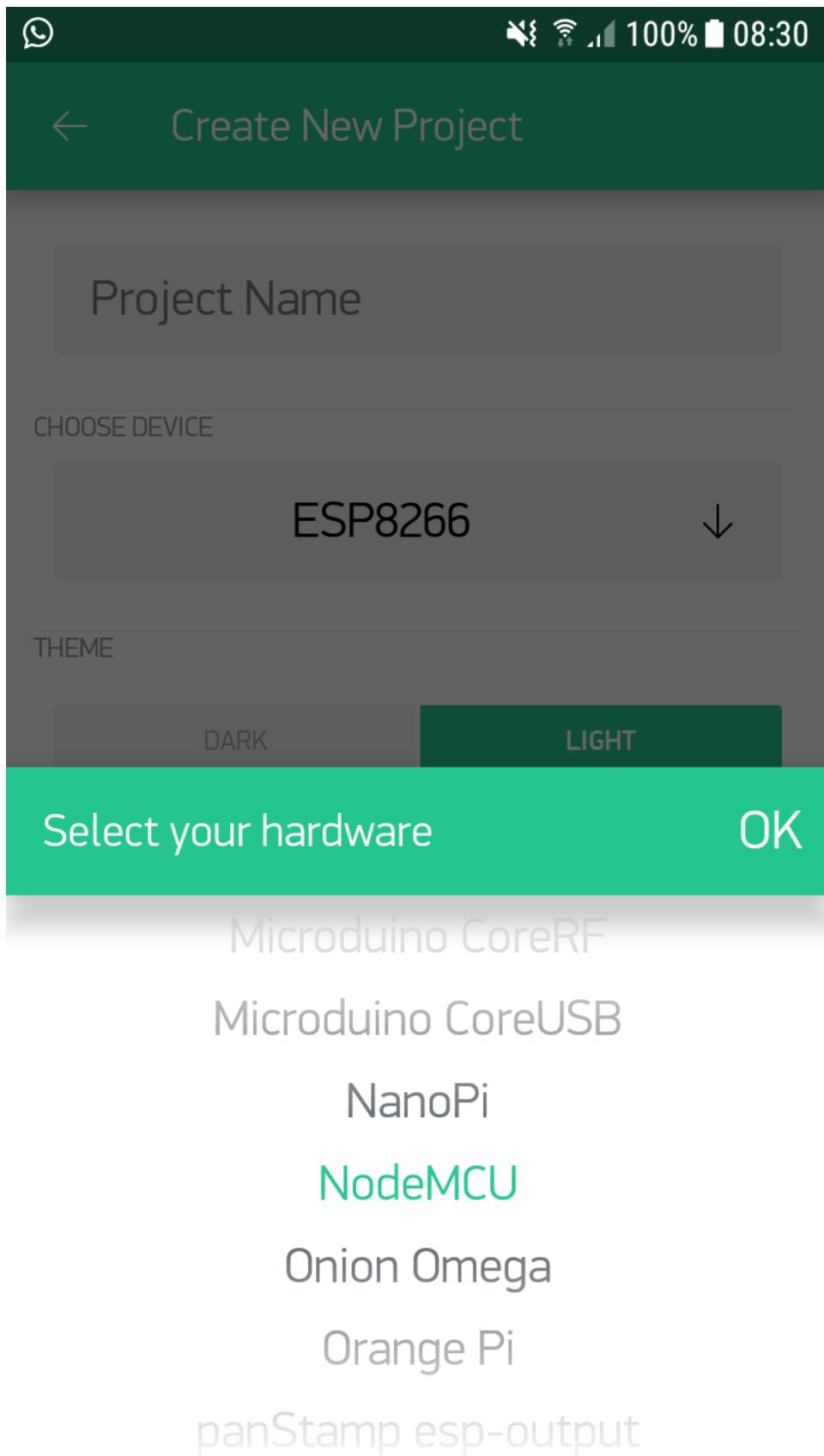
Setting up the Blynk App

Now it's time to set up the Blynk App to monitor sensor values. The first step is to [install the Blynk App](#) on your smartphone.

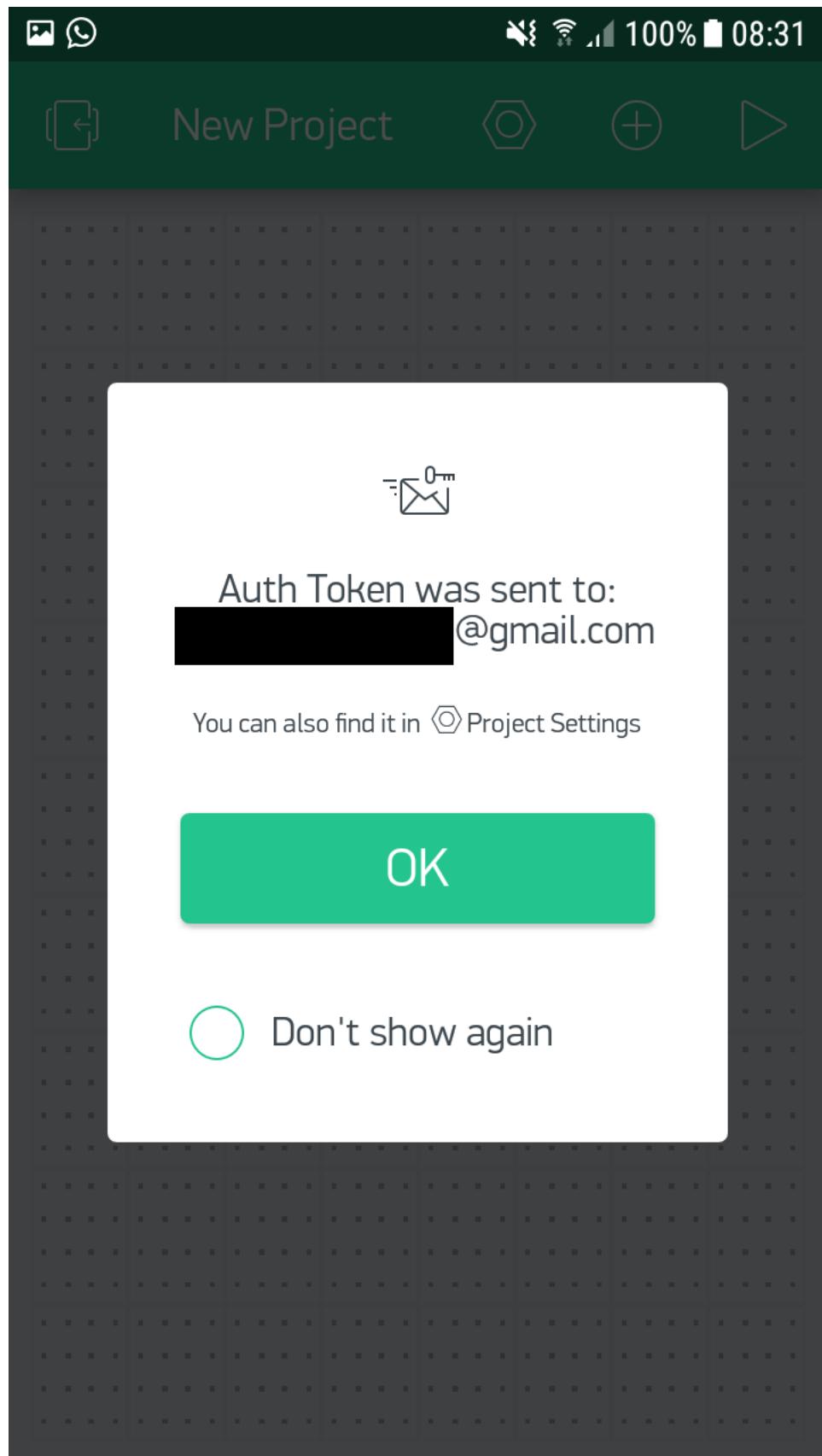
Upon installation, you'll need to create a new Blynk account. This step is important as the app will send a unique authentication code to your email. You'll need this to continue using the Blynk app.



After creating a new account, create a new project and select your hardware.

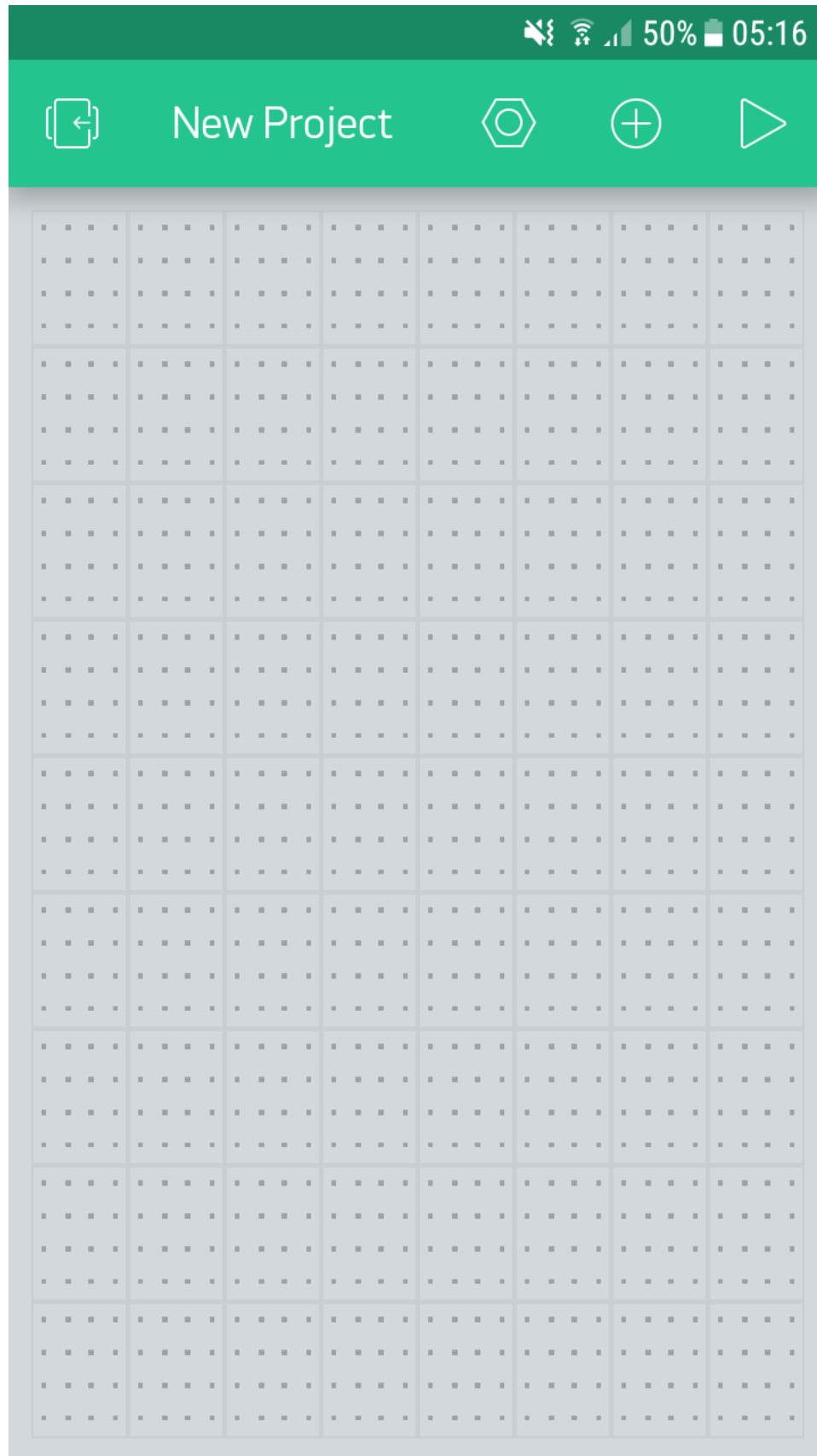


The Blynk app automatically sends you a unique authentication code for every new project you create. You need this code to connect your smartphone to the hardware. The authentication code will be sent to the email address you used for registration.

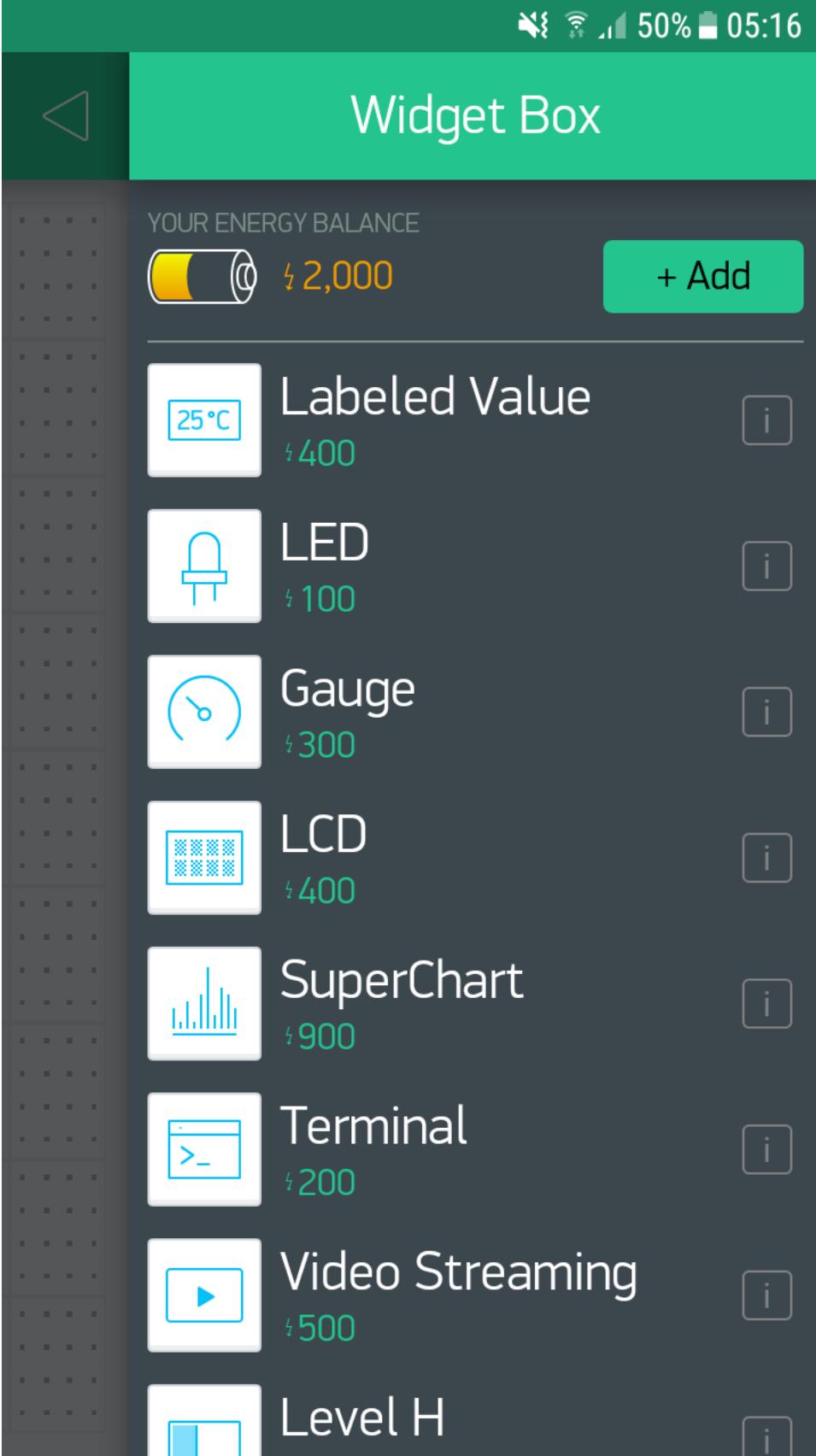


Creating our Smart Garden Monitoring System in Blynk

Now you have an empty canvas as shown below. We need to add four gauge widgets to monitor temperature, humidity, water level, and soil moisture level.



Tap on the canvas to open up the widget box. Now pick a gauge.



50% 05:16



Widget Box

YOUR ENERGY BALANCE



+ Add



Labeled Value

£ 400



LED

£ 100



Gauge

£ 300



LCD

£ 400



SuperChart

£ 900



Terminal

£ 200



Video Streaming

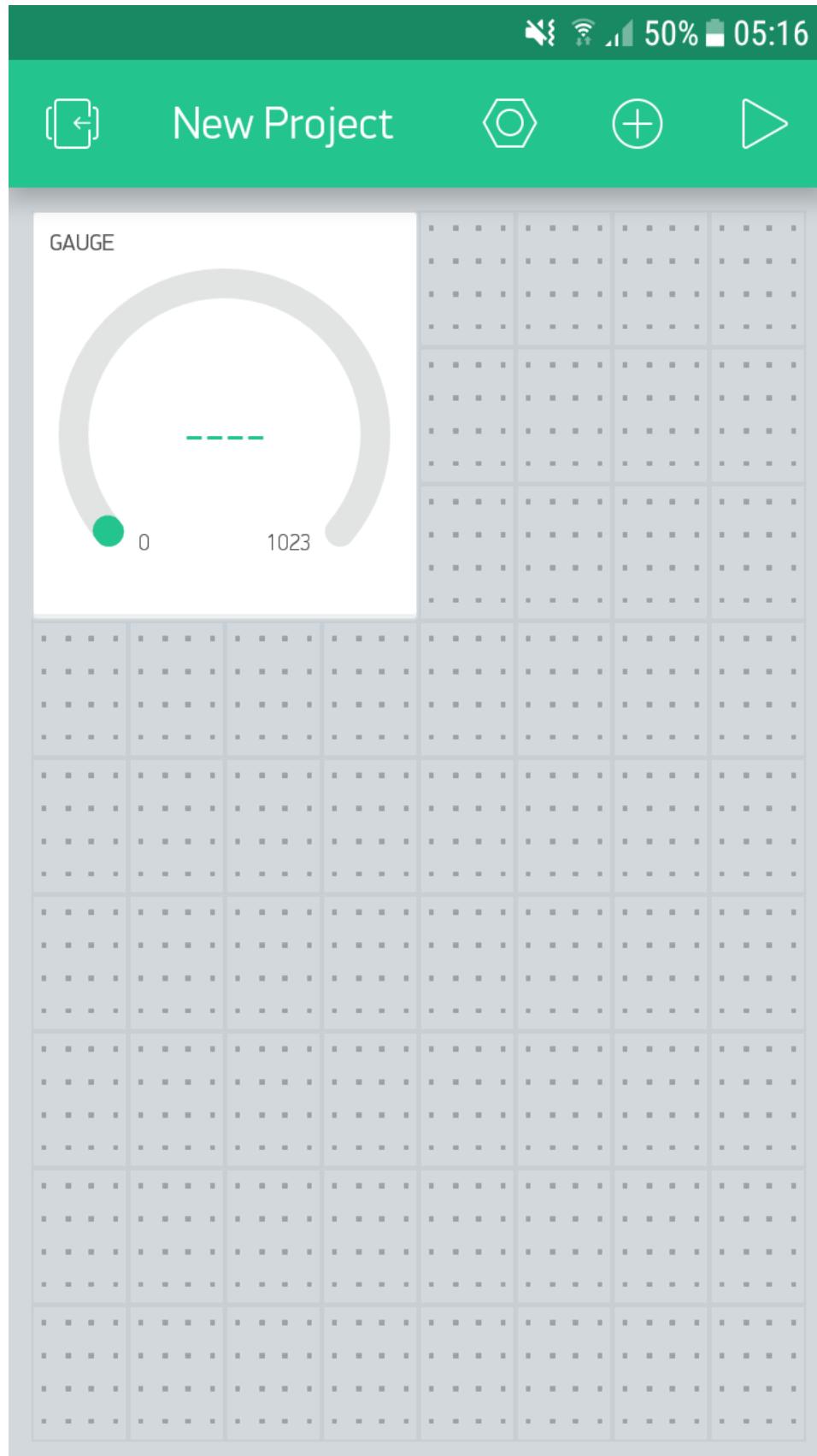
£ 500



Level H



Tap and drag the widget on to the canvas.



Tap on the widget to change its settings. There, you can change the name, size, text size, and color of the gauge.

Gauge

INPUT

PIN

0 1023

LABEL

e.g: Temp: /pin/ °C

DESIGN

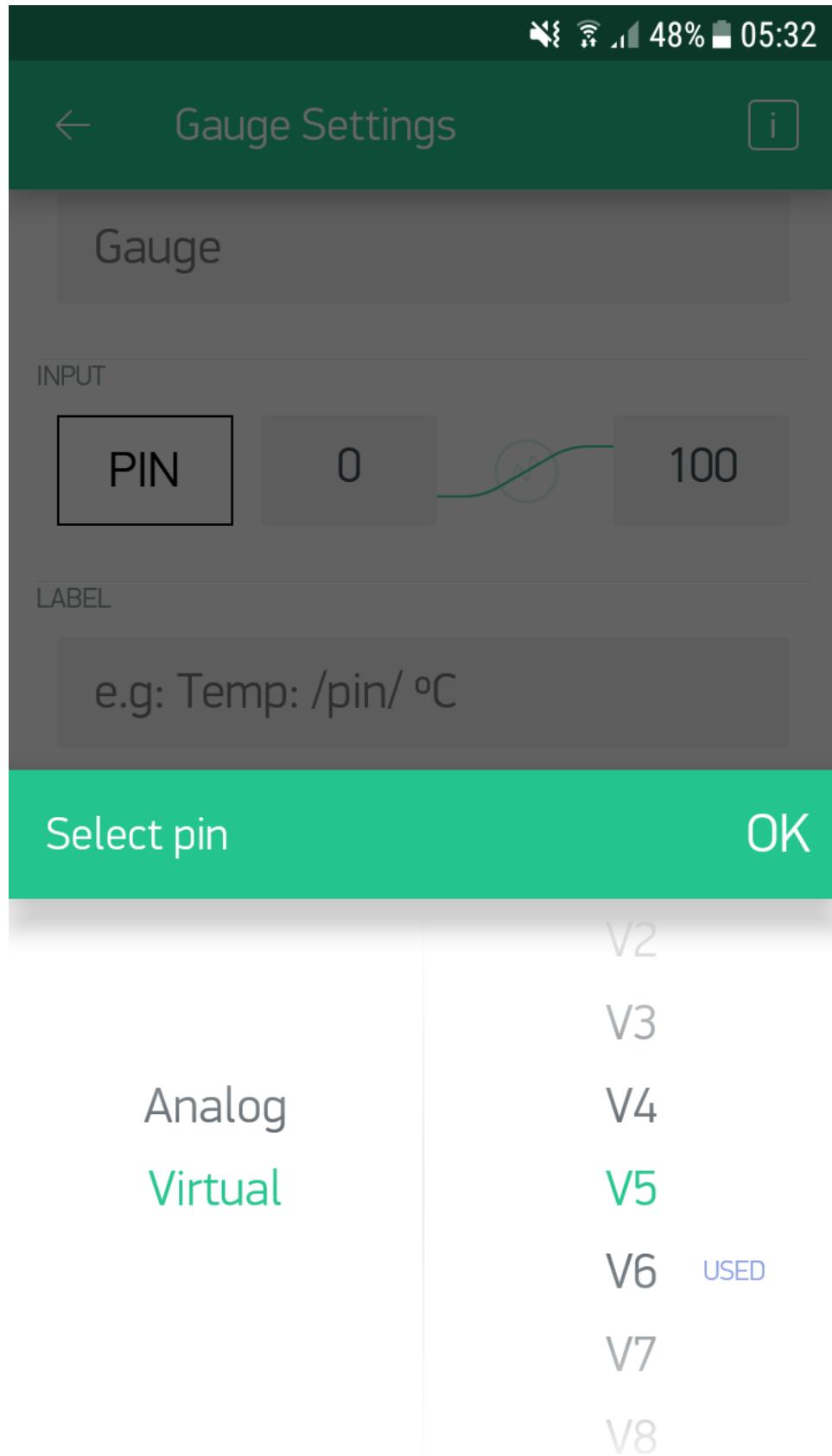
FONT SIZE

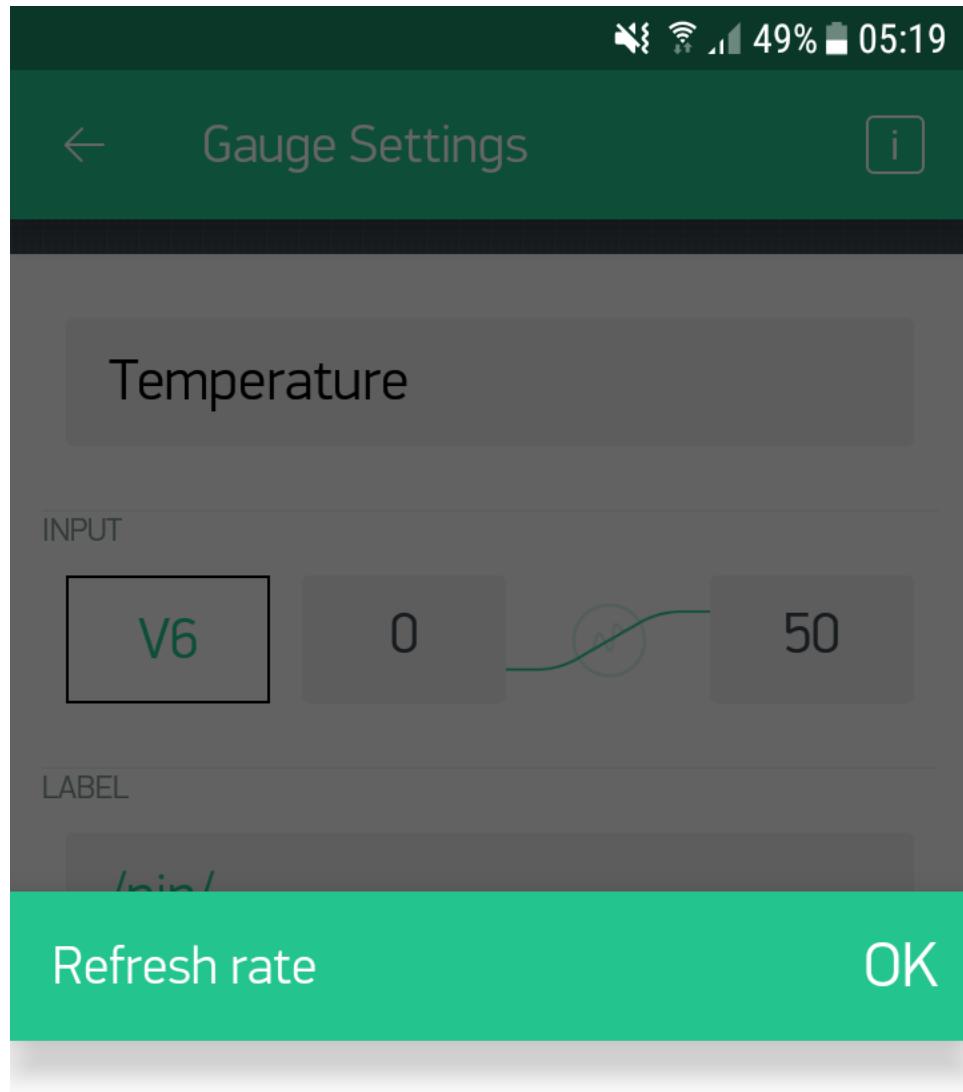
T T T

TEXT

READING RATE

The two most important settings are the pin and the refresh rate. Set a virtual pin for each gauge with two seconds of refresh time for temperature and humidity values. Set five seconds of refresh time for water level and moisture level.





PUSH

1 sec

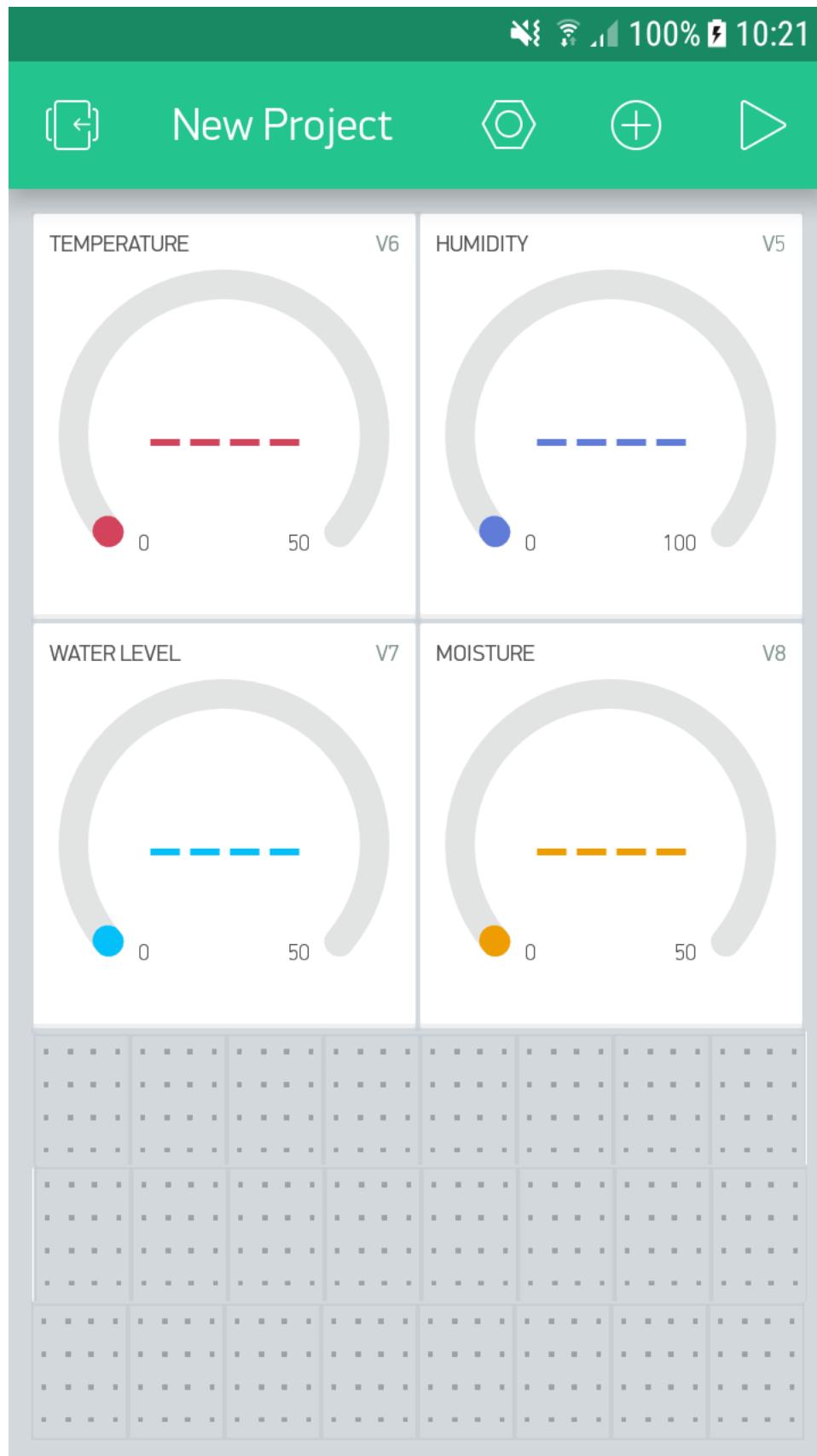
2 sec

5 sec

10 sec

15 sec

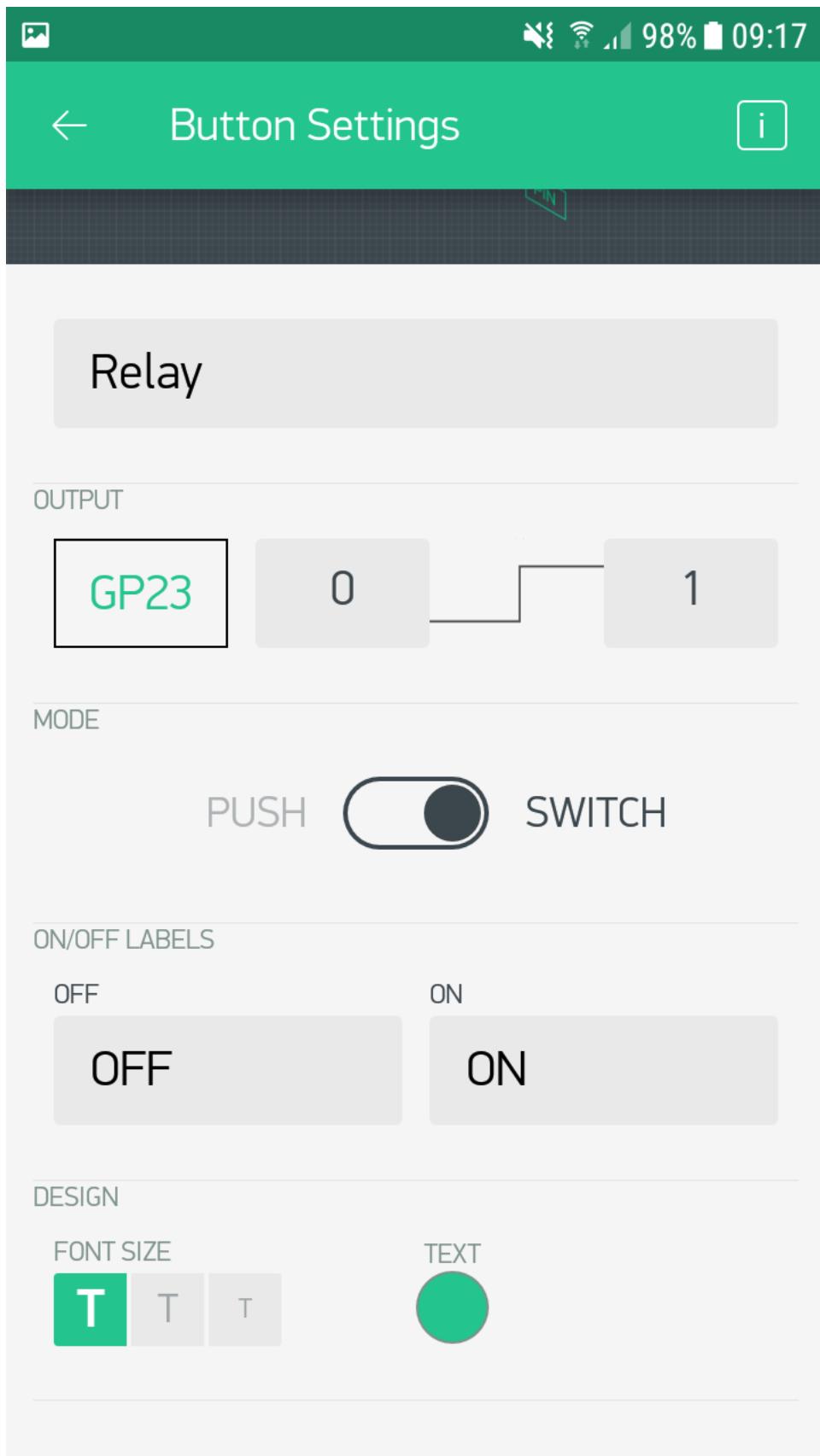
Now your canvas should look like this:



Finally, add a button widget to control the relay. There, change the name of the button, button type, and pin. The pin is the GPIO pin of the ESP32 where the relay is connected.

The image shows the 'Button Settings' screen from the Blynk app. At the top, there's a header bar with a back arrow, the title 'Button Settings', and an information icon. Below the header is a diagram of a button component with a ground connection (G) and a pin connection (PIN). The main area contains several configuration sections:

- Button**: A large section labeled 'Button' with a red border.
- OUTPUT**: A digital output configuration with a 'PIN' selector (set to 'PIN'), a value '0', a switch symbol, and a value '1'.
- MODE**: A mode selection section with a 'PUSH' label, a toggle switch, and a 'SWITCH' label.
- ON/OFF LABELS**: Buttons for 'OFF' and 'ON' states.
- DESIGN**: Font size and text color selection.



Now the Blynk App project configuration is finished and you are ready to go!

Arduino Code

Download the Arduino [source code](#).

The Arduino code is pretty straightforward and easy to understand. I have used the DHT11 sensor library to get the temperature and humidity. Water and soil moisture levels are just analog readings. According to your environment and sensor constraints, these analog values can be mapped into a range.

There are two functions to send data to the Blynk App. The first function sends temperature and humidity values and the second one sends water and soil moisture levels.

```
void sendSensor1()
{
    float h = dht.readHumidity();
    float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit

    if (isnan(h) || isnan(t)) {
        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(V5, h);
    Blynk.virtualWrite(V6, t);

}

void sendSensor2()
{
    waterLevel = analogRead(waterLevelSensor);
    waterLevel = map(waterLevel, 0, 2000, 0, 20);
    moistureLevel = analogRead(moistureLevelSensor);
    moistureLevel = map(moistureLevel, 0, 4000, 0, 20);

    Blynk.virtualWrite(V7, waterLevel);
    Blynk.virtualWrite(V8, moistureLevel);
}
```

The Arduino Blynk library has a method called **virtualWrite()** to write the sensor reading into virtual pins we have already configured in our Blynk app.

Before sending the values, we must configure the Wi-Fi settings.

```
BlynkTimer timer;

// Go to the Project Settings (nut icon).
char auth[] = "YOUR AUTH CODE";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "YOUR WIFI SSID";
char pass[] = "YOUR WIFI PASSWORD";
```

The Blynk timer object sends temperature and humidity readings every two seconds and water and soil moisture levels every five seconds. That data will update in the gauges of the Blynk App.

```
Blynk.begin(auth, ssid, pass);

dht.begin();

// Setup a function to be called every second
timer.setInterval(2000L, sendSensor1);
timer.setInterval(5000L, sendSensor2);
```

Running the System

Press the Run button to start the Blynk App. You can see that your gauges will start to display the sensor readings and these values update every two and five seconds. You can turn ON and OFF the relay by pressing the button remotely from anywhere.





[Jinani Wijesinghe](#)

I am a mechatronics engineer and an electronics hobbyist. I love to share innovative projects with DIY enthusiasts across the globe.

Author

Comments (3)

[Jim Wignall](#) 9 months ago
0 replies

Thank-you for the great tutorial. But, I have a question. Could I have the software send me an email when one of the sensors, say moisture, goes below a certain value? My tendency toward neglecting my garden at critical times has cost me in the past. What code would be necessary to alert me via email to go look things over?



[Wayne Smith](#) 9 months ago
0 replies

Hello Jinani,

Great tutorial. Thank You!

One question. Between the moisture sensor and the ESP2, your Fritzing drawing shows a 'module'. What is purpose of this module? Part #? Source?.

God Bless,

Wayne



[Emmanuel Banda](#) A month ago
0 replies

hello when i upload the cord to my ESP 32 only Temperature and Humidity date is what is being Displayed in Blynk app . Soil moisture and water lever shows o. i have check my wiring and setting on both sides it fine. any assistance will be highly appreciated

Comment on this project

[Submit](#)

Categories

- [Sensors](#)
- [Automation](#)
- [Cloud Computing](#)
- [Industrial](#)
- [IoT](#)
- [Weather](#)

Tags

- [Arduino](#)
- [humidity sensor](#)
- [temperature sensor](#)
- [Esp32](#)
- [Blynk](#)
- [Gardening](#)

Related Blynk Projects & Tutorials



[Build an IoT-Controlled Robot With ESP8266 and Blynk](#)

- [Blynk](#)
- September 4, 2019

 [Reginald Watson](#)

2 1 4.8k



[How to Make an Audio Player With Speaker Using the Arduino Uno](#)

- [Arduino](#)
- March 27, 2016

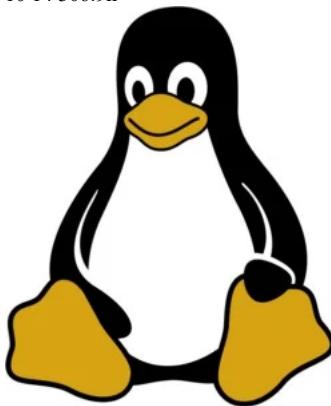
 [Sarvagnya Purohit](#)

51 7 90.5k



[How to Connect a Raspberry Pi to a Laptop Display | Headless Setup](#)

- [Raspberry Pi](#)
- March 21, 2018

 [Aqib](#)
10 14 306.9k

[Basic Linux Commands for Beginners](#)

- [Linux](#)
- March 21, 2018

 [Alok Naushad](#)

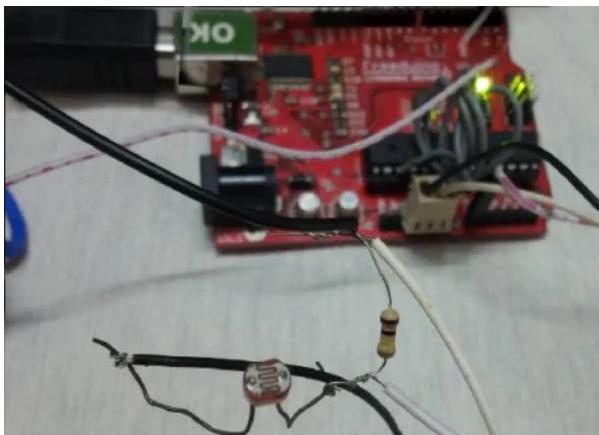
73 39 1.4m



[How to Build an Arduino Self-Balancing Robot](#)

- [Arduino](#)
- May 2, 2017

 [Reginald Watson](#)
80 26 124.4k

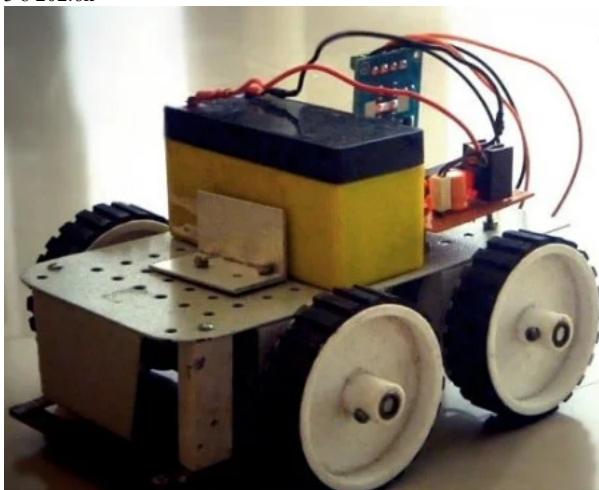


[How to Use an LDR Sensor With Arduino](#)

- [Arduino](#)
- March 21, 2018

 [Arvind Sanjeev](#)

3 8 202.6k



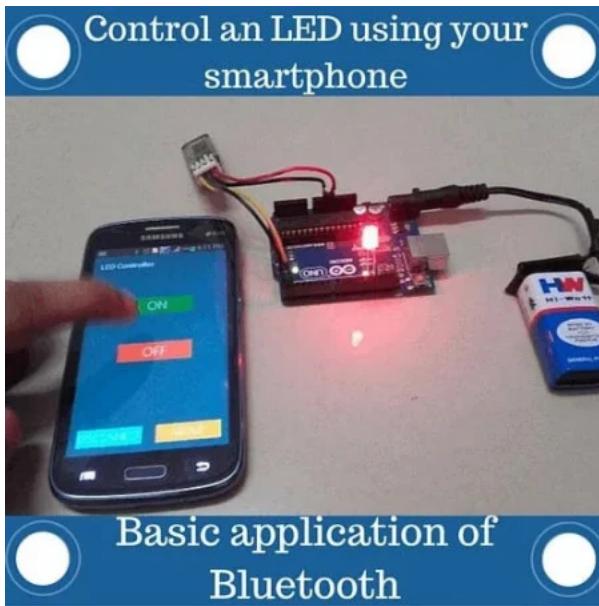
[How to Make Your Own Remote-Controlled Car](#)

- [PCB](#)
- January 8, 2016



[Mayoogh Girish](#)

318 14 95.5k



[Bluetooth Basics: How to Control an LED Using a Smartphone and Arduino](#)

- [Arduino](#)
- March 23, 2018



[Mayoogh Girish](#)

12 7 177k

[Show Footer](#)

We Are



We provide a place for makers like you to share your designs, collaborate with one another, and learn how to take your product to market.

Network sites

[AllAboutCircuits.com](#) [EEPower.com](#) [Mikrocontroller.net](#) [ElectronicsPoint.com](#)

Categories

- [3D Printing](#)
- [Amateur Radio](#)
- [Audio](#)
- [Augmented Reality](#)
- [Automation](#)
- [Automotive](#)
- [Cloud Computing](#)

- [Computers & Peripherals](#)
- [Consumer Electronics](#)
- [Cyber Security](#)
- [Displays](#)
- [Drones](#)
- [Health & Fitness](#)
- [Home Automation](#)
- [Industrial](#)
- [Industrial IoT](#)
- [IoT](#)
- [Lighting](#)
- [Machine Learning](#)
- [Mobile](#)
- [Motor Control](#)
- [Power](#)
- [Robotics](#)
- [Security / Identification](#)
- [Sensors](#)
- [Smart Grid/Energy](#)
- [Telecom](#)
- [Virtual Reality](#)
- [Wearables](#)
- [Weather](#)
- [View All](#)

Platforms

- [Linux](#)
- [Raspberry Pi](#)
- [Arduino](#)
- [ESP8266](#)
- [Custom](#)
- [PIC](#)
- [PCB](#)
- [MicroPython](#)
- [View All](#)

Connect with Us

- [Facebook](#)
- [Twitter](#)
- [YouTube](#)
- [Pinterest](#)

- [About Us](#)
- [Contact Us](#)
- [Write for Us](#)
- [Help Center](#)

EETech Media, LLC. All rights reserved

- [Privacy](#)
- [Terms of Service](#)

[Top](#)