Module 9: IOT Platform - Blynk

Introduction

Blynk is a platform that allows you to create and control IoT applications using a smartphone, tablet, or web browser. You can use Blynk to connect various hardware devices, such as Arduino, Raspberry Pi, ESP8266, ESP32, and others, to the internet and interact with them through a graphical user interface. Blynk provides a widget library that you can drag and drop to design your application, including buttons, sliders, displays, charts, and more. You can also use Blynk to monitor and analyze data from your sensors, send notifications, and integrate with third-party services.

By using Blynk with the ESP32, you can easily and quickly prototype your IoT projects without writing complex code or dealing with network protocols. You can use Blynk to control ESP32 outputs, such as LEDs, relays, motors, and more, as well as read inputs from sensors like temperature, humidity, light, and others. Blynk also allows you to create custom functions and logic for your ESP32, such as timers, triggers, and actions.

Setting Up Blynk

Please download the Blynk app on your smartphone. For more information on how to register an account, create a template, configure datastreams, set up the web dashboard, and configure the mobile dashboard, you can watch this video tutorial.

Blynk Syntax

Blynk events are actions that occur when you interact with your Blynk application or when your hardware sends or receives data from the Blynk server. You can use Blynk events to control the logic and behavior of your ESP32, such as turning devices on or off, reading or writing data, performing calculations, and more. You can handle Blynk events using Blynk library functions and Arduino sketches.

The Blynk library provides several functions you can use to handle Blynk events, such as:

- **Blynk.begin()**: This function initializes the Blynk connection and connects your ESP32 to your Blynk project. You need to call this function in the `setup()` function of your Arduino sketch and provide your WiFi credentials and Blynk authentication token as parameters. Example:

```
void setup() {
  Serial.begin(115200);
  // Initialize Blynk connection
  Blynk.begin(auth, ssid, pass);
}
```

- **Blynk.run()**: This function runs the Blynk process and handles communication between your ESP32 and your Blynk project. You need to call this function in the `loop()` function of your Arduino sketch, ensuring it is not blocked by long-running code. Example:

```
void loop() {
  // Run Blynk process
  Blynk.run();
}
```

- **BLYNK_WRITE()**: This function is a macro that defines a callback function executed when a widget writes a value to a pin on your ESP32. You can use this function to handle events triggered by your Blynk app, such as pressing a button, moving a slider, or sending a command. You need to specify the pin number as a parameter and use the `BlynkParam` object to access the value. Example:

```
// Define a callback function for pin V1
BLYNK_WRITE(V1) {
  // Get value from widget
  int value = param.asInt();
  // Do something with the value
  Serial.println(value);
}
```

- **Blynk.virtualWrite()**: This function writes a value to a virtual pin in your Blynk project. You can use this function to send data from your ESP32 to your Blynk app, such as sensor readings, status updates, or custom messages. You need to specify the virtual pin number and the value as parameters. You can also specify the data type, such as `int`, `float`, `string`, or `array`. Example:

```
// Write value to pin V2
Blynk.virtualWrite(V2, 123);
// Write string to pin V3
```

```
Blynk.virtualWrite(V3, "Hello Blynk");

// Write array to pin V4

int array[] = {1, 2, 3, 4, 5};

Blynk.virtualWrite(V4, array, 5);
```

- **BLYNK_READ()**: This function is a macro that defines a callback function executed when a widget reads a value from a pin on your ESP32. You can use this function to handle events triggered by your Blynk app, such as sensor value requests, display updates, or chart refreshes. You need to specify the pin number as a parameter and use the `Blynk.virtualWrite()` function to send the value. Example:

```
// Define a callback function for pin V5

BLYNK_READ(V5) {

  // Read value from sensor

  int value = analogRead(A0);

  // Send value to widget

  Blynk.virtualWrite(V5, value);

}
```