## Code on vscode (esp32 connected to IR sensor only)

blynkapp > src > main.cpp > loop()

```cpp
#define BLYNK_TEMPLATE_ID "TMPL2fncfpLtN"
#define BLYNK_TEMPLATE_NAME "esp"
#define BLYNK_AUTH_TOKEN "iT3RMf_ITB3yMAL-haL6_rsw7NIrEKz3"

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#define IR_SENSOR_PIN 4 // Pin connected to IR sensor
#define LEDPIN LED_BUILTIN       // Pin connected to LED
// Blynk credentials
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "network";
char pass[] = "12345678";
void setup() {
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  pinMode(LEDPIN, OUTPUT); // Set LED pin as output
  pinMode(IR_SENSOR_PIN, INPUT); // Set IR sensor pin as input
}
void loop() {
  Blynk.run(); // Run Blynk
  // Read the IR sensor value
  int irValue = digitalRead(IR_SENSOR_PIN);
  // Send IR sensor data to Blynk app
  Blynk.virtualWrite(V4, irValue); // Send IR sensor data to virtual pin V4
  delay(1000); // Delay between reads
}
// This function is called every time the value of V3 changes
BLYNK_WRITE(V3) {
```

**`#define BLYNK_TEMPLATE_ID`**: Defines the unique identifier for the Blynk template you're using. It ensures that the correct settings and widgets are associated with your project.
**`#define BLYNK_TEMPLATE_NAME`**: Provides a name for the template, helping you identify it within the Blynk platform.
**`#define BLYNK_AUTH_TOKEN`**: This is your authentication token, unique to your project, allowing your ESP32 to communicate with the Blynk server.

**`#include <WiFi.h>`**: Includes the Wi-Fi library, enabling your ESP32 to connect to a Wi-Fi network.
**`#include <BlynkSimpleEsp32.h>`**: Includes the Blynk library for ESP32, enabling communication between your ESP32 and the Blynk cloud.

**`#define LEDPIN LED_BUILTIN`**: Uses the built-in LED of the ESP32.

**`char auth[] = BLYNK_AUTH_TOKEN;`**: Stores the Blynk authentication token in a character array.

`char ssid[] = "network";`: Stores the SSID (name) of the Wi-Fi network you want to connect to.

`char pass[] = "12345678";`: Stores the password for the Wi-Fi network.

```
void setup() {
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  pinMode(LEDPIN, OUTPUT); // Set LED pin as output
  pinMode(IR_SENSOR_PIN, INPUT); // Set IR sensor pin as input
}
```

`Blynk.begin(auth, ssid, pass);`: Connects the ESP32 to the Blynk cloud using the provided authentication token and Wi-Fi credentials.

`pinMode(LEDPIN, OUTPUT);`: Sets the built-in LED pin as an output, allowing you to control it (turn it on/off).

`pinMode(IR_SENSOR_PIN, INPUT);`: Sets the IR sensor pin as an input, allowing the ESP32 to read its value.

```
BLYNK_WRITE(V3) {
  int ledState = param.asInt(); // Get the value from the Blynk app (V3)
  digitalWrite(LEDPIN, ledState); // Control LED based on Blynk command
}
```

`BLYNK_WRITE(V3)`: This function is triggered every time the value of Virtual Pin V3 changes in the Blynk app.

`int ledState = param.asInt();`: Reads the value from the Blynk app (V3) and converts it to an integer. This value determines the LED state (0 or 1).

`digitalWrite(LEDPIN, ledState);`: Sets the LED state based on the value received from the Blynk app (turns it on or off).

```
void loop() {
  Blynk.run(); // Run Blynk
  // Read the IR sensor value
  int irValue = digitalRead(IR_SENSOR_PIN);
  // Send IR sensor data to Blynk app
  Blynk.virtualWrite(V4, irValue); // Send IR sensor data to virtual pin V4
  delay(1000); // Delay between reads
}
```

`Blynk.run();`: Keeps the Blynk connection active, allowing continuous communication with the Blynk cloud.

`int irValue = digitalRead(IR_SENSOR_PIN);`: Reads the current value from the IR sensor. It returns `HIGH` or `LOW` depending on the sensor's state.

`Blynk.virtualWrite(V4, irValue);`: Sends the IR sensor value to Virtual Pin V4 in the Blynk app, allowing you to monitor the sensor's status remotely.

`delay(1000);`: Introduces a 1-second delay between each sensor read to avoid overwhelming the Blynk server with data.

## Needed Library

```
[env:esp32doit-devkit-v1]
platform = espressif32
board = esp32doit-devkit-v1
framework = arduino
monitor_speed = 115200
lib_deps =
    blynkkk/Blynk@^1.3.2
```

## Using the Blynk App

1. **Virtual Pins Setup:**
    - **V3**: Controls built in LED in ESP32.
    - **V4**: Displays the status of the IR sensor (0 means object is in front of sensor , 1 means no object found) .
2. **Adding Widgets:**
    - **Button Widget** (for LED control):
        - Add a Button widget to your dashboard.
        - Link it to Virtual Pin V3.
        - Set it to Switch mode (ON/OFF).
    - **Label or Value Display Widget** (for IR sensor):
        - Add a gauge to display data.

- Link it to Virtual Pin V4.

Wokwi simulation but with ultrasonic sensor
https://wokwi.com/projects/406130966268004353