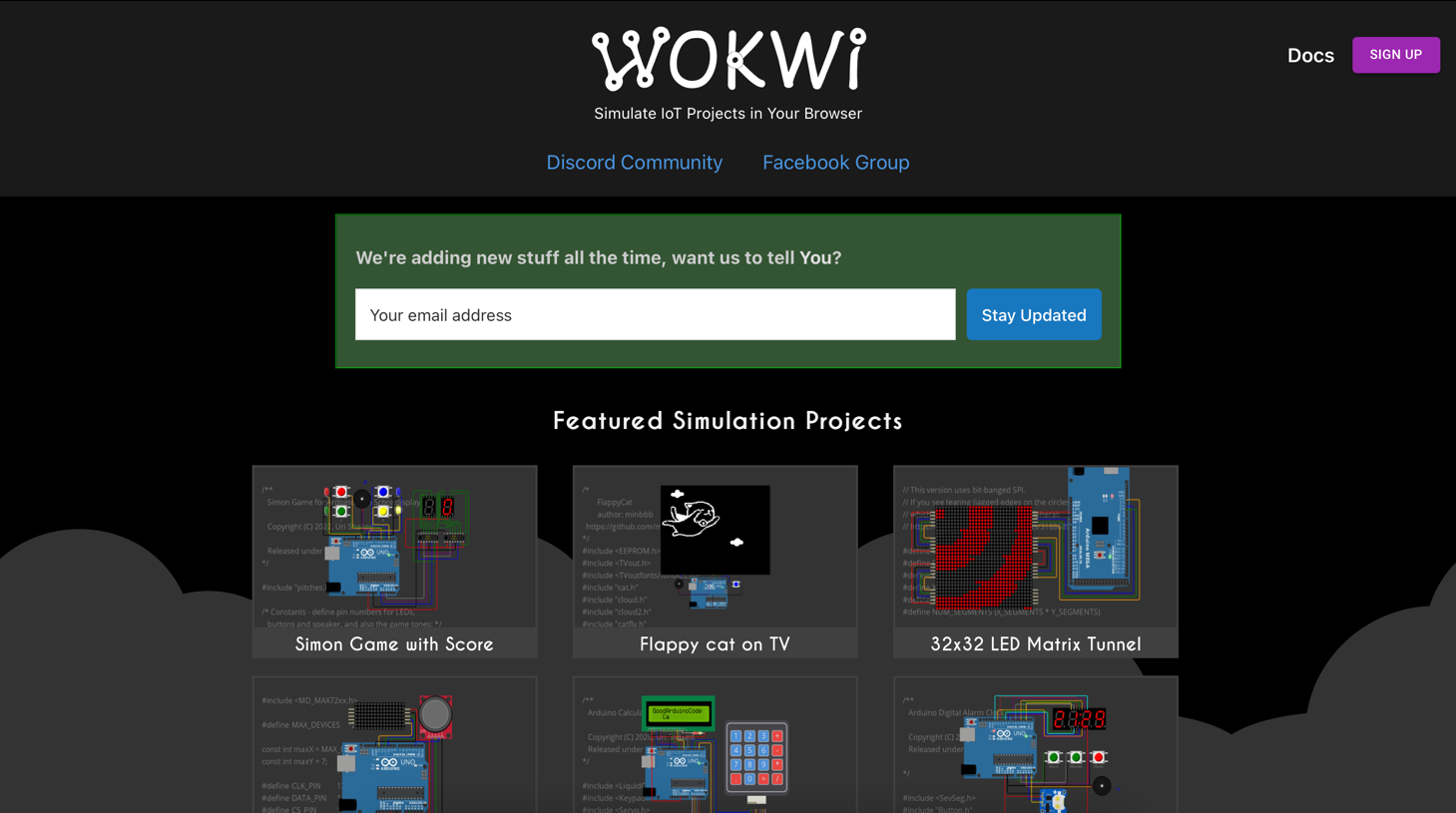
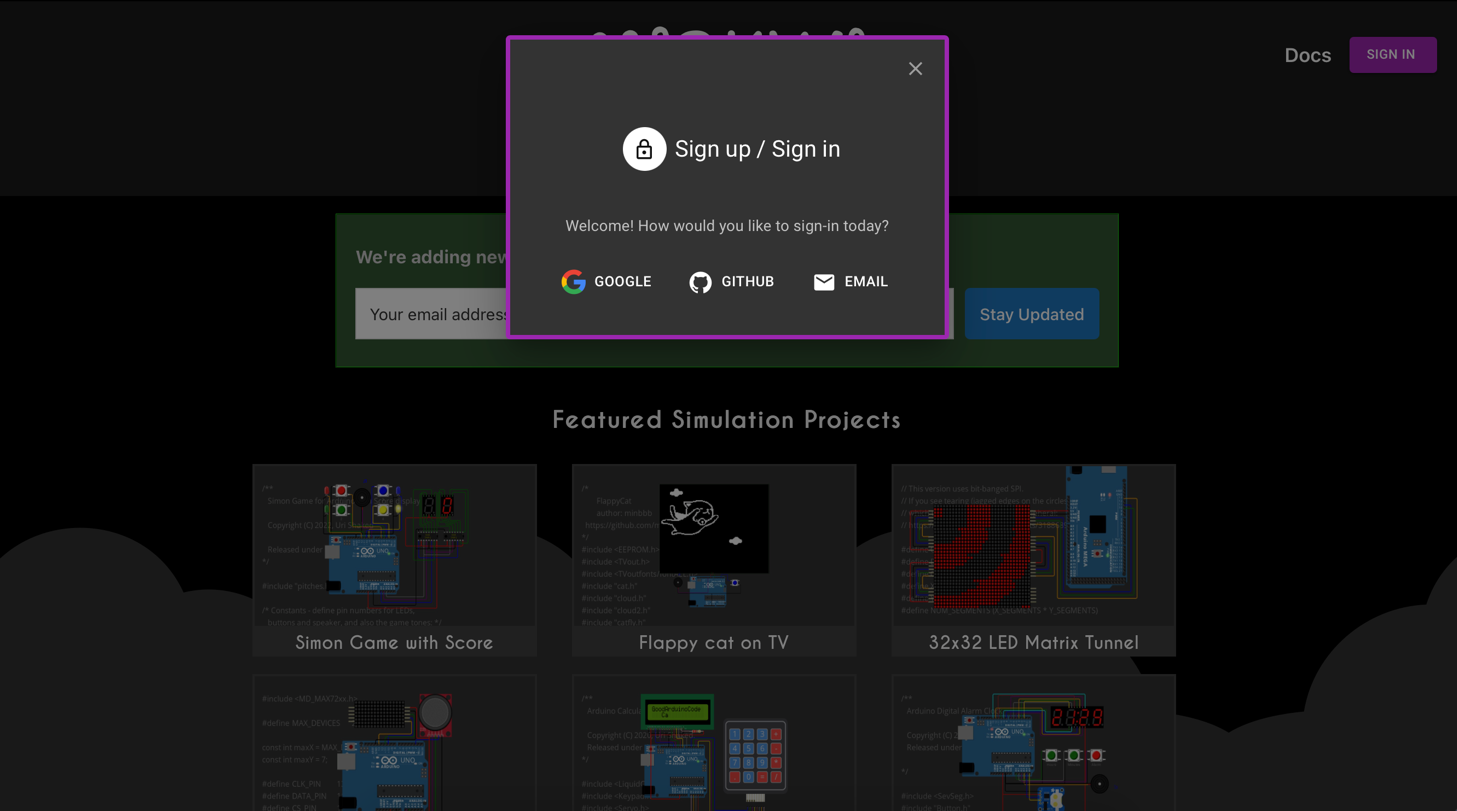
**Sign Up**

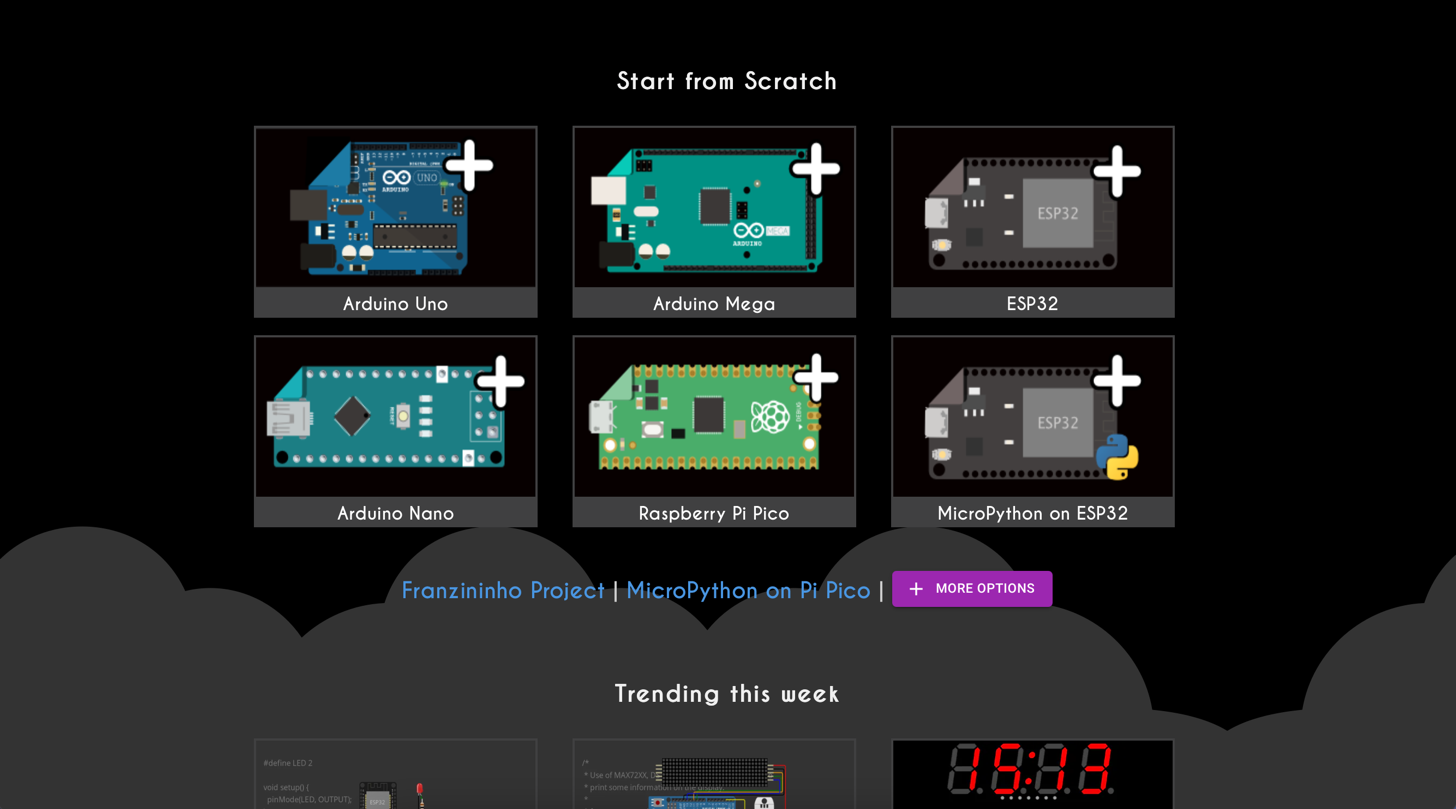
Open <https://wokwi.com> “Wokwi”, an online embedded systems simulator.



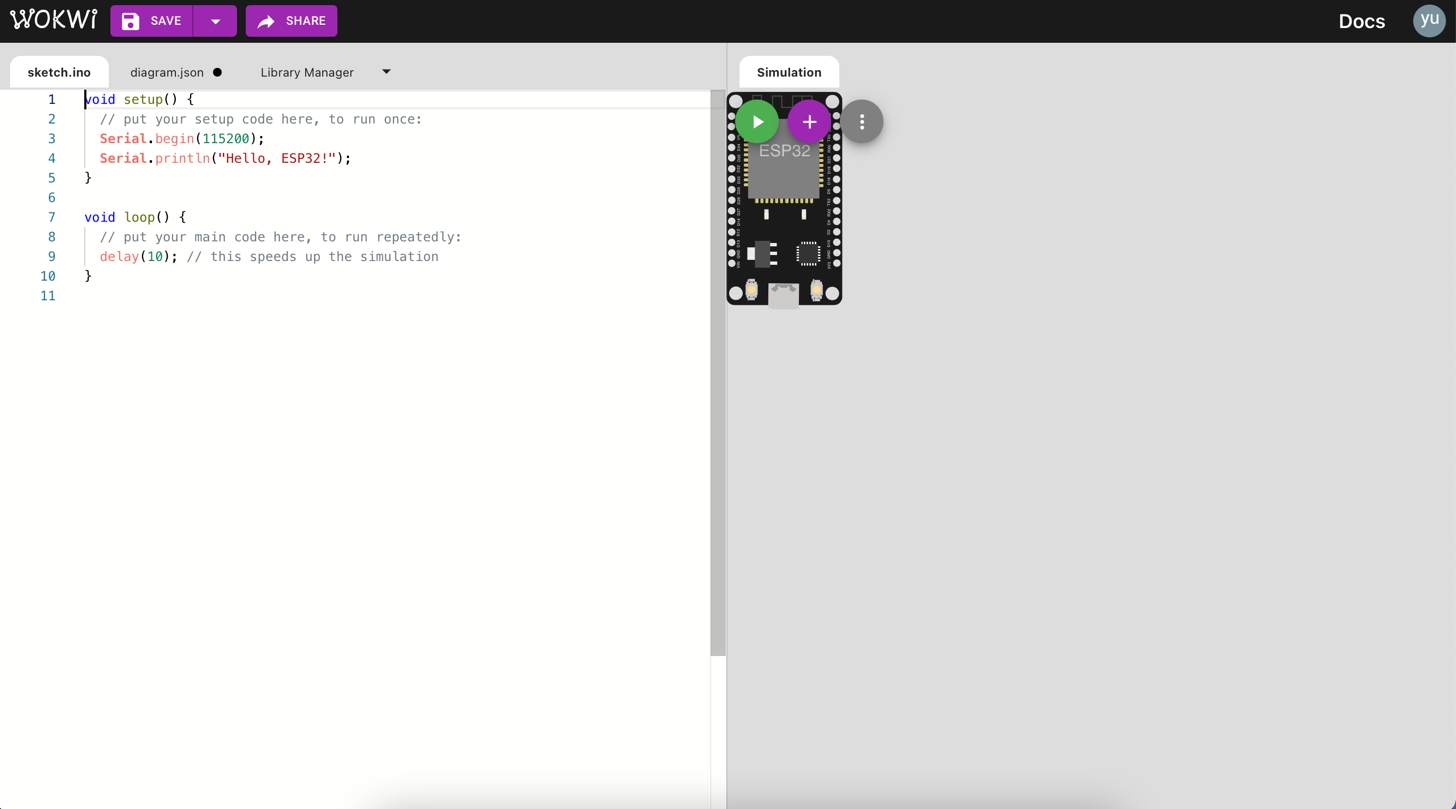
Sign up to Wokwi with your preferred way (GOOGLE, GITHUB, EMAIL).



After Sign up, go to “Start from Scratch” section and choose “ESP32” board.



You will see as follow.



**Exercise 1**

Click on the run button , what will happen? Show your screenshot result.

Graphical user interface, text, application

Description automatically generated

**Exercise 2**

How to do in order to get “Your Name and Your student ID” as followed output? Show your screenshot result.

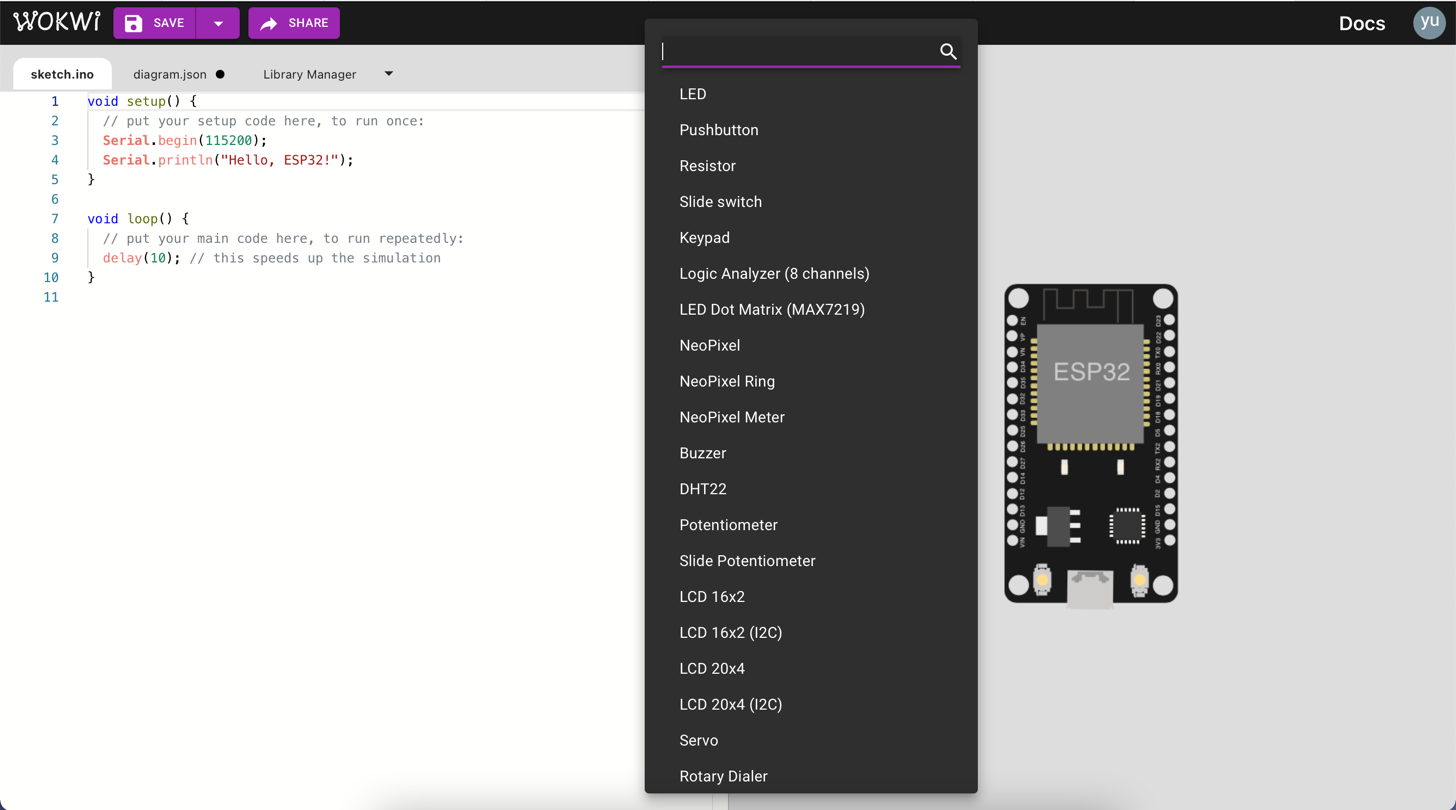


Graphical user interface, text, application

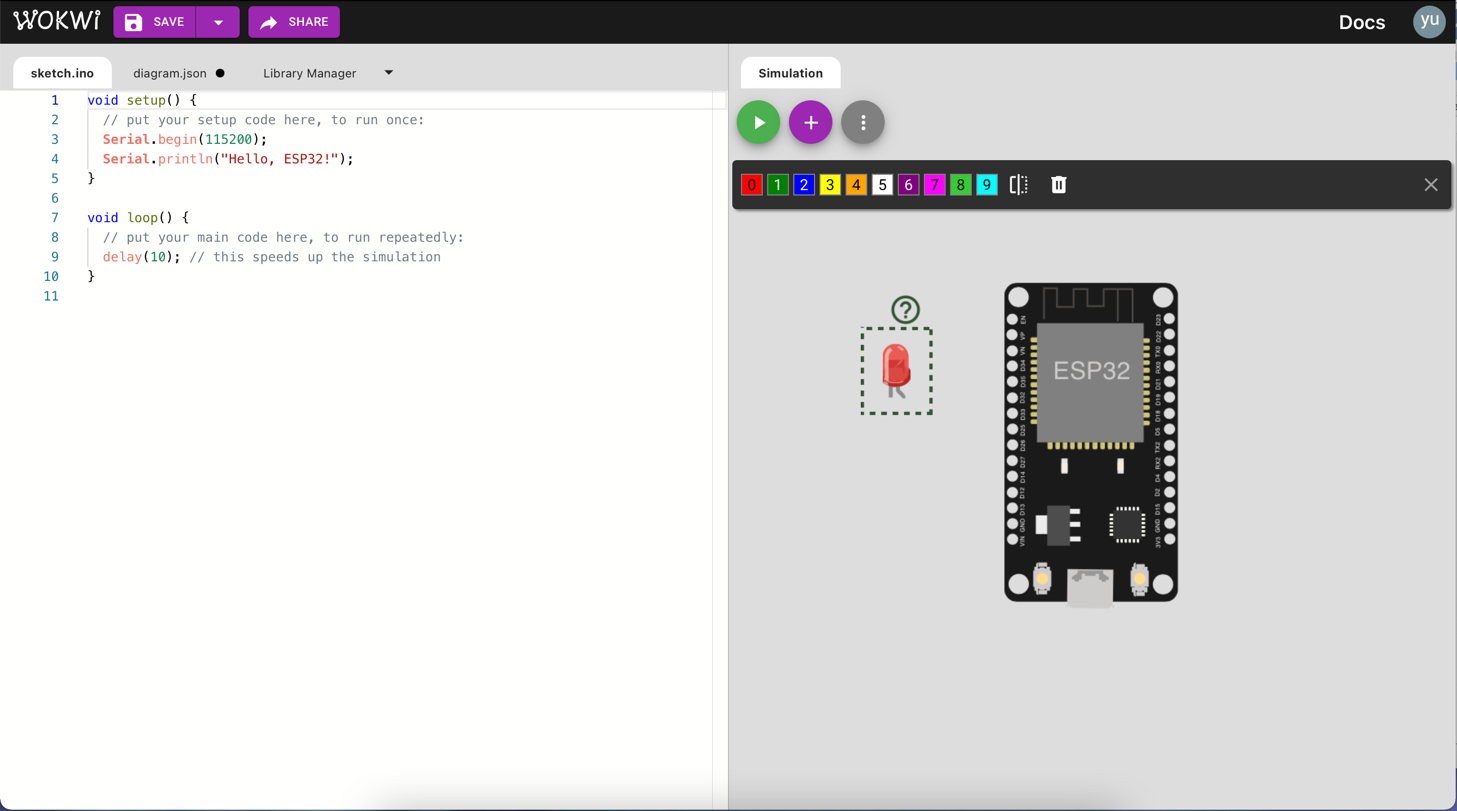
Description automatically generated

**LED blinking**

Add “LED” by clicking this plus sign  and choose LED.

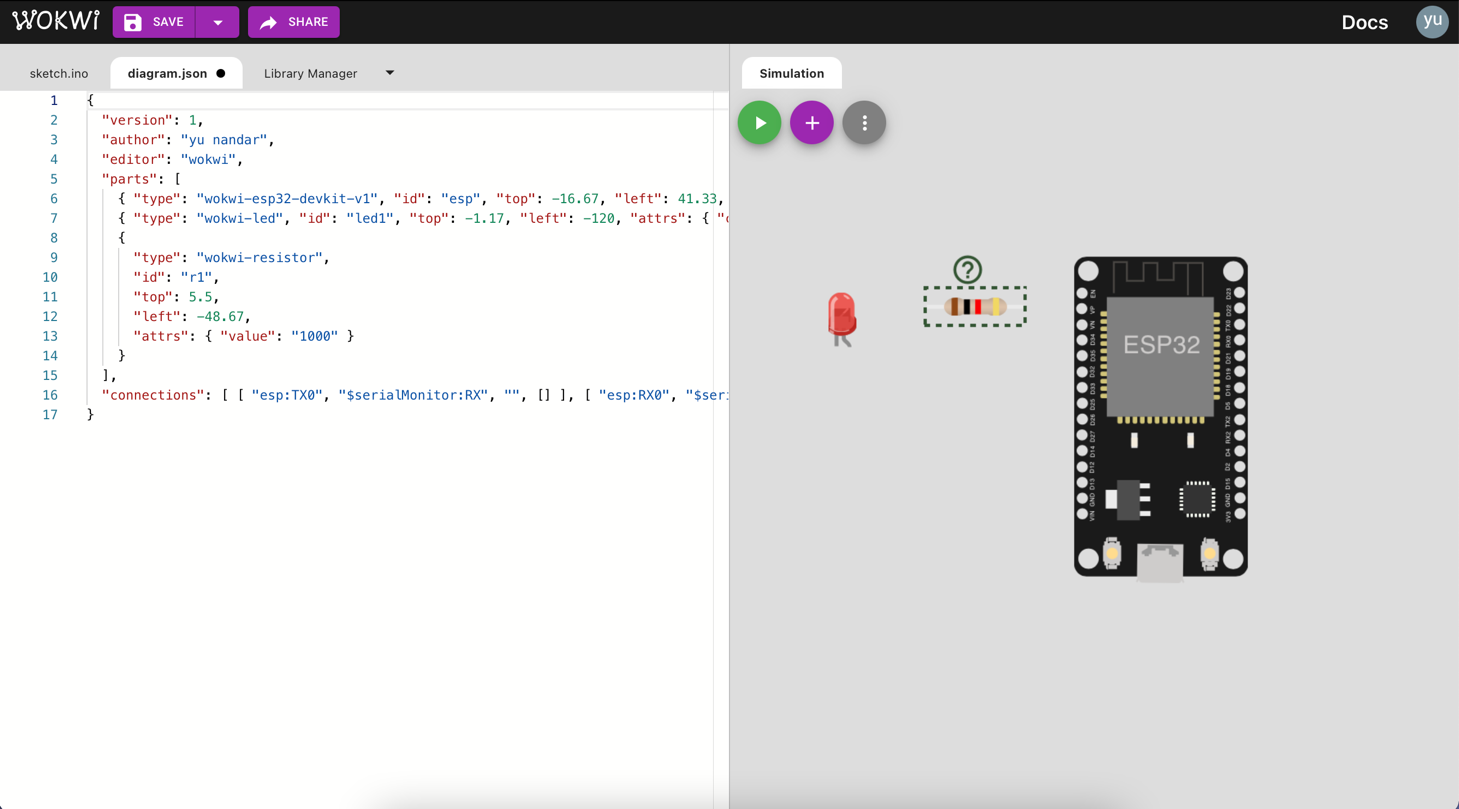


You can change the color of LED as follow.

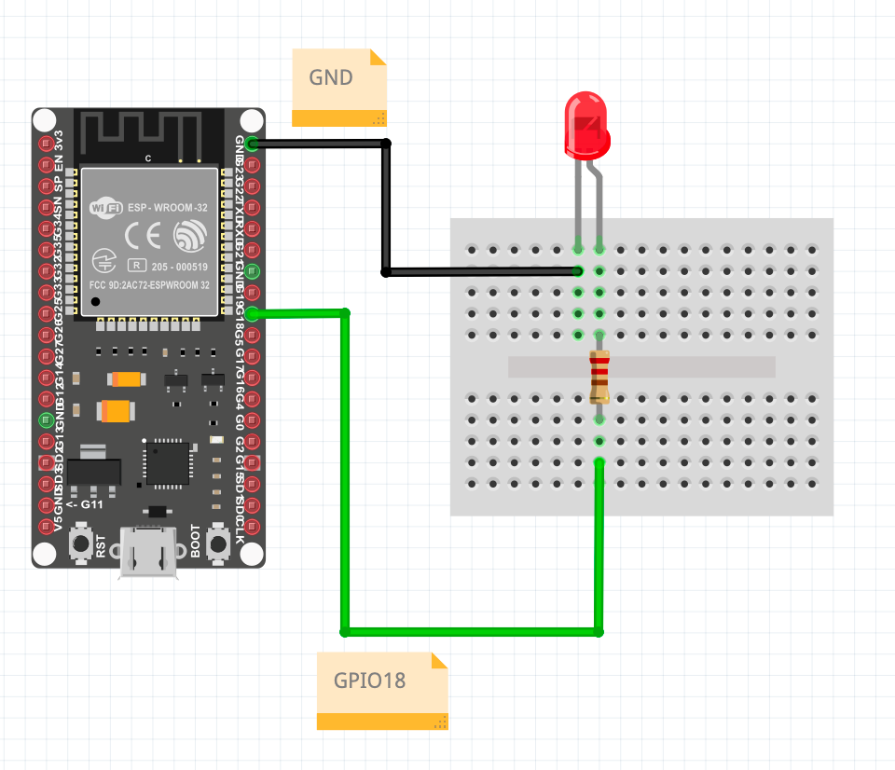


Add “Resistor” also. To modify the value of resistance, go to “diagram.json” tab.

The default value of resistance is in “Ohms”. For LED blinking exercise, we need 220 Ohms resistor. So, change its value to 220.



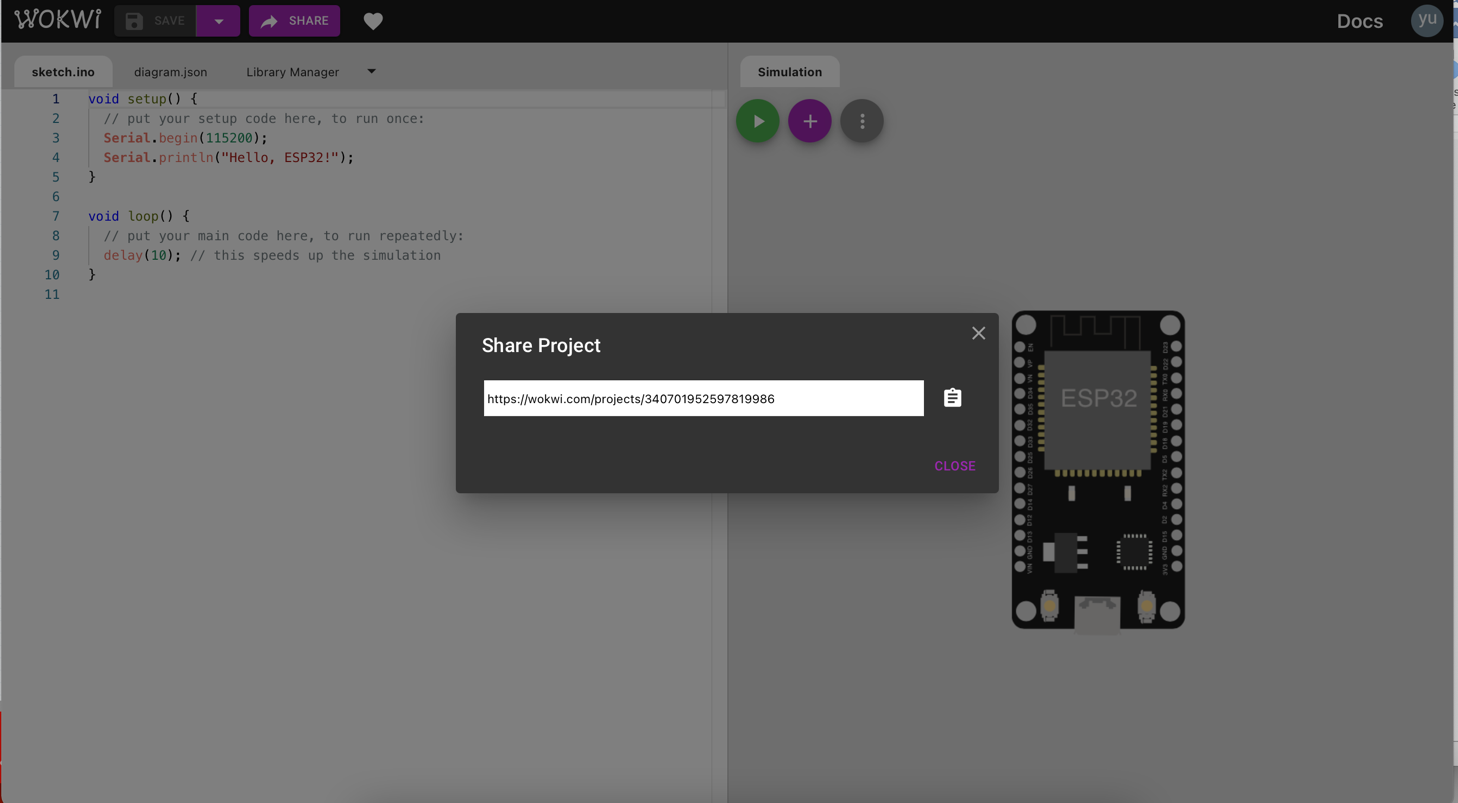
Connect ESP32, LED and resistor according to the following schematic.



Here is the code for LED blinking exercise.

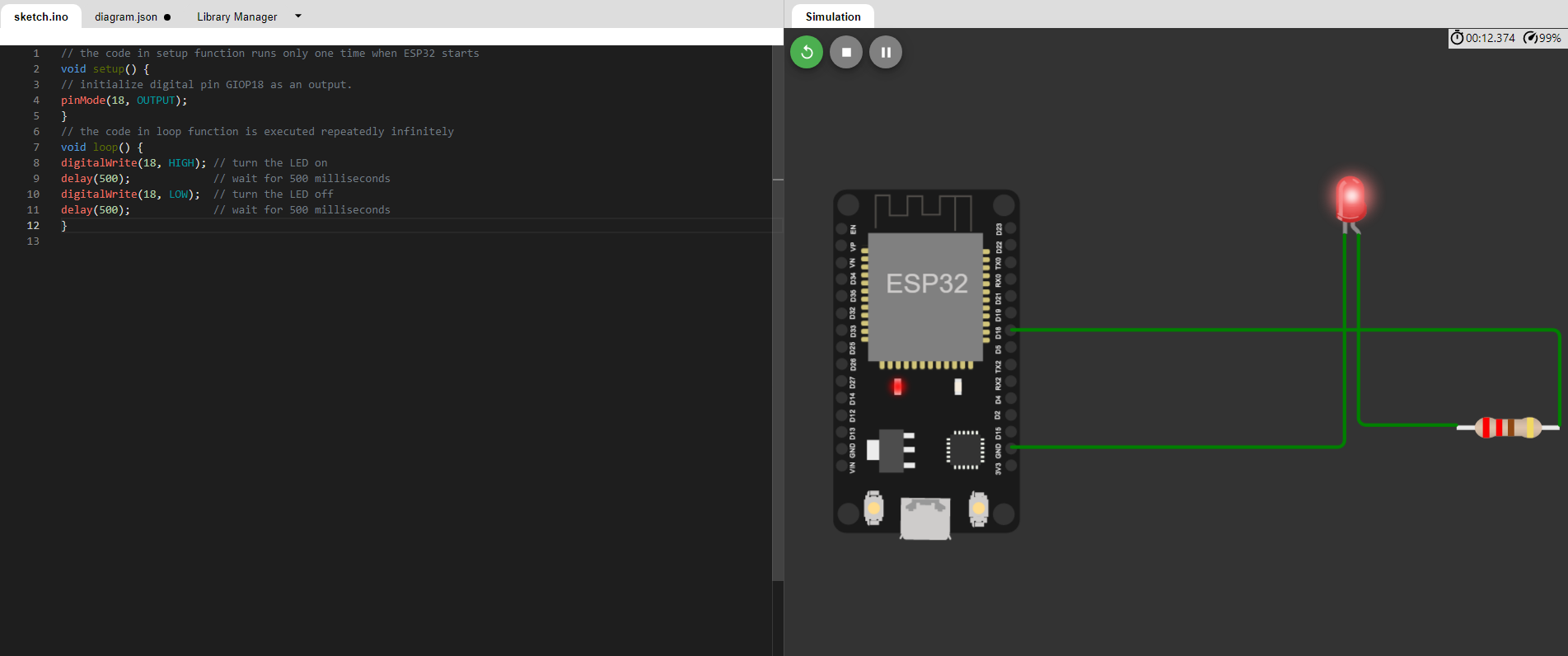
1. // the code in setup function runs only one time when ESP32 starts
2. void setup() {
3. // initialize digital pin GIOP18 as an output.
4. pinMode(18, OUTPUT);
5. }
6. // the code in loop function is executed repeatedly infinitely
7. void loop() {
8. digitalWrite(18, HIGH); // turn the LED on
9. delay(500); // wait for 500 milliseconds
10. digitalWrite(18, LOW); // turn the LED off
11. delay(500); // wait for 500 milliseconds
12. }

To share your Wokwi project, save  your project first and then you can share by clicking this share button .



**Exercise 3**

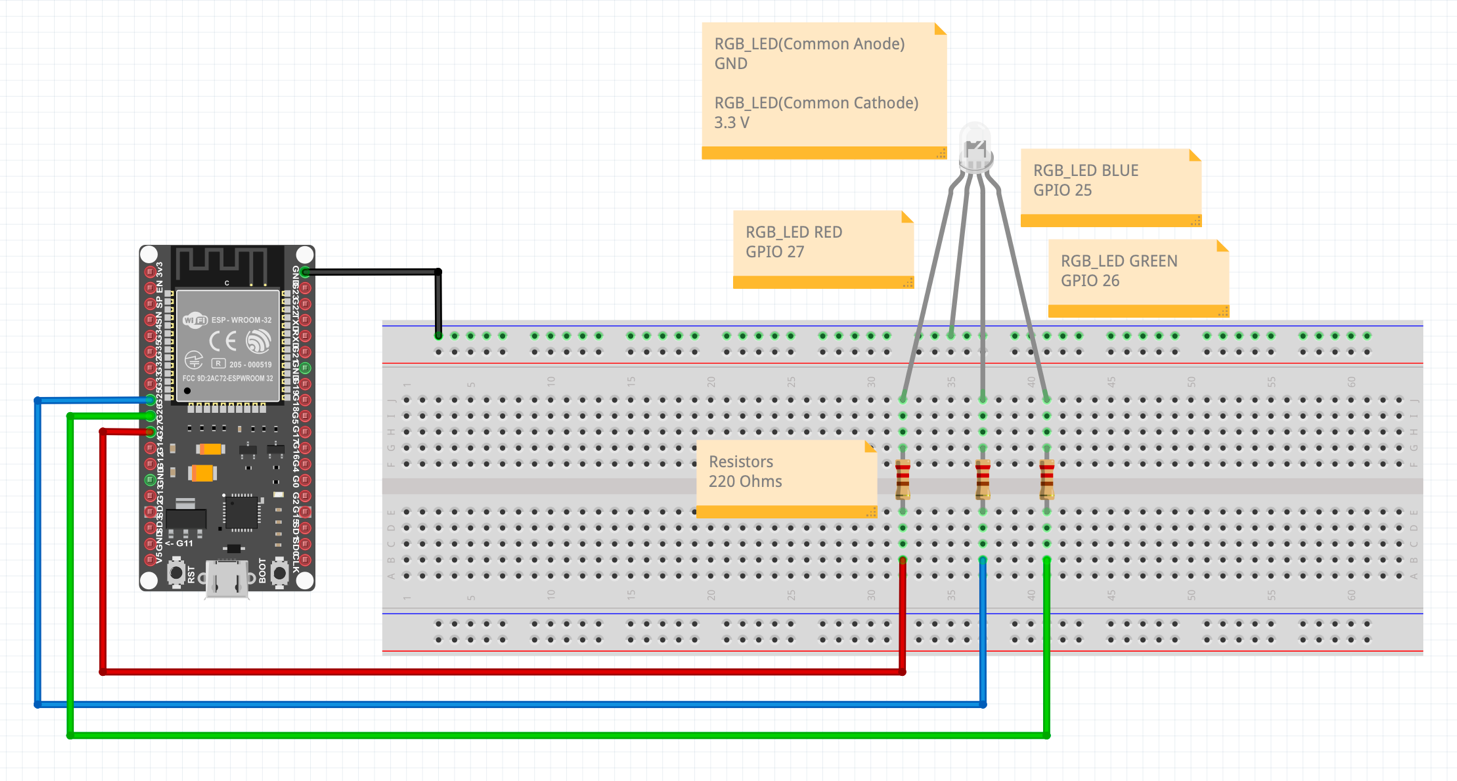
Your Wokwi project link for “LED blinking”.



https://wokwi.com/projects/340784110368719444

**RGB-LED blinking**

Circuit schematic



Code

1. #define PIN\_RED 27 // GIOP27
2. #define PIN\_GREEN 26 // GIOP26
3. #define PIN\_BLUE 25 // GIOP25
4. void setup() {
5. pinMode(PIN\_RED, OUTPUT);
6. pinMode(PIN\_GREEN, OUTPUT);
7. pinMode(PIN\_BLUE, OUTPUT);
8. }
9. void loop() {
10. // color code #00C9CC (R = 0, G = 201, B = 204)
11. analogWrite(PIN\_RED, 0);
12. analogWrite(PIN\_GREEN, 201);
13. analogWrite(PIN\_BLUE, 204);
14. delay(1000); // keep the color 1 second
15. // color code #F7788A (R = 247, G = 120, B = 138)
16. analogWrite(PIN\_RED, 247);
17. analogWrite(PIN\_GREEN, 120);
18. analogWrite(PIN\_BLUE, 138);
19. delay(1000); // keep the color 1 second
20. // color code #34A853 (R = 52, G = 168, B = 83)
21. analogWrite(PIN\_RED, 52);
22. analogWrite(PIN\_GREEN, 168);
23. analogWrite(PIN\_BLUE, 83);
24. delay(1000); // keep the color 1 second
25. }

**Exercise 4**

Your Wokwi project link for “RGB-LED blinking”.

Graphical user interface

Description automatically generated

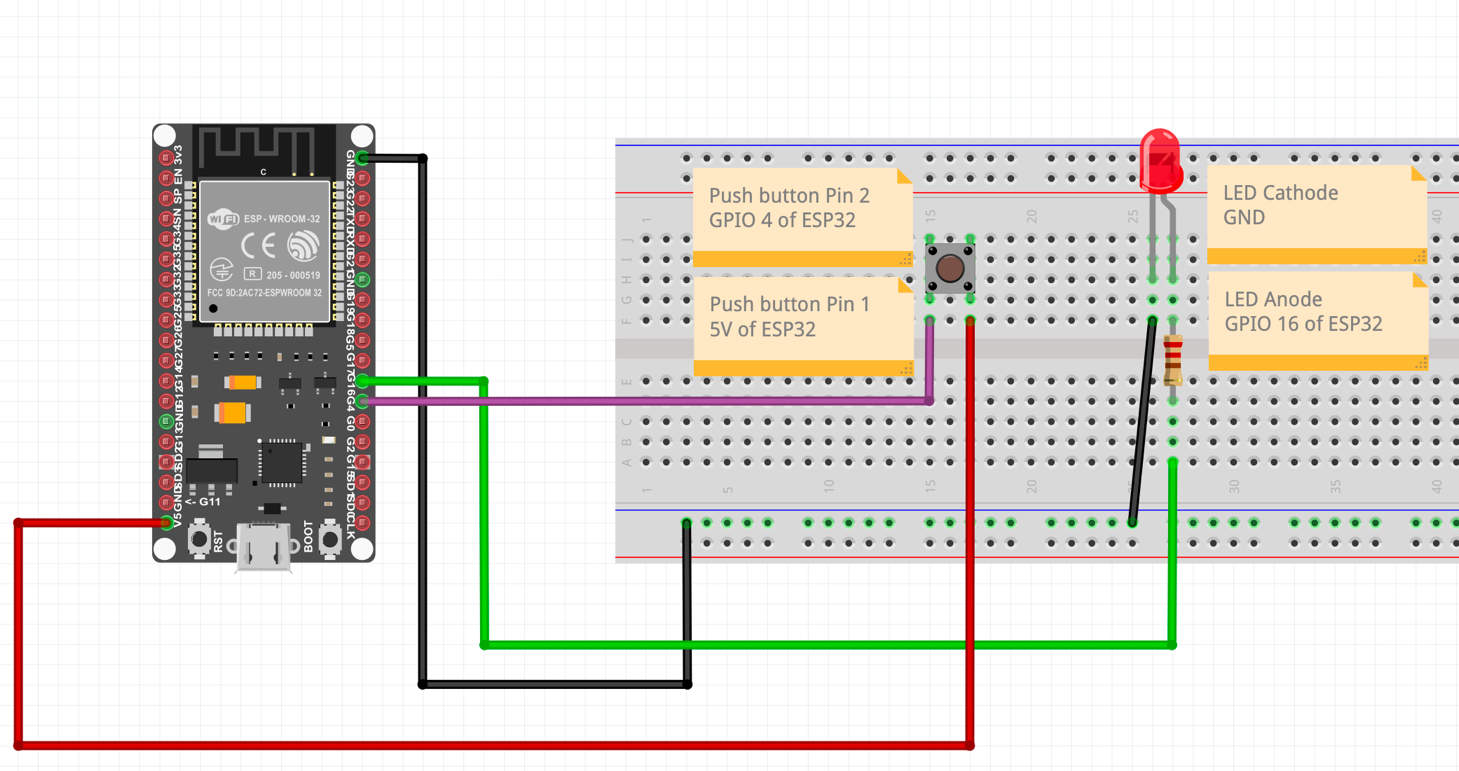
https://wokwi.com/projects/340786905469682260

**Remark**: Color code R,G,B values are used from W3 school tutorial

<https://www.w3schools.com/colors/colors_hexadecimal.asp>. You can also change or add more colours.

**Pull up/ Pull down**

Circuit schematic



Code

1. #include <Toggle.h>
2. const byte buttonPin = 4;
3. const byte ledPin = 16;
4. const unsigned int blinkMs = 100; // blink duration (ms)
5. const byte blinkMode = 1; // 0: on change, 1: on press (default), 2: on release
6. Toggle button(buttonPin);
7. void setup() {
8. button.begin(buttonPin);
9. button.setInputInvert(1);
10. pinMode(buttonPin, INPUT\_PULLDOWN);
11. pinMode(ledPin, OUTPUT);
12. }
13. void loop() {
14. button.poll();
15. digitalWrite(ledPin, button.blink(blinkMs, blinkMode));
16. delay(1);
17. }

**Exercise 5**

What will happen, if we change “INPUT\_PULLDOWN” to “INPUT\_PULLUP” at Line 10? How about if we change “INPUT\_PULLDOWN” to “INPUT” at Line 10 also? Explain your findings. Your Wokwi project link for “Pull up/ Pull down”.

https://wokwi.com/projects/340787276766249555

Graphical user interface

Description automatically generated

Ans: When change to Pullup, nothing happens but if change to input it will light. Since, input must be on click as the standard.

**Remark**: Not all GPIOs on an ESP32 have pullup and pulldown resistors.

**References**

<https://www.hackster.io/Hack-star-Arduino/esp32-online-systems-simulator-you-were-looking-for-2022-327d42>

<https://docs.wokwi.com/guides/esp32>

<https://esp32io.com/tutorials/esp32-rgb-led>

<https://esp32io.com/tutorials/esp32-led-blink>

<https://www.w3schools.com/colors/colors_hexadecimal.asp>

<https://github.com/Dlloydev/Toggle>