**Praktik Pengendalian LED dengan menggunakan 3 Input Button di Wokwi dengan Bahasa Pemograman Arduino di Visual Studio Code**

*Muhammad Fau Zan Sabani1*

*Fakultas Vokasi, Universitas Brawijaya*

*Email:* [*mfauzan18@student.ub.ac.id*](mailto:mfauzan18@student.ub.ac.id)

# Abstrak:

# Pada laporan praktikum ini dibuat bertujuan untuk menjelaskan dan membahas pembuatan sistem pengendalian LED berbasis Arduino dengan tiga tombol sebagai input. Tujuan dari proyek ini adalah untuk mengimplementasikan kontrol sederhana terhadap tiga buah lampu LED dengan pola kedipan yang berbeda berdasarkan tombol yang ditekan. Dengan semestinya pemograman Arduino digunakan sebagai pengendali utama yang mengatur pola nyala dan kedipan LED sesuai dengan logika yang telah diprogram sebelumnya.

Uji coba praktikum dimulai dengan pembuatan kode Arduino yang dirancang untuk mengatur respons setiap tombol. Tombol pertama memicu kedipan lampu merah sebanyak lima kali, tombol kedua mengaktifkan kedipan bergantian antara lampu merah dan hijau, sementara tombol ketiga memicu kedipan bergantian di antara lampu merah, kuning, dan hijau. Setiap output LED diuji dengan skenario yang telah ditentukan, dan perilaku sistem diamati melalui simulasi dan pengujian rangkaian fisik.

Hasil utama dari eksperimen ini menunjukkan bahwa sistem berhasil berfungsi sesuai dengan perancangan, di mana setiap tombol dapat mengaktifkan pola kedipan lampu LED yang berbeda. Proyek ini menunjukkan potensi aplikasi kontrol sederhana dalam pengendalian perangkat elektronik berbasis input digital, yang dapat diterapkan dalam berbagai skenario otomatisasi dasar.

Kata Kunci *LED, Tombol, IoT, Arduino, ESP32, Wokwi, Laporan Praktikum*

# Abstract:

This practicum report aims to explain and discuss the creation of an Arduino-based LED control system with three buttons as input. The purpose of this project is to implement simple control of three LED lights with different flashing patterns based on the button pressed. Arduino programming should be used as the main controller that regulates the LED flashing pattern according to the pre-programmed logic.

The practical trial begins with the creation of Arduino code designed to regulate the response of each button. The first button triggers the red light to blink five times, the second button activates alternating blinks between red and green lights, while the third button triggers alternating blinks between red, yellow, and green lights. Each LED output was tested with predefined scenarios, and the system behavior was observed through simulation and physical circuit testing.

The main results of this experiment show that the system successfully functions as designed, where each button can activate a different LED flashing pattern. This project demonstrates the potential of simple control applications in digital input-based control of electronic devices, which can be applied in various basic automation scenarios.

Keywords *LED, Button, IoT, Arduino, ESP32, Wokwi, Practical Report*

# 1. Pendahuluan

* 1. **Latar Belakang**

Pengendalian perangkat elektronik, seperti lampu LED, menjadi topik yang umum namun penting dalam berbagai aplikasi teknologi modern, mulai dari sistem otomatisasi rumah hingga sistem peringatan visual. Dalam konteks pembelajaran dasar mikrokontroler, pengendalian lampu LED menggunakan tombol sebagai input merupakan salah satu cara efektif untuk memahami konsep logika dasar dan interaksi antara perangkat input dan output. Sistem ini dapat memberikan gambaran awal tentang bagaimana sebuah perangkat digital menginterpretasikan masukan dan menghasilkan keluaran sesuai dengan algoritma yang telah diprogram.

Arduino memberikan fleksibilitas dalam pengendalian perangkat elektronik melalui pemrograman sederhana, memungkinkan pengguna untuk mengimplementasikan berbagai fungsi otomatisasi dalam kehidupan sehari-hari. Dalam eksperimen ini, penggunaan tiga tombol untuk mengendalikan tiga lampu LED dengan pola kedipan yang berbeda memberikan dasar pemahaman tentang prinsip pengendalian logika digital.

Selain itu, sistem ini juga berpotensi menjadi pondasi bagi aplikasi yang lebih kompleks, seperti sistem peringatan visual atau bahkan penerapan dalam perangkat IoT (Internet of Things). Dengan memahami bagaimana mikrokontroler bekerja untuk mengatur output berdasarkan input tertentu. Oleh karena itu, percobaan ini menjadi langkah awal yang penting dalam memahami konsep dasar pengendalian perangkat elektronik berbasis mikrokontroler, serta membuka peluang untuk pengembangan lebih lanjut di masa depan.

* 1. **Tujuan Eksperimen**

Tujuan dari eksperimen simulasi traffic light menggunakan Arduino dan Wokwi ini adalah sebagai berikut:

* Merancang dan mengimplementasikan sistem pengendalian tiga LED menggunakan tiga tombol input untuk menghasilkan pola kedipan yang berbeda sesuai dengan tombol yang ditekan.
* Mengembangkan keterampilan dalam memprogram Arduino untuk mengendalikan output digital berdasarkan input dari tombol, dengan logika dan alur kontrol yang tepat.
* Memahami cara kerja dasar mikrokontroler dalam mengelola perangkat input (tombol) dan output (LED) untuk menciptakan pola interaksi yang terstruktur.
* Menguji kemampuan sistem dalam merespons masukan secara real-time dan memastikan bahwa LED berkedip sesuai dengan pola yang telah ditentukan berdasarkan logika pemrograman.
* Menyediakan landasan pengetahuan untuk pengembangan lebih lanjut dalam aplikasi mikrokontroler dan sistem otomatisasi sederhana, serta membangun pemahaman yang kuat dalam mengontrol perangkat elektronik melalui pemrograman dasar.

# 2. Metodologi

* 1. **Alat dan Bahan**

Dalam perancangannya, untuk membuat simulasi pengendalian LED dengan Button berbasis IoT menggunakan Wokwi dan Arduino dibutuhkan beberapa alat dan bahan, diantaranya:

* 1 buah mikrokontroller ESP32,
* 3 buah lampu LED (dengan warna diubah ke warna merah, kuning, dan hijau),
* 3 buah Tombol (Button),
* 3 buah resistor,
* Platform Wokwi, yang digunakan untuk simulasi rangkaian dan pemrograman secara virtual.
* Software Arduino IDE/Extension PlatformIo (jika menggunakan Visual Studio Code).
  1. **Langkah Implementasi**

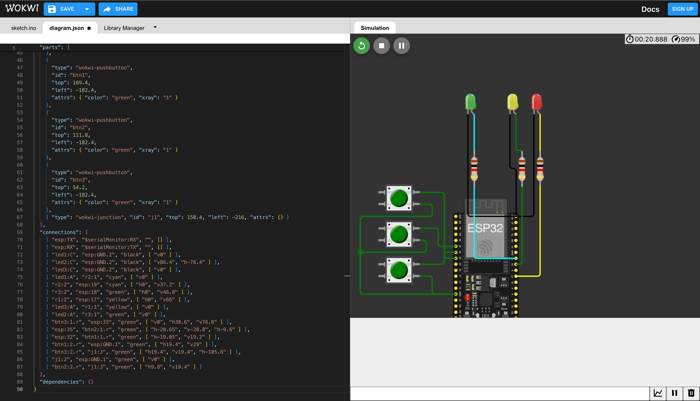
Berikut langkah-langkah implementasi simulasi lampu lalu lintas dengan menggunakan Mikrokontroler ESP32 di Wokwi/Arduino:

* Install Arduino IDE atau jika menggunakan VsCode install Extension PlatfromIO di bagian ekstensinya. Buka juga website wokwi untuk tampilan simulasi nya secara virtual (jangan lupa untuk install ekstensi Wokwi juga di VsCode).



* Buat proyek baru di Wokwi dengan memilih ESP32 sebagai mikrokontroler. Tambahkan juga komponen LED sebanyak 3 buah (atur setiap LED sesuai seperti warna merah, kuning, dan hijau) dan tambahkan juga 3 buah resistor pada layout Wokwi, serta hubungkan dengan sisi kiri lampu LED ke pin Ground (GND) dan dari sisi kanan hubungkan ke resistor lalu menyalur ke ESP32 menggunakan kabel jumper virtual. Lalu hubungkan juga button ke mikrokontroller ESP32.

A screenshot of a computer

Description automatically generated

* Untuk pemogramannya menggunakan bahasa Arduino (C/C++) di Arduino IDE/PlatformIO (bisa juga langsung dimasukkan ke file ‘main.c’ di website Wokwi). Kodenya kita atur untuk momen ketika kita memencet tombol 1 nyala lampu merah akan berkedip sebanyak lima kali, lalu untuk tombol kedua ketika di pencet maka akan menyalakan lampu merah dan hijau secara bergantian, dan untuk yang terakhir Ketika kita memencet tombol ketiga maka lampu merah, kuning dan juga hijau akan menyala secara bergantian. Seperti berikut kode pemogramannya:

#include <Arduino.h>

// Deklarasi pin tombol

const int tombol1 = 33;

const int tombol2 = 35;

const int tombol3 = 32;

// Deklarasi pin LED

const int lampuMerah = 17;

const int lampuKuning = 18;

const int lampuHijau = 19;

void setup() {

Serial.begin(115200);

Serial.println("Program LED dan Tombol ESP32");

// Konfigurasi tombol sebagai input dengan pull-up internal

pinMode(tombol1, INPUT\_PULLUP);

pinMode(tombol2, INPUT\_PULLUP);

pinMode(tombol3, INPUT\_PULLUP);

// Konfigurasi LED sebagai output

pinMode(lampuMerah, OUTPUT);

pinMode(lampuKuning, OUTPUT);

pinMode(lampuHijau, OUTPUT);

}

void loop() {

if (digitalRead(tombol1) == LOW) {

// Tombol 1: Lampu merah berkedip 5 kali

for (int i = 0; i < 5; i++) {

digitalWrite(lampuMerah, HIGH);

Serial.println("Merah Menyala");

delay(500);

digitalWrite(lampuMerah, LOW);

Serial.println("Merah Padam");

delay(500);

}

}

if (digitalRead(tombol2) == LOW) {

// Tombol 2: Lampu merah dan hijau berkedip bergantian

for (int i = 0; i < 5; i++) {

digitalWrite(lampuMerah, HIGH);

digitalWrite(lampuHijau, LOW);

Serial.println("Merah Menyala");

delay(500);

digitalWrite(lampuMerah, LOW);

digitalWrite(lampuHijau, HIGH);

Serial.println("Hijau Menyala");

delay(500);

}

digitalWrite(lampuHijau, LOW); // Matikan LED setelah selesai

}

if (digitalRead(tombol3) == LOW) {

// Tombol 3: Lampu merah, kuning, dan hijau berkedip bergantian

for (int i = 0; i < 5; i++) {

digitalWrite(lampuMerah, HIGH);

Serial.println("Merah Menyala");

delay(500);

digitalWrite(lampuMerah, LOW);

digitalWrite(lampuKuning, HIGH);

Serial.println("Kuning Menyala");

delay(500);

digitalWrite(lampuKuning, LOW);

digitalWrite(lampuHijau, HIGH);

Serial.println("Hijau Menyala");

delay(500);

digitalWrite(lampuHijau, LOW);

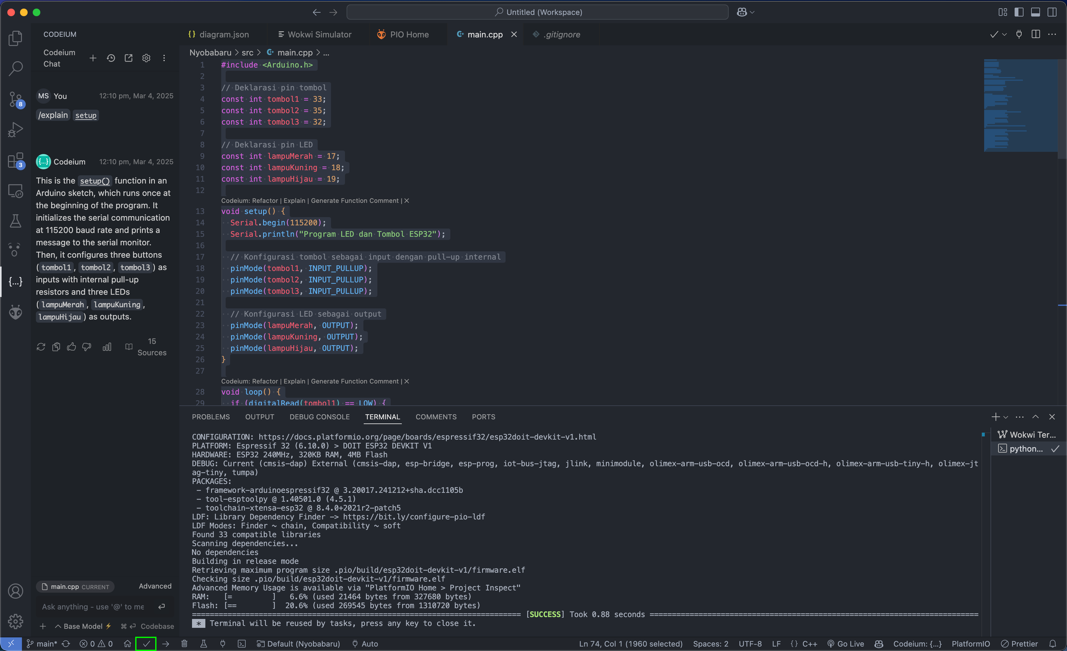
}

}

}

}

* Jika di platform Wokwi file ‘main.c’ tidak bisa ter-compile/dijalankan maka kita bisa menguji apakah sukses atau tidak program kode tersebut di PlatformIO dengan menggunakan tombol centang dibawah.



* Jika kode di file ‘main.cpp’ sukses di *Build* maka langkah selanjutnya untuk dapat melakukan virtualisasi simulasi lampu merah adalah membuat file bernama ‘diagram’ yang nanti diisi oleh kode dari file ‘diagram.json’ di projek Wokwi sebelumnya. Lalu kalau sudah, ubah nama file ‘diagram’ sbelumnya menjadi ‘diagram.json’ seperti di platform Wokwi sebelumnya.

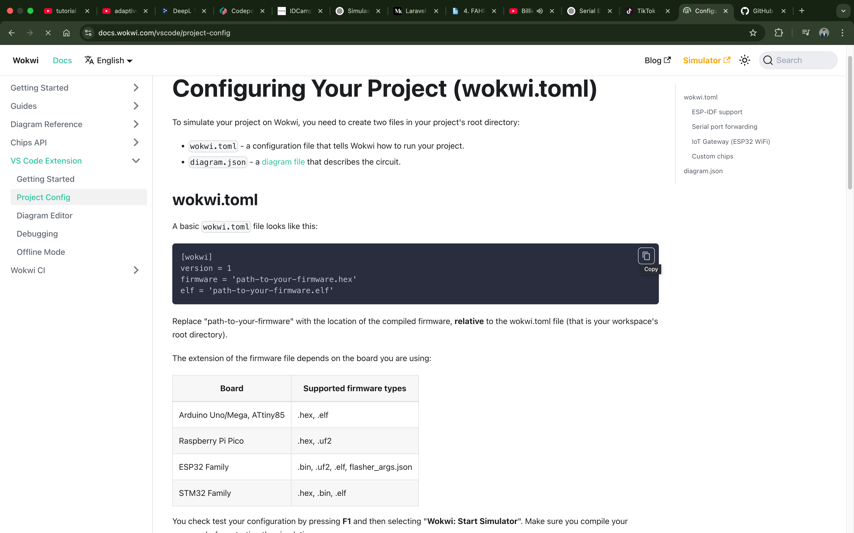
A screenshot of a computer

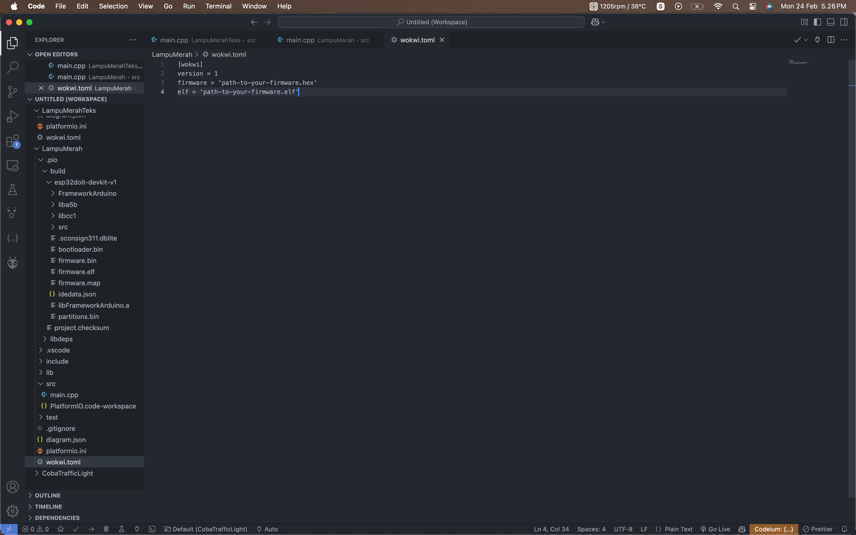
Description automatically generated

A screenshot of a computer program

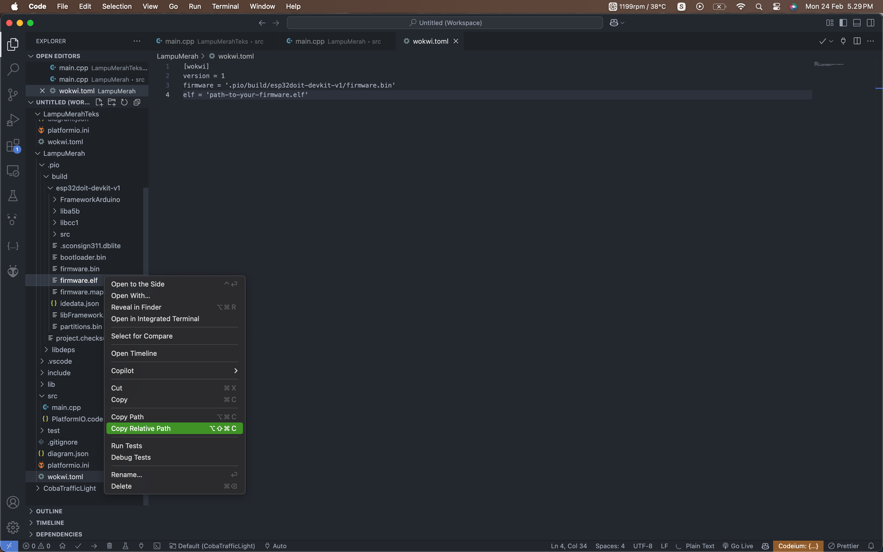
Description automatically generated

* Untuk dapat menjalankan simulasi dari file ‘diagram.json’ tersebut maka kita harus menambahkan file yang kita namakan ‘wokwi.toml’. Langkah berikutnya cari di web browser dengan keyword ‘Wokwi.toml’ dan salin kode konfigurasinya yang berasal dari halaman web dokumentasi wokwi tersebut ke file ‘wokwi.toml’ yang sudah kita buat.

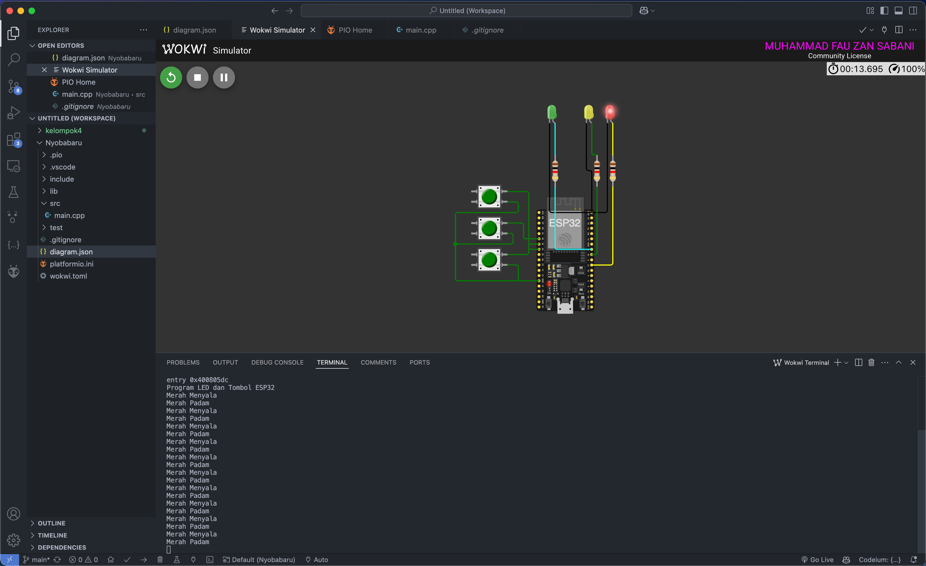




* Langkah terakhir untuk dapat mengkonfigurasi file ‘wokwi.toml’ tersebut agar dapat membantu menjalankan file diagramnya maka kita perlu untuk mengganti path ‘firmware’ dan ‘elf’ path file yang berada di dalam rute folder ‘pio/build/esp32doit-devkit-v1’. Salin tiap file dengan format file yang bertuliskan ‘.hex’ dan ‘.elf’ dengan menggunakan klik kanan dan klik ‘Copy relative path’ lalu tempel ke path yang harus diisi di file ‘wokwi.toml’.



* Jalankan ulang file ‘diagram.json’ untuk dapat melihat hasil akhir simulasi yang sudah kita buat.

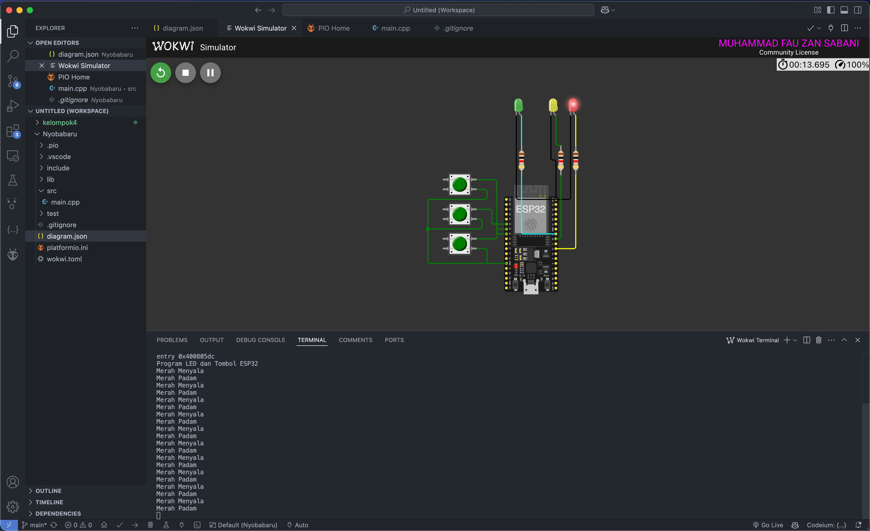


# 3. Hasil dan Pembahasan

1. **Hasil Eksperimen**

Dari simulasi yang sudah dibuat, di dapatkan hasil yang sudah sesuai dengan durasi yang kita inginkan. Berikut hasil simulasinya:

* Untuk tombol pertama Ketika kita klik maka lampu merah:



* Untuk tombol kedua di klik maka lampu merah dan hijau akan menyala secara bergantian:

A screenshot of a computer

Description automatically generated

* Untuk tombol ketiga Ketika di klik maka ketiga lampu akan menyala namun secara bergantian:

A screenshot of a computer

Description automatically generated

# 4. Lampiran

* Kode Program diagram.json:

{

"version": 1,

"author": "Anonymous maker",

"editor": "wokwi",

"parts": [

{ "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 76.8, "left": -62.36, "attrs": {} },

{

"type": "wokwi-led",

"id": "led1",

"top": -99.6,

"left": -53.8,

"attrs": { "color": "limegreen" }

},

{

"type": "wokwi-led",

"id": "led2",

"top": -99.6,

"left": 13.4,

"attrs": { "color": "yellow" }

},

{ "type": "wokwi-led", "id": "led3", "top": -99.6, "left": 51.8, "attrs": { "color": "red" } },

{

"type": "wokwi-resistor",

"id": "r1",

"top": 24,

"left": 47.45,

"rotate": 90,

"attrs": { "value": "1000" }

},

{

"type": "wokwi-resistor",

"id": "r2",

"top": 24,

"left": -58.15,

"rotate": 90,

"attrs": { "value": "1000" }

},

{

"type": "wokwi-resistor",

"id": "r3",

"top": 24,

"left": 18.65,

"rotate": 90,

"attrs": { "value": "1000" }

},

{

"type": "wokwi-pushbutton",

"id": "btn1",

"top": 169.4,

"left": -182.4,

"attrs": { "color": "green", "xray": "1" }

},

{

"type": "wokwi-pushbutton",

"id": "btn2",

"top": 111.8,

"left": -182.4,

"attrs": { "color": "green", "xray": "1" }

},

{

"type": "wokwi-pushbutton",

"id": "btn3",

"top": 54.2,

"left": -182.4,

"attrs": { "color": "green", "xray": "1" }

},

{ "type": "wokwi-junction", "id": "j1", "top": 158.4, "left": -216, "attrs": {} }

],

"connections": [

[ "esp:TX", "$serialMonitor:RX", "", [] ],

[ "esp:RX", "$serialMonitor:TX", "", [] ],

[ "led1:C", "esp:GND.2", "black", [ "v0" ] ],

[ "led2:C", "esp:GND.2", "black", [ "v86.4", "h-76.4" ] ],

[ "led3:C", "esp:GND.2", "black", [ "v0" ] ],

[ "led1:A", "r2:1", "cyan", [ "v0" ] ],

[ "r2:2", "esp:19", "cyan", [ "h0", "v37.2" ] ],

[ "r3:2", "esp:18", "green", [ "h0", "v46.8" ] ],

[ "r1:2", "esp:17", "yellow", [ "h0", "v66" ] ],

[ "led3:A", "r1:1", "yellow", [ "v0" ] ],

[ "led2:A", "r3:1", "green", [ "v0" ] ],

[ "btn3:1.r", "esp:33", "green", [ "v0", "h38.6", "v76.8" ] ],

[ "esp:35", "btn2:1.r", "green", [ "h-28.65", "v-28.8", "h-9.6" ] ],

[ "esp:32", "btn1:1.r", "green", [ "h-19.05", "v19.2" ] ],

[ "btn1:2.r", "esp:GND.1", "green", [ "h19.4", "v29" ] ],

[ "btn3:2.r", "j1:J", "green", [ "h19.4", "v19.4", "h-105.6" ] ],

[ "j1:J", "esp:GND.1", "green", [ "v0" ] ],

[ "btn2:2.r", "j1:J", "green", [ "h9.8", "v19.4" ] ]

],

"dependencies": {}

}