**Practice Simulation LED Traffic Light on ESP32 using Wokwi and Visual Studio Code**

*Andreika Luna Alghivari*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: andreikaluna@student.ub.ac.id*

**Abstract**

Traffic lights are an important element in the transportation system to regulate vehicle flow and improve road safety. This research aims to design and implement an ESP32-based traffic light system with simulation using Wokwi and code testing through Visual Studio Code (VS Code). ESP32 was chosen due to its capability in data processing and wide connectivity. Software development is done using Arduino programming language and tested using VS Code with PlatformIO extension for debugging as well as Wokwi simulation for validation before physical implementation. The system uses LEDs as red, yellow, and green light indicators with color change logic controlled through programmed code. The test results show that the system works according to the predetermined scenario, including the duration of color changes that comply with traffic rules. The combination of ESP32, Wokwi, and VS Code provides efficiency in development as well as flexibility in testing before implementation on physical devices.

Keywords: Traffic light, ESP32, Wokwi, Visual Studio Code, PlatformIO, Arduino.

**1. Introduction**

* 1. **Background**

Traffic lights are an important component in modern transportation systems that function to regulate the flow of vehicles and pedestrians to improve traffic safety and efficiency. Along with the development of technology, traffic light systems are increasingly sophisticated and can be controlled automatically using microcontrollers. ESP32 is one of the microcontrollers that is often used in IoT-based projects because it has good processing capabilities as well as WiFi and Bluetooth connectivity support.

In the process of developing an ESP32-based traffic light system, simulation and testing are required before physical implementation to ensure that the system works according to the scenario that has been designed. Wokwi is a web-based simulator that allows users to test code and hardware designs virtually, thus saving costs and time in system development. In addition, the use of Visual Studio Code (VS Code) with PlatformIO or Arduino IDE extensions provides flexibility in writing, editing, and debugging program code before uploading to the ESP32.

With this simulation, developers can understand how microcontroller-based traffic lights work, test light color change algorithms according to traffic rules, and optimize code before implementation on physical devices.

* 1. **Objective**

1. Implement a traffic light system using ESP32.
2. Write and test the program code for the traffic light in VS Code.
3. Ensure the system works according to the traffic light color change rules.

**2. Methodology**

**2.1 Tools & Materials**

Wokwi, ESP32, Arduino IDE, and Visual Studio Code.

**2.2 Implementation Steps**

1. Create new project in Wokwi
2. Select ESP32 and create a traffic light by adding 3 red yellow green leds.
3. Change the main.c code with the code that is already available in the module.
4. There are 2 important files that will be used (firmware.bin and firmware.elf files) then copy the relative path of each file into the wokwi.toml file
5. Create a diagram.json file then copy paste the code from the json diagram on the wokwi
6. Start simulation

**3. Results and Discussion**

**3.1 Experimental Results**

The ESP32-based traffic light system simulation using Wokwi and tested in Visual Studio Code has been successfully run. The system can manage red, yellow, and green light color changes according to predetermined traffic rules. The use of Wokwi as a simulator allows testing to be done without the need for hardware, thus speeding up the debugging and code optimization process. Thus, this simulation proves that the ESP32 can be effectively used to control traffic lights, and the results can be used as a basis for implementation on physical devices.

1. Main.cpp

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

1. Diagram.json

A computer screen shot of a computer program

Description automatically generated