



# PROJECT II – VIRTUAL ENGINE LAUNCH

## Introduction

This project primarily serves to assess your skills, though for most of you, it will be a formality. At the same time, it's not just a "shelf" project; we will be actively using it in the team. You might already be familiar with the ESP platform—it's essentially a more powerful Arduino with built-in WiFi and Bluetooth. Later, you won't be working with them in the team, but for now, they serve as training wheels, and we don't want to scare you with STM32 IDE. Meanwhile, your colleagues, hardware specialists/novices, are working on a custom PCB that you will use later.

This document describes the project requirements (what you need to do) and its implementation (how you need to do it). The implementation section will include many links to tutorials, so even if you don't know something, don't worry—you'll learn everything. If you get stuck on something, don't hesitate to contact me (via Teams, email), and I'll be happy to help. Also, try to assist each other in the team as much as possible; we need as many team members as possible :).

Each week, plan meetings where you'll organize task fulfillment and recruit older team members for help.





## Requirements

You will inherit data from Project I, where the status of the launch briefcase is measured. Your task will be to process this data and then “launch” a rocket model. In this model, there will be an LED strip, and you will change its color and intensity according to the flame temperature of a real rocket launch.

Hardware requirements:

- ESP32
- LED strip
- Power supply unit (PSU) for the LED strip
- Holder for the LED strip in the model

## Recommended project execution

1. **IDE setup:** Your first task is to set up an IDE that works with ESP32 chips. You can use either VSCode with PlatformIO or Arduino IDE. I strongly recommend using VSCode as it will be more useful for you in the long run. Start by flashing the built-in LED. Coordinate with your team since projects can't be directly backported from PlatformIO to Arduino.
2. **Git Setup:** The second task is to set up a Git account and create a repository for your project. We'll conduct a workshop to guide you through this.
3. **LED connection:** The third task is to connect the LED strip to your ESP32. Try blinking with several colors. Try making it as bright as possible.
4. **Color gradient:** The fourth task is to program the most accurate color transition of a launching rocket flame. Try to write this as algorithmically as possible, using loops, and aim for the smoothest transition between individual colors.
5. **Data Connectivity:** The fifth task is to ensure connectivity with other projects. You will need to handle receiving data in the format that your colleagues in Project I choose. Process the data locally, and based on them, the "rocket launch" will be triggered. The data will be received via Bluetooth.
6. **Documentation:** The last task is to create documentation for the entire project. This should be a maximum of 3 A4 pages in English. I suggest you start drafting documentation during your meetings to keep things on track.



## Literature

<https://randomnerdtutorials.com/esp32-digital-inputs-outputs-arduino>

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

<https://randomnerdtutorials.com/esp32-bluetooth-low-energy-ble-arduino-ide>

<https://randomnerdtutorials.com/vs-code-platformio-ide-esp32-esp8266-arduino>

<https://randomnerdtutorials.com/installing-esp32-arduino-ide-2-0>

<https://randomnerdtutorials.com/esp32-digital-inputs-outputs-arduino>

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

<https://randomnerdtutorials.com/esp32-bluetooth-low-energy-ble-arduino-ide>