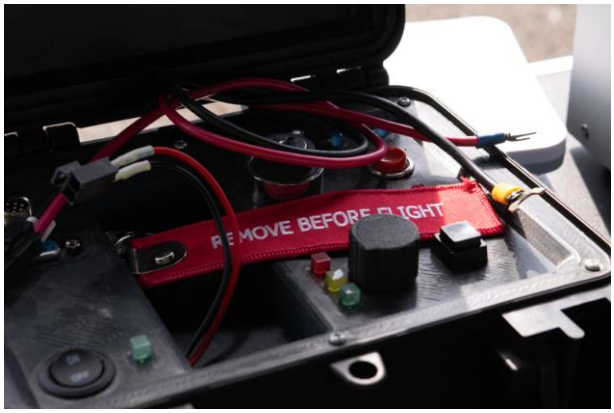
PROJECT IV. – ADVANCED LAUNCHBOX STATUS MEASUREMENT

Introduction

This project is for more advanced individuals interested in joining the team. Unlike your colleagues, you already have some experience with firmware programming, so we don't want to underestimate you with a simple project. You won't be using the ESP32 like your colleagues; instead, you'll directly try programming on the STM32. You will have access to a PCB provided by a more senior team member and will follow the same project assignment as those in Project I. By learning this platform earlier than the other newcomers, you will have a competitive advantage, but it's also expected that you will assist them when they transition to STM32.

Meanwhile, our fellow hardware engineers and novices are working on a custom PCB, which you'll eventually use in future phases.



This document outlines both the project objectives (what to do) and the project execution (how to do it). The execution section includes a number of linked tutorials, so don't worry if you're unfamiliar with certain topics—you'll have plenty of opportunities to learn. If you get stuck, feel free to reach out to me via Teams or email. I'll be happy to help. Also, support one another as much as possible; the more members we have contributing, the better. Be sure to schedule weekly meetings to organize tasks and involve senior team members for guidance.

Requirements

The goal of this project is to develop a system that collects data from our launch box and wirelessly transmits it for use in other projects. The data to be collected includes:

- Firing status (red, yellow, green) with a timestamp
- Key position
- Battery voltage
- Launch button status

Hardware Requirements:

- RocketLink Ground PCB
- Launch case: //Refer to the schematic
- Interconnection materials, resistors

Recommended project execution

1. **IDE setup:** The first task is to get the STM32 IDE up and running. There will be a seminar for this, so make sure to attend!
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. **Understanding the Launch Case:** Your third task is to understand the wiring diagram of the briefcase. Take a multimeter and the briefcase to the office, and under supervision, measure the values within the case. Identify where to measure each data point. DO NOT DISCONNECT OR MODIFY ANYTHING IN THE CASE!
4. **Project setup/ PCB familiarization:** The fourth task is to analyze the RocketLink schematic and determine which physical pins are connected to what on the processor. Then, create a project in the IDE, and using CubeMX, set up the individual pins and define what you will measure on them.
5. **Measuring data:** The fifth task is to connect your STM to the briefcase and start measuring values. The battery will need to be measured using ADC (Analog-to-Digital Converter). Be careful, as the processor pins can only handle voltage in the range of 0-3.3V. You'll need to reduce the voltage using a resistor divider before measuring. **Beware—if something burns out, you're in trouble!**
6. **Data Connectivity:** The sixth task is to ensure connectivity with other projects. The board contains an Ethernet port, and we would like to send data through it in a civilized format. This task is a bit beyond the basic scope, so if you can't manage it, we will help you.

7. **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://ohmslawcalculator.com/voltage-divider-calculator

Launchbox schematic - if the link doesnt work, message me on Teams

All other documentation can be found here

https://github.com/CTU-Space-Research/RocketLink

https://wiki.st.com/stm32mpu/wiki/How_to_get_started_with_STM32CubeIDE_from_scratch

https://wiki.st.com/stm32mcu/wiki/Getting_started_with_ADC

https://wiki.st.com/stm32mcu/wiki/Getting_started_with_GPIO

https://community.st.com/t5/stm32-mcus/how-to-create-a-project-for-stm32h7-with-ethernet-and-lwip-stack/ta-p/49308