Avionics bay design documentation

Afghan Goat, Fohurka

January 7, 2026

Contents

0.1 Avionics bay

0.1.1 Overview of the subsystem

The goal of this document is to contain the plans, procedures, all physical and digital statistics and the architecture for the avionics bay.

The main goal of the avionics is to aid the rocket towards a successful recovery. The idea is that we measure the height using a gyroscope and a barometer and after the apogee, command a main chute open.

Concrete implementations will be described in the corresponding subsystem.

0.1.2 The flight computer

The primary goal of the flight computer is to perform calculations which helps determine when to activate the recovery systems (parachute). The board operates with 9V provided by the Power Supply Unit.

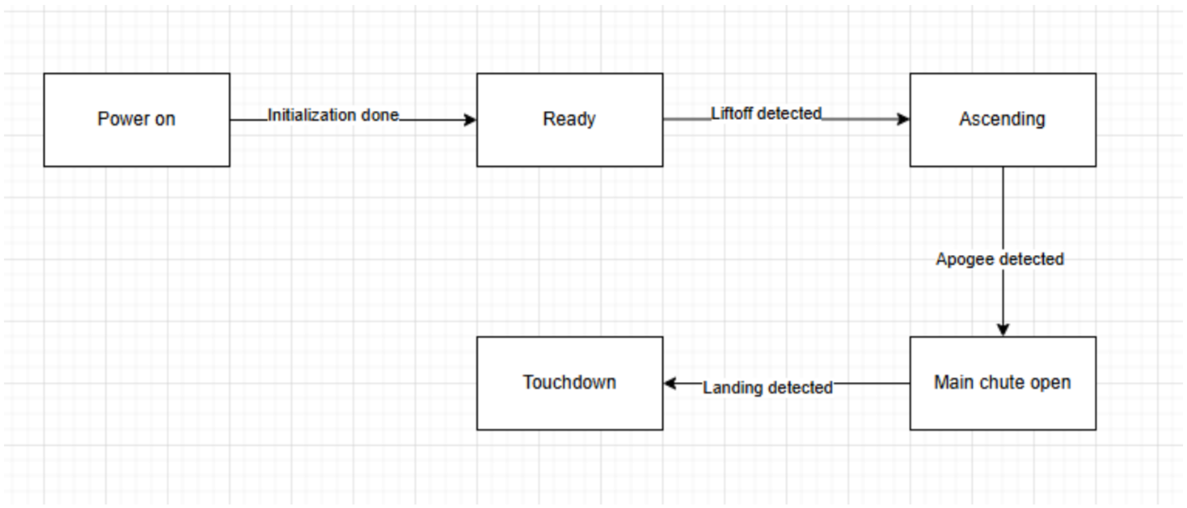

Figure 1: This diagram shows the process of a flight. The computer needs to detect the apogee in order to be able to open the main chute at the right time.

The computer determines when to open the chute.

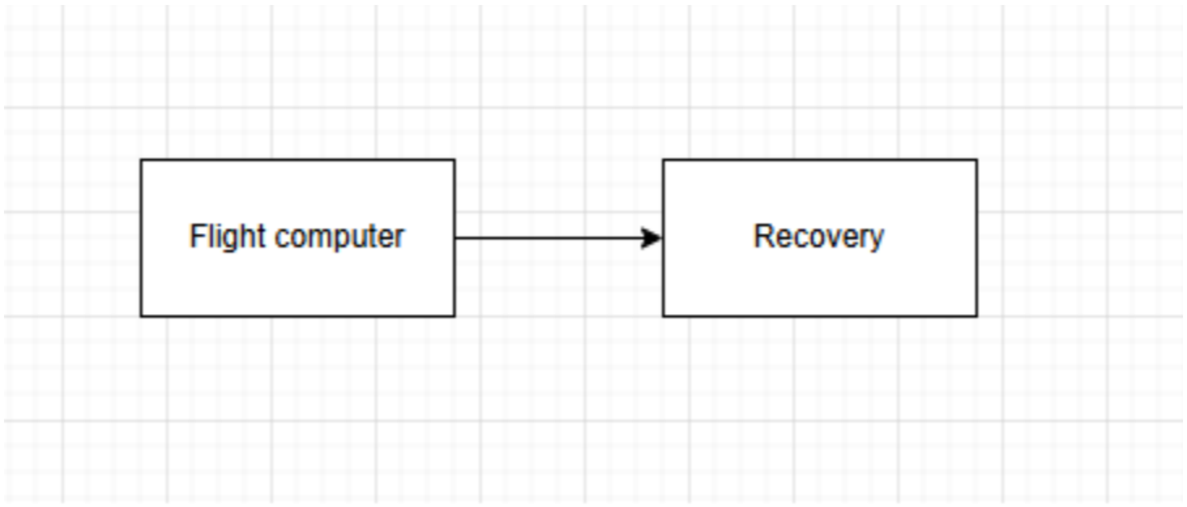

Figure 2: This diagram shows what components of the rocket the avionics logics (or the flight computer) interacts with

0.1.3 The software framework

The source code of the software can be found here: https://gitlab.com/kristof.jani06/aladeenrocket

The software does:

1. Reads the gyro,
2. Based on the reads, it opens the chute after X seconds of the apogee.

0.1.4 Design constraints

The avionics bay's diameter can not exceed the diameter of 5 centimeters.

0.1.5 Detailed design description

The avionics bay contains elements for logging barometric pressure, acceleration and systems which will activate the recovery chute, namely:

0.1.6 The schematic layout of the Avionics electrical logic

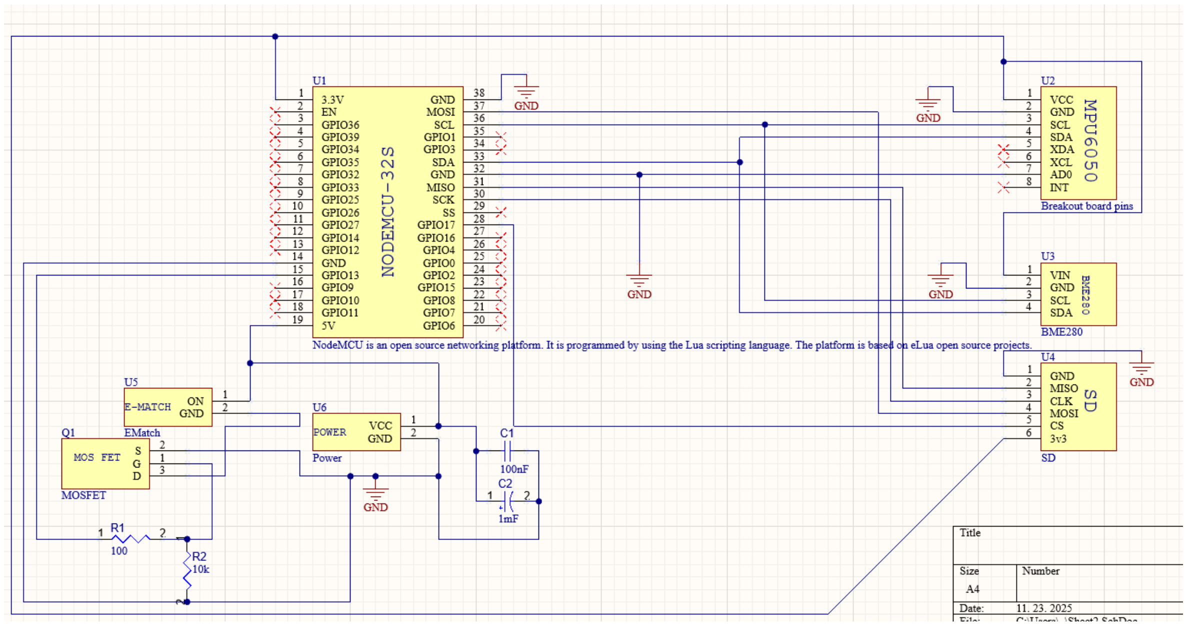

Figure 3: The electronic schematic of the avionics bay components.

We originally planned with a PCB design but then needed to switch to a test circuit design. We also made the bay compatible with the PCB and the test circuit design too. 4 holes are drilled to the circuit which will aid it in it's stability. The test circuit design consists of 2 separate test circuits held together by the ESP32 module of the hardware subsystem.

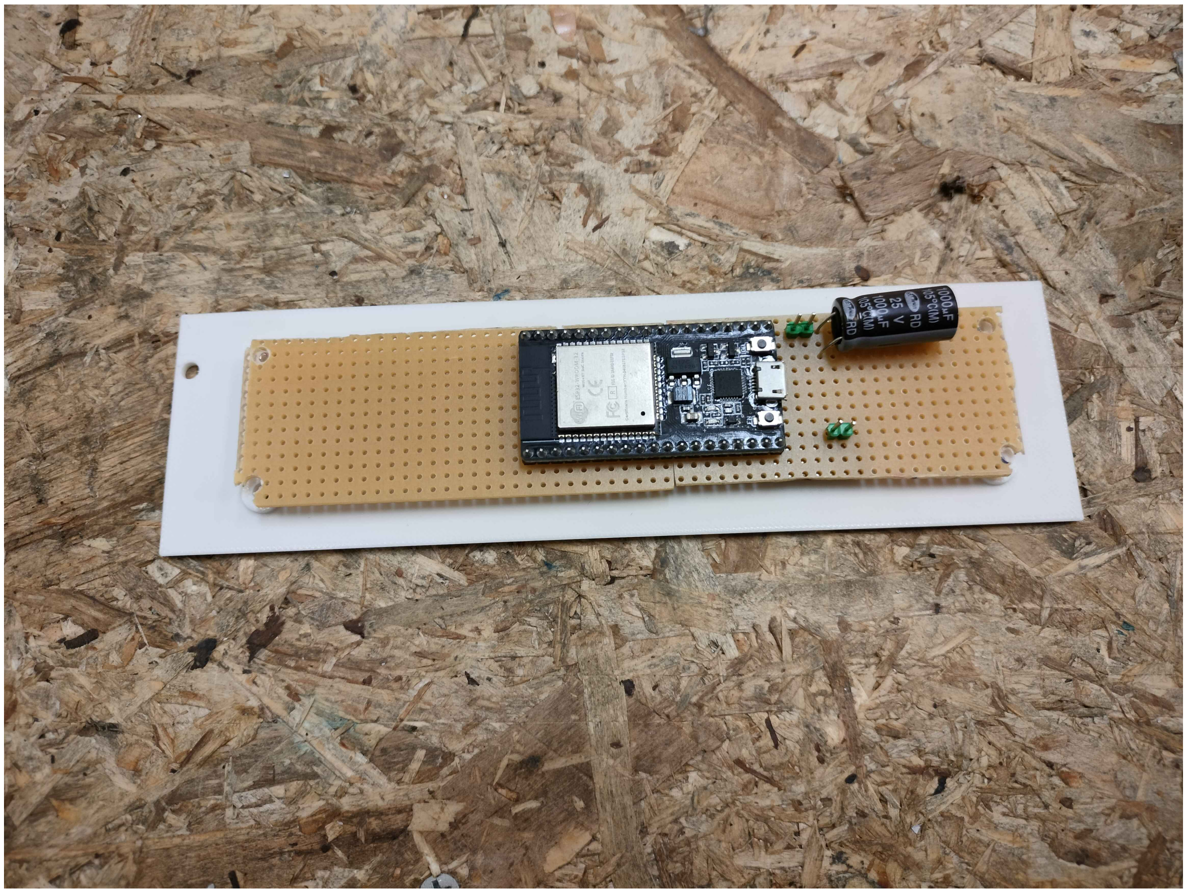

Figure 4: The layout of the test circuitry. (Originally designed with the PCB version)

Originally, the circuitry would be fixated with ruthex only but due to the lack of length of the circuitry, it would alone not be stable enough. Additional glueing should also applied to the 4 holes.

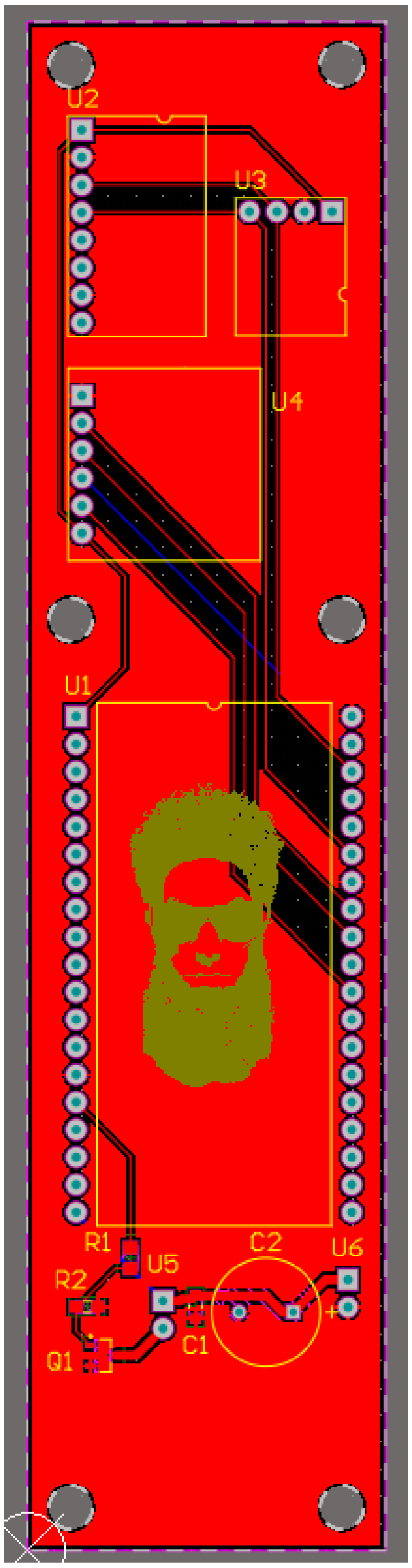

Figure 5: The layout of the PCB. (Originally designed with this)

0.1.7 General architecture

1. The avionics bay will be split by a plate and 2 pillars which will help in the stabilization of the bay.

2. The plate will have a hole, where cables can connect the two sections of the bay.
3. The plate will be marked as a green cuboid on the images and on the 3d model.
4. The pillars will be marked as red objects on the 3d model.
5. The NodeMCU-32S is visible as-is on the 3d model.
6. The BME280 is visible with a grey color on the 3d model. This part will be responsible for measuring the barometric pressure and humidity.
7. The MPU-6050 is visible as-is on the 3d model. (The usage of the Adafruit is optional). The GY-521 version can be seen as dark blue on the 3d model. This part will be responsible for measuring the acceleration of the rocket.
8. The 9v battery pack is visible as an orange cube on the 3d model.
9. The SD card is visible as a purple cuboid on the 3d model.

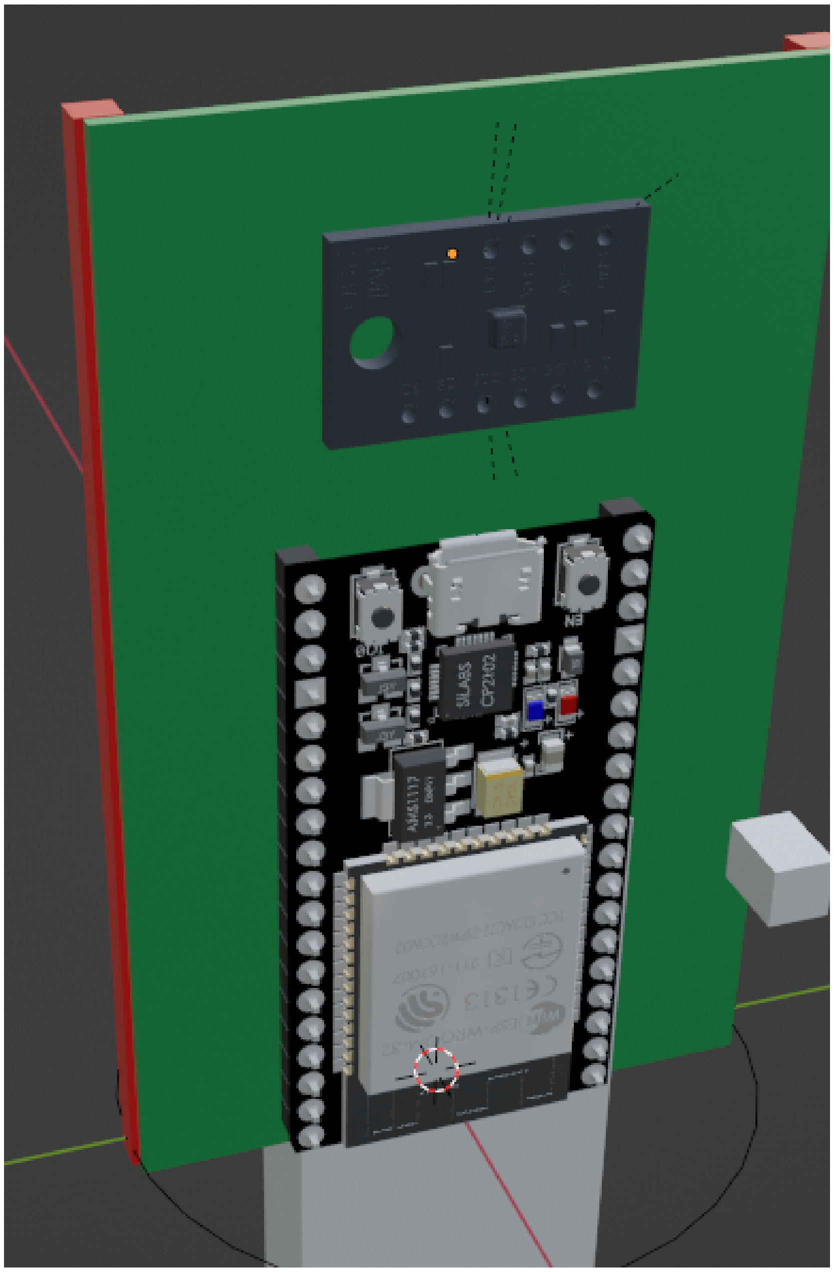

Figure 6: The first side of the bay

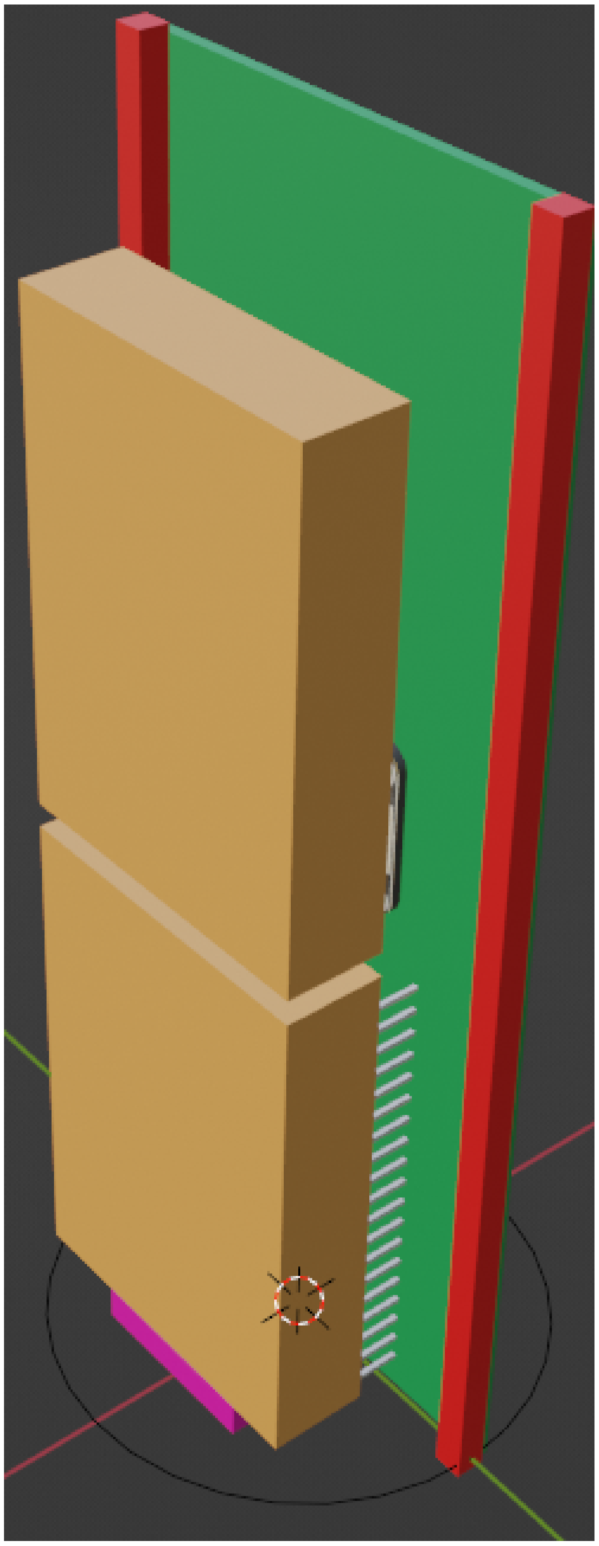

Figure 7: The second side of the bay.

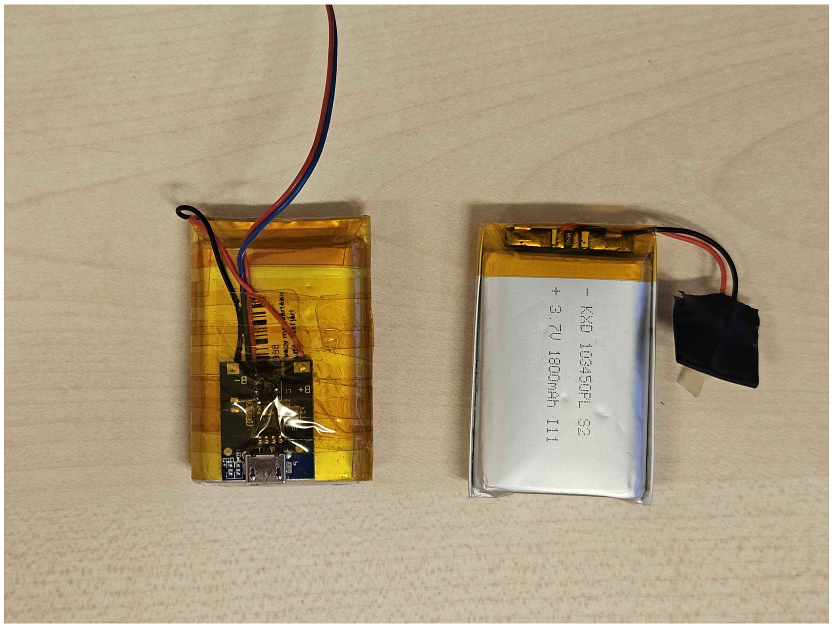

Figure 8: The battery packs locations are marked with orange color on the model.

The pin ends will be removed to save space.
NOTE: The green plate is NOT the PCB, just the plate.

0.1.8 Cost estimation

Below, the price of the elements will be listed:

1. SD card (JESW Micro 128G Ultra Memory SD Card...) - HUF3089.00
2. NodeMCU-32S - HUF3306.70
3. GY-521 (MPU6050) 6DOF - HUF1582.00
4. SparkFun Atmospheric Sensor Breakout - BME280 - HUF648.00
5. SD card module - HUF335
6. IRLML2502 MOSFET - HUF126
7. Ribbon cable - HUF318
8. Pin header - HUF163
9. Gate pull-down - HUF44
10. Gate res - HUF45
11. Bulk condensator - HUF182
12. Ceramic condensator - HUF46
13. Black wiring - HUF139
14. Red wiring - HUF137
15. 2x Test circuit - HUF400 x2

Total cost of the avionics bay electronics: HUF10960

0.1.9 Subsystem requirements

Here are the subsystem requirements split to each module:

Requirement ID	**Description**
BMESUB-ALADEEN-AV-RQT-0001 **Recovery system electronics**	
Requirement	Launch vehicles shall implement recovery system electronics, including sensors/flight computers and “electric initiators”, with a separate battery power supply.
Rationale	Ensures that the recovery system is fully tested, since no redundancy is present.
Notes	
Verification category	Review of Design
Verification method	Simulation
BMESUB-ALADEEN-AV-RQT-0002 **Data logging module**	
Requirement	The barometric pressure and the acceleration at each time frame shall be logged.
Rationale	
Notes	
Verification category	Review of Design
Verification method	Simulation
BMESUB-ALADEEN-AV-RQT-0003 **Data logging module**	
Requirement	The barometric pressure and the acceleration at each time frame shall be logged.
Rationale	
Notes	
Verification category	Review of Design
Verification method	Simulation
BMESUB-ALADEEN-AV-RQT-0004 **The holder plate**	
Requirement	The plate shall hold the PCB and have holes on it.
Rationale	It should not break.
Notes	
Verification category	Review of Design
Verification method	Simulation
BMESUB-ALADEEN-AV-RQT-0005 **The structural plate holder rods**	
Requirement	The rods should hold the plate in place and provide stability.
Rationale	They should not break.
Notes	
Verification category	Review of Design
Verification method	Simulation

Table 1: Subsystem Requirements Table

0.1.10 FMEA Analysis

Columns: Failure mode, Probability, Severity, Criticality, Actions (How the criticality drives the further development, additional verifications, and/or operations action)

Failure mode	**Probability**	**Severity**	**Criticality**	**Actions**
Ascending	Inprobable	Very Severe	Very Critical	The rocket will not reach the desired apogee and the structure may be damaged.

Table 2: FMEA Analysis Table

0.1.11 Interface description

Interface definition: Predetermined categories of interfaces eg mechanical, thermal, electrical, human etc. Everybody categories its interfaces according to these interfaces. Interface compatibility is a management level task!

0.1.12 Verification plan

Verification plan → Sablons (Verification method description, success criteria, TIMELINE) Here, the verifications described by the requirements and the verifications found by the FMEA complies and results in one comprehensive verification plan.

Tests:

1. The avionics gyroscope and the BME will be tested by running around in a large building and rapidly changing height.
2. The endurance of the hardware will be tested by shaking.
3. The software will be tested by custom unit tests, which will test the response of the BME and the gyro by in-house simulations.

0.1.13 Dimension and weight estimation

Below, the dimension and weight will be listed for each avionics part.

Part name	**Dimensions (Width × Length × Height)**	**Weight**
JESW Micro 128G Ultra Memory SD Card TF Flash Card ...	12mm × 15mm × 4mm	0.5g
GY-521 (MPU-6050) 6DOF	20mm × 16mm × 2.57mm	1.6g
NodeMCU-32S	48mm × 25mm × 10mm	10g
SparkFun Atmospheric Sensor Breakout – BME280	15mm × 11mm × 5mm	1g
Adafruit MPU-6050	17.8mm × 26mm × 4.6mm	1.8g
li-po accumulators	34mm × 50mm × 10mm (x2 parts)	70g
Cabling, solder, etc.	—	13g
2x 0806 resistors	—	2g
PCB	141mm * 33mm	20g

Table 3: Dimensions and weight of avionics components

The estimated total weight for one instance of the required parts is: **119.5 grams**, excluding the pillars and the plate, and also excluding the weight of the PCB.

This weight was calculated with the GY-521 MPU-6050 variant.

0.1.14 Verification needs

In the verification phase, all software functionality should be verified, namely:

1. The recovery parachute opening logic
2. The apogee verification logic
3. The data logging logic

The soldering also needs to be verified as the correctness of that is critical to the working of the avionics part.

0.1.15 Operational modes per state

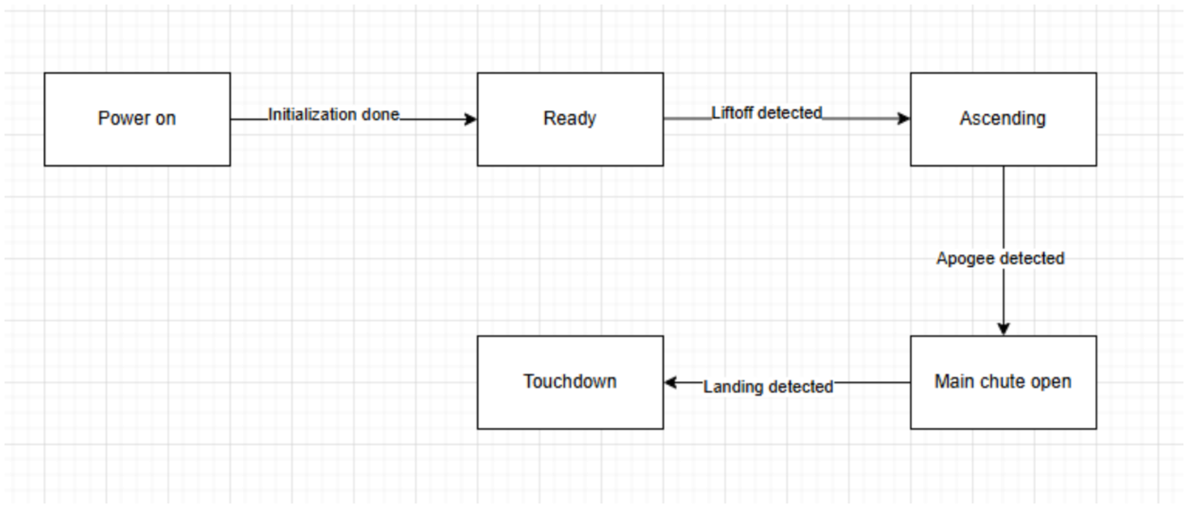

Figure 9: This diagram shows the process of a flight.

The subsystem behaves through:

1. Prelaunch: The avionics bay is calibrated and is given power. The initialization of the software is here.
2. Ascent: The avionics bay collects and logs information about pressure and height. It also determines the apogee position and when to open the chute
3. Descent: The avionics bay collects and logs information about pressure and height.
4. Recovery: The avionics bay contains an SD card where data is logged.

0.1.16 Command Telemetry definitions

The avionics communicates with the recovery bay. It tells the recovery bay when to open the chute based on the barometric and acceleration calculations.

0.1.17 Safety considerations

Proceed with extreme caution of the parachute did not open when the rocket is on ground again. The avionics logic may falsely indicate that the parachute needs to be opened even after flight.

0.1.18 Programming constraints

The software of the module shall be done in C++ and be C++17 standard compliant. The Arduino.h library shall be used for making the mainstream layout of the code flow. Additionally, the arduino design pattern shall be abided.

Additionally, these language features should be constrained:

1. Recursion shall be avoided. Only systems where recursion is inevitable, shall be used.
2. Constant compile time values shall be marked as constexpr and consteval shall be used accordingly. Additionally, if these are not possible replace them with preprocessor directives.
3. No C++ exceptions shall be used EVER. The methods and functions should indicate their success with either their return value or by modifying a global variable. (Based on the F-35 software manual.)

0.1.19 Environmental constraints

The avionics bay was designed to abide these constraints:

1. A maximum temperature of 80C°inner temperature.
2. A maximum height of 500 meters. Altough this has not been tested, the maximum height support can theoretically be much more.

0.1.20 High-level assembly procedures

The assembly of the avionics bay is done in this order:

1. Assemble and fit the PCB or the hardware mainframe.
2. The soldering of the subsystem modules. First, the surface mounted modules shall be assembled for ease-of-use. Next, the condensators and then the resistors. After all, the remaining hole mounted submodules should be soldered.
3. The soldering of the traces.
4. The mounting to the stability plate. (Then glueing if necessary)
5. The fitting of the avionics bay.

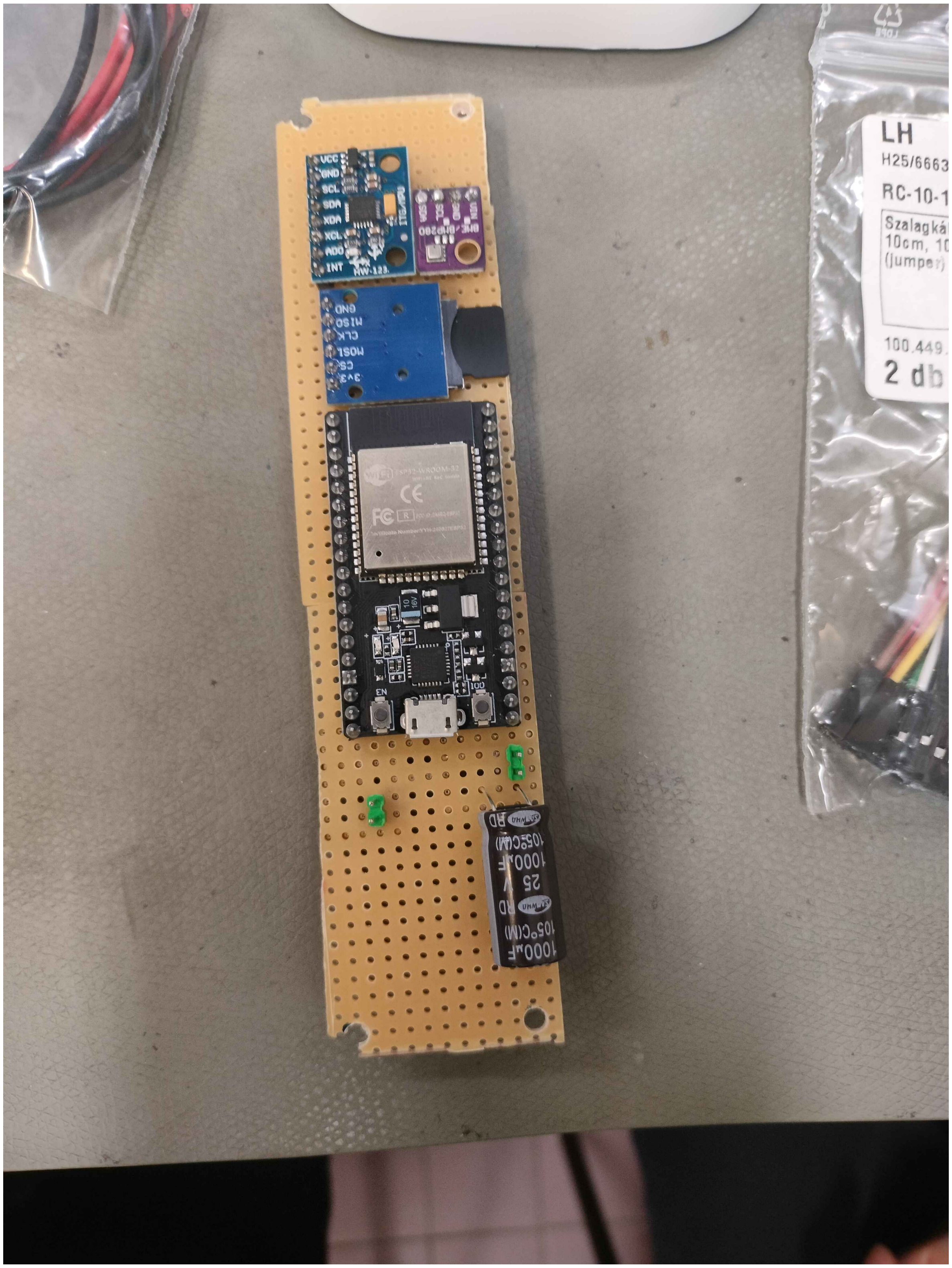

Figure 10: The panel's front side. Here, we can access the SD card and the ESP-32.

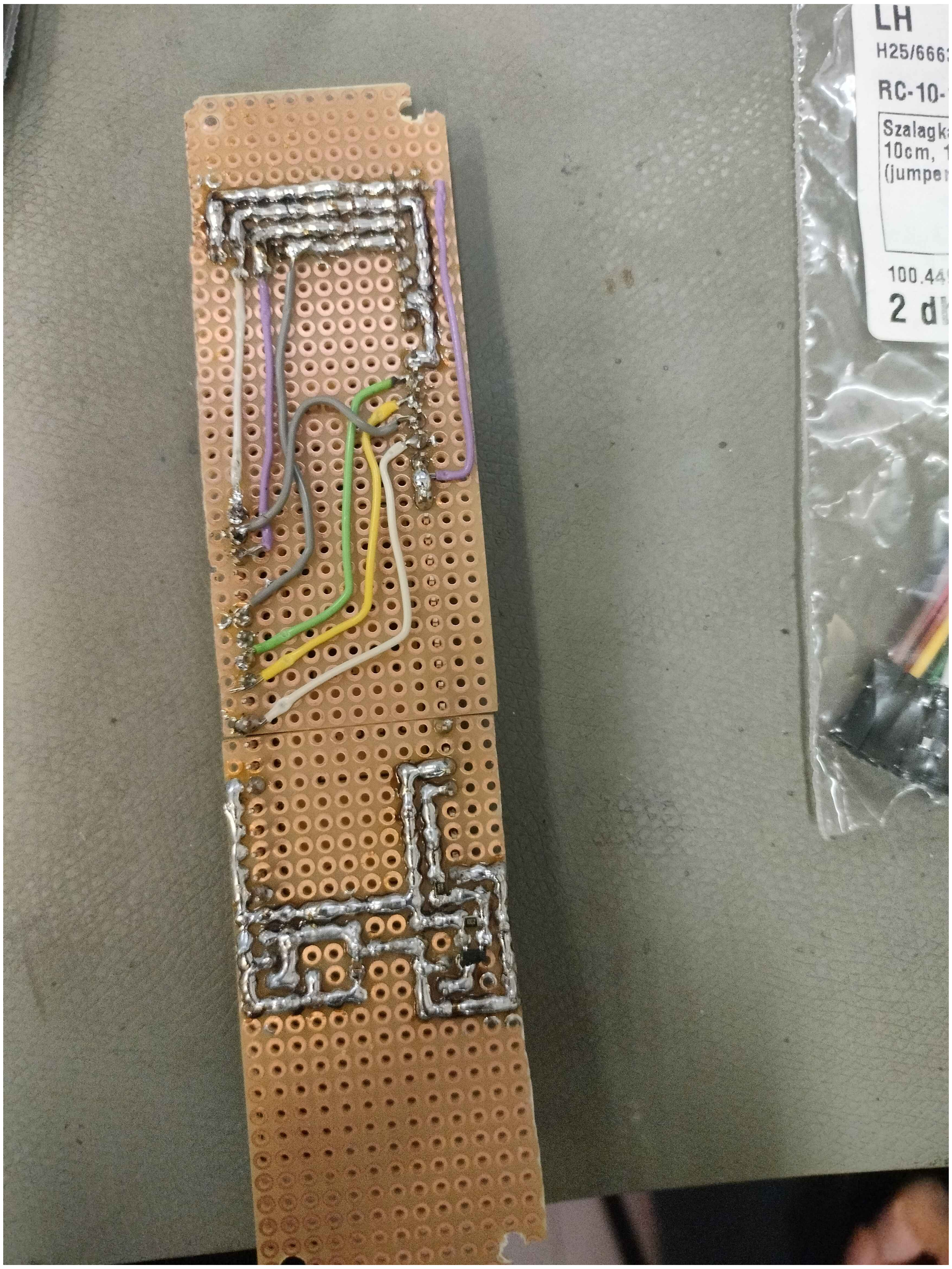

Figure 11: The panel's back side. Here, the traces and jumpers are visible.

0.1.21 Calibration procedures

The pressure and altitude sensing and their calibration is done be the software. For reference, see the main source code of the control software.

0.1.22 Lessons learned / Future work

The most important design lesson is that always prepare the avionics bay to support ANY hardware mainframe, not just PCB. This was a problem because we couldn't make our own PCB in time and thus we needed to resort to test circuitry.

A more well-padded hardware mainframe on the Y axis should also be a future improvement as of now we can only provide enough stability via glueing.