Arduino Rocket Flight Computer

Philippos Moschidis,
National and Kapodistrian University of Athens,
Aircraft Engineering Department,
Psachna, Greece,
mosxidis.filip@gmail.com

Petros S. Bithas,
National and Kapodistrian University of Athens,
Digital Industry Technologies Department,
Psachna, Greece,
pbithas@dind.uoa.gr

Abstract- This paper describes the construction of a flight computer, using a small model rocket, which is built with materials that can be easily obtained. In this context, a comparison of the flight data with those of the simulation has been performed. Based on this framework, an introduction to the flight computer will be given and then its construction together with all its sensors and its programming will be described.

Moreover, a detailed analysis of the rocket used will be given. In particular, the design concept will be presented, with the help of a computer program (OpenRocket), in conjunction with the materials used, the procedure followed for its construction, and finally, the test launches carried out up to the final launch. Entering the last chapter, the flight data of the simulation will be presented, followed by the data of the launch that was carried out and analysed using a numerical computational and programming language. Finally, a comparison between the two data sets will be made.

Keywords- Arduino, Flight Computer, Matlab, Microcontroller, Project.

I. Introduction

Model rocketry is an activity that involves designing, constructing, and launching self-made rockets. Model rockets vary greatly in size, shape, weight, and construction from detailed scale models of professional rockets to lightweight and highly finished competition models. Thousands of rockets ranging from 10 cm high miniatures to large models reaching altitudes over 10 km are launched annually. Model rocket motors with thrusts from a few Newtons up to several kilo-Newtons are readily available. This activity is relatively popular and is often cited as a source of inspiration for children to become engineers and scientists [1].

Amateur rocketry has been started in the 1950s when hobbyists wanted to experiment with their skill of building rockets. Designing, building, and firing self-made motors was, however, extremely dangerous, and the American Rocket Society has estimated that about one in seven amateur rocketeers during the time were injured in their hobby. In 1958, the first commercially-built model rocket motors became available. Having industrially-made, reasonably-priced, and safe motors available, removed the most dangerous aspect of amateur rocketry. This along with strict guidelines for the design and launching of model rockets formed the foundation for a safe and widespread hobby [2][3].

The subject in this paper is the construction of a flight computer and a model rocket. With the help of the flight computer, the data from the launch were collected from an SD card after the rocket had returned from the launch and then the data were analysed on Matlab. However, in order to perform this analysis, the rocket had to safely return from the launch without any damage. In model rocketry, safely retrieving a rocket after it has been launched is usually a primary objective, in which the recovery system is a device, mechanism, or process that allows the rocket to return safely to the ground [4]. To achieve that the mission profile is separated into a few steps.

To start, the launch happens when the engines fire up, 2.1 seconds later there's the main engine cutoff that's when the engine stops burning. From that point, the rocket is coasting for 5 seconds until it reaches apogee and the stage separation begins. More specifically, the nose is separated from the rest of the body of the rocket with the help of a small engine fire-up. A few seconds later, the parachute is deployed and after that, the rocket reduces speed until it reaches the ground and performs a smooth landing.

In another paper [5] the goal was to compare some commercial and self-made solid rocket motors. In particular, the paper highlights that the Klima D9 engine (the same engine used in the proposed rocket) underperformed in terms of total impulse and maximum thrust, differing on average from the nominal values of 6.8% and 16.5%, respectively, while they overperformed in terms of average thrust by 4.1%, and showed good agreement on the average burning time. Explaining some of the results that are present later on. In similar investigations in the past, the scope was to construct a small-sized model rocket and an introduction for the OpenRocket computer program was given[6]. Furthermore, another paper on the construction and description of flight computers is [7]. In another project, the coding and analysis of the flight computer were given in [8]. Finally, in [9] the design, analysis, and test of a High-Powered Model Rocket were described.

The methodology presented in this paper can be used to familiarise with amateur rocketry. The remainder of this article is as follows. In Section II, we present the flight computer and its sensors. In Section III, a brief intro for the rocket is given, while in Section IV we describe the flights that were performed. In Sections V to VII, we present the simulation and the experimental data and how they were analysed. Finally, in Section VIII, we will compare the simulation data with the experimental data and after that in Section IX, a conclusion of the paper will be given.

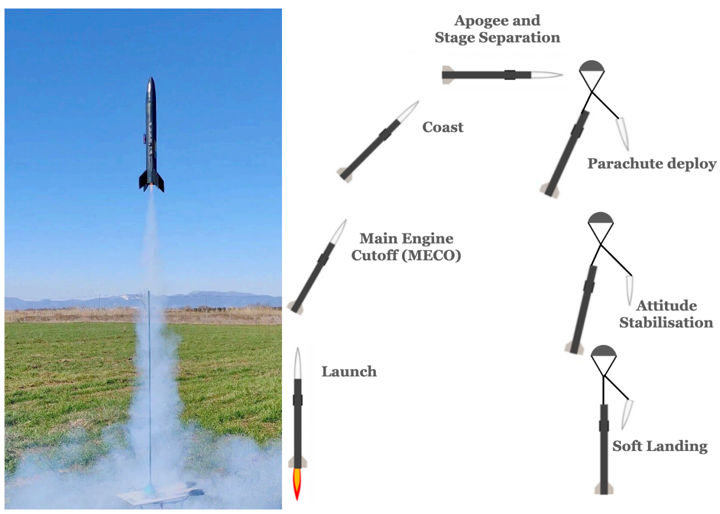

Fig. 1. Model rocket and mission profile

II. Flight Computer

The purpose of the flight computer is to calculate and collect the data that will be available during the flight. To do this, a computer is needed which can give commands to the sensors to collect and calculate data: temperature, pressure, speed, altitude, acceleration, orientation, and displacement. In this case, because the duration of the flight is short and the rocket is small, an Arduino microprocessor is used. Specifically the Arduino Nano is almost as powerful as the Arduino Uno microprocessor. Additionally, the Arduino hardware is easy for beginners and the IDE is simple to understand. But the main reason for the selection of the Arduino nano is it's size, weight and low cost as it is one of the smallest yet very powerful. For the assemble of the PCB I mostly used a soldering iron to create solder lines and to connect cables from the microcontroller to the sensors.

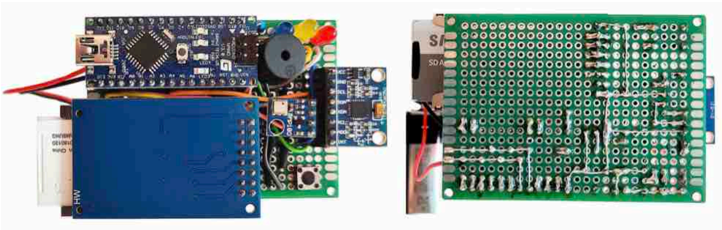

Fig. 2. Flight computer

The first sensor is the BMP 180, it was ready to use out of the box, there was no need for fine-tuning. The sensor is designed to measure barometric pressure from 30000 to 110000 Pa (9000 m to -500 m above sea level) and temperatures from -40°C to 85°C with an accuracy of ±1°C[10][11]. The unit comes with an integrated ML6206 3.3V regulator so that it can be used with a 5V microcontroller such as the Arduino Nano[12].

The second sensor used is the MPU 6050 which consists of a three-axis accelerometer and a three-axis gyroscope. This helps us to measure acceleration, velocity, orientation, displacement, and other parameters related to the rocket's motion. To test the MPU 6050, Jeff Rowberg's example code was used, which includes a nice visual demonstration of the IMU's output. To measure acceleration, the MPU6050 uses its on-chip accelerometer with four programmable full-scale ranges of ±2g, ±4g, ±8g, and ±16g. For the rotation measurement the MPU6050 measures the angular rotation of the rotation axis of the vehicle with programmable full-scale ranges of ±250°/s, ±500°/s, ±1000°/s, and ±2000°/s [13][14].

The microprocessor powers and coordinates the two sensors mentioned. To store the data during the flight, since they will not be displayed on a screen at that time, a Micro SD adapter will be needed to collect the data so that it can be analysed offline.

III. Flight Computer Diagram

As seen in Figure 3, all the flight computer components are connected to the Arduino, which powers them and also determines the operation of each sensor, while the Arduino itself is powered by three 3.3V batteries. The BMP 180 and MPU 6050 sensors communicate with the Arduino using the I2C protocol (it is a bus interface connection protocol incorporated into devices for serial communication [15]), so the SCL and SDA pins of the two sensors are connected to each other and from there a cable goes to the A5 and A4 pins respectively, while both sensors are powered by the Arduino's 3.3V pin. There are also three LEDs, which indicate which is the current mode of operation of the computer. The red LED indicates that the computer is connected to the sensors, if the LED remains on it means that there is a problem with one connection. If everything is okay, the yellow LED turns on.

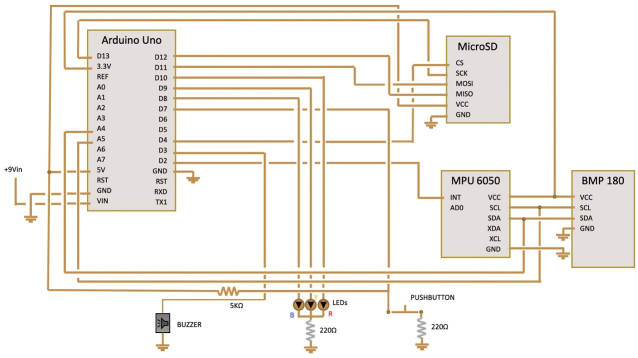

Fig. 3. Flight computer diagram

This means that the computer is collecting data. Finally, when the push-button is pressed, the blue LED turns on and the computer changes from the "data logging" state to "stop data logging and safe disconnect''. The software presented in [16] is used by the flight computer for data logging and is written in C++ programming language and loaded into the Arduino Nano with the help of the Arduino IDE.

III. The Model Rocket

The general design of the rocket consists mainly of two parts, the main body of the rocket and its nose. The rocket is divided into two bodies as after launch and while landing, the nose is separated from the rest of the body to open the parachute to reduce the landing speed. In the launches that were carried out, more than one rocket was used and some improvements were made after the unsuccessful launches.

The engines used are the D9-5 from Klima [17]. These have a total thrust of 20Ns, a thrust of 9N, a total thrust duration of 2.1s, and a delay of 5s. The 5-second delay means how many seconds after the engines have fired there will be an additional burn of the engines to separate the nose of the rocket and deploy the parachute. For the process just described to work properly, the rocket body, nose cone, and parachute must be connected. This was achieved by attaching a rubber band to the inner wall of the rocket, while at the other end of the rubber band, a ring was fitted which is connected to the parachute as well as to the rocket's nose.

The first rocket (Aurora V1) used in the first launches without the flight computer had a D9-5 engine, the weight of the rocket was 0.322kg. As shown in the figure below, the overall length of the rocket is 76 cm and its diameter is 0.68m. In the lower left of Figure 5 the total height (112m)

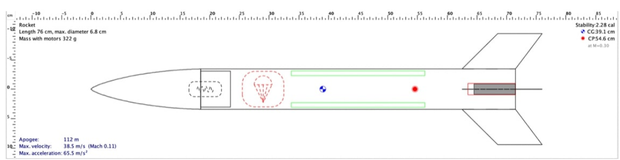

Fig. 4. Aurora V1 (First model rocket) without the flight computer

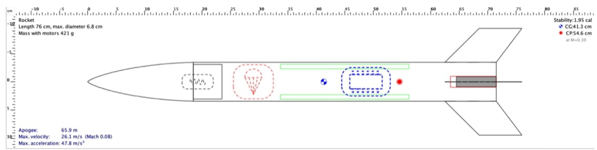

Fig. 5. Aurora V1 with the flight computer

that the rocket would reach, the maximum velocity (38.5m/s) and the maximum acceleration (65.5m/s) are shown.

Four trapezoidal blades (0.09m root chord, 0.075m tip chord, 0.06m height, and 45° arrow angle) were used at the base of the rocket and the engine was stabilised with the help of four rings.

The installation of the flight computer inside the rocket increased the total weight from 0.322kg to 0.421kg. This resulted in the maximum speed dropping to 26.1m/s and the maximum acceleration to 47.8m/s, while the maximum altitude would be 65.9m meaning that the margin to deploy the parachute is minimal. Since it is not possible to reduce the total weight of the rocket, the next solution is to use two engines in the rocket to achieve a higher altitude for the parachute to open.

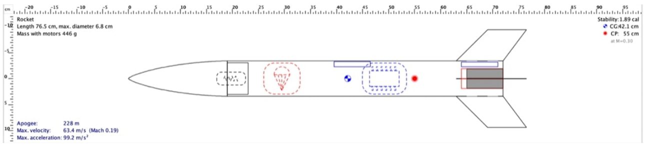

Fig. 6. Aurora V2 (First model rocket) with the flight computer

Following this conclusion, the second rocket was built under the name Aurora V2. The only difference in the design of the rocket is that a second engine was added. The height and diameter remained the same. Looking at Figure 6, we can see that the total altitude was increased to 228m (an ideal value for the parachute to deploy and reduce the landing speed as much as possible), the maximum speed is 63.4m/s and the maximum acceleration is 99.2m/s².

IV. Model Rocket Launches

In total there were eight launches, the first five with the Aurora V1 Rocket. In the first three launches, the flight computer was not inside the rocket as the aim was to check that all the components that had been installed would work properly.

The first flight, which took place on the 21st of October 2021, had some stabilising issues. 8 days later the second launch took place. The launch started correctly but quite early on, the rocket took a sideways inclination (due to wind), resulting in insufficient time for the nose to disengage and deploy the parachute. After the new rocket nose was built, the first successful flight followed on the 25th of November 2021. The launch went exactly as planned and during the landing, the nose was disconnected, the parachute deployed and the rocket returned at a relatively low speed.

Once everything worked as planned, the next step was to place the flight computer inside the rocket to record the data. To achieve this, a hole was made in the rocket so that the computer could fit inside and was stabilised inside its body with a fibrous insulating material (polypropylene).

At the fourth (31 December 2021) and fifth (6 January 2022) launches it appeared that the problem of the rocket during the flight was the low altitude and the wrong weight distribution, we decided to build a new rocket with two engines. This new design required more space inside the rings to accommodate two engines. To solve the problem of the centre of gravity the flight computer was placed at a lower point inside the rocket and three 3.3V clock type batteries connected in series replaced the 9V battery used in the first flights (less weight).

The sixth launch, which happened on the 27th of January 2022, achieved a satisfying altitude but the parachute did not open, fortunately, the rocket was not seriously damaged so the repair was quick and simple. The seventh flight (10 February 2022) was successful, the parachute deployed at the correct altitude resulting in an extremely smooth landing. However, a problem with the SD card resulted in no data being recorded. The eighth launch was the last and took place on the 11th of February 2022. There was no problem during the launch, but unfortunately, the parachute was late to deploy resulting in a high landing speed. This did not prevent successful data collection.

V. Launch Data

The flight computer can collect the data thanks to its two sensors. The launch data are time, temperature pressure which is understandable, but the other values, namely ax, ay, and az (acceleration on each axis) and rx, ry, and rz (Euler angles indicating the angular position of the sensor [17]) are in raw form and it is not easy to draw a conclusion so they will be analysed in the following chapter. Some of the data collected during the flight are presented in the following table.

VI. Launch Simulation

With OpenRocket (a fully featured model rocket simulator that allows you to design and simulate your rockets before you build and fly them) [18] at our disposal, it is possible to simulate the launch and create diagrams. There is a wide range of diagrams to choose from, but since the flight computer on the rocket calculates altitude, temperature, pressure, and velocity, we will limit ourselves to these. It is also possible to select weather variables and geographic coordinates so that the results obtained will be more accurate.

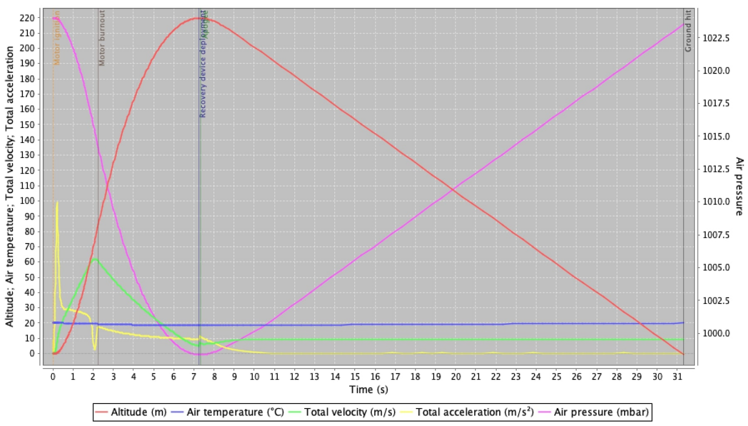

Fig. 7. Altitude, Temperature, Total velocity, Total acceleration and Air pressure

VII. Launch data analysis

The flight computer can collect the data thanks to its two sensors. Some of the data like time, temperature, and pressure are clear, but the other values, namely ax, ay, and az (acceleration on each axis) and rx, ry, and rz (Euler angles) are in raw form and it is not easy to conclude so they were analysed with the help of Matlab [19][20]. An important data that the flight computer did not record is the altitude at which it was launched. Using the data from the air pressure sensor on the BMP 180, we will determine the approximate altitude of the flight computer using the relationship between air pressure and altitude (in Paschal's and meters respectively [21].

Looking at the altitude plot in diagram 8, we see that the rocket reached almost 160 meters at close to 50ts (note that 50 is not the seconds of flight but the fiftieth data record, a timestamp is about 0.16 seconds). Then, it appears to be a sharp decrease in height, while near 120ts the reduction is smoothed out. This is because the parachute did not open immediately so the rocket was free-falling.

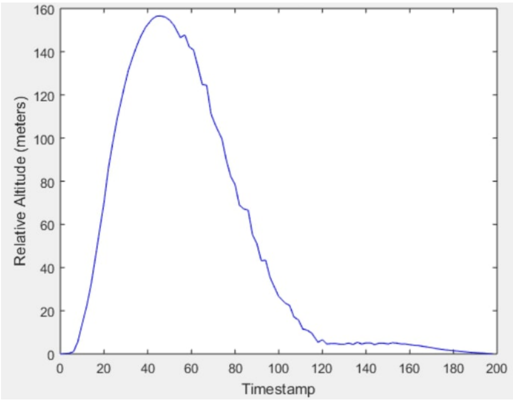

Fig. 8. Altitude diagram

In diagram 9 the temperature is shown, at 60ts there is a sharp increase in temperature, which shows that at this point the ejection charge phenomenon is observed, i.e. the nose of the rocket opens to deploy the parachute. While the pressure plot (diagram 10) is exactly the opposite of the altitude plot, since as the pressure decreases, the altitude increases.

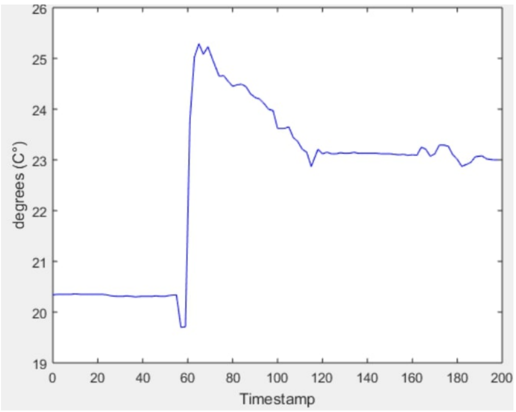

Fig. 9. Temperature diagram

Utilising the raw values of the MPU6050, the rocket's velocity, as well as the motion in the x, y and z axes, can be analysed. The way the flight computer is mounted inside the rocket, the x-axis is the up-down motion, the y-axis is the right-left motion and the z-axis is the in-out motion. Looking at the three motion diagrams (diagram 11), we can see that up to 60ts the values of all three axes are constant.

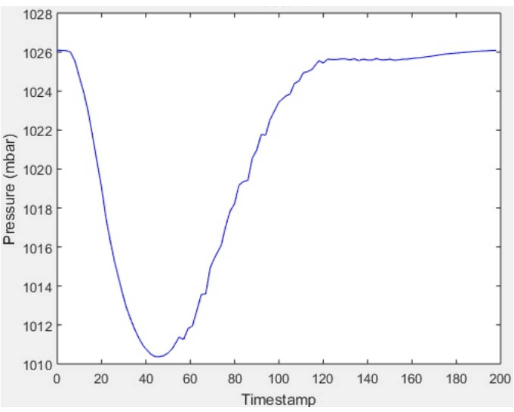

Fig. 10. Pressure diagram

However, afterward, there is a lot of turbulence due to the fact that the rocket has finished burning the engines and starts to return to the ground. As the parachute has not yet opened, the rocket has several turbulences in all directions up to the landing point.

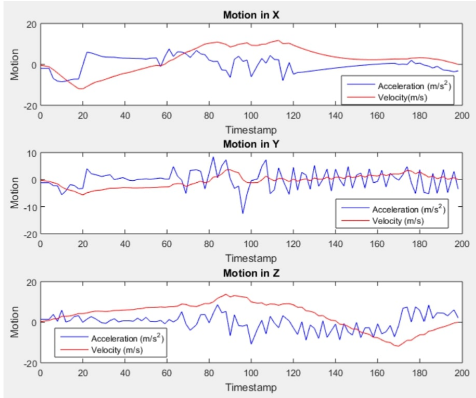

Fig. 11. Axle movement diagram

The last graph (diagram 12) shows the speed. Up to 20ts, we see that there is a continuous increase in speed as the engines are used up to that point. Then up to 40ts, there is a continuous decrease in speed because at this point the rocket reaches its maximum altitude. There is then a further increase in speed (up to 90ts) as the rocket falls from the maximum altitude it has reached and begins its return to the ground. Finally, from 170ts until the rocket touches the ground there is a sharp decrease in speed which occurs because the parachute has opened.

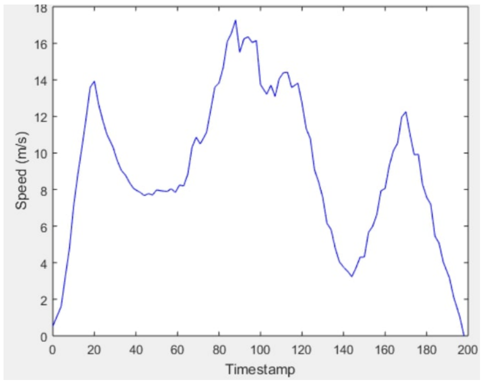

Fig. 12. Speed diagram

VIII. DATA COMPARISON

Now that we have the plots from the simulated launch (right) and the actual launch (left), we can compare the values. Starting with the altitude (diagram 13), we note that initially in both diagrams there is a sharp increase in altitude, but in the simulation, the maximum altitude was 220 meters, whereas, in reality, it was 160 meters. During the descent of the rocket in the simulation diagram after the parachute has opened, there is a steady decrease until landing, whereas in the actual launch diagram the parachute deployed very late and therefore a sharp decrease is observed.

The temperature graphs (diagram 14) differ considerably from each other. Because the sensor is located inside the rocket, the values are not very accurate due to the combustion of the engines. In the simulation, there is a continuous decrease until it reaches the maximum altitude and from there a steady increase until it returns to the initial temperature. Below in the pressure (diagram 15) diagrams, we see that the exact opposite of what happened at altitude occurs. In the simulation when the parachute opened a gradual increase in pressure is shown, whereas, in reality, the pressure increase is steeper up to 120ts where the parachute has opened, resulting in a steady increase in pressure.

Finally, in the velocity diagrams (diagram 16) we can see that the simulation and the reality are very different. Whereas at the beginning of both diagrams there is a constant increase in speed and a relative decrease after the end of engine combustion, in the simulation there is a constant decrease in speed until the parachute opens, but in reality, there is a new increase. Until the rocket lands, there are noticeable fluctuations in the diagram with the actual launch which are imperceptible in the simulation. The difference between the two diagrams is due to the fact that the simulation shows only the lateral velocity.

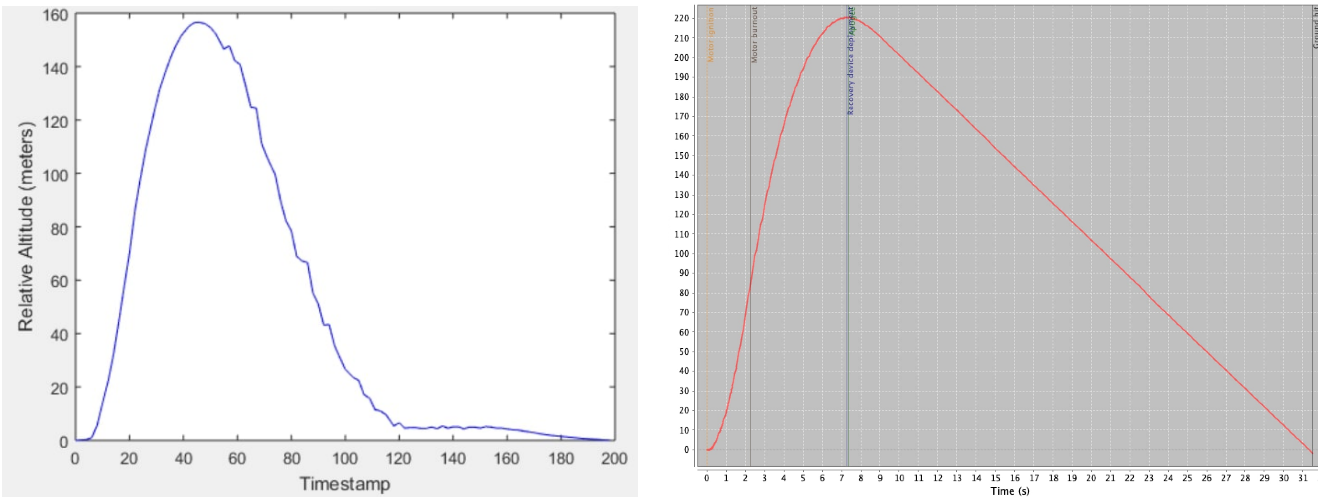

Fig. 13. Altitude comparison diagram

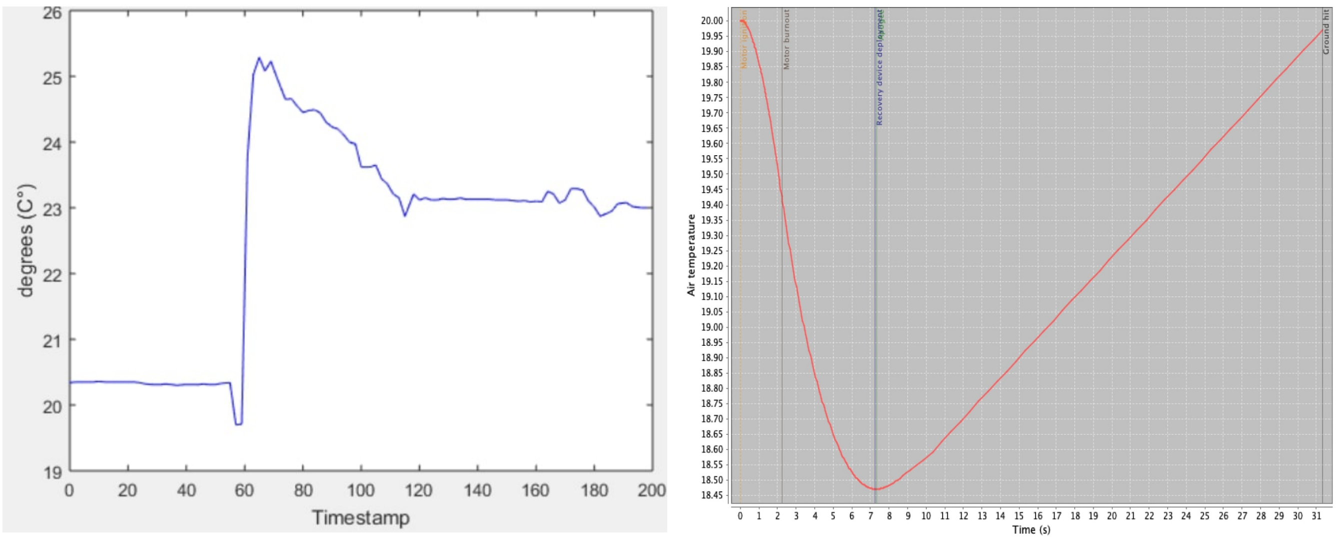

Fig. 14. Temperatur comparison diagram

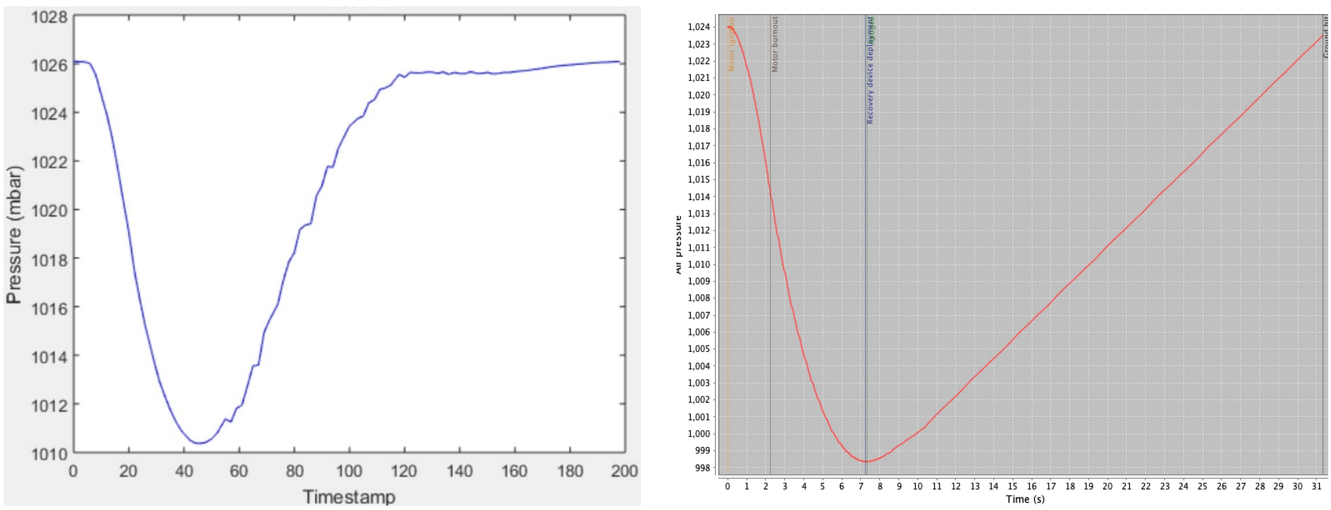

Fig. 15. Pressure comparison diagram

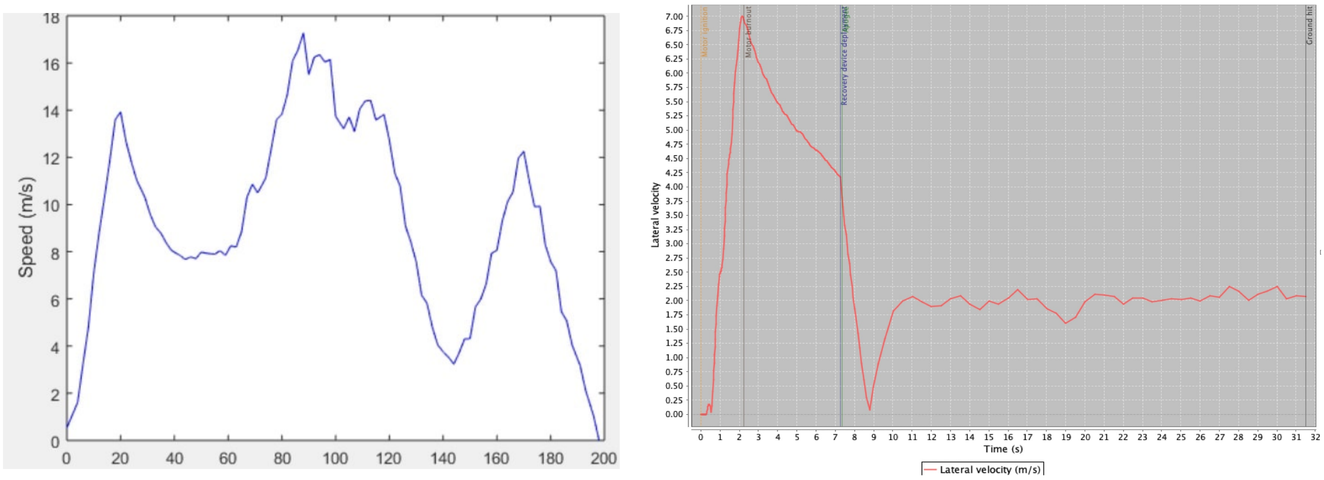

Fig. 16. Speed comparison diagram

IX. CONCLUSIONS

Building the rocket was easier than expected. However, because of the use of different materials, it was difficult to evenly distribute the weight. This could have been achieved by using a 3D Printer. The most difficult part of the rocket construction was the stabilisation of the flight computer inside the rocket, which also had to be protected from the engines. The solution was found by using a piece of acetate paper which had to be cut with great precision to allow the nose to be separated from the rest of the body of the rocket and a coating of aluminium foil which served as an insulating material. Another issue was the difference between some diagrams showing the actual launch versus those showing the simulation. The flight computer data could be much more detailed if more accurate sensors were used, which could not be done due to high costs.

Thus we conclude that there is room for improving the measurements and the construction by using better sensors and more robust materials for the rocket. This conclusion and the fact that the difficulties encountered in the course of the work were not discouraged but increased the interest, are opening up the prospect of possible further work on this subject.

X. REFERENCES

1. Model Rocket Workshop: a Problem-Based Learning Experience for Engineering Students J.I. Rojas, X. Prats, A. Montlaur and E. García-Berro Universitat Politècnica de Catalunya (UPC), Escola Politècnica Superior de Castelldefels (EPSC). December 2008. International Journal of Emerging Technologies in Learning (iJET)
2. Stine, H., Stine, B., Handbook of Model Rocketry, 7th edition, Wiley, 2004.
3. OpenRocket technical documentation For OpenRocket version 13.05 2013-05-10 by Sampo Niskanen
4. T. Milligan, "The Different Rocket Recovery Techniques," Apogee Components, July 2017.
5. Student firing tests and launches with commercial and self-made solid rocket motors: Mario Tindaro Migliorino, Manuel Aiello, Matteo Berti, Marco Rotondi, Simone D'Alessandro, Daniele Bianchi, Munzer Jahjah, Marco Pizzarelli. Department of Mechanical and Aerospace Engineering, Sapienza University of Rome. Italian Space Agency (ASI). January 2022
6. D. Rohini, C. Sasikumar, P. Samiyappan, B. Dakshinamurthy, Neeraja Koppula. Design & analysis of solid rocket using open rocket software. May 2022. International Conference on Advanced Materials for Innovation and Sustainability.
7. William Brown, Michael Wiesneth, Thomas Faust, Nghia Huynh, Carlos Montalvo, Kent Lino and Andrew Tindell. Measured and simulated analysis of a model rocket. February 5, 2018. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering
8. https://www.instructables.com/Model-Rocket-Flight-Computer-With-Arduino/
9. Julia Bigwood, Caroline Kuhnle, Robert Connor, Alec Mitkov, Colby Gilbert, Nathaniel Rutkowski, Christian M. Schrader, James Ternent. Design, Analysis, and Test of a High-Powered Model Rocket-2. April 2021. Worcester Polytechnic Institute
10. https://lastminuteengineers.com/bmp180-arduino-tutorial/
11. https://components101.com/sensors/bmp180-atmospheric-pressure-sensor
12. https://store.arduino.cc/products/arduino-nano-every?queryID=undefined
13. https://lastminuteengineers.com/mpu6050-accel-gyro-arduino-tutorial/
14. https://components101.com/sensors/mpu6050-module
15. https://www.geeksforgeeks.org/i2c-communication-protocol/
16. Code written by Matthew Minogue, Copyright 2020
17. Rocket motors: https://www.raketenmodellbau-klima.de
18. Openrocket, https://openrocket.info
19. https://en.wikipedia.org/wiki/Euler_angles
20. https://www.mathworks.com/products/matlab.html
21. https://www.engineeringtoolbox.com/air-altitude-pressure-d_462.html