SWAROG PROJECT DOCUMENTATION

C'SPACE 2017

Hadrien BILLIAU	Project leader
Thomas CLAQUIN	Member – Mechanic team
Julien GALIDIE	Member – Electronic team
Dimitri GUERSON	Member – Mechanic team
Victor PERTEL	Member – Electronic team

Redacted by Hadrien Billiau, 10/09/2017

Summary

I. Project presentation

A. The beginning

The project first started as an academic project, a scholar study of a complete system. At this stage, the goal wasn't to produce anything, just to design an abstract system, develop the concept by imagining how it could be made and what kind of circuits and mechanisms would run it.

After a great brainstorming session with the project team, we decided to design a device that would make an echo to the search of exoplanets and habitable planet. We are searching habitable planets outside our solar system, and exploring other solar systems to find out extraterrestrial life. The key to probe planets is still to gather data about them; this is why we took the decision to design a small probe. A tiny CanSat is small and light enough to be delivered far away, and cheap enough to let students make simple missions as a training repeatedly.

Our first idea once we started the project was to realize it with the ESATACA Space Odyssey, the rocketry and space club of the ESTACA engineering school. The main topic of the project has been quickly defined as a space probe measuring common useful parameters at the surface of a planet. We worked with the project Solinas's team to get the maximum external dimensions we had to fit all of our project's systems: Solinas is the experimental rocket (FUSEX) that will launch our project, as well as a captive balloon during the CanSat challenge we wanted to participate to.

B. The functionalities

Some of our ideas have been canceled to gain space. We first stated to brainstorm about what was necessary for our project – or for a space probe on an unknown planet. Hence, we kept the mandatory mission (the deployment of a part of the CanSat), the recording of the atmospheric parameters, and the camera. With these systems, there was not enough space for the GPS and the telecommunication system.

Therefore, our project is designed to film the ejection of the CanSat, the fall, and the landing, while it records some atmospheric data: the temperature, pressure and humidity, of the atmosphere, as well as the composition of the ambient light (with the RGB – Red Green Blue – components).

C. The first designs

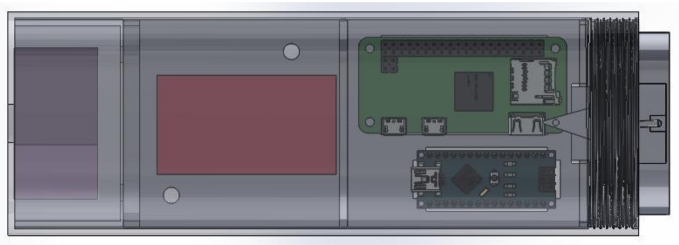
When we designed the CanSat, we decided to have an internal structure – the skeleton of the electronical part – and a protecting shell – the skin – so the equipment would not be damages by the landing or meteorological hazards.

We explored the possibilities to machine a metallic internal structure, but it wouldn't be easy and quick to produce. It was a bit expensive, especially in case we had to make a prototype first, and would also be dangerously heavy. Of course, it would be complicated to modify it after it has been made.

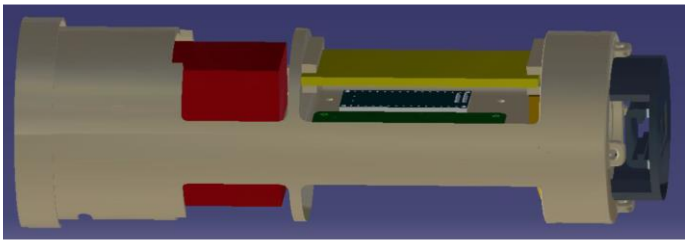
A 3D-printed structure was a great solution to obtain a light, tough enough, quick and easy to produce one. It was made with PLA plastic, a common and quite cheap material. This material meets our expectations in terms of quality, toughness, price and accessibility. It has been tested and improved with the ESTACA FabLab, where we used the 3D printer and were advised by Mr. Pierre-Emmanuel Mangin.

Two structures were designed with Catia software, and a monolithic structure was chosen with some of the innovative ideas from the other one. Some modifications to make this design more resistant and more convenient for the assembly were made. The final structure was indeed too complex to machine it without the 3D printing technology, but also weaker than a full metal structure, so we reinforced the impact lower zone with an aluminum ring screwed to the external shell.

The external shell was made of polymer-reinforced carbon fibers (we will abridge it with PRCF). This is a great material as it can resist to a rocket crash, and is lighter than aluminum or polymer-reinforced glass fibers so it is perfect for our use.



The first design made with 3 separated containers as a main structure and an external protective shell. This one was not selected, but some ideas from it were.

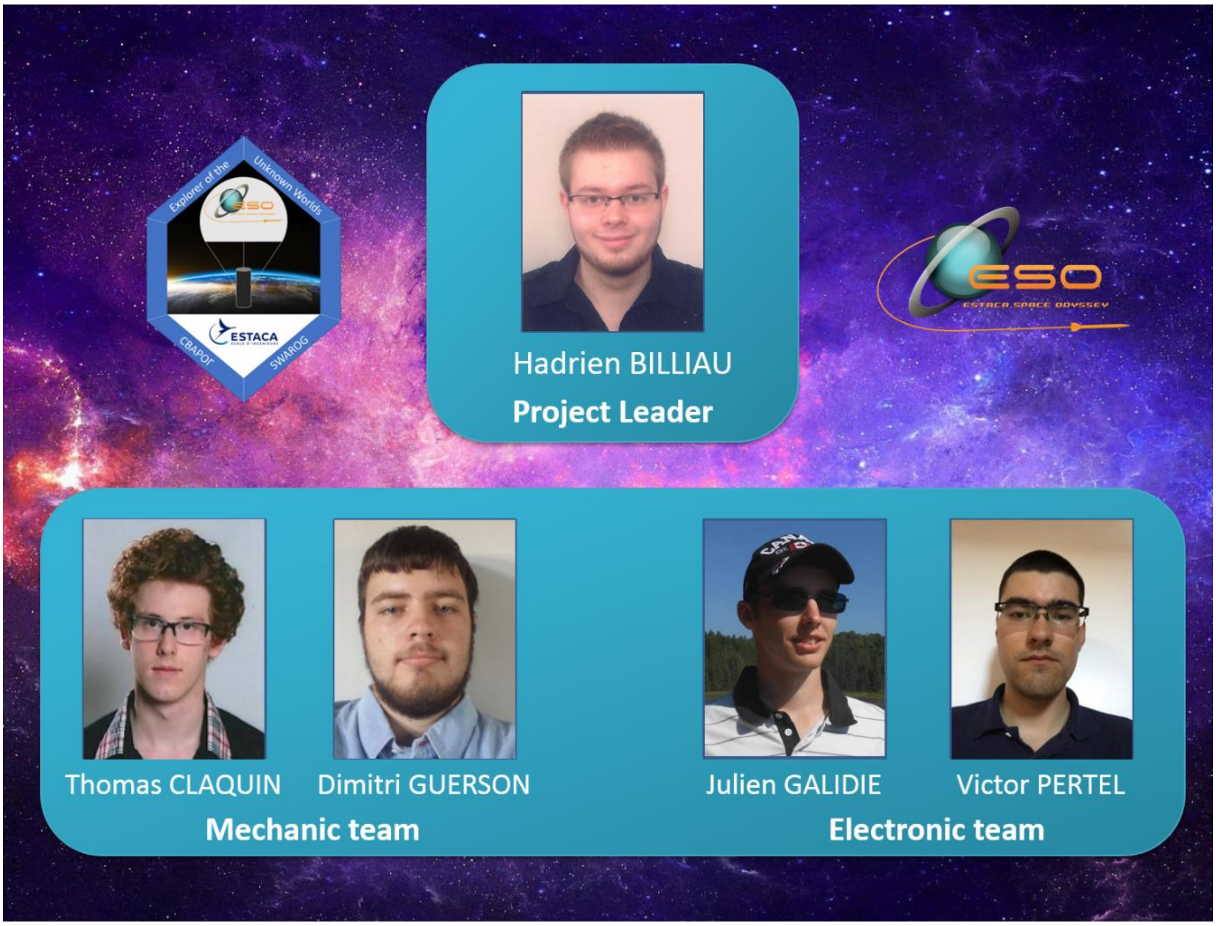


The second design with a monolithic structure, which has been improved later with the selected ideas from the other design.

In addition, here is the final design of the labels that can be found on the polymer reinforced carbon fiber external protective shell:

D. Team presentation

Here you can find some information about the team who developed this project since the very beginning, from scratches to the functional device.

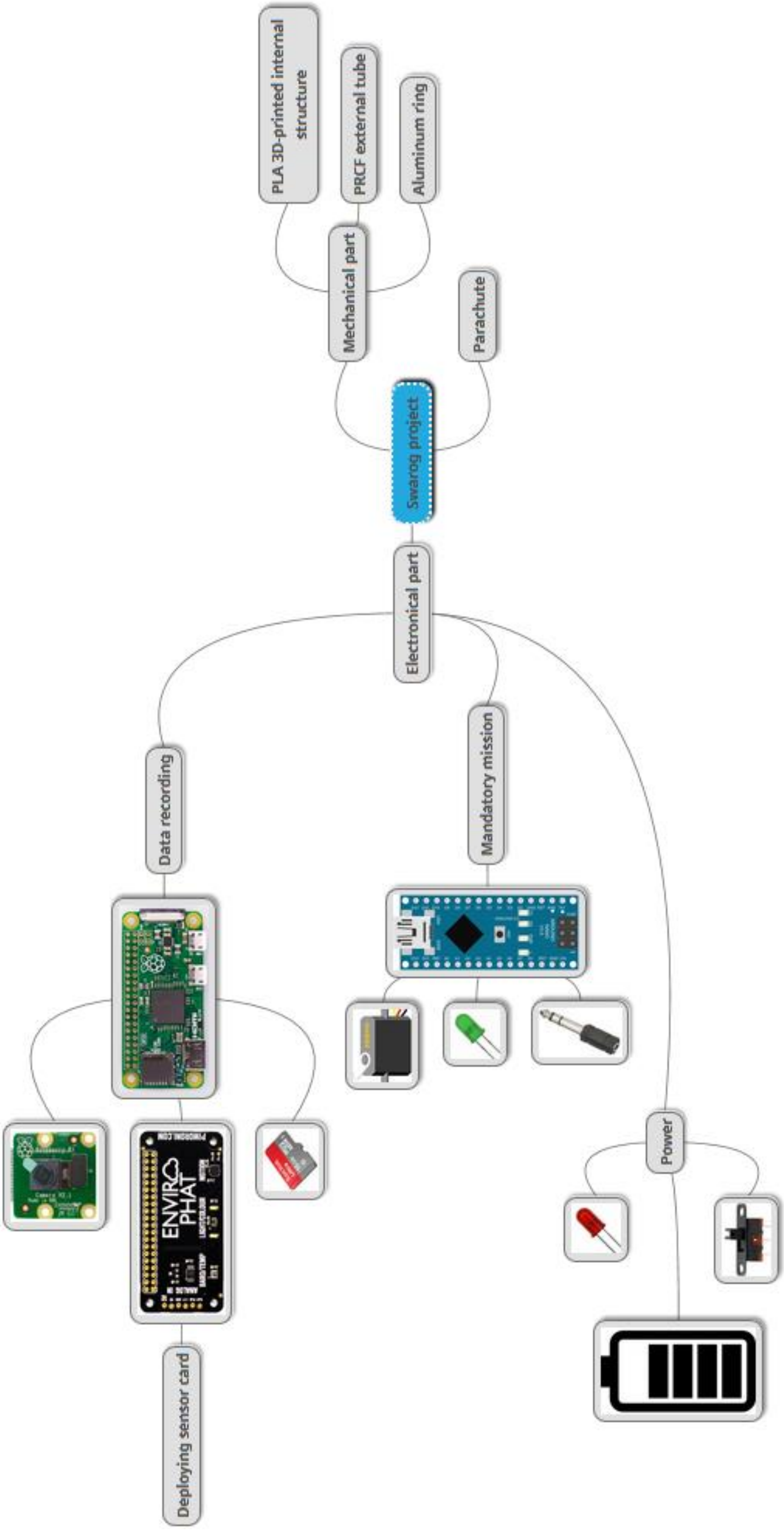


E-mail contact:

hadrien.billiau@estaca.eu	- worked on the management, report, documents, electronic and coding part.
thomas.claquin@estaca.eu	- worked on the 3D modeling and structure machining.
dimitri.guerson@estaca.eu	- worked on the 3D modeling and structure machining.
julien.galidie@estaca.eu	- worked on electronic cards realization and tests.
victor.pertel@estaca.eu	- worked on electronic cards tests and electrical architecture.

II. Concept diagram

The diagram on the following page will detail which component is connected to which other.We can find on it the mechanical and the electronical part, which is divided into 2 subparts, as explained later.


Swarog project
Mechanical part
PLA 3D-printed internal structure
PRCF external tube
Aluminum ring
Parachute
Electronical part
Data recording
Mandatory mission
Power
Deploying sensor card
ENVIRO PHAT

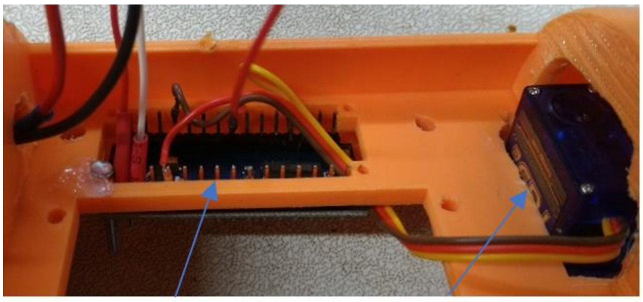
III. Electronical architecture

A. Mandatory mission

This year the challenge's mandatory mission is to realize a deployment mechanism which expends itself out of the CanSat volume. We decided to deploy some sensors in order to first protect them during the fall and the landing, and second to get more accurate measures. Indeed, these sensors will be directly in contact with the external medium, the atmosphere we want to gather data from.

To achieve this, we decided to make an upper container. It was made to fit at the top of the structure, and it is pushed by a servomotor and a lever, a simple and tiny enough solution to activate the deployment.

This servomotor is controlled by an Arduino Nano. This part is autonomous and disconnected to the electronical part of the free mission, so it is easier to secure both the measuring part and the electromechanical part. Any problem occurring on one part will not affect the other.

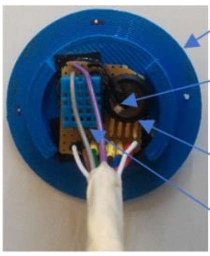


A blinking green LED indicates that the CanSat initiated its deployment mission to check if this part works before the flight, for visual inspection purpose. A jack connector is the way we control the CanSat is no longer in the launching pod or the FUSEX: when the CanSat will be launched, the jack connector will be unplugged and start powering the Arduino Nano. Once it is running, a timer will activate the servomotor after the landing. The servomotor will push outside the upper compartment which contains a humidity sensor, a temperature sensor, and a pressure sensor.

B. Free mission

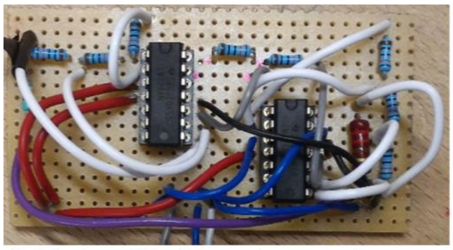
Sensors used

To measure and record the characteristics of an Earth like planet, we use some analogical sensors and some numerical ones.



First, the upper compartment contains a DHT11 humidity sensor, a P0K1.232.6W.B.007 temperature sensor, and a MPX2202A absolute pressure sensor.

In addition, we use an Enviro pHat which concentrate a motion sensor, a light sensor (or RGB colors sensor), an absolute pressure sensor, a temperature sensor and an ADC. The measures of the temperature are slightly above the test values, which is due to the heat generated by both the Raspberry Pi and the Enviro pHat itself. Of course, some heat is also generated by the batteries, the power card, and the amplification card, with a minor impact compared to the Raspberry Pi Zero.



Finished amplification card – some wires are not connected yet.

On the lower part of the 3D printed structure, we embedded a camera which will film the ground during the fall of the CanSat. This kind of camera plugs itself on the camera port of the Raspberry Pi Zero (a compatible f is needed).

Converting the analog signals to digital ones

An amplification card is connected to the outputs of the sensors, so the measures can be sent to the core of the CanSat, and be read by an analog to digital converter (ADC). The converted signal is processed by the Raspberry Pi Zero V.1.3 and recorded on a micro SD card. The Raspberry Pi Zero V.1.3 is the nano-computer which processes the signal to regroup all the measures on a CSV file and records it on the embedded micro SD card. This card contains the OS of the Raspberry Pi Zero, Raspbian (a modified and light version of Debian Operating System).

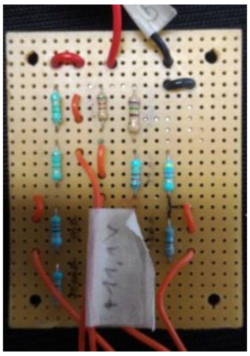
Software part

The main software was coded with Node-RED. This allowed us to use visual scripting, a quick way to code via diagrams and programable boxes; it was faster to code and to debug this way. The library of the Enviro pHat has been used to communicate via I²C bus, and was very useful to read the measures from the sensors and the ADC.

The camera was controlled by using Picamera software. It runs on Raspbian OS and was designed to manage Raspberry Pi – compatible cameras of any kind (without IR filter, normal kind, webcams …). It provides variety of tools to modify, preview, record, or take pictures with such a camera using any kind of Raspberry Pi.

C. Power management

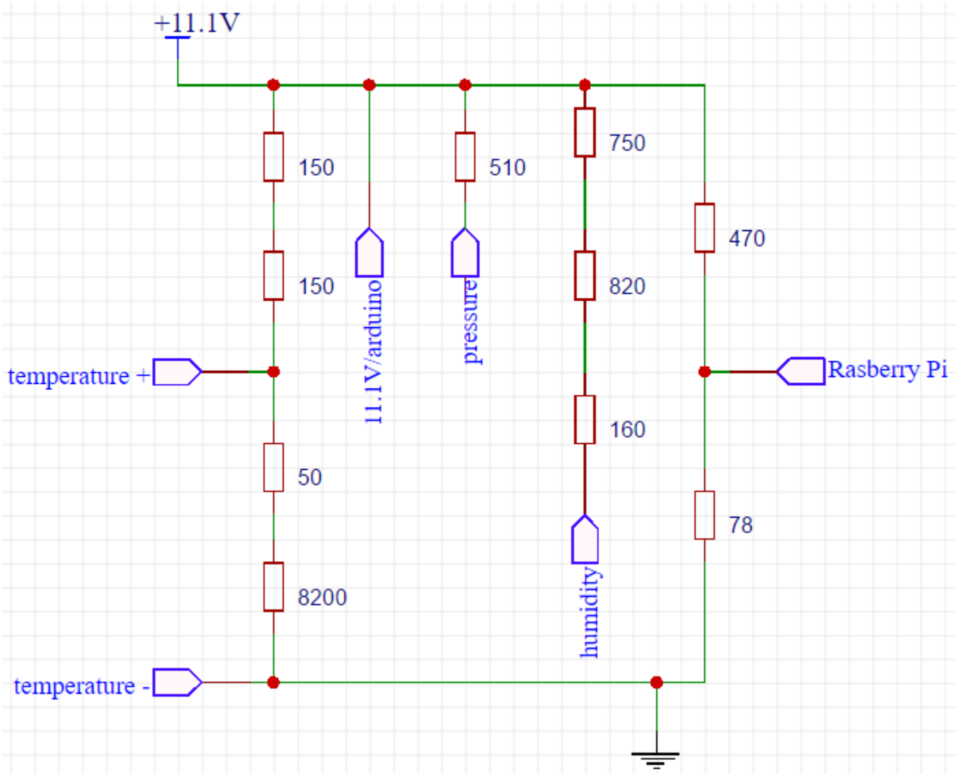
A LED indicates that the power is on, and a sliding switch allows turning it on or off. Everything is powered by the 2 batteries in parallel, 11.1V, 0.5 A (5000 mAh) each. 2 diodes prevent dysfunctions or one to charge the other.



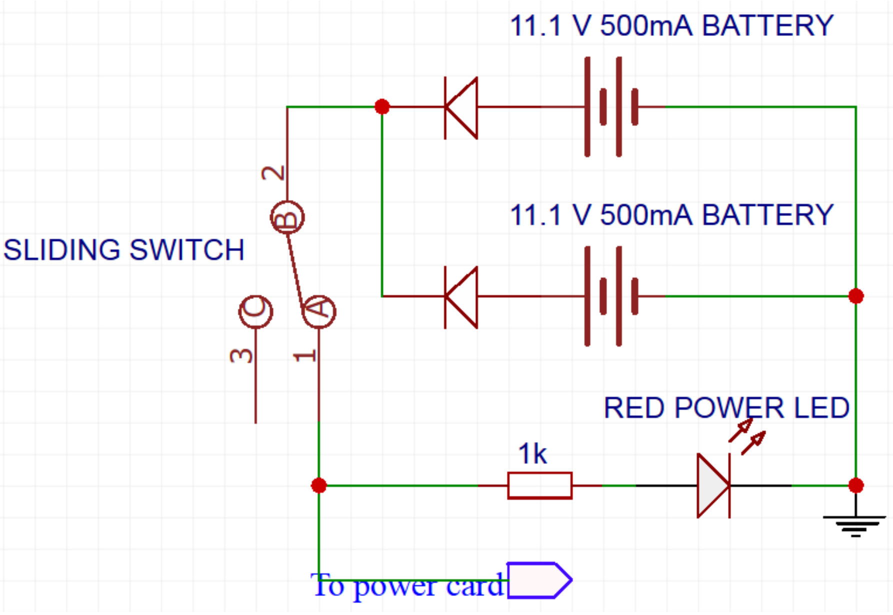
The power card redistributes power at the right voltages and amperages to the components: it steps down the input voltage to every used voltage using resistances. The Arduino powers and controls the servomotor and the blinking LED. The amplification card is powered, as the sensors and the Raspberry Pi Zero by the power card. The camera, using the flexible cable, is powered via the Raspberry Pi Zero.

D. Electronic cards schematics

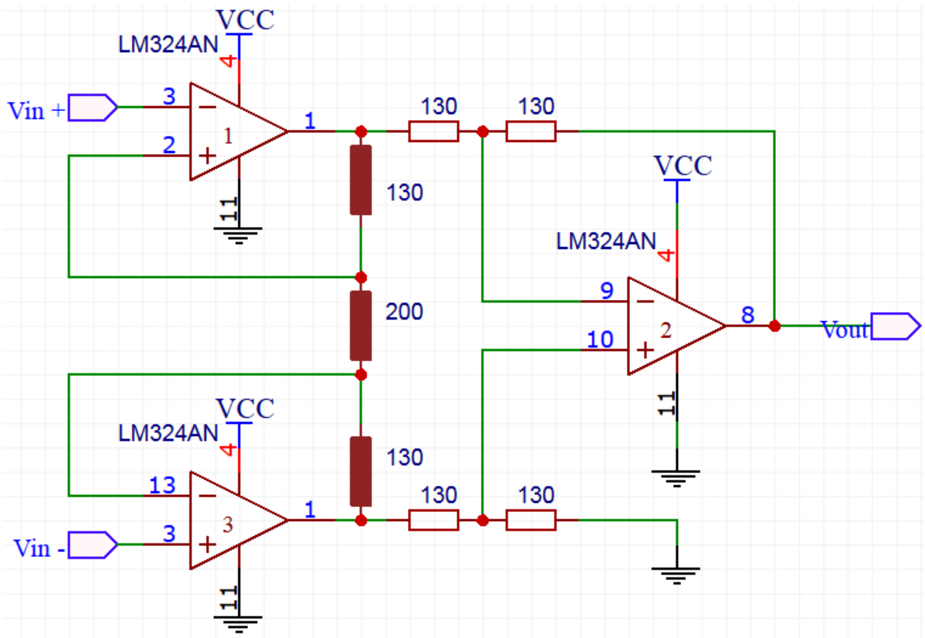
Power card



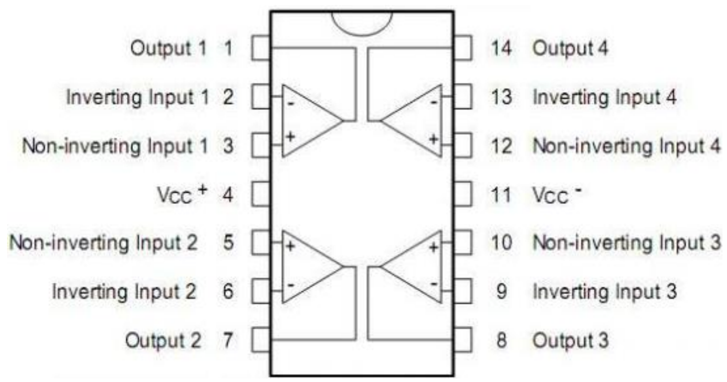
Power indicator and management



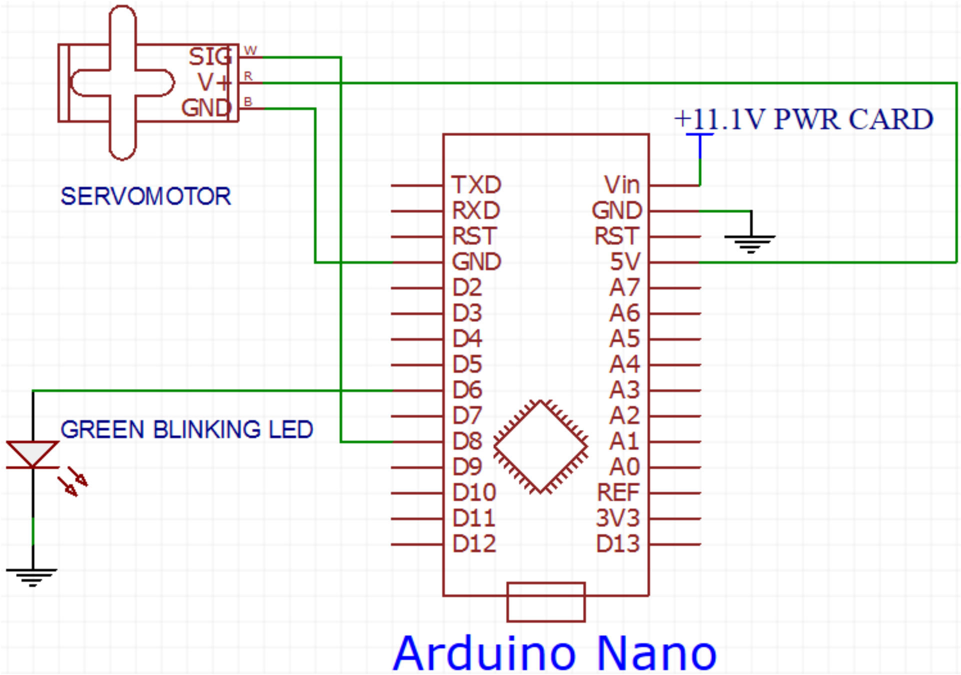
Amplification card



As the LM324AN is an integrated circuit gathering 4 operational amplifiers, the pin numbers next to the amplifiers' pins comes from the datasheet as shown on the schematic below.



Arduino and servomotor part



IV. Softwares

A. Camera recording software

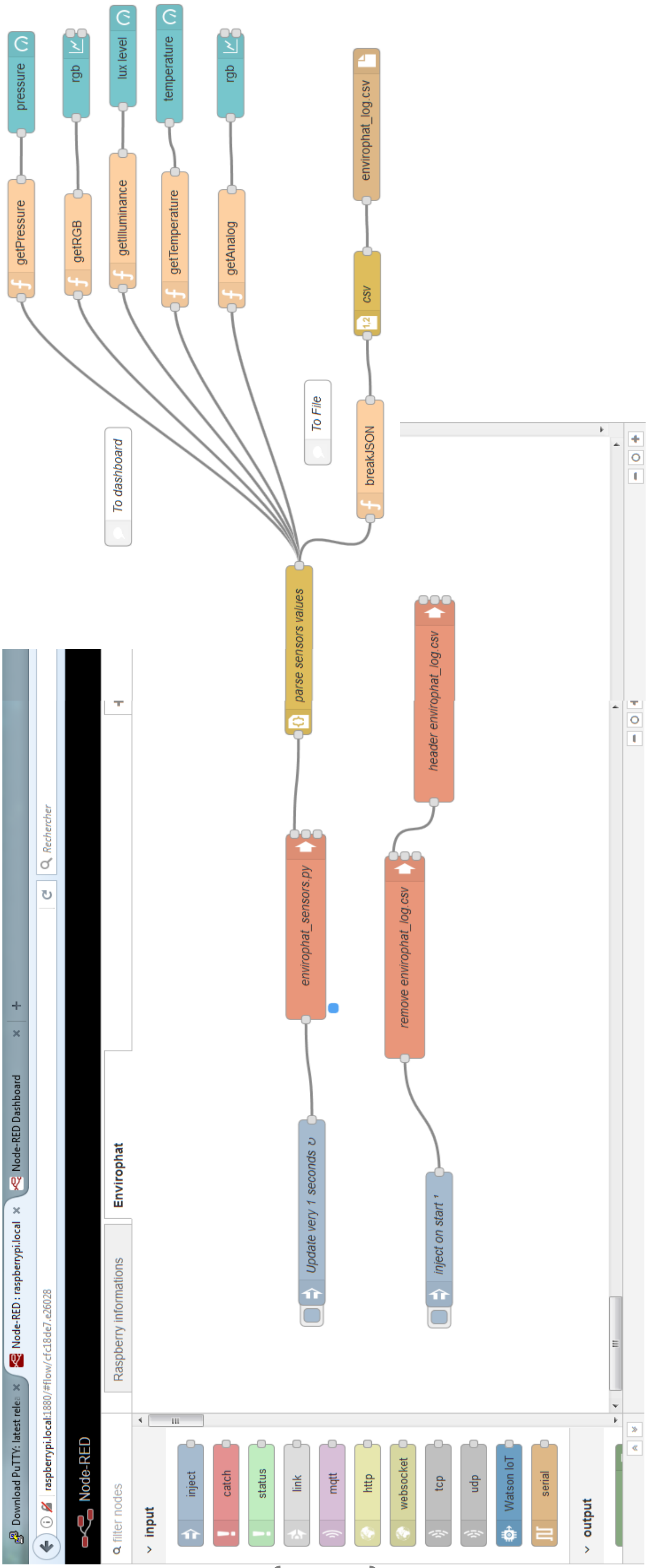
```
#! /usr/bin/python
# -*- coding: utf-8 -*-
import picamera
import time
import datetime as dt#data structure for date and time
import math

from envirophat import light, weather, motion, analog

recordingTime = 30 #seconds will be 20 minutes
numbersOfFiles = 1 #replaced by 40 for real situation


#
# Init camera
#
camera = picamera.PiCamera()#object/data structure
camera.resolution = (1280, 720)
camera.framerate = 60

camera.annotate_text_size = 16
camera.annotate_background = picamera.Color('black')
camera.annotate_text = dt.datetime.now().strftime('%Y-%m-%d %H:%M:%S')

#
# Magnetometer
#
north = motion.heading()


for f in range(numbersOfFiles) :
        #
        # Starting to record !
        #
        filename = str(int(time.time()*10))+ "_" + str(f) + '_swarog_camera.h264'
        camera.start_recording(filename)
        start = dt.datetime.now()

        #
        # looping and printing sensors values
        #
        while (dt.datetime.now() - start).seconds < recordingTime :
                rgb = light.rgb()
                analog_values = analog.read_all()
                x,y,z = motion.accelerometer()
                corr_heading = (motion.heading() - north) % 360
                camera.annotate_text = str(dt.datetime.now().strftime('%Y-%m-%d $
                camera.wait_recording(0.2)#values checking refreshing time
        camera.stop_recording()#end of program
        print('exported to ' + filename)

#EOF
```

B. Arduino software (Servomotor control)

```
#include <Servo.h>
int pos = 0;
int compteur=0;
int LED=5;
Servo motswarog;

// Setup code:

void setup() {
 motswarog.attach(9);
}

// Main code to loop:

void loop() {
 motswarog.write(90); //"Resting" position
 while(compteur<60)
 {
  digitalWrite(LED,HIGH); // High level (5V) – turns on the LED
  delay(500); // pause (500 ms)
  digitalWrite(LED,LOW); // Low level (0V) – turns off the LED
  delay(500); // pause (500ms)
  compteur++;
 }
 digitalWrite(LED,HIGH);
 delay(1080000); // 18 minutes before deployment
 motswarog.write(pos); // Upper compartment deployment
 delay(600000); // 10 minutes sensors exposure
 motswarog.write(90); // Back to initial position
 while(1) // Use here to stop the instructions while in a loop
 {
 }
}
```

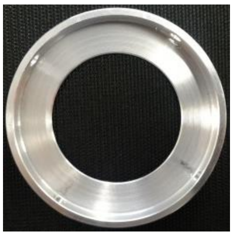
C. Node-RED visual code (measurements recording)

Here are the main boxes composing the main software of the Swarog project.


pressure
getPressure
rgb
getRGB
lux level
getIlluminance
temperature
getTemperature
rgb
getAnalog
To dashboard
To File
envirophat_log.csv
csv
breakJSON
parse sensors values
header envirophat_log.csv
envirophat_sensors.py
remove envirophat_log.csv
Update very 1 seconds ↻
inject on start '
Node-RED : raspberrypi.local
Node-RED Dashboard
raspberrypi.local:1880/#flow/cfc18de7.e26028
Node-RED
Raspberry informations
Envirophat
filter nodes
input
inject
catch
status
link
mqtt
http
websocket
tcp
udp
Watson IoT
serial
output

V. Mechanical architecture

The mechanical structure consists in a 3D printed structure (the internal PLA structure), a 3D-printed container (the PLA upper container), an aluminum ring and a PRCF tube (the protective external shell). The 3D printed structure fits inside the PRCF tube and is glued on the small free room between the 2 structures.



The aluminum ring is maintained by 4 screws to the PRCF tube (with 90° of angular interval one to another, as shown by the picture).

The 3D printed structure is the bone of the project: it contains the batteries, the Arduino Nano, the Raspberry Pi Zero, the camera, the Enviro pHat and the servomotor.

The tube is the skin of the project, protecting everything it contains. It is linked to the metallic ring with screws and nuts. A second nut is used to fix 4 cords to the parachute; the 4 others are glued to the PRCF tube and reinforced with tape.

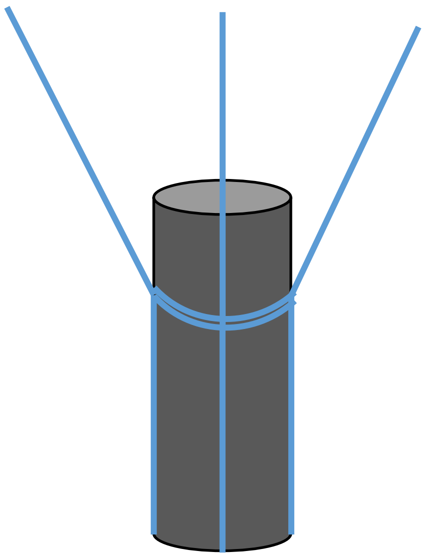
The upper container is maintained to the structure thanks to nylon reinforced wires, and deploys itself with the servomotor and the lever, as explained earlier.

All the 8 cords are tied to the parachute as shown by the picture. 4 of them are tied to the bolts and nuts at the bottom of the CanSat, and the 4 others are tied and glued to the top of the CanSat, protected by kevlar reinforced tape, as the following paragraph shows.

The cords are tied at the top of the CanSat to form a kind of harness around it, as we can see on the schematic on the right.

The lower aluminum ring is used as a base but it is also used to attach the cords to the lower part of the CanSat. 4 cords are attached this way, and 4 others are attached to the harness. This is reinforced with glue and Kevlar-reinforced adhesive tape.

This architecture has been tested, and was validated once the CanSat was launched for the drop test at the challenge's eve. It landed without any damage, showing that this can be reused for another similar CanSat.



VI. Work organization and management

A. Team organization

At first, we gathered the competences we had and then made two groups, depending on what we preferred to work on: the mechanic team and the electronic team.

The first mission of the mechanic team was to design on Catia software the main structure after defining with all the members its necessary characteristics during a brainstorming meeting. Then, after the first try, we selected the best ideas to improve the first 3D structure and edit the definitive structure. The external dimensions have been rectified as well with the reduction of the maximum external volume caused by the FUSEX rocket Solinas.

Their second mission was to create and adapt the PRCF tube so both could fit with a small free room, enough to fix it with glue. Once this was done, we focused on integrating the components, and also readjusting the internal structure to let the cables inside the 3D-printed structure. This was a very important step because due to the 3D-printing technique, we needed to reshape correctly the structure and also some parts to let screws and nails pass through.

Their last mission was to machine the PRCF tube by drilling where LEDs, the switch and the jack connector will be glued.

The first mission of the electronic team was to find practical solutions to power the different cards and elements of the CanSat, as well as defining which card would best suit the mission we wanted to accomplish. This included the search of the batteries, and the design of the power card. To find these, we summed up the power consumption of every card and element of the project.

Their second one was to program the mandatory mission part (the Arduino Nano) and the free mission part (the Raspberry Pi Zero). This was a defining, long and hard part of the project; this is why we needed some help for a part of it. The Arduino IDE was used to program the Arduino Nano using C-based language, the camera was programmed with Picamera software, and the Raspberry Pi runs under Raspbian OS and runs node-RED based software.

Their last mission was to integrate and test all the components while finishing the assembly of the CanSat.

B. Management

This project used participation as a key to let innovative ideas and solutions emerging. Every main topic and technical choices project was first debated in groups. As we all were active and implicated in our project, this kind of creative process was quite natural for us, and then, when we started the realization of parts of the project, we switched to a more directive form of management, using online tools. We used Dropbox mostly (a shared folder kept synchronized permanently to grant access to important files), mailboxes and Discord (a Skype-like platform where we posted messages about what kind of tasks were left, ideas and solutions, …). We used Discord as a forum, a chat, and a telephone.

We used a retro planning to have a global vision of the advances and the great steps, a detailed list of what has been done, what is under progress, and what was planned to do on Excel. We kept the budget in target funds via an Excel file, and also a dedicated folder on Dropbox.

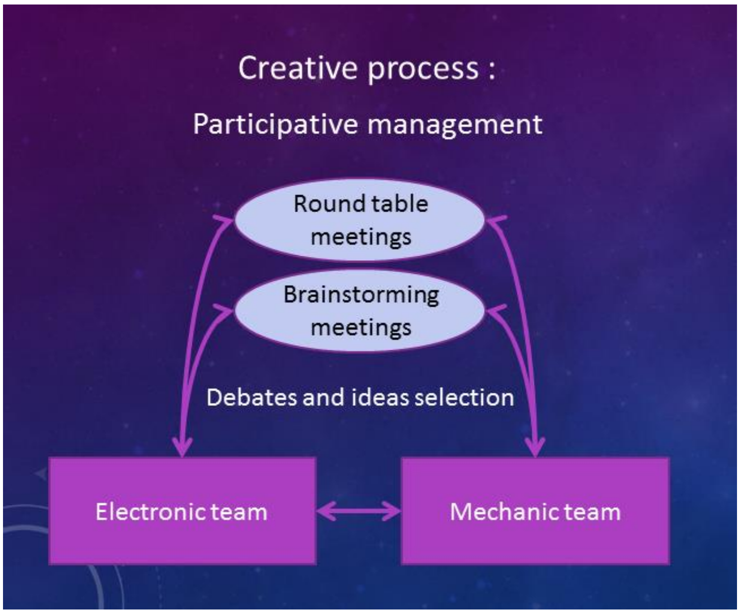
The repartition of work was quite natural, as everyone had a predilection topic to work on. As a first step, we made a great mindmap to organize and share our ideas. We all wrote what we had in mind before any choice. This was an effective way to find some useful ideas such as the 3D printed structure, and the deployment of some of the sensors.

First, the mechanic team provided ideas and concepts about the structure and the internal components layout, and the electronic team did the same for the components and firsts algorithms. Then the ideas were put together during a brainstorming review, and we selected the best ideas from it. We repeated the process until one choice seemed obvious and was validated by everyone.

Once the components were selected, we ordered them and tested them. We learned by ourselves how to test those sensors so we all could work on this part.

The cleansing of the 3D printed structure and the first integration tests were done by the mechanic team, while the electronic team made the power card and the amplification card on a breadboard to test and validate the schematics and components.

The integration was done time after time and some improvements were implemented during the process. We worked together and gave help when a problem or a delay happened.



C. External help

We needed help for some part of the project. First, we needed help with the main program. Mr Clément Bordeaux helped us coding it. Finally, we needed help during the final assembly part, Mr Marcel Billiau help us to integrate the components and to machine the internal 3D-printed structure. Others also helped and gave us advices through the development of the project, and they can be found on the "Acknowledgement" part of this report.

D. Planning

Here is the planning roughly detailed. As it started as an academic project, its specific steps are not detailed here; the realization begun after the report due date. (The planning is on the next page, the 21st due to lack of space on this one. It will also appear on the presentation.)

E. Budget

The initial budget was only 100€, but we negotiated an extra budget, bringing the definitive budget to 200€. This was a common budget with the Solinas project as our CanSat was to be launched with this FUSEX rocket. We bought some extra sensors, and we had to, because of the package we found online. We also had to buy an extra Arduino Nano, because the one we first used was nowhere to be found after the RCE3 Meeting, causing some delay to test the mandatory mission part. Despite these extra sensors, we still managed to maintain the good budget balance to finish our project in the meantime and without budget troubles. The total costs, with details can be found on the page after the planning, the 14th one.

The green components are the ones we had for free by our partners.

F. Partners

ESTACA FabLab

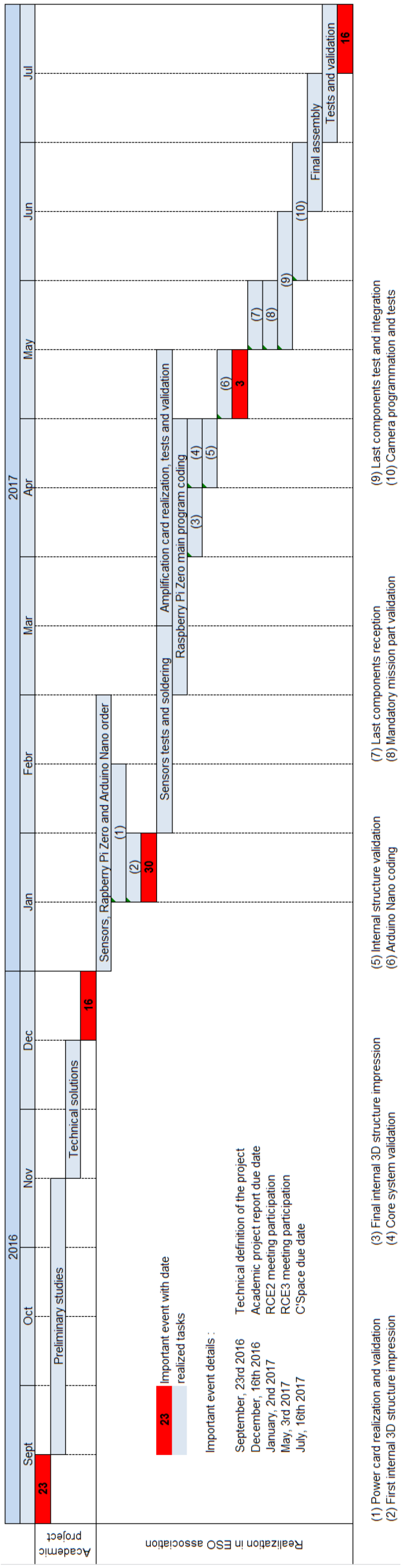
The ESTACA FabLab provided us advices and the first 3D printed internal structure we used on tests and validation purpose. The ESTACA FabLab also allowed us to use their 3D printer. We also made the upper container with one of the 3D printer available there.

ESTACA Workshop

The ESTACA Workshop provided us some tools and machined for us the aluminum ring based on our schematics.

Envie De Chute Libre 78

This association provided to the ESATACA Space Odyssey association an old parachute in which we cut at the right dimensions our CanSat's parachute.



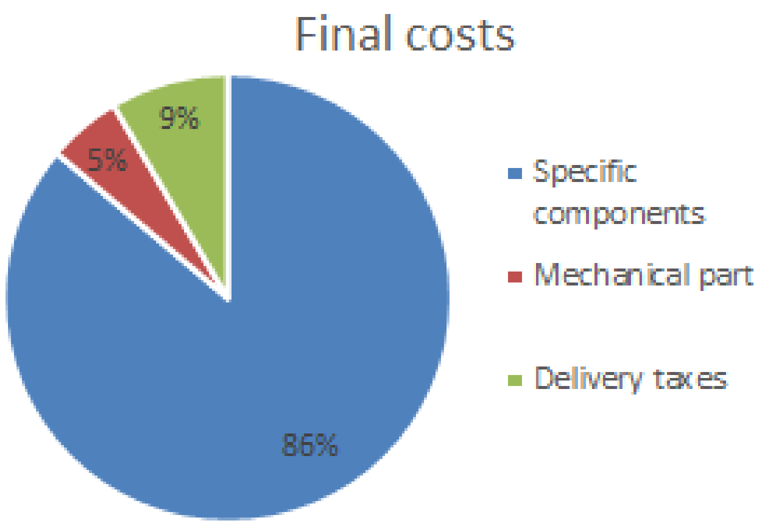
2016
2017
Sept
Oct
Nov
Dec
Jan
Febr
Mar
Apr
May
Jun
Jul
Academic project
Realization in ESO association
23
Preliminary studies
Technical solutions
16
Sensors, Rapberry Pi Zero and Arduino Nano order
(1)
(2)
30
Sensors tests and soldering
Amplification card realization, tests and validation
Raspberry Pi Zero main program coding
(3)
(4)
(5)
(6)
3
(7)
(8)
(9)
(10)
Final assembly
Tests and validation
16
23 Important event with date
realized tasks
Important event details :
September, 23rd 2016 Technical definition of the project
December, 16th 2016 Academic project report due date
January, 2nd 2017 RCE2 meeting participation
May, 3rd 2017 RCE3 meeting participation
July, 16th 2017 C'Space due date
(1) Power card realization and validation
(2) First internal 3D structure impression
(3) Final internal 3D structure impression
(4) Core system validation
(5) Internal structure validation
(6) Arduino Nano coding
(7) Last components reception
(8) Mandatory mission part validation
(9) Last components test and integration
(10) Camera programmation and tests

	Component	Quantity	Price (€)	
Specific components	Raspberry Pi Zero V.1.3	1	4,78	Components bought on internet
	Raspberry Pi Camera	1	29,13	
	Camera flexible cable	1	7,59	
	Enviro pHat	1	19,42	
	Arduino Nano	2	8,96	
	Micro SD card	1	9,90	
	Sensors	8	48,36	
	Servomotor	1	2,32	
	Batteries	2	30	
	Battery connector	2	2,32	
Basic components	LEDs *	2	0	Components already available in the workshop
	Diodes *	2	0	
	Resistances *	20	0	
	PCB *	1	0	
	Wires *	8	0	
	Operational Amplifiers *	2	0	
Mechanical part	3D printed structure	2	10	(1)
	Aluminum ring	1	0	(2)
	PRCF tube	1	0	The ESTACA workshop provided these for free
	Screws *	16	0	
	Nuts *	20	0	
	Nails *	4	0	
	Braces *	8	0	

* : Already available at the workshop
(1) The first try was provided by ESTACA FabLab
(2) The workshop machined it for free

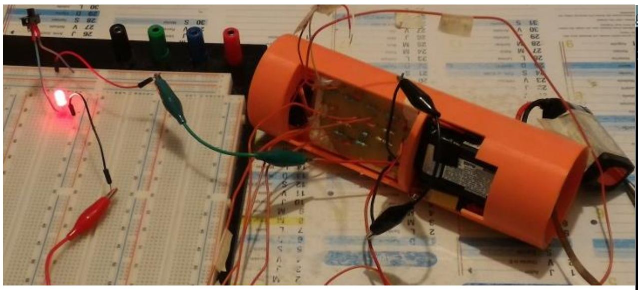
Total delivery taxes	16,11 €
Total costs	188,89 €

Initial budget	100 €
Extra budget	100 €
Extra costs	- 11,11 €



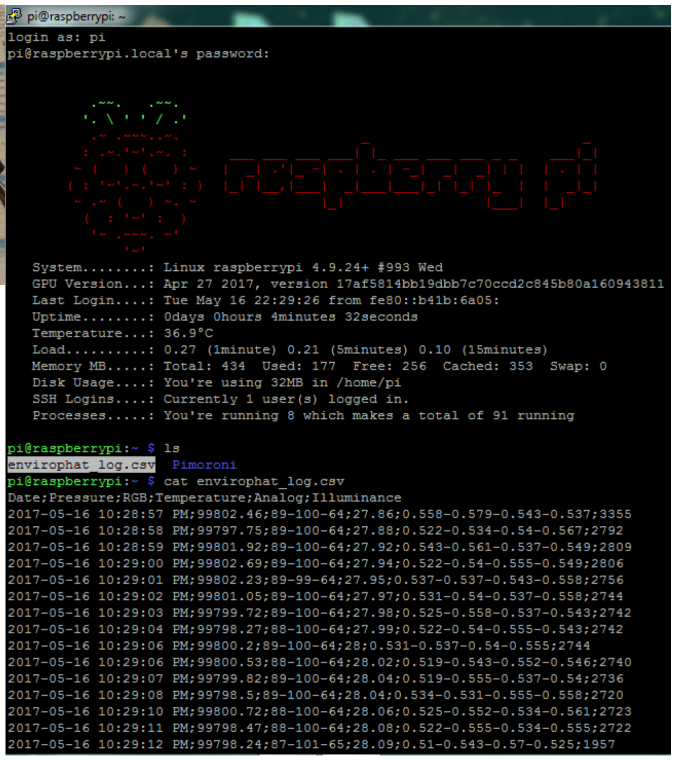
VII. Experiments and expected results

During the realization of the project, we tested each solution we applied to verify we haven't damaged the circuit or put a bug in a program. We tested every circuit before soldering them, and this was also the case for the Raspberry Pi Zero and the Arduino Nano we tested before starting to code. We also protected with thermic insulator the exposed parts of wires and components to prevent any short-circuit.



Power test with batteries ↑

Data recording test →

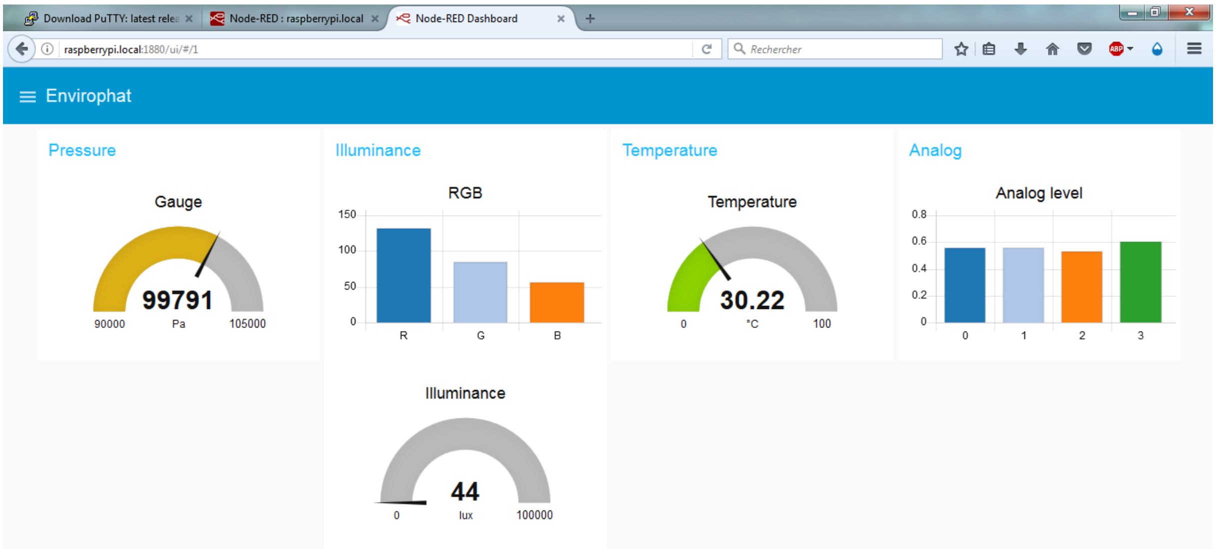


↑

The video is recorded in .h264 format, and a timestamp was added not to overwrite the previous records.

An overlay allows us to check in live the variations of some environmental parameters with a timestamp.

The overlay of Node-RED was used to test the reactivity and the functioning of the sensors. We could test in live with temperature tests (approaching a cold or a hot source), pressure test (testing it near a barometer), approaching a lamp to test the light sensor …



We expect the following values, for the launch conditions on the test field in Tarbes, France:

Data	Equipment	Expected values range
Temperature	Enviro pHat	Between 20°C to 40°C (direct reading)
Pressure	Enviro pHat	Between 800 hPa to 1300 hPa
Acceleration	Enviro pHat	Between -5 m/s and +5 m/s
Angular position	Enviro pHat	Between 0° and 359° during the mission
RGB color	Enviro pHat	Between 0 and 255 for each primary color
Video	Camera	Video of the ejection, fall, and landing
Humidity	DHT11 sensor	Between 20% to 80% (direct reading)
Temperature	P0K1.232.6W.B.007 sensor	Between 1,5V to 3V
Pressure	MPX2202A sensor	Between 1,5V to 3V

The CanSat during the final integration with windows to fit components, closed with Kevlar-reinforced tape.

VIII. Results of the experiment

A. Results

Unfortunately, we don't have any file of the drop test or the final launch. As the visual indicators (the two LEDs) were working, we can say that the mandatory mission was successful, but the free mission failed. The root causes can be 3 main incidents, or a combination of the followings :

- Cable problems during the integration
- False manipulation before the launch
- Power card failure

For the first identified possible cause, because the cables were close to the inner surface of the external shell, some wires may have been disconnected in the area of the Raspberry Pi Zero. The power cables soldered to the Raspberry Pi Zero were thin cables, so even if they were glued not to teared them off, they could have been twisted inside the external shell or could have been damaged.

The second possible cause is because no one from the team could stay during the whole campaign more than the week-end due to summer internships. As documents, presentations, scientific extract and explanations have been provided, the presentations were done by an external member, and the final assembly by another member. We can't be sure every remaining stage of the final assembly worked perfectly, and hence this source of doubt is a potential root cause.

The last possible root cause comes from the power card itself : it is made of basic voltage dividers, without any regulator. No regulator was possible inside because of the heat it would have create : there were risks of melting the PLA internal structure and risks of components over-heating beside false readings of the temperature recorded by the Enviro pHat.

The delivered voltage was precisely 5.07 V for the Raspberry Pi Zero, and with a power consumption of about 180mA maximum, the Raspberry Pi Zero should have booted, but as we had no data, a problem occurred before it could start gathering data. As it can still work properly, it hasn't been exposed to over voltage / current. So maybe the voltage decreased too quickly to power the Raspberry Pi Zero or there was a problem with the +5V and GND GPIOs (General Purpose Inputs Outputs).

Further investigations are required to identify the root cause(s), by testing the electrical continuity of the wiring and replacing the power source of the Raspberry Pi Zero.

B. Potential future improvements

We want to carry on the project, with a learning center for new members or people willing to learn technologies from Swarog project, and an innovative part where the team can improve the concept and present the project to the CanSat challenge.

The main improvements of the future version of Swarog can be adding a GPS, different sensors, or adding wireless data transmission. We can also use other materials for the internal structure (like aluminum) to gain weight and space inside the CanSat. The project can also be optimized: we decided to separate the mandatory mission and the free mission, but we could have piloted the servomotor from the Raspberry Pi Zero. In addition, we can choose smaller components and the latest version of the Raspberry Pi Zero, the Raspberry Pi Zero W (W for wireless) which includes Wi-Fi. Our ambition is to use Swarog as a base project for future CanSat teams inside ESO association. This project will be presented during the presentation and the vote for the projects of the new academic year.

IX. Problems we encountered

A. Mechanical issues

The first problem we came across was a mechanical one: the external dimensions had to be reduces to fit inside the FUSEX. To avoid any further problem about this matter, we decided to take the minimum external diameter with the maximum height so the parachute could also fit inside.

Another problem we quickly managed to solve was about the internal structure: the first try with reinforcements to attach parachute's cords to was not tough enough at some areas, and the reinforcements were not solid enough to maintain the CanSat flying without breaking. So we decided to attach the cords to 2 elements: the screws and the external PRCF tube.

Finally, the integration was the last mechanical problem: we simply re-machined the internal structure so every component had a large enough place to fit in. This was quite easy to solve, but took many time to be done perfectly, as it was harder to machine than expected.

B. Electronic issues

With the beginning of the prototype, we had some problems with the breadboard realization, and simply testing again with different wires were all it took to make the power card run. Another problem occurred with the diodes: it was stepping down voltage from the batteries, so we modified the resistances impacted by this change.

To solve the amplification card problem, we had to make extra steps. We first identified the defective component by testing them individually, and we changed the OP-AMP to make a first amplification stage working. For the second one, we also had to change a resistance because the input voltage was not the required one. At this stage, we implemented it with the sensors, and saw the signal was still too high to be read by the ADC. This last problem was solved by adding a voltage divider bridge at the output of the card.

C. Coding issues

We had some problems with the programming of the Raspberry Pi Zero. As it only had a micro USB data connector, we had to use the OTG (On The Go) mode to access it via USB, using SSH protocol. First, we installed Raspbian OS, and Node-RED to enable visual scripting programming on the Raspberry Pi Zero. The first configuration took time, but was working fine with the first version of the python script we use to record data. The second time, it didn't work on another computer. After a long time debugging Node-RED, we also installed on the computer PuTTY, a free SSH client to access the Raspberry Pi Zero from the computer to simplify the maintenance.

The last problem we had was about the camera: we needed to download it from internet, but none of our try worked and the Raspberry Pi Zero couldn't connect itself to internet despite the right parameters. Our last chance was to manually install Picamera on the micro SD card. We used WinSCP to upload the program, and we finished to code it using Node-RED and a python script.

X. Swarog Project Checklist

30 minutes to launch	
Visual checks	☐
External carbon protective shell	
No fragment	☐
No hit	☐
No hole (except known drillings)	☐
Upper container	
No fragment	☐
No hit	☐
No hole (except known drillings)	☐
Cords	
Good condition, uncut.	☐
Parachute	
Good condition, uncut.	☐
Camera case	
Protected with plastic film	☐
Uncut, clear, clean plastic film	☐
Labels	
Labels "ON" (switch and red LED), "OFF" (switch), "In case it's lost", "Largage " (green LED), and logo	☐

20 minutes to launch	
Functional checks	☐
Turning the CanSat on	
Slide the switch on "ON" position	☐
Red LED "ON" lit	☐
"dropped in flight" configuration test	
Unplug the jack plug	☐
Green LED "Largage" blinking (check for 5 seconds)	☐

15 minutes to launch	
Reset	☐
Reset to launch configuration	
Plug the jack plug	☐
"Largage" green LED off	☐
Turning the CanSat off	
Slide the switch on "OFF" position	☐
Red LED "ON" off	☐

10 minutes to launch	
Setting the CanSat up	☐
Wrap the parachute around the CanSat	☐
Slide the switch on "ON" position	☐
Put the CanSat in the rocket slot	☐
Plug the CanSat to the rocket	☐

XI. Conclusion

The CanSat is now finished. The different elements are working fine and the whole project works perfectly. The last stage of the project will be its launch and its mission. This project was a fantastic opportunity to design and build a full functional project starting from zero. This was a challenge to make it real with our studies, our other works and the delays to get everything delivered, and the different problems we encountered. We managed to finish all this on time.

This working prototype will be another concept to explore remotely other worlds, and we hope it will be a source of inspiration for other students or projects makers.

XII. Acknowledgement

We would like to thank for their help and their participation:

Mr. Pierre-Emmanuel Mangin and **FabLab ESTACA** for the 3D printing technology and the advices

Mr. Claude Caruel for the supervision of the school project part

ESO association for the advices, materials and the tools

Envie De Chute Libre 78 for the parachute cloth

Mr. Clément Bordeau for all the help with his coding skills and the advices

Mr. Marcel Billiau for all the help for the assembly

Mr. Marcel Tran-Phu-Tri for all the help and the advices

Ms. Françoise Massemin for the seams of the parachute

Mr. Marin Sangouard for the help at the beginning of the project