El propósito principal de este proyecto es el diseño e implementación de de una plataforma de calibración de un sensor Barométrico usando en un globo estratosférico. Además de dicha plataforma, un prototipo electrónico ha sido desarrollado para una satisfatoria consecución de los datos de Presión y Altura. Junto con un simulador que permite estimar la altura máxima de vuelo y las condiciones del mismo.

Álvaro Hidalgo Romera nació en Granada, España, en Agosto de 1991. Con este trabajo finaliza el Grado en Física de la Universidad de Granada

Andrés María Roldán Aranda es el profesor ingeniero a cargo del presente proyecto, así como el tutor del alumno. Actualmente es profesor del departamento de Electrónica y Tecnología de Computadores de la Universidad de Granada.

TRABAJO FIN DE GRADO

Diseño e Implementación de Plataforma de Calibración de Altimétro Para Globo Estratosférico

Álvaro Hidalgo Romera

Física

2016/17

UNIVERSIDAD DE GRANADA

Grado en Física

TRABAJO FIN DE GRADO

Diseño e Implementación de Plataforma de Calibración de Altimétro Para Globo Estratosférico

Álvaro Hidalgo Romera

Año académico 2016/2017

Tutor: Andrés María Roldán Aranda

Trabajo Fin de Grado en Física

Diseño e implementación de plataforma de calibración de altímetro para globo estratosférico

Álvaro Hidalgo Romera
Universidad de Granada
Junio de 2017

UNIVERSIDAD
DE GRANADA

Tutor: Andrés Roldán Aranda
Departamento de Electrónica y computadores
Universidad de Granada

Acknowledgments:

I would like to thank to my brother and friends from ”‘Choriviota y sus cochinos”’ and ”‘miami lovers”’ Also to all the people I have met in Kaiserlautern.

Finally, I would like to thank to my advisor, Andrés María Roldán Aranda for his provided timely and instructive comments, not only for this project, but for my future professional life.

Abstract

La altura a la que se eleva un globo estratosférico durante un vuelo se puede medir de manera directa a través de un altímetro basado en un sensor barométrico. La medida de la altura del globo mediante este sensor requiere de una calibración previa en una cámara de vacío donde se pueda reproducir las condiciones de presión y realizar el ajuste del sensor.

La altura alcanzada durante el vuelo depende del tamaño del globo, del peso del conjunto globo, sistema de retención y góndola con instrumental y masa y tipo del gas de impulsión. Se va a realizar una aplicación en Python que permita estimar la altura máxima esperada de vuelo y las condiciones del mismo. Se va a realizar una aplicación en Python que permita estimar la altura máxima esperada de vuelo y las condiciones del mismo.

Contents

1 Introduction

High Altitude Balloons provide an exceptional and low cost way to accomplish missions in near-space environment. In the last years, the potential use of high altitude balloons as observation platforms for atmospheric studies studies or research purposes has attracted increasing interest.

The general theory of how a high altitude balloon works is quite simple. A payload is appended to a balloon which is filled with helium. Once the balloon is released it will rise because the Helium is lighter than air pulling the payload with it. As the balloon rises the outside pressure decreases leading the helium inside the balloon to expand, which causes the balloon to stretch. Although the latex of which the balloon is made is really stretchy, it gets to a point in which it can't stretch any more and bursts. Once this event happens, the payload begins to fall to earth pulled by gravity. As it falls the parachute opens up slowing its descent and it gently glides down to Earth[14].

Figure 1: High altitude sounding balloon [8].

1.1 High Altitudes Balloon

The general structure of the high altitude balloon can be seen in Figure 2. The scheme of it is quite simple. It has the balloon at the top, the parachute between the balloon, with the payload at the bottom.

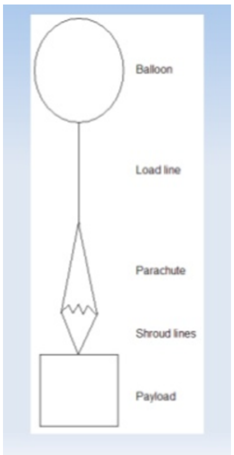

Figure 2: High Altitude balloon scheme [8].

Subject to the size of the payload, scientific objectives, and the the scale program, they are many types of balloons.

Latex balloons, known as $sounding or weather balloons$ as well, are designed to reach an altitude between $35-40km$ and burst, after which a parachute is deployed to safely carry the payload back to Earth. They are low-cost and easily available. The flight of this kind of balloons should last for a few hours.

There are also thin plastic balloons, called $zero-pressure balloons$, which can float at high altitudes for days. THey can reach altitudes around the $50km$ and carry payloads of several tons. They are very expensive for a college budget, requiring special manufacturing and launch towers. They are normally employed by large-scale institutions.[3]

Figure 3: Zero-pressure Balloon [9].

1.1.1 Payload Module

Inside the payload module can be placed whatever is needed to accomplish the purpose of the mission. The payload should be protected from low temperatures and shocks.

There are endless possibilities to what can be flown into the atmosphere. Most balloons have a sensor system to record measurements. a. This sensor system will usually use smallest possible space and consume the least amount of power possible. It shall be composed of a microcontroller, a GPS, and a pressure sensor

The GPS, because of the restrictions[10] it is used for recovery purpose of the payload. So as to measure the altitude of the balloon during the flight we need a pressure sensor.[15]

1.1.2 Profits of High altitude balloons

They are several advantages of using high altitude balloons to perform scientific observations or technological experiments over using rockets and satellites that could be summarized as follows:

1. Low cost: High altitude balloon provide a affordable and attainable way to study atmosphere, For a reasonable price one can run a simple prototype and satisfy his own curiosity. In relation of more sophisticate balloon such as the zero-pressure, since the launch facilities are particularly basic, its operation and maintenance costs are also substantially lower, on a rough estimation of the basics costs of the main balloon, and common equipment, the outlay required for a balloon experiment is at least a factor of 10 lower than that required for a small sounding rocket, at least a factor of 100 lower than that required for a large rocket and a small satellite, and at least a factor of a 1,000 lower than that required for a large satellite.

2. Able to mount heavy payloads: Since it was said, the zero-pressure balloon can carry payloads of several tons. Furthermore, in contrast with satellites, which are transported in the restricted space of nose cones of rockets, balloons permit a lot of flexibility in the external shape of experimental equipment.

3. Recovery and reuse of the payload : Since the observational instrumentation and experimental equipment mounted on a balloon can return to earth by parachute and be recovered, the same equipment can be used multiple times, resulting in significant cost benefits. This is advantageous when it is desired to collect consistent data using the same equipment and when it is desired to perform better observations and experiments by making improvements to the recovered payload.

4. High flexibility in launch site selection: Since balloon facilities are relatively small scale, balloons may be launched from standard mid-latitude regions.

5. "In-situ observations" of the upper atmosphere: Only balloons are capable of performing "in-situ observations" of the atmosphere at altitudes of 30–40 km in the stratosphere. Aircraft cannot reach these altitudes, and sounding rockets pass through them in a short time. Atmospheric observations from an orbiting satellite are limited to remote sensing measurements made from a long distance.

6. Pilot experiments for space technologies: Scientific balloons for research and development of rocket and satellite technology as well. For example, in the first stage of development of a space vehicle that returns to the earth from orbit, an experiment may be conducted into investigating its high-speed flight characteristics by dropping a model from balloon altitude and allowing it to free fall at high speed.

7. Means to develop the next generation of researchers: These days, it is widely recognized that fostering and securing students and young researchers who are interested in space science and technology is a critical issue. Balloon experiments are highly valuable for such a purpose. Everyone who participates in a balloon experiment can be involved in many different kinds of activities at every stage of the project. Usually the participant is involved in the development of on-board systems from its design and development to the final test and launch. However, with satellites there is little such opportunity, since the project is usually large-scale and expensive, and consequently tasks are performed in narrow areas of specialization.[1]

Now the physics principles that make the balloon rise will be presented.

1.2 Principle of Buoyancy

The fact that balloons float in the atmosphere is based on the principle of buoyancy discovered by Archimedes in the third century B.C. It states that "an object submerged in a fluid experiences an upward force that is equal to the weight of the same volume of fluid"

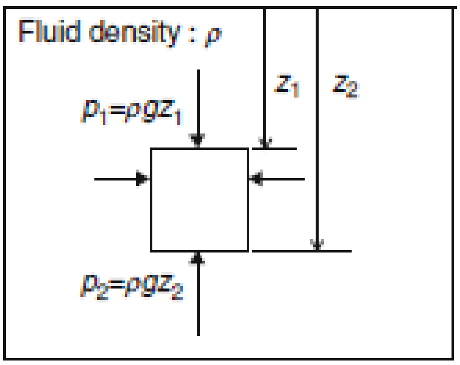

Figure 4: Buoyancy Principle [1].

As shown in Fig. 2, a rectangular solid is lowered into a fluid. According to Pascal's law, the object is subject to pressure from the fluid pressing on its surfaces in directions normal to its surfaces. The pressures on the sides are counterbalanced, so that the pressures acting on the upper and lower surfaces p_1 and p_2 are given by

$$p_1 = \rho g z_1 \tag{1.1}$$

$$p_2 = \rho g z_2 \tag{1.2}$$

where z_1 and z_2 are the respective depths of the upper and lower surfaces, ρ is the fluid density, and g is the acceleration due to gravity. Hence, if we take the area of the upper and lower surfaces to be S and the volume of the rectangular body to be V, the total sum of forces F in the upward direction that is imparted to the rectangular solid by the fluid is given by [1]

$$F = p_2 S - p_1 S = \rho g (z_2 - z_1) S = \rho g V \tag{1.3}$$

1.2.1 Effect of buoyancy from a gas

Consider the increase in the upward force if a buoyant gas of volume V is injected into a balloon. If we take ρ_a and $\rho_g$to be the densities of the external air and the internal buoyant gas, respectively, the increase in upward force acting on the balloon ΔF becomes

$$\Delta F = (\rho_a - \rho_g) g V \tag{1.4}$$

The $\rho_a g V$ term on the right-hand side is the buoyant force from Archimedes' principle, and $\rho_g g V$is the weight of the gas inside the balloon. Thus, the contribution of the gas to the upward force is given by the difference between the density of the external air and that of the internal buoyant gas. For this reason, in subsequent descriptions, $\Delta \rho g V$is referred to as the effective buoyant force and it is a function of the difference in the densities $\Delta \rho = \rho_a - \rho_g$.[1]

1.3 Barometric Formula

The barometric formula is used to model how pressure of the air changes with altitude. In spite of its simplicity, it applies reasonably well to the purpose of this project.

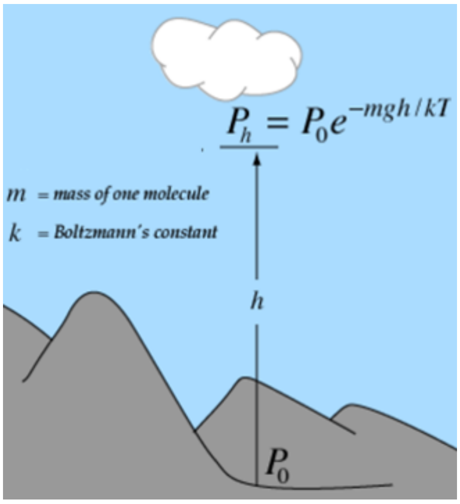

Figure 5: Barometric Formula[16].

They are several ways of arriving at the barometric formula. We derived it from the Hydrostatic derivation. Considering a still gas contained in a vessel oh height H, in equilibrium the pressure in a given height z is

$$P(z) = P(H) + M(z) g \tag{1.5}$$

Where $M(z)$ is the mass of the gas in a column of unit area that extends from z to H

$$M(z) = \int_H^z \rho_m(u) du \tag{1.6}$$

Where ρ_m is the mass density.

From the perfect gas equation

$$PV = NKT \tag{1.7}$$

Where N is the number of molecules contained in the volume V, one can obtains.

$$\rho_m(z) = \frac{Nm}{V} = \frac{mP(z)}{KT} \tag{1.8}$$

obtaining the following integral equation:

$$p(z) = P(H) + \frac{m}{k} \int_H^z \frac{gp(u)}{T} du \tag{1.9}$$

whose solution is:

$$P(z) = P_0 \cdot e^{-Mgz/kT} \tag{1.10}$$

Where P_0 is the pressure at see level[2]

1.4 Pressure sensor and LCD screen

In this section will be presented the devices used for the electronic prototype As mentioned in previous sections, GPS can not handle high altitudes. This is a significant problem because it is necessaty to know the altitude of the balloon for reliable tracking. Without the altitude reading one cannot know if the balloon is going up, if it has reached the highest point, or if it is coming down.[13]

The pressure sensor selected was the $BMP085 pressuresensor$ in the module $gy - 65$.

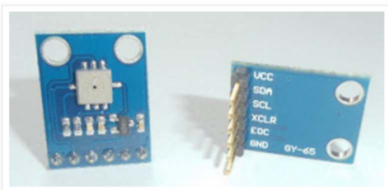

Figure 6: GY-65 BMP085 module [27].

1.4.1 BMP085

It is a new generation oh high precision digital pressure sensors for consumer applications. Some characteristics of the device are now presented[26]

- supply voltage: 1.8 3.6V
- LCC8 package:
  - Robust
  - Ceramic lead-less chip carrier (LCC)Package
  - Small footprint 5.0mmx5.00mm

  - super-flat $1.2 mm height$

- Low power $5\mu A$ at 1 sample/ sec in standard mode
- temperature measurement included
- I^2C interface

The way in which the sensor measures pressure and temperature is shown it the following flowchart

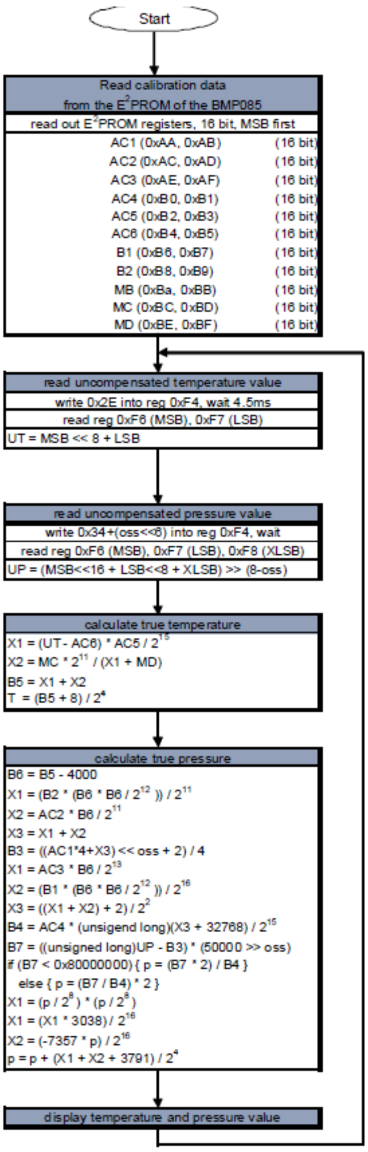

Figure 7: BMP085 Flowchart [26].

Once the pressure is measured, the height is calculated by the Barometric formula

1.4.2 LCD Screen

A display was required in order to show the data measured. Is used an LCD screen of 2 lines by 16 characters interfaced to an I^2C daughter board. This $I2C$ module only requires 2 data connections (SDA and SCL) besides a power of $5V$ and GND to operate. [28]

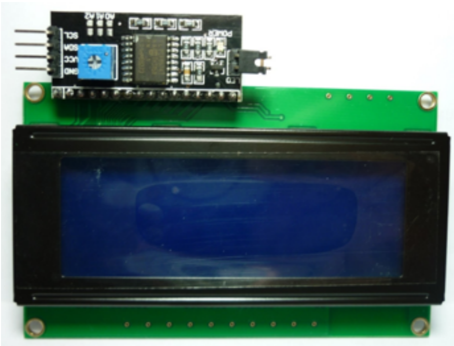

Figure 8: 2x16 LCD Screen [29].

1.5 Electronic and Communication Tools

This section introduces the microcontroller used, and the protocols of communications used between the different devices.

1.5.1 Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.[17]

The pinout of the arduino board is

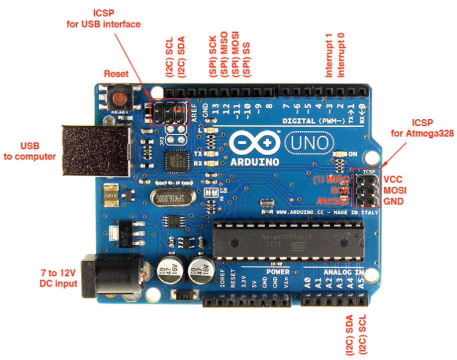

Figure 9: Arduino pinout [18].

1.5.2 I^2C Communication

I^2C is a multi-master, multi-slave serial computer bus invented by Phillips Semiconductor. It is used for attaching lower-speed peripheral to proccesors and microcontollers.

Is uses only two bidirectional open-drain lines, Serial Data Line (SDA) and Serial clock Line (SCL) pulled up with resistors. Besides voltage and ground.

- SDA is used to transmit the data
- SCL is an asynchronous clock that point out when read the data

Each device connected to the bus has an exclusive direction of 7 bits.

One of them has to act as master, i.e, it controls the clock which does not require a specific speed. It is multi-master, which means that the master can change, but only one can be acivated and the rest must configure as slave. The master provides a protocol of control and collision detection.

The main idea is that every component is connected in parallel to the two lines of the bus, SDA and SCL, as one can see in figure 10.

In the Arduino UNO board, the SDA and SCL pins can be found in the analogic pins A4 and A5 respectively.[19]

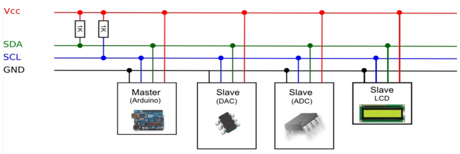

Figure 10: I^2C Communitacion [19].

1.5.3 Serial Communication

Serial communication is the process of sending data one bit at a time, sequentially, over a communication channel or computer bus.

Serial communication is used for all long-haul communication and most computer networks, serial computer buses are becoming more common even at shorter distances, as improved signal integrity and transmission speeds in newer serial technologies have begun to outweigh the parallel bus's advantage of simplicity [12]

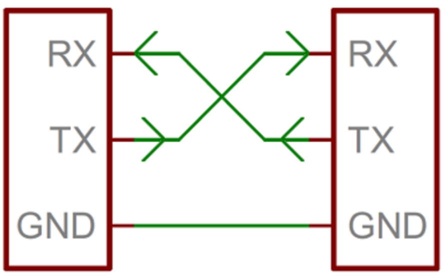

Figure 11: Serial Communication [12].

2 Methodology

As mentioned, the height of the balloon will be measured through the barometric sensor which requires a previous calibration. In order to calibrate the sensor, a math model was required to determine the change of pressure with height. The PTC Mathcad Prime software was used to develop this mathetmatical model. Furthermore, it was necessary to design and build a vacuum chamber.

An electronic prototype was developed to record these measurements.

Lastly, a burst simulator was created with Python to simulate and determine the maximum height that the balloon can reach.

2.1 MathCAD Simulation

Knowledge of quantities, such as pressure and temperature, are required for proper calibration of pressure altimeters. Because the real atmosphere doesn't stay constant, it was needed to use a hypothetical model as an approximation of what was expected.

For this purpose software PTC MathCad Prime was used.

PTC Mathcad is engineering math software that allows one to present all of the calculations computed with plots, graphs, text, and images in a single document.[20]

To model temperature, bibliographic research was required to find the most suitable model for reference. The model chosen was the 1976 Standard atmosphere[7], which divides the atmosphere into layers with an assumed linear distribution of absolute temperature with altitude and ignores the fact that on any given day it could be colder or hotter than standard, or have a higher or lower pressure than standard. There is no weather variations in the model at all.

6 different layers can be distinguished. Due to the the desired range of the burst

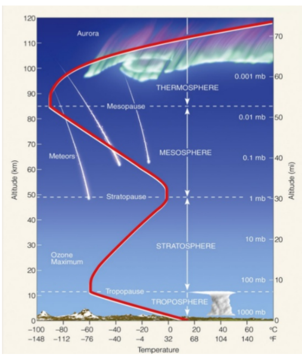

Figure 12: Atmosphere Layers [8].

height, the temperature was simulated up to 37km.

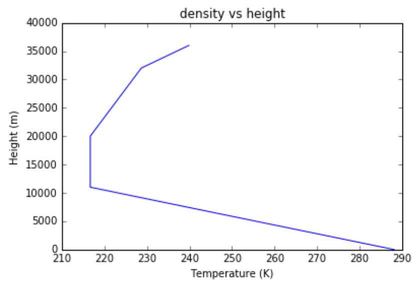

Figure 13: Temperature Simulation

To model pressure, the 1976 Standard Atmosphere model was not used. Instead, the model based on the barometric formula was used. The reason this model was used is because the pressure sensor uses a similar algorithm based on this model to calculate height, according with the data sheet.[26]

2.2 Vacuum chamber

In order to be able to produce a vacuum, a vacuum pump was borrowed from the Granasat Aerospace Rearch group

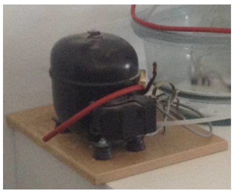

Figure 14: Vacuum Pump

At this point of the project it was required to design and build a vacuum chamber. Two different models were designed.

First, a vacuum desiccator with a RIN valve[35] was purchased. The valve was removed from the design, because communication between the microcontroller and the PC was needed. Hence, to accomplish this, a rubber lid [36] with two holes was placed there instead. A couple of copper tubes, provided by the "'Facultad de ciencias"', were inserted across these two holes. Due to the difference between the radius of the holes and the radius of the copper tubes, patience and oil was required to fulfill this task.

The purpose here was to connect the vacuum pump through one of them, and the electronic prototype through the other.

In order to get a proper connection between the sensor and the microcontroller, which was placed outside the vacuum chamber, five long male-female wires were placed through the copper tube.

Soldering skills where required to create such wires. The male pin header and the female pin header were put in both ends of a wire long enough for our needs.

Due to the copper tubes, vacuum couldn't be produce. To achieve the vacuum, epoxi glue was used at one end of the tube. With this, air losses were reduced to a minimum, enabling us to handle "'high altitudes".

The electronic prototype, with the exception of the arduino board, were placed inside the vacuum chamber. According to the documentation, the first vacuum chamber requires 14L of volume. The price of the chamber was 250 euros.

Figure 15: Vacuum Chamber First Model.

For the second model a bell jar with a top knob was purchased [37]. Because the bottom of the bell was not covered, a piece of wood large enough was requested from the carpentry service of the "'Facultad de Ciencias"'. A couple of holes where made inside the radius of the bell in order to pass through them another two copper tubes. Again, both of them were provided by the services of "'Facultad de ciencias"'.

As it was done for the first model, epoxi glue was applied on both sides of the wood so as to not have air losses.

Another four more holes were made at the four corners of the wood. Four rubber legs were set in those corners.

In order to make the six holes, static and hand-use drilling skills were developed in the GranaSat Lab.

Once the platform was completed, only four male-female wires were passed through the copper tube. The reason four cables were required is because, for this model, everything except the pressure sensor had to be placed outside of the vacuum chamber for future improvements and better visibility.

So as to not have air losses, wax for the vacuum [38] was applied around the radius of the bell jar.

The company did not provide the volume capacity of the bell, but it did provide the diameter and height of the bell.

To calculate the volume capacity, it was assumed the bell is a cylinder with an approximate capacity of 20L. The price of the bell was 180 euros.

Figure 16: Vacuum Chamber Second Model.

2.3 Electronic prototype

As soon as it was possible to create a vacuum, the pressure within the bell was able to be manipulated. Therefore, it was necessary to design a prototype to measure the pressure.

The microcontroller selected was an Arduino UNO [17].

Along this part of the project, three different phases could be differentiated:

2.3.1 Knowing the sensor and Early design

The steps can be laid out as follows:

- Firstly, it was need to know how the bmp085 works. It was possible to do so after a deep reading of the data sheet.[26]
- The next step was to be able to read the data of pressure. To achieve this, the official library of the sensor was used.[30]
- Once the pressure reading was taken, an atltitude reading was required. That was quite simple to do, because the library has its own instructions for this purpose.
- With the pressure and altitude readings, a display was required, so the user could see the data easily. To accomplish this, a 16x2 LCD screen was used.[28]
- The LCD screen and the bmp085 work with I^2C communication, so connecting both devices to the SDA and SCL pins of the arduino UNO,it would be possible to print the data on the LCD screen. After some research, the library "Liquid Crystall"[31] was chosen to control the LCD screen.

2.3.2 User Interface

At this point of the project, it was possible to see ”in situ” the readings through the LCD screen.

To improve this early prototype, a user interface was required to enable interaction with the device.

So the idea was that once the serial monitor was opened, a list of commands would be shown where it would be possible to select the desired option by pressing the right key on the keyboard.

2.3.3 Plot the data

Now it was required to plot the data.

Communication between the PC and the Arduino was accomplished using serial communication via a USB wire. Matlab was the software chose for this purpose.

Using this communcation, it was possible to use the variables from the arduino sketch in a Mathlab[21] sketch.

2.3.4 MicroPirani 925 Vacuum Pressure Transducer

In order to improve our system, a Vacuum Pressure Transducer was added.

The 925 MicroPirani vacuum transducer offers a wide measuring range from $1 \cdot 10^{-5}$ Torr to atmosphere and is based on measurement of thermal conductivity. The MicroPirani sensor consists of a silicon chip with a heated resistive element forming one surface of a cavity. A cover on top of the chip is forms the other surface of the cavity. Due to the geometry of the sensor convection cannot take place within the cavity and consequently the sensor is insensitive to mounting position. Gas molecules are passed by diffusion only to the heated element where the heat loss of the gas is measured. The sensor element is very robust and can withstand high G-forces and instant air inrush.[32]

To connect the devices and the system, a copper tube with a hole in the center of it was added to the transducer. That way it was possible to measure the pressure inside the vaccuum chamber.

2.4 Burst Simulator

Once the platform for the pressure sensor bmp085 was completed, a burst simulator was required. The programming language employed for this simulation was Python[24]. The reason was because Python is an open source language with uncountable information and tutorials which can be found on the web. The fact that I could teach myself to use it was a key factor for the election of python to develop the application. Moreover, the module matplotlib[25] offers a very intuitive way to plot all the data without the use another external software.

When the simulator was ready, the data obtained would be compared with the online burst simulator.

Throughout the development of the simulator 3 phases can be distinguish:

2.4.1 Phase one. Burst Simulator

During this phase, a program was required, which given a launch volume, a balloon weight, and a payload weight the gross lift, ascent rate, burst height, and time needed to burst could be obtained. To accomplish this task a bibliographic[4], [5],[6], search was required. All through this research, a great many math models for simulation of high altitude balloon systems were found, but were beyond the scope of the task. At the end, information needed was found by the hand of Steve Randall and the UK High altitude Society[14].

Afterward, a simple burst simulator was designed.

2.4.2 Phase two. Pressure model comparation

In this second phase, in order to see if the bmp085 pressure sensor algorithm to compute height was good enough, two models were developed. One based in the 1976 us standard atmosphere and one based on the barometric formula.

If the error among these two models would be really high, one could concluded that the bmp085 wouldn't be a good pressure sensor to be located in a High altitude mission. So as to model the pressure, a temperature model also designed.

2.4.3 Phase three. Density model and data acquisition

In phase three, once that pressure and temperature was modeled, a density air model was requested to observe how the the ascent rate would change with the air density.

Moreover, it was also requested that all the results be printed in one .txt file.

3 Results

This section will present the outcome of the steps followed in section 2 and the difficulties found in our path in order to get the final results.

Along the elaboration of this project two phases can be identified:

- **Phase I Sensor calibration**: The results obtained from the construction of the calibration platform and everything related to the arduino code and the mathalb code used to obtain the pressure and altitude data will be presented.
- **Phase II Python Application**: Here will be discussed everything related to the burst simulator and the results obtained to verify if the bmp085 is a good sensor for high altitude missions.

3.1 Phase I Sensor calibration

As it was said in section 2, once the vacuum was produced, it was possible to change the pressure inside the vacuum chamber, and for this reason a software capable of measureing pressure and consequently height was created.

Throughout this subsection, a discussion over the development of the arduino and matlab code, along with the obstacles discovered when attempting to add the micropirani925 to the calibration platform, shall be performed.

3.1.1 Schematic

The schematic for the prototype was designed according to the datasheet.

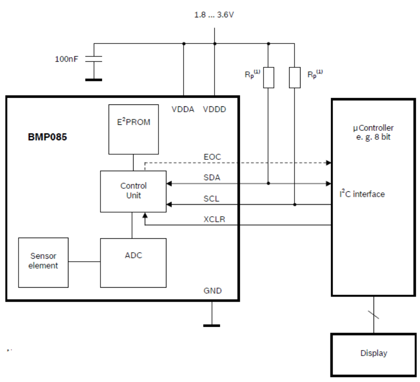

Figure 17: Prototype Scheme [26].

The idea was to connect the SDA and the SCL of the display and the pressure sensor with the arduino. The I^2C pins of the arduino are found in pins A4 and A5. The microcontroller connected to the PC, also provides the power and the ground.

It is important to mention that the bmp085 pressure sensor works with a voltage of 1.8 3.6V. So in order to not crush the pressure sensor, a voltage of 3.3V was supplied also from the arduino.

With a simple scheme it was possible to run the whole measurement system.

3.1.2 Measurements of pressure with Arduino

An arduino sketch was developed to measure pressure. In order to program the bmp085 pressure sensor and the LCD screen, the $Adafruit - BMP085$[30] and $LiquidCrystal - I2C$ [31] libraries were respectively used. In addition, the library "'wire"' was also needed because it allows for I^2C communication.

The idea of it as one can see was:

- Firstly, it checks if the sensor is connected. And if it is not connected, it will continuously check it in an infinite loop.

```
if (!pressure_sensor.begin ()) {
  lcd.print ( "Sensor BMP085 not found, check the connections!" );
  while ( 1 ) {}//pause forever, please reset
}
```

Figure 18: Arduino Scretch.

- Secondly, it will take measurement readings of pressure with the instruction $lcd.print(pressure - sensor.readPressure())$
- Once it is measured, the pressure then is stored in the variable "'Pressure"'

```
if (!pressure_sensor.begin ()) {
  lcd.print ( "Sensor BMP085 not found, check the connections!" );
  while ( 1 ) {}//pause forever, please reset
}
```

Figure 19: Arduino Scretch.

- The same process is used to measure height.

```
lcd.print("Press =");
lcd.print(pressure_sensor.readPressure ());
pressure=pressure_sensor.readPressure (); //th
```

Figure 20: Arduino Scretch.

- The whole process will run in infinite loop

Once this task was fulfilled, a user interface was required. The initial idea was that when the user opens the *serialmonitor* of the arduino, the individual is shown a list of instructions. So depending on which key is pressed, the user will receive one thing or another, i.e, communicate the microcontoller with the computer.

By means of *serialcommunication* it was possible to do this.

Therefore, the code was improved to accomplish this task.

The idea of how this part of the code operates is:

- Once it is checked that the sensor is connected, the function $help - command$ is called.
  - this function only prints the information in the serial monitor

```
if (!pressure_sensor.begin ()) {
  lcd.print ( "Sensor BMP085 not found, check the connections!" );
  while ( 1 ) {}//pause forever, please reset
}

help_command(); //function that creates an interface in the serial monitor
```

Figure 21: Arduino Scretch.

- Afterward, it is checked if the serial communication is available.
  - It will be when the serial monitor is open

```
if (Serial.available() > 0) { //Chec
  delay(50); //wait
  instruction = Serial.read(); //Rea
  lcd.clear();
  keyState = false; //we set the log
```

Figure 22: Arduino Scretch

- the variable *instruction* is created, which will store and read what key is pressed
- It was done with a $switch - case$ statement to label the instruction associated with the key
  - if c is pressed then the temperature reading is printed
  - if d is pressed then the altitude reading in SI is printed
  - if h is pressed then the command list is printed again.
  - if g is pressed then the pressure and altitude reading is printed in $mmhg$ and ft
  - if j is pressed then the pressure reading in SI and $mmhg$ is printed
  - if any other key is pressed, nothing happens
  - if the q is pressed, it stops to take measurements
- The whole process will run in infinite loop

```
if (Serial.available() > 0) { //Check if there is any dat
  delay(50); //wait
  instruction = Serial.read(); //Read each character one
  lcd.clear();
  keyState = false; //we set the logic state false
  // Do a switch case to identify what kind of data is se
  switch (instruction) {

    //If it is send a 'c', we print the values of tempera
    case 'c':
      lcd.clear();
      while (keyState == false) { //take data until the e
        Serial.print("Temperature = ");//we print it in t
        Serial.print(pressure_sensor.readTemperature());/
        Serial.println(" *C");
        lcd.setCursor(0, 0); //we do the same in the lcd
        //        "                 "
        lcd.print("Temp =");
        lcd.print(pressure_sensor.readTemperature());//re
        lcd.print("*C");
        lcd.setCursor(0, 1);
        //        "                 "
        scrollInFromLeft(1, "www.granasat.com");//we prir
        delay(1000);
        char outKey = (char)Serial.read(); //in case want
        if (outKey == 'q') {
          keyState = true; //in case of receive a 'q' cha
        } else keyState = false; //in case of receive anc
      }
      instruction = 'z';
      break;

    //if it send a 'd', we print the pressure in Pascals
    //in the lcd we will print the pressure and the heigh
    case 'd':
      lcd.clear();
      while (keyState == false) {
```

Figure 23: Arduino Scretch

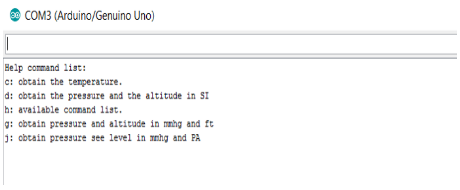

Figure 24: Serial Monitor

3.1.3 Plot the data measured with Matlab

At this point, it was possible to see "'in situ"' the measurement readings in the LCD screen, but they weren't stored. So the main goal was to plot the data obtained in real time. To do so, the major difficulty was to use the data acquired, save it in a file, and use excel to plot it. But the file would not have been used after that because they are only needed for testing. Besides that, arduino memory is limited, and maybe the program could exceed and crash. So the stored data should be **transmitted to the PC**. If the data would be sent and stored to the laptop, why not just transmit and use it directly in order to make a graph also "'in situ "'.

To accomplish this, Matlab was the software selected for this task. The program operates as follows:

- Firstly, two vectors were created to store the data
- After that, the port is opened

- A counter is initialized as well
- The graph is configured
- While the counter is less than the maximum of measurements, the data in the vector will be plotted
- Afterward, the port is closed.

Now the graph is acquired. It is possible to compare the data obtained with the data previously simulated with MathCad.

Both graphs are

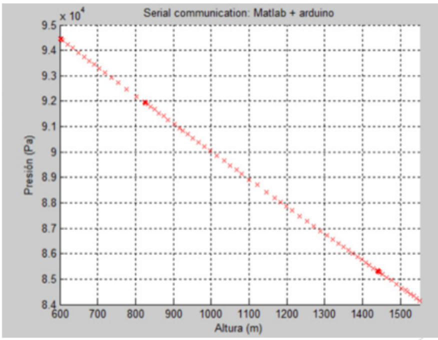

Figure 25: Experimental Pressure acquired with Matlab

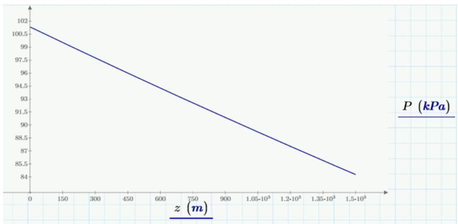

Figure 26: Theoretical Pressure acquired with MatCad

The whole calibration platform can be seen in the picture

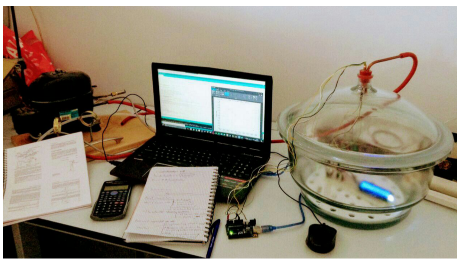

Figure 27: Calibration Platform

3.1.4 Transducer Micropirani925

In order to improve the calibration, the Micropirani925 transducer was used.

The idea was to add it to the platform using a copper tube with a little hole in the middle so the transducer would receive the same pressure as the vacuum chamber.

The micropirani925 has its own software to communicate with the computer by means of serial communication and its own data logger.

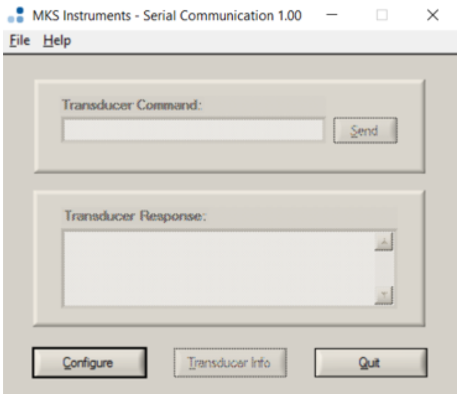

Figure 28: MicroPirani 925 Serial Communication software

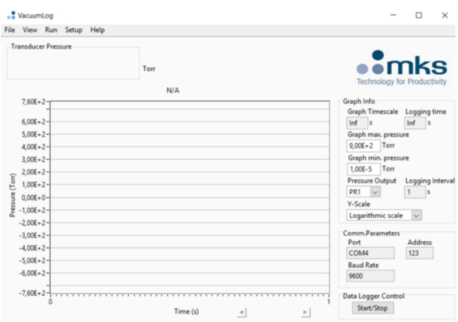

Figure 29: MicroPirani 925 Data Logger

But sadly it was impossible to establish communication between the computer and the transducer.

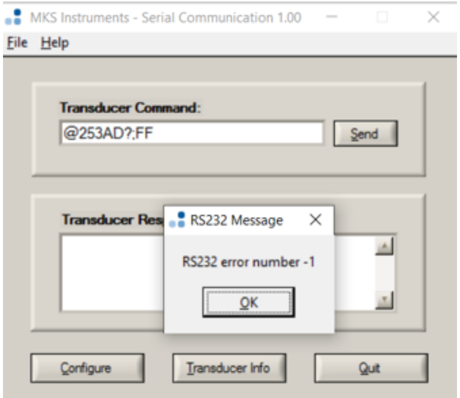

Figure 30: Error 1.

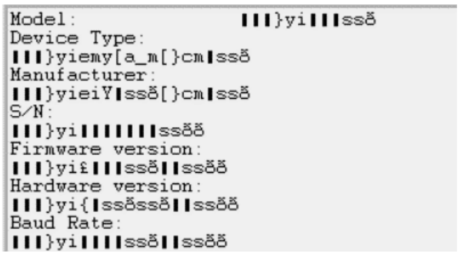

Figure 31: Error 2.

Figure 32: Error 3.

Figure 33: Error 4.

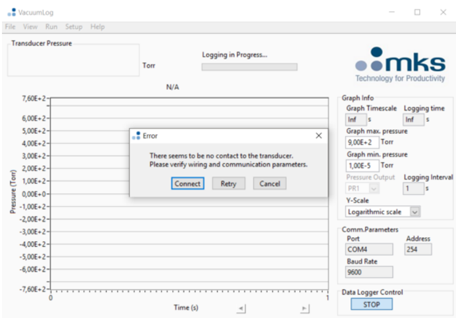

Figure 34: Error 5.

In order to fix the problem, all the wires where checked and the rx and tx pins, following the documentation of the transducer[32]. It was soldered according to the documentation again but the problem persisted.

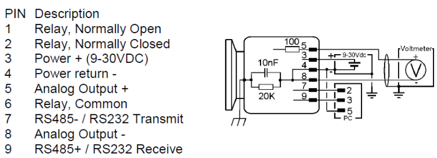

Figure 35: Transducer Micropirani 925 Pinout and scheme [32].

All the attempts were done under the supervision of the tutor but the problem persisted.

3.2 Phase II Python Application

This phase will discuss the results achieved with the burst simulator.

Before starting to program the code, a deep bibliographic research was required in order to simulate the flight of the balloon.

It must be noted that in this simulator, the change of the shape of the balloon was not considered, nor the thermal behavior or how solar radiation can affect the skin of the balloon. There are large types of phenomena that can affect the flight of the balloon which will be studied in the future.

3.2.1 Burst simulator

A Totex Balloon burst simulator was developed based on the notes of Steve Randall and the Kaymont Totex Sounding Balloon Data.[14]

The idea of how the program works is quite simple because it just required some math. There were no conditional statements.

- Firstly, variables for helium density, air density (288K and 101kPa), air density model and balloon data were declared in addition to launch volume, balloon and payload weight, and drag balloon average
  - air density model is a natural log multiplier to give a result similar to $NRLMSISE-00$ [39] over the range $0-80km$
  - drag balloon average was calculated from manufacturer data and flight data results
- making use of this data it computes burst height, gross lift, ascent rate and the time needed to burst.

During the research, contact was established with the provider in order to get proper documentation[22] about the product they sell. Unfortunately, due to confidentiality policies it was impossible to get it.

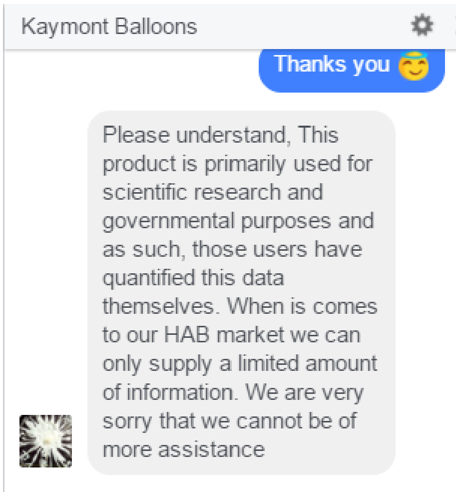

Figure 36: Conversation with Kaymont Balloon supplier.

The results obtained were compared with one of the flight predictors more commonly used in the website[11].

```
The burst height is:   36929.143 m
The amount of helium need need to lift is  7.185 kg
The ascent rate is:   270.906 m/min
The time need to burst is:   136.317 min
```

Figure 37: Burst Simulator outcome.

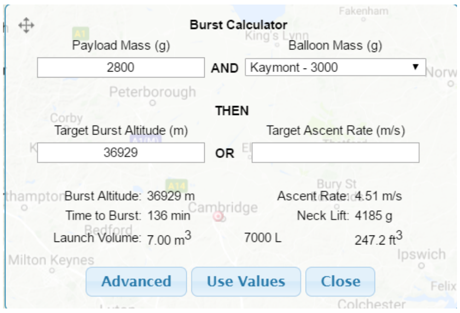

Figure 38: Web Burst Simulator [11].

As one can see the results match perfectly with the our burst simulator.

3.2.2 Density, temperature and pressure simulator

As it was said in section 2. a pressure simulator was developed in order to verify if the bmp085 would be able to give a pressure more similar possible to the one simulated based on the us 1976 standard model.

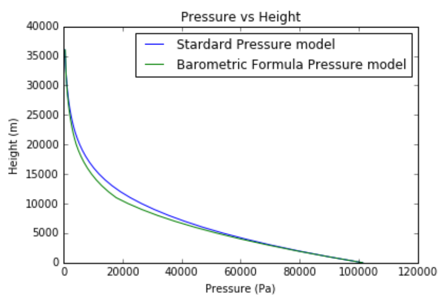

Figure 39: Pressure Models.

As can be seen in the graphic, both models work similarly, and it tends to be equal. They are less similar between the 11-20km, where they have an acceptable error of 20%.

Two density models were also developed, one based in the us 1976 standard model and the other based in gas ideal law.

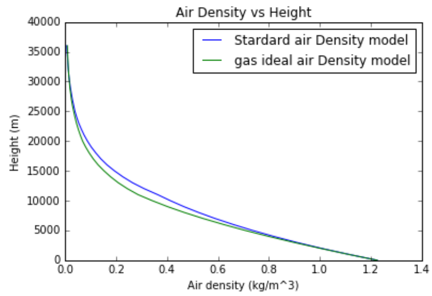

Figure 40: Air Density Model.

One can see in the picture that both models also match very well. The idea of this simulation was to see if the ascent rate would increase during the uplift.

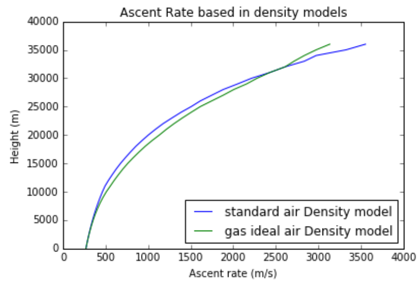

Figure 41: Ascent Rate Model.

Even though that the ascent rate increases during its ascent, this data cannot be accepted as correct because it is incredibly large.

The error in this simulation is because it wasn't taking into the account some effects that would affect the ascent rate.

The balloon drag is modeled as a sphere, which is highly dependent on Reynolds number, and if the balloon passes through the turbulent flow, Reynolds number values change during its ascent. The ascent rate is also affected by the balloon shape changes during the flight.[40]

Is order to check if the simulation of the us 1976 was well done, the results obtained were compared with this web simulator[23].

It is important to mention a special gratitude to Steve Randall [41] for answering my e-mails and providing me guidance.

4 Conclusions

In this section I am going to distinguish among two different subsections. First, a general conclusion will be discussed about the results of this project and future improvements. Second, a discussion about the personal obstacles and difficulties found along the way to develop it and some personal thoughts.

4.1 General conclusions

High altitude balloons provide an opportunity to do science in a really simple way. Despite being seemingly simple, they provide incredible opportunities for continued learning and can provide amazing data, photos, and videos upon successful retrieval. Also, they bring the opportunity and experience of sending up a device made by oneself to near space.

For this project, it was needed to learn a great amount of knowledge and develop skills just to put into practice two simple physical phenomena such as the Buoyancy Principle and the Barometric formula. Taking into the account the task firstly demanded, the design and building of a calibration platform for the pressure sensor, we can conclude that such an early design was successfully fulfilled. The first model, due to it being practically ready to run, offered us the chance to focus all our efforts into the configuration of the electronic device design in order to get the pressure reading of BMP085 module. The graph obtained with MatLab, plotting the data transmitted via serial communication between the arduino and the PC, looks quite similar with the one previously simulated with MathCad. Therefore, it was concluded that the **bmp085 pressure sensor was correctly calibrated**.

So as to get a more precise pressure reading and improve the design, the micropirani925 transducer was introduced. But as was said, the failure of the device, and not recognition of it, made impossible the communication between the pc and the instrument. This made it unable to take a measurement of the pressure inside the vacuum chamber.

In what refers to the burst simulator, the simulation was later compared with the hab predictor. And introducing the same data in both simulators, it was obtained **similar results** in what is related to burst height, ascent rate, time needed for it to burst.

In addition to this achievement, the simulation of pressure, temperature, and air density was also compared with the the 1976 USA simulator, which also matches, concluding that the 1976 us standard atmosphere were well simulated.

height (m),	temperature(K),	pressure(Pa),	density air(kg/m^3)
0	288.15	101325.0	1.225
1000	281.65	89874.136	1.112
2000	275.149	79494.428	1.006
3000	268.649	70107.478	0.909
4000	262.149	61638.954	0.819
5000	255.649	54018.472	0.736
6000	249.148	47179.479	0.66
7000	242.648	41059.129	0.589
8000	236.148	35598.168	0.525
9000	229.648	30740.817	0.466
10000	223.147	26434.653	0.413
11000	216.647	22630.498	0.364
12000	216.65	19327.242	0.311
13000	216.65	16507.614	0.265
14000	216.65	14099.338	0.227
15000	216.65	12042.402	0.194
16000	216.65	10285.551	0.165
17000	216.65	8785.004	0.141
18000	216.65	7503.37	0.121
19000	216.65	6408.713	0.103
20000	216.65	5473.753	0.088
21000	217.651	4677.126	0.075
22000	218.651	3999.114	0.064
23000	219.651	3421.833	0.054
24000	220.651	2929.958	0.046
25000	221.651	2510.549	0.039
26000	222.651	2152.673	0.034
27000	223.651	1847.085	0.029
28000	224.651	1585.96	0.025
29000	225.651	1362.673	0.021
30000	226.651	1171.608	0.018
31000	227.651	1008.004	0.015
32000	228.652	867.817	0.013
33000	231.454	748.063	0.011
34000	234.254	645.977	0.01

Figure 42: Simulator Data outcome.

Altitude [m]	Temperature [Kelvin]	Pressure [pascal]	Density [kg/m3]
0.00000	288.150	101325	1.22500
1000.00	281.650	89874.6	1.11164
2000.00	275.150	79495.2	1.00649
3000.00	268.650	70108.5	0.909122
4000.00	262.150	61640.2	0.819129
5000.00	255.650	54019.9	0.736116
6000.00	249.150	47181.0	0.659697
7000.00	242.650	41060.7	0.589501
8000.00	236.150	35599.8	0.525168
9000.00	229.650	30742.5	0.466348
10000.0	223.150	26436.3	0.412707
11000.0	216.650	22632.1	0.363918
12000.0	216.650	19330.4	0.310828
13000.0	216.650	16510.4	0.265483
14000.0	216.650	14101.8	0.226753
15000.0	216.650	12044.6	0.193674
16000.0	216.650	10287.5	0.165420
17000.0	216.650	8786.68	0.141288
18000.0	216.650	7504.84	0.120676
19000.0	216.650	6410.01	0.103071
20000.0	216.650	5474.89	0.0880349
21000.0	217.650	4677.89	0.0748737
22000.0	218.650	3999.79	0.0637273
23000.0	219.650	3422.43	0.0542803
24000.0	220.650	2930.49	0.0462674
25000.0	221.650	2511.02	0.0394658
26000.0	222.650	2153.09	0.0336882
27000.0	223.650	1847.46	0.0287769
28000.0	224.650	1586.29	0.0245988
29000.0	225.650	1362.96	0.0210420
30000.0	226.650	1171.87	0.0180119
31000.0	227.650	1008.23	0.0154288
32000.0	228.650	868.019	0.0132250
33000.0	231.450	748.228	0.0112620
34000.0	234.250	646.122	0.00960889
35000.0	237.050	558.924	0.00821392

Figure 43: 1976 Standard Atmosphere Simulator [23].

Consequently, taking a sight of both plots of pressure, one made under the 1976 USA standard model and the other under the barometric formula, one can appreciate how similar both curves are, with an maximum error of 20%. Hence, it was concluded that the pressure procured by the bmp085 in such circumstances of height and temperature will be **reasonable good**

To bring this to a close, the BMP085 is an amazing sensor that will work perfectly in the high altitude balloon, becauase besides pressure, it can measure temperature as well, and only for a price of 8 euros.

4.2 Future Improvements

Moreover, for all the work in this project, there is still space to improve. Inside this section will be a discussion of the improvements to the pressure sensor calibration, burst simulator, and future work.

Furthermore, to write this paper as it was demanded by the tutor at the beginning of the project, another document will be written with no page restriction.

4.2.1 Sensor calibration

Firstly, the use of the transducer MicroPirani925 would fit perfectly in this project, offering a pressure measured more precisely than the BMP085.

Figure 44: Transducer MicroPirani 925 [33].

Secondly, the development of a PCB board in which to put the sensor alongside the GPS and radio module for the transmission of data and for the recovery of the payload. This PCB board will be placed within the payload. This assignment will give us an opportunity to work in a more profound way with serial communication, alongside enabling us to learn more about PCB design. This knowledge was not offered in the degree, not even as an elective subject. And due to the theoretical character of this project, there was no time to formally acquire it.

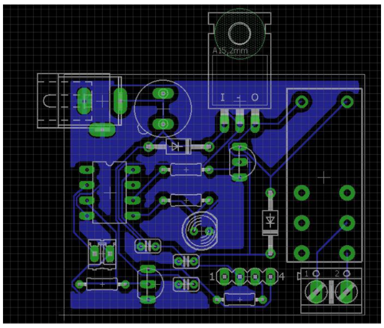

Figure 45: PCB Design example.

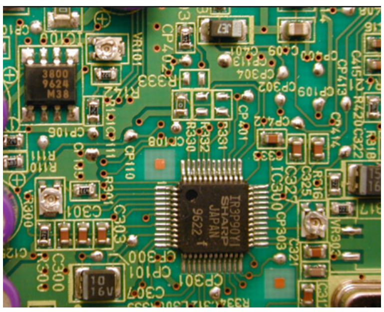

Figure 46: PCB board example [34].

4.2.2 Burst simulator

Regarding the simulator, it did not take into account the thermal behavior, and how solar radiation can affect the skin of the balloon. It also did not take into account how the shape itself would change during the flight.

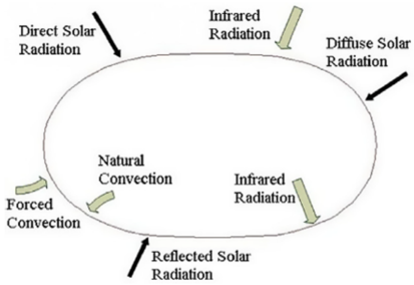

Fig. 1. Thermal environment of high altitude balloon.

Figure 47: Thermal environment High Altitude Balloon [6].

In addition to that, a motion simulator can enlarge the actual model. So one can visualize the path that our balloon should take.

Neither was considered winds aloft, launch location, launch time, if the sky is cloudy, ground albedo, cloud albedo...etc; i.e a great number of phenomena which affects the the balloon flight. [4], [5],[6],

As one can see there is a room for improvement.

4.2.3 Future work

At this point, one can realize that in these 4 months, skills and knowledge were introduced but there is still a lot work to do.

Furthermore of all the improvements previously mentioned, it would be nice to work with all the sensor systems in the payload module, such as:

- Accelerometer: to take a look of acceleration readings, to see what the g-forces look like when the balloon explodes
- Magnetometer: To get orientated in space, this module can give an orientation of where are we pointed.
- Light sensor to detect IR or UV light at heights where it can be measured
- A camera: to take our own pictures of near space

This sensor are just an example, the payload can carry whatever other sensors are required to study of the atmosphere at such altitudes.

To come to an end of this section, the aim is to face and overcome all the challenges that high altitude ballooning can put in our way.

And last but not least, ***To reach the sky***

Figure 48: Picture taked from High Altitude Balloons [13].

4.3 Personal conclusions

My motivation to study Physics was always relative with the studies of Electrical Engineering. That is why, back in 2014, I asked to professor Andrés Roldán to conduct my Bachelor Thesis and he encouraged me to study abroad with the program Erasmus in Kaiserlautern, Germany. Thanks to this advice I enjoyed one of my best years of my life in which learn more that I could read in a book.

Once my Erasmus experience was over, we started this project in which I had the change to break the ice with electronic devices, and Andrés taught me the basis of how to work in this field. To be able to grow by myself.

Along this project, I didn't learn just how the High Altitude Balloons works, I also learned to talked with maintenance service of Facultad the Ciencias, how to cut a copper tube, how to drill properly, how to solder, how to write a proper code. Besides all the software tools like Arduino, Python, MathCad and Mathlab.

Basically, to do to feel science as real, not just in the notebook or the computer.

And for this and all the experience earn I am very thankful.

References

[1] N. Yajima, N. Izutsu, T. Imamura and T. Abe,
Scientific Ballooning technology and applications of Exploration balloon floating in the Stratosphere,
Springer, 2009

[2] On the Barometric Formula
Mário N. Berberan-Santos, Evgeny N. Bodunov and L. Pogliani,
American Associattion of Physics Teachers,

[3] An Overview of High-Altitude Balloon Experiments at the Indian Institute of Astrophysics
M. Safonova, A. Nayak, A. G. Sreejith, J. Mathew, M. Sarpotdar,S. Ambily, K. Nirmal, S. Talnikar, S. Hadigal, A. Prakash and J. Murthy

[4] J. F. Kreider,
AIAA 5th Aerodynamic Deceleration Systems Conference,
Albuquerque, New Mexico/November 17-19, 1975.

[5] R. Farley
AIAA 5th Aviation, Technology, Integration, and Operations Conference,
Arlington, Virginia/September 26-28, 2005.

[6] Q. Dai, X. Fang, X. Li, L. Tian
Performance simulation of high altitude scientific balloons,
Advances in Space Research **49** (2012) 1045-1052.

[7] W. Diehl
Standard Atmosphere- Tables and Data,
Bureau of Aeronautics, Navy Department, Report No.218.

[8] LimaSat Project,
https://www.slideshare.net/Scientix/scientix-11th-spne-brussels-18-mar-2016-limasat

[9] Columbia Scientific Balloon Facility,
https://www.nsbf.nasa.gov/balloons.html

[10] GPS Tracking
http://ravtrack.com/GPStracking/cocom-gps-tracking-limits/469/

[11] High altitude burst predictor,
http://predict.habhub.org/

[12] Serial Communication,
https://learn.sparkfun.com/tutorials/serial-communication

[13] High Altitude Balloon Launch,
https://www.sparkfun.com/tutorials/180

[14] UK High Altitude Society,
https://ukhas.org.uk/start

[15] Community Balloon Challenge,
https://community.balloonchallenge.org/t/payload-overview/660

[16] Barometric Formula,
http://hyperphysics.phy-astr.gsu.edu/hbase/Kinetic/barfor.html

[17] Arduino
https://www.arduino.cc/

[18] Arduino Pinout,
http://www.electroschematics.com/7958/arduino-uno-pinout/

[19] www.prometec.new,
http://www.prometec.net/bus-i2c/

[20] MathCad,
https://www.ptc.com/en/engineering-math-software/mathcad

[21] MathCad,
https://es.mathworks.com/products/matlab.html

[22] Kaymont Balloons,
http://kaymontballoons.com/

[23] 1976 Standard Atmosphere Calculator,
https://www.digitaldutch.com/atmoscalc/tableatmosphere.htm

[24] Python
https://www.python.org/

[25] Matplot Documentation,
https://matplotlib.org//

[26] BMP085 Digital pressure sensor Data sheet
https://www.sparkfun.com/datasheets/Components/General/BST-BMP085-DS000-05.pdf

[27] BMP085 Digital pressure sensor Data sheet
http://blog.circuits4you.com/2016/04/arduino-temperature-and-bmp085-pressure.html

[28] I2C 1602 Serial LCD Module Data sheet
https://opencircuit.nl/ProductInfo/1000061/I2C-LCD-interface.pdf

[29] I2C 1602 Serial LCD Module Data sheet
https://www.aliexpress.com/popular/i2c-lcd-display.html

[30] BMP085 Digital pressure sensor library
https://github.com/adafruit/Adafruit-BMP085-Library

[31] I2C 1602 Serial LCD Module Library
https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library

[32] Vacuum pressure transducer 925 MicroPirani
https://www.mksinst.com/Docs/R/925man-revh.pdf

[33] Vacuum pressure transducer 925 MicroPirani picture
https://www.mksinst.com/product/product.aspx?ProductID=648

[34] EMC considerations in PCB Design and Layout
http://www.hottconsultants.com/outlines/pcb.html

[35] Auxylab, Productos de laboratorio
http://www.auxilab.es/es/productos-laboratorio/desecador-c-llave-rin-24-29-d-300mm/

[36] Auxylab, Productos de laboratorio
http://www.auxilab.es/es/productos-laboratorio/tapon-conico-2-orificios-26x19-mm-h-28-mm/

[37] Auxylab, Productos de laboratorio
http://www.auxilab.es/es/productos-laboratorio/campana-boton-vidrio-300x400mm-simax/

[38] Auxylab, Productos de laboratorio
https://www.labbox.com/es/productos/F0400/x400/S025/

[39] NRLMSISE-00
https://en.wikipedia.org/wiki/NRLMSISE-00

[40] Drag of a sphere
https://www.grc.nasa.gov/www/k-12/airplane/dragsphere.html

[41] Steve Randall
https://ukhas.org.uk/ukhas:members