Sharif Al-Husein Raie is a telecommunications engineering student from Granada, Spain. He has started his career in Telecommunications Technologys Engineering at the University of Granada since 2016, with a major in Electronic Systems. With this thesis the career comes to an end.

Andrés María Roldán Aranda is the academic head of the present project, and the student's tutor. He is a professor in the Departament of Electronics and Computers Technologies.

Bachelor'S THESIS

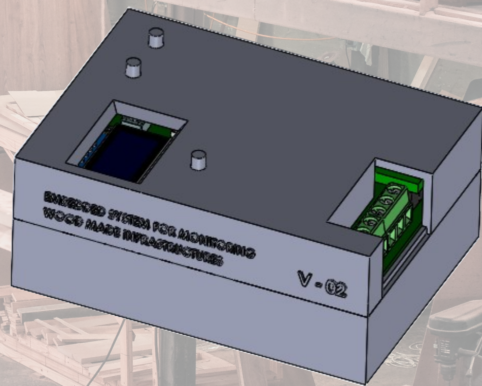
Embedded system for monitoring wood made infrastructuses

Sharif Al-Husein Raie

TELECOMMUNICATION ENGINEERING

UNIVERSITY OF GRANADA

Bachelor in Telecommunication Engineering



Embedded system for monitoring wood made infrastructures

Sharif Al-Husein Raie

2020/2021

Tutor: Andrés María Roldán Aranda

Printed in Granada, September 2021.

“Embedded system for monitoring wood made infrastructures”

BACHELOR IN
TELECOMMUNICATION ENGINEERING

Bachelor's Thesis

"Embedded system for monitoring wood made infrastructures"

ACADEMIC COURSE: 2020/2021

Sharif Al-Husein Raie

BACHELOR IN TELECOMMUNICATION ENGINEERING

"Embedded system for monitoring wood made infrastructures"

AUTHOR:

Sharif Al-Husein Raie

SUPERVISED BY:

Andrés María Roldán Aranda

DEPARTMENT:

Electronics and Computers Technologies

D. Andrés María Roldán Aranda, Profesor del departamento de Electrónica y Tecnología de los Computadores de la Universidad de Granada, como director del Trabajo Fin de grado de D. Sharif Al-Husein Raie,

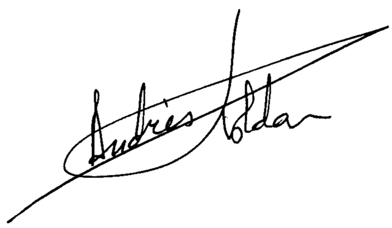
Informa:

Que el presente trabajo, titulado:

"Embedded system for monitoring wood made infrastructures"

ha sido realizado y redactado por el mencionado alumno bajo mi dirección, y con esta fecha autorizo a su presentación.

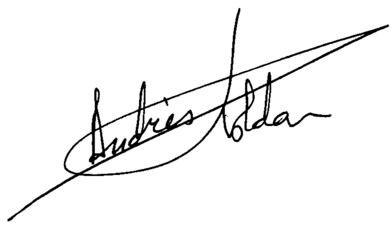
Granada, a 1 de septiembre de 2021



Fdo. Andrés María Roldán Aranda

Los abajo firmantes autorizan a que la presente copia de Trabajo Fin de Grado se ubique en la Biblioteca del Centro y/o departamento para ser libremente consultado por las personas que lo deseen.

Granada, a 1 de septiembre de 2021



Fdo. Sharif Al-Husein Raie

Fdo. Andrés María Roldán Aranda

Embedded system for monitoring wood made infrastructures

Sharif Al-Husein Raie

KEYWORDS:

Poplar, Strain Gauge, Glulam, Altium Designer® 21, Co-design hadware-software, SolidWorks®, EDA, Electronics, ADC, PCB design, Python, Qt designer, Visual Studio Code.

ABSTRACT:

Starting from a context in which sustainable construction and the efficient use of natural resources are being promoted more every day, the COMPOP project appears where it is intended to develop new composite materials for construction combining laminated poplar wood (Glulam) and reinforcing it with carbon fiber.

The COMPOP project has been active for a few years and has reached a stage in which they want to demonstrate the effectiveness of these new materials in real constructions. With this, the need arises to monitor the stresses supported by these structures that are still in the prototype phase.

This Final Degree Project aims to develop a device that meets this need, that is, it measures the stresses that each beam of a structure supports and analyzes these data, checking that the deformations suffered are within the operating margin at all times.

The development and implementation of this project is performed following methodologies of **Systems Engineering**, giving realism and getting the student closer to professional techniques, widely recognized in the job market. Furthermore, the complexity and multidisciplinary scope of this Bachelor's Thesis allows covering not only the different specialties of the Bachelor in **Telecommunication** Engineering but also acquiring knowledge and transversal abilities from other fields of the Engineering, such as **Mechanical** or **Building Engineering**. Besides specific software of each of the mentioned areas, advanced techniques of **machining**, **manufacturing** or **characterization** of different devices, among others, have been analyzed and applied.

Embedded system for monitoring wood made infrastructures

Sharif Al-Husein Raie

PALABRAS CLAVE:

Chopo, Galga extensiométrica, Glulam, Altium Designer® 21, Codiseño hadware-software, SolidWorks®, EDA, Electrónica, ADC, Diseño de PCB, Python, Qt designer, Visual Studio Code.

RESUMEN:

Partiendo de un contexto en el que cada día se fomenta más la construcción sostenible y el uso eficiente de los recursos naturales, aparece el proyecto COMPOP donde se pretende desarrollar nuevos materiales compuestos para construcción combinando madera laminada de chopo (Glulam) y reforzándolo con fibra de carbono.

El proyecto COMPOP lleva activo unos cuantos años y ha llegado a una fase en la que se quiere demostrar la eficacia de estos nuevos materiales en construcciones reales. Con ello surge la necesidad de monitorizar los esfuerzos que soportan estas estructuras que están aún en fase de prototipo.

Este Trabajo Final de Grado pretende desarrollar un dispositivo que cubra esta necesidad, es decir que mida los esfuerzos que soportan cada viga de una estructura y analice esos datos comprobando que en todo momento las deformaciones sufridas están dentro del margen de operación.

El desarrollo e implementación de este proyecto se lleva a cabo siguiendo metodologías de **Ingeniería de Sistemas**, dotándolo de realismo y acercando al alumno a técnicas profesionales de amplio reconocimiento en el mercado de trabajo. Asimismo, la complejidad y ámbito multidisciplinar de este Trabajo Fin de Grado le permite cubrir, no sólo las diferentes especialidades del Grado de Ingeniería de **Telecomunicación**, sino también adquirir conocimientos y habilidades transversales o específicos de otros campos de la Ingeniería, como la **Mecánica** o la **Edificación**. Así, además de software especialista de cada uno de los campos mencionados, se han analizado y aplicado técnicas avanzadas de **mecanizado**, **fabricación** (soldadura utilizando técnicas de *reflow*) o **caracterización** de diferentes dispositivos, entre otros.

'Tough and competent'

Acknowledgments:

This work has been possible thanks to a small amount of people, but to whom I owe a great acknowledgment; although the one I can show you in these lines will undoubtedly be insufficient, serve this brief space as such.

The first of them must be for my family, my parents, Hasan and Fatima, my sisters Batul and Tasnim and my brother Obada. Ever since I can remember, their unconditional support in every aspect has been of primary importance for every of my undertakings to come to fruition. The support provided during this project has been only another proof of this.

To Francisco and Irene, for the technical support, the laughs and warnings to stop doing what you are doing. To Francisco Rescalvo, for attending me patiently in the materials lab and repeating the tests over and over again without complaint.

This Bachelor's Thesis would not have been born without the persistence of my tutor, Andrés Roldán Aranda, in the face of my initial reticence. His ambition has got me closer to numerous projects throughout all these years, and this work unbeatably closes this unforgettable stage.

Everyone of the mentioned is part of this Thesis. To all of them, and to those who could not see it, thank you.

Agradecimientos:

Este trabajo ha sido posible gracias a un número reducido de personas, pero a las cuales debo un gran agradecimiento; aunque el que pueda mostrarle en estas líneas será sin duda insuficiente, sirva este breve espacio como tal.

El primero de ellos ha de ir destinado a mi familia, mis padres, Hasan y Fatima, mis hermanas Batul y Tasnim y mi hermano Obada. Desde siempre, su apoyo incondicional en todos los aspectos ha resultado de importancia capital para que toda empresa acometida llegara a buen puerto. El apoyo mostrado durante el periodo de desarrollo de este proyecto no ha sido sino una muestra más del mismo.

A Francisco y Irene por el apoyo técnico, las risas y los avisos para dejar lo que estás haciendo. A Francisco Rescalvo por atenderme pacientemente en el laboratorio de materiales y repetir los ensayos una y otra vez sin quejarse.

Este Trabajo Fin de Grado no habría nacido sin la persistencia de mi tutor, Andrés Roldán Aranda, frente a mi reticencia inicial. Su ambición me ha acercado a numerosos proyectos a lo largo de todos estos años, y este trabajo cierra de manera inmejorable esta inolvidable etapa.

Cada uno de los mencionados son partícipes de este Trabajo. A todos ellos, y a los que no pudieron verlo, muchas gracias.

Index

List of Figures

List of Tables

Glossary

ADIME Is a consolidated group of the College of Building Engineering of the University of Granada, whose research focuses on the diagnosis of materials and structures through mechanical tests and non-destructive methods (NDT)..

Altium Designer® 21 EDA software used to design PCB from schematics. It allows 3D Design, as well as electronics simulation.

CircuitCAM CAM program for PCB and other related technologies from China. It is supporting technologies for both prototyping and line production..

clone I-214 Type of poplar that comes from Populus x Euramericana, a hybrid of Populus deltoides and Populus Nigra..

COMPOP Composite & Poplar, a project of building engineering school, consisting of the development of engineering products made from poplar boards and sheets with composite material inserts for use in construction.

CubeSat Miniaturized satellite normally for space research, with dimensions of $1\,\mathrm{dm}^3$ and mass lower than 1.33 kg per unit.

GranaSAT GranaSAT is an academic project from the University of Granada originally consisting in designing and developing a picosatellite (CubeSat). Coordinated by the Professor Andrés María Roldán Aranda, GranaSAT is a multidisciplinary project with students from different degrees, where they can acquire and enlarge the knowledge necessary to face an actual aerospace project.

SolidWorks® CAD Software from Dessault Systèmes for 3D Mechanical Design.

Acronyms

ADC Analog to Digital Converter.

ADIME Acústica y Diagnóstico de Materiales y Estructuras.

CAD Computer Aided Design.

CAM Computer Aided Manufacturing.

CNC Computer Numeric Control.

CSA *Construcción Sostenible de Andalucía*.

EDA Electronic Design Automation.

FDM Fused Deposition Modeling.

FR4 Flame Retardant 4 (4 means fiber glass epoxy).

FRP Fiver Reinforced Polymer.

Glulam Glued Laminated Timber.

GUI Graphical User Inteface.

HP-GL Hewlett and Packard Graphics Language.

I2C Inter-Integrated Circuit.

IC Integrated Circuit.

IDIE Research and Development in Building Engineering, of its acronym in Spanish (Investigación y Desarrollo en Ingeniería de Edificación).

IFAPA *Investigación y Formación Agraria y Pesquera de la Junta de Andalucía*.

LVL Laminated Veneer Lumber.

MCU Micro-Controller Unit.

NDT Non-Destructive Test.

PC Personal Computer.

PCB Printed Circuit Board.

PGA Programmable Gain Amplifier.

PLA Polylactic Acid.

PROPOPULUS PROmovere POPULUS (from Latin).

SCH Schematic.

SoC System-on-Chip.

SOP-16 Small Outline Package, 16 pins.

Chapter 1

Introduction

This Bachelor's Thesis is presented as a compilation of the knowledge acquired throughout the years of the bachelor's degree and specially, during this project period. It aims to reflect the engineering process behind the design, development, prototyping and verifying stage of a product. The overall goal of the project is developing an embedded system for monitoring wood made infrastructures, to be used in future projects related with monitoring constructions, specially wood made constructions.

This Bachelor's Thesis is carried out in collaboration with GranaSAT and ADIME groups, both belonging to the University of Granada. Moreover, this project fits within COMPOP research field, this will be explained in detail in Subsection 1.1.1.

GranaSAT is a group dedicated to different multidisciplinary projects which gathers people from a variety of fields who are committed to acquiring new knowledge related to Electronics and Aerospace Engineering. Since its origins, one of its main purposes has been getting a CubeSat in orbit, however, today its goals goes far beyond, and a wide range of devices and projects are developed in collaboration with different students and enterprises, such as the subject of this project.

Figure 1.1 – *GranaSat Logo*

ADIME is a consolidated group for Research and Development in Building Engineering (IDIE) from the University of Granada, whose research focuses on the diagnosis of materials and structures through mechanical tests and non-destructive methods (NDT) with vibro-acoustic techniques (acoustic emission, ultrasound and vibrations). The group conducts research and multidisciplinary knowledge transfer on the border between Materials and Structures, Physics, Electronics and Signal Processing. It has 4 active doctors, 2 researchers in training and solid collaborations with different departments and companies. Currently, the group has a Testing Laboratory for Structural Diagnosis and a Wood Workshop.

Figure 1.2 – *IDIE-ADIME Logo*

1.1 Prior art. Problem Statement

1.1.1 COMPOP Project

COMPOP is a project whose acronyms refer to **Composite & Poplar**. The project aims to develop and mechanically evaluate new engineering products derived from the combination of poplar wood and fiber-reinforced composite materials (FRP). **COMPOP** is developed by the University of Granada (Building Engineering College) and the Granada headquarters of the *Instituto de Investigación y Formación Agraria y Pesquera de la Junta de Andalucía* (IFAPA), and has the support of the companies MAPEI SPAIN SA, BASALTEX, the *Bonsai Arquitectos* studio and the Cluster of *Construcción Sostenible de Andalucía* (CSA), moreover it is supported by The European Poplar Initiative, PROPOPULUS.

For decades, construction in Granada has turned its back on a local resource highly valued in other countries and that already it have demonstrated its validity for structural use, poplar wood. Structural technical wood, represented in its clearest expression by glued laminated wood (Glulam), nevertheless offers an enormous opportunity for the development of a proximity industry at the service of a construction aimed at: 0 waste, almost zero energy consumption, use of proximity products and maximum carbon sequestration. *Las Vegas* of Granada and its avenues, as a sign of cultural identity of our territory for more than a century, are the ideal raw material for this.

In the past and to a large extent even today, lumber from tree trunks has been used as beams and pillars for floors and roofs. However, the beams and pillars of sawn wood contain numerous defects of wood as a biological material (knots, cracks, ridges, etc.) and involve less efficiency forest use. Nowadays, however, wood construction in Europe and Spain has clearly opted for industrial technical wood products made by joining small boards or veneers joined together by toothed joints and structural glues. Thanks to that, a product with better mechanical properties is achieved by eliminating large defects, and it favors a greater efficient use of the trunk, then fast-growing species, like poplar, produce trunks with smaller diameter take more relevance. Using this technology, there is currently a construction boom with laminated wood (Glulam) and Laminated Veneer Lumber (LVL) in Spain and Europe.

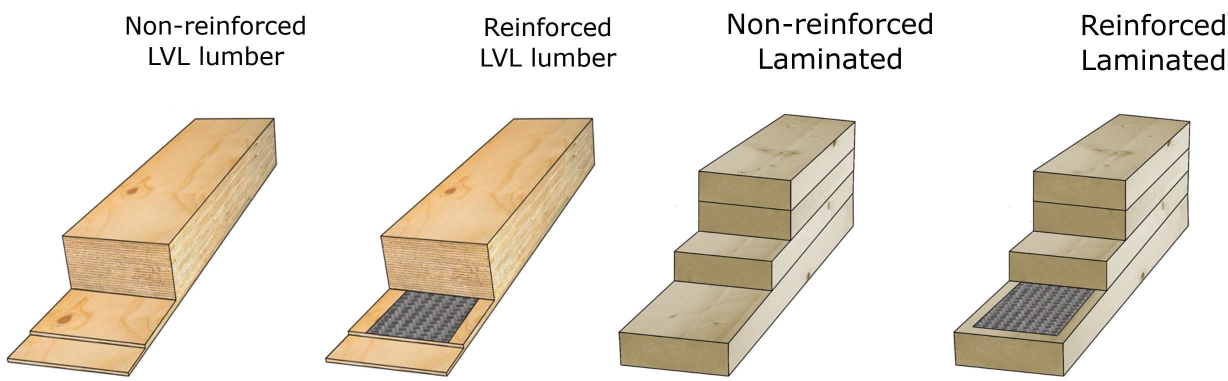


Figure 1.3 – *Laminated Beams [13]*

The goal of this project is to investigate, by means of experiments and analytical calculations, the mechanical properties of new products made with poplar wood and integrated composite material of carbon or basalt and analyze their viability for use in the construction sector. For new products it is expected to have this properties:

- Reduced weight.
- Allowing the use of wood with less mechanical performance, compensating with the performance of the other materials.
- Better resistance, rigidity and ductility than non-reinforced materials.
- Less mechanical variability, reducing the influence of defects.
- Well resistance to fire, as the reinforcing element is embedded.

Following this goals they have develop different two meter long prototypes of glued

1

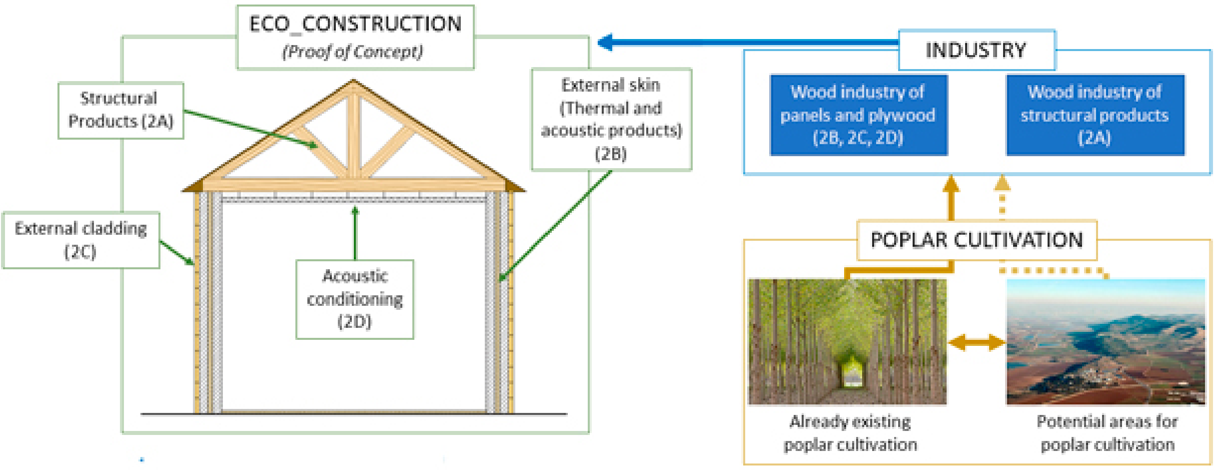


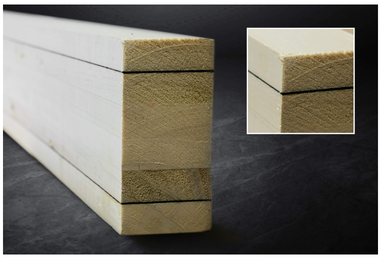
Figure 1.4 – *Sustainable poplar construction [13]*

laminated poplar wood beams with carbon fiber fabrics embedded between the wooden boards

The first strategy that has been followed has been to include in its manufacture carbon fiber fabrics typically used in the aeronautical industry, which give it a relevant reinforcement of its mechanical properties. Researchers at IDIE have shown that it is possible to improve the stiffness of rolled poplar beams by between 5% and 44%, depending on the amount, type and location of the carbon fiber.

The sawn wood from this plantation provided an average modulus of elasticity of 7000 MPa (low value to be used as structural wood), the Glulam wood without carbon fiber contributed an average modulus of 8500 MPa, which is a resistance class GL20h , whose structural use is totally viable. In addition, if carbon fiber fabric is introduced into the poplar beam, it has been shown that the modulus of elasticity is 9300 MPa if it is placed only in the lower zone of the beam and 10500 MPa if it is placed in the upper and lower zone. In addition, the carbon fabrics provide an improvement in the ductility of the structural element of up to 115%, thus being much safer against seismic events.

(a) *Glulam single side reinforced*



(b) *Glulam double side reinforced*

Figure 1.5 – *Poplar laminated beams [13]*

The project has used as raw material poplar wood from clone I-214 extracted from a 9-year-old plantation located in *Vega de Granada*, in an experimental farm managed by IFAPA, and whose plantation, growth and production of wood have been scientifically analyzed during its years of life.

It is clear that by combining different materials it is possible to achieve better results taking advantage of the particular properties of each material separately, but why use poplar wood in particular instead of another one?

1.1.1.1 Poplar wood

The species of the genus Populus have a series of botanical traits of great importance, which, together with the development of appropriate cultural techniques and industrial technology, are the key to the generalization of poplar cultivation worldwide, known generically as populiculture.

Poplar is one of the most efficient trees in terms of sustainability. It is one of the fastest growing species in Europe and one hectare of poplar captures 11 tons of CO_2 every year. Poplar also can be planted on otherwise useless land, thus optimizing land use and increasing timber supply with a renewable source of raw material. It's speed of growth also makes it an economically sound investment as revenue comes in relatively short cycles. On the other hand, carbon emissions are reduced as the transport footprint diminishes. [15]

Figure 1.6 – *Poplar Cultivation [16]*

Poplar's big advantage is its fast growth cycle when compared to other trees. It is also a "local" species, as it blends optimally with agro-forestry and poplar plantations positively impact local economies, benefitting rural development and employment.

As the substitution of non-renewable or unsustainable materials, like fossil fuels or concrete, with renewable ones, such as wood, is a must in the journey to a more sustainable economy and society, it is foreseeable that the industry will demand more and more timber in the future, and the high rotation of poplar makes it an extremely efficient source of raw material.

Furthermore, the fact that poplar can be planted in plantations guarantees the supply of locally produced wood, thus avoiding the need for the industries to import material from third countries, keeping resources in our rural areas and so, improving local economies while helping mitigation of climate change, as the logistics carbon footprint diminishes.

For local economies, as poplar cultivation makes a profitable business, it pulls on employment and development requiring both skilled and unskilled labour to grow it and to transform it into products. Although this could stand true for any species of tree, poplar plantations stand out in this issue as the poplar cycle is 12 to 15 years, a very short span of time if we compare it with the 100+ cycle of oak! Actually, poplar cultivation gets quite close to agriculture due to its speed of rotation. Lastly, the environmental benefits of poplar plantations are most important since these green filters can hold up to 70% to 90% of nitrates and 75% of the sediments. This provides an improvement of infiltration in river areas due to the roots of the poplar trees which help avoid erosion during floods.

1

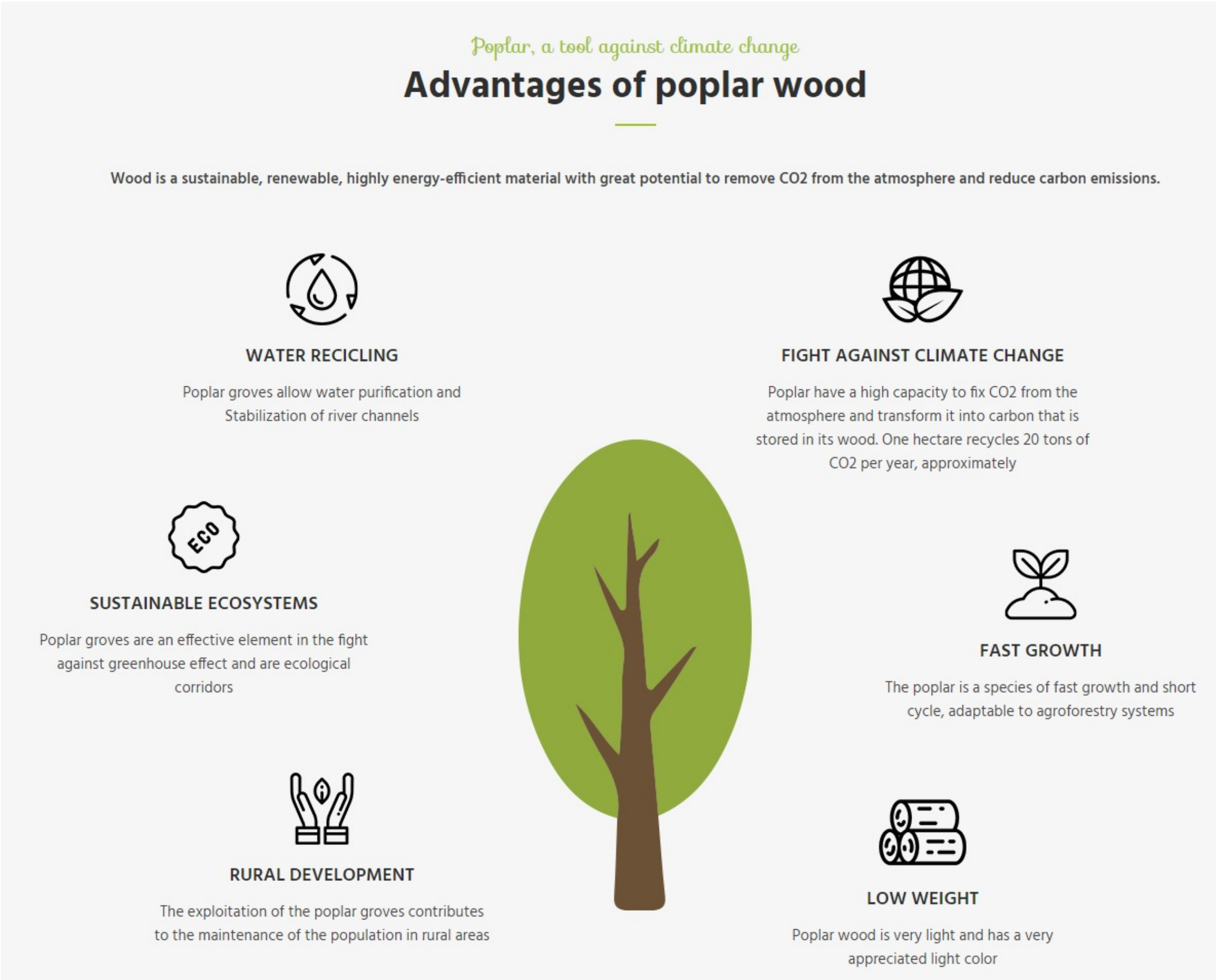


Figure 1.7 – *Poplar Wood Benefits [13]*

1

1.2 Project Goals and Objectives

In relation to the COMPOP project and having reached a phase where it is intended to verify the viability of the materials developed in real constructions, there is a need to monitor in real time the efforts supported by each element of the structure. Until now, professional laboratory equipment has been used to characterize the stresses supported by a beam, but in case of wanting to carry out measurements on a beam belonging to a construction, this equipment is impractical due to its size and cost.

Therefore, ADIME researchers ask us for an electronic product capable of monitoring an infrastructure by accurately measuring the supported forces through the use of strain gauges. Moreover, the device must has reduced dimensions and weight.

Therefore, objectives listed in Table 1.1 must be understood as the author's expected results in academic and professional terms of the execution of this project.

Ref.	Objective
Obj.1	Become familiar with the whole process of designing and manufacturing an engineered product according to customer. specifications
Obj.2	To familiarize with professional Requirements Refinement techniques by applying the Engineering Model Process.
Obj.3	To generate enough and clear documentation of the whole process, which may be required during the same project or just useful for future references or designs.
Obj.4	To dynamically acquire multidisciplinary knowledge of different specialties as needed during the execution of the project, featuring Mechanical and Building Engineering concepts and techniques.
Obj.5	To demonstrate the knowledge acquired during Bachelor's degree in Telecommunication Engineering, as well as multidisciplinary abilities gathered during the execution of this Bachelor's Thesis.
Obj.6	To successfully overcome the subject Bachelor's Thesis'.

Table 1.1 – *Top-level objectives of the project*

1.3 Project Requirements

Let us consider a certain structure, a bridge, a building or something similar, made with wood or some other material. This structure could be supporting different loads at each moment due to different factors (people passing over it, a gust of wind, etc.) In this case it would be interesting to know how much load the entire structure or a single beam is holding at each instant and how much deformation each element of the structure supports. Knowing these data in real time makes innumerable tasks easier, such as the maintenance of the structure, its repair if necessary or even in the event of an accident where the internal structure of the building ends damaged, it could be evacuated immediately, reducing the losses as much as possible.

To address this problem and solve it by the most efficient way, we should satisfy this requirements:

- Embedded system.
- Low power consumption.
- Reduced size and weight.
- Possibility of feeding it through different sources, such as solar panels o directly from the mains.
- Wireless data transmission
- Real time data processing.
- GUI (Graphic User Interface), to make easier the user interaction with the device.

1.4 Project Structure

1

This project, divided into six chapters and an addendum, progressively analyzes the system under development from different points of view, addresses the design tasks and finalizes with the successful completion of the product.

These chapters are:

- **Introduction**. This chapter, which is intended to be an introduction and show the general objectives and the reasons which motivate this project. The prior art is also included in this chapter.
- **System Analysis**. This section addresses some proposals to meet the requirements discussed in previous section and analyzes each element to be included in the final solution, in addition to the project schedule.
- **System Design**. It translates the technological solutions analyzed in the previous chapter to actual systems able to execute the tasks required. The blocks structure of the project introduced in the previous chapter is followed again, and each of them is extensively treated, including details at all levels of the design task.
- **System Manufacture**. The fourth chapter deals with **system design**. It translates the technological solutions analyzed in the previous chapter to actual systems able to execute the tasks required. The blocks structure of the project introduced in the previous chapter is followed again, and each of them is extensively treated, including details at all levels of the design task.
- **Validation**. Once the device is designed and manufactured, chapter five addresses a series of validation tests, in order to check that the system meets the **Formal Requirements** defined and, consequently, the Functional Requirements.
- **Conclusions and Future Lines**. Finally, chapter six includes the main conclusions extracted from this Bachelor's Thesis, as well as some future lines of work which have naturally emerged during the design process.
- **Appendix A**, in which the budget and associated cost of this project are detailed.
- **Appendix B**, with Altium Designer® 21 detailed schematics and PCB views.
- **Appendix C**, that contains the FW and SW code in C++ and python respectively.
- **Appendix D**, where is inserted 3D views of the PCB and the last version of the full device.

Chapter 2

System Analysis

This second chapter tackles another phase related to analysis and requirements definition. We will address some proposals to meet the requirements discussed in previous section and analyzes each element to be included in the final solution, in addition it will include the project schedule.

We start with a first proposal with which it is intended to meet the client's requirements, Figure 2.1 shows a general schematic of this possible proposal. There we can identify three main blocks. First, there is the power block, it include all the necessary for feeding the rest of the device, in this case there is a solar panel which charge a simple battery. Also we need some electronics to protect the battery from over discharge or overload, in addition to a voltage regulator to provide the needed voltage to power the device. This block should give an electrical tension as clear and continuous as possible to avoid dirtying the analog signal measurements. Another solution is using a switching power supply plugged directly into AC power.

The next block is the PCB, the main board, where all the needed electronics is housed. The transducer also should be connected to this board, the one most used for measuring deformations is the strain gauge for more details see Section 2.2. Moreover, on this PCB we can find the signal conditioner circuit whose main function is filtering the noise from the signal and amplify it to adapt this signal to the next block input range, in this case is an ADC, analog to digital converter. After that we can read the digital signal using a MCU, package it and send it to the server using the transmitter module. If the device is located in a place with mobile phone coverage, we can send the data by GPRS.

The last block is the receiver that gets the sent data and give it to the server. The server is the responsible of analyze this data and show the results to the end user.

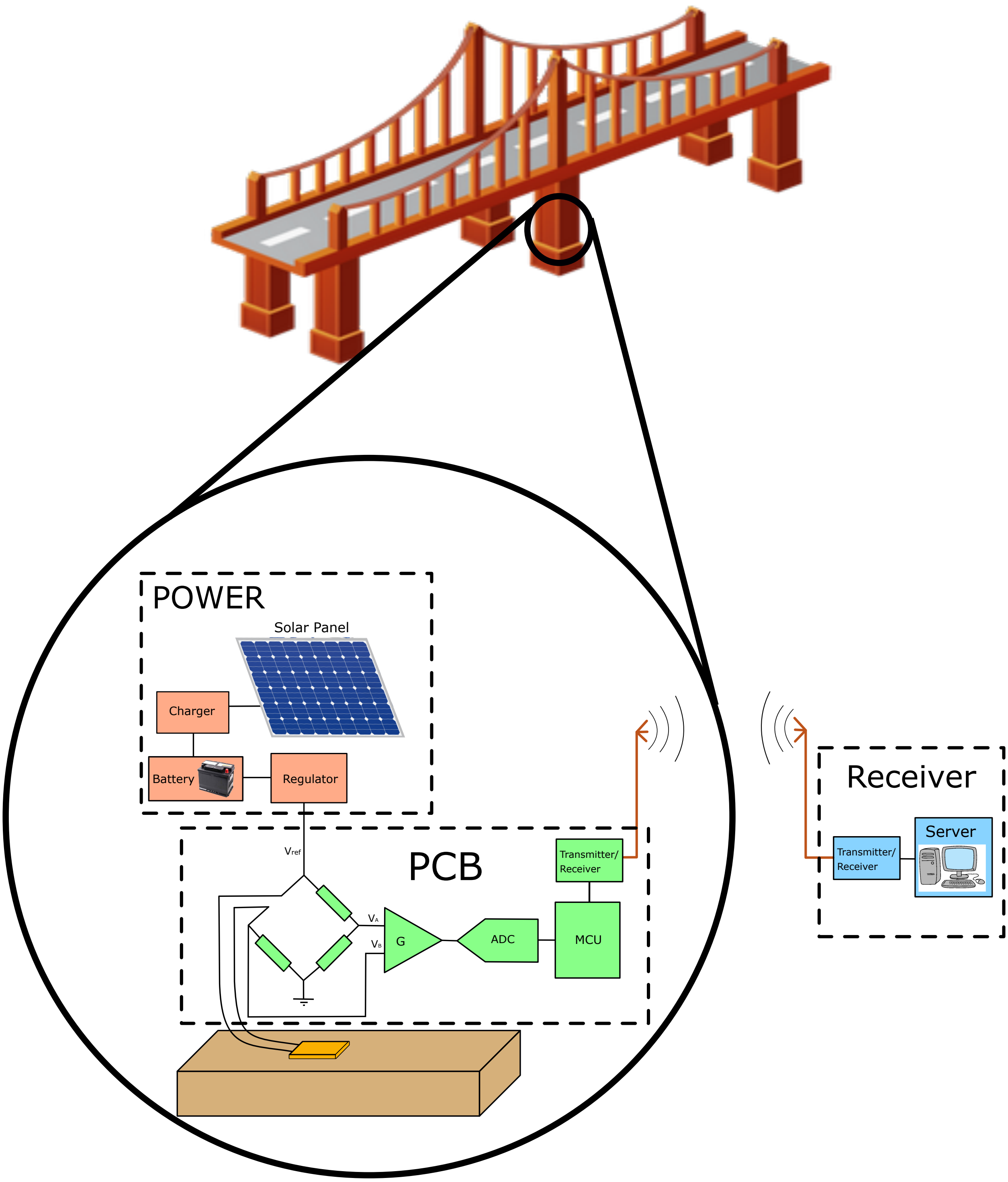


Figure 2.1 – *Schematic First Proposal*

However, in this thesis due to the short time available, only the electronics block could be developed in addition to a simple server that shows the data collected in real time using a GUI. The rest of the blocks (power and wireless transmission) will be left for future projects.

In following sections, each element will be analyzed separately and explained the reason for choosing a specific element instead of a different one that performs the same function.

2.1 Micro-Controler Unit

The MCU is a fundamental element in the system and is in charge of monitoring all the processes, therefore, among all the different options that we can find in the market, we must choose one that has enough performance to process the amount of data required in the process, however we should not take one with much higher performance because it would increase the cost of the product without being sense.

2.1.1 PIC-18F4550 micro-controller

The first option we will discuss is using a PIC-18F4550 micro-controller with all the external electronics needed and integrate all this in a PCB. This allow us to make a more flexible and compact design with cheaper cost. However it will take more time in the design and manufacture.

Figure 2.2 – *PIC-18F4550 [26]*

2.1.2 Arduino

This open hardware/software development board with abundant documentation on internet, it is another good option to test the different components of the system easily and quickly in this first prototyping phase. It allow us reducing the designing and developing time, although prototype will be less compact and larger than other solutions.

Figure 2.3 – *Arduino UNO [24]*

2.1.3 ESP-32

Is a low-cost, low-power SoC micro-controllers with integrated Wi-Fi and dual-mode Bluetooth. The ESP32 series employs either a Tensilica Xtensa LX6 microprocessor, Xtensa LX7 dual-core microprocessor or a single-core RISC-V microprocessor and includes built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power-management modules. [40]

This board is very versatile and it can allow us wireless communication using the Wi-Fi.

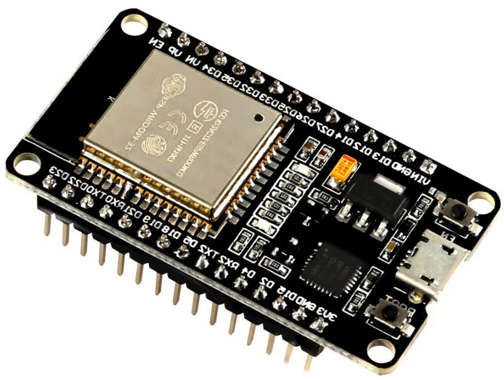


Figure 2.4 – *ESP-32 [25]*

In this first version of the prototype, we are going to give more priority to the correct operation of the transducer, as well as the signal conditioning circuit and the sampling and retention circuit.

Once these components been correctly characterized and studied the noise and interference levels, we can proceed with a second version where we integrate a micro-controller in a PCB like the PIC-18F4550, but this will be the object of study of

MCU	PIC18F4550	Arduino UNO	ESP32
RAM (KB)	2	2	520
Flash (KB)	32	32	448
EEPROM (B)	256	1240	- -
GPIO	35	20	34
CLK (MHz)	48	16	30
Price (€)	1.76	22.50	2.15

Table 2.1 – *MCU comparative*

another project.

In this case and looking at the Table 2.1, the ESP-32 provides better performance in addition to the possibility of transmitting data by Wi-Fi. However, as I mentioned, we will prioritize design and manufacturing speed in order to have the device operational and be able to make all the required tests. That is why we decided to **use** an **Arduino UNO**, in addition to the abundant documentation that we can find on Internet makes the process much easier.

2.2 Strain Gauge Analisys

The experimental stress analysis is carried out by measuring the deformation of the part under load, in order to infer the local forces from there. The knowledge of the mechanical stresses supported by a structure is essential for appreciating the safety of its operation. The tension generates deformations of the material to which they are subjected and the relation between the magnitudes: strength and deformation, can be explicated by the theory of Strength of materials or the Structure Calculation. The most used transducer for measuring material deformation is the strain gauge, it sticks to the structure, being subjected to the same deformation as that. The most used kind of strain gauges are the resistive one.

The strain gauges have many applications, not only for measuring the stress, but also in the measure of other physical quantities, in particular mechanical, in which the action on a test body causes deformation that can be measured by strain, such as pressure, force, acceleration, torque, etc. However, for this project we will focus only on stress measure which is the most simple way for monitoring a structure. [1]

Figure 2.5 shows an linear strain gauge which can measure the strain in one direction. This is one of the most used type of strain gauge.

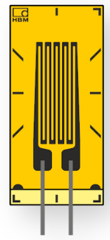


Figure 2.5 – *Linear Strain Gauge*

2.2.1 Physical Fundamentals

Consider the bar shown in Figure 3.1, which is applied an axial load, T. Without the load, it has a length L1, its diameter is D1, and the cross-sectional area is A. If a load is applied without exceeding the elastic limit of the material, the axial strain is given by:

$$\varepsilon_a = \frac{L_2 - L_1}{L_1} = \frac{\Delta L}{L} \quad ; \quad L = L_1 \quad ; \quad \Delta L = L_2 - L_1 \tag{2.2.1}$$

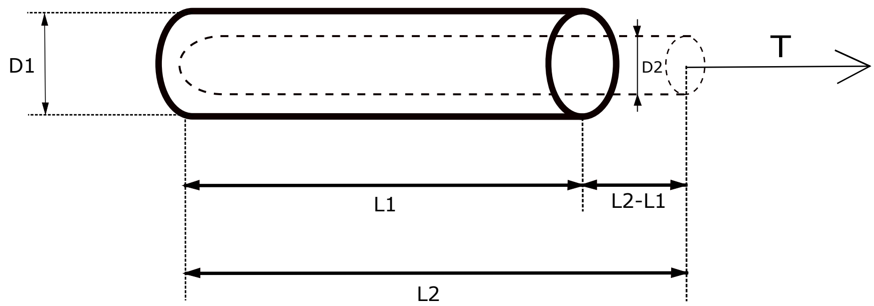


Figure 2.6 – *Axial load aplied to a bar*

This represents the strain per unit length and it is related to the modulus of elasticity or Young's, E, of the material, to the strength, T and the area A by the equation:

$$\varepsilon_a = \frac{\Delta L}{L} = \frac{T/A}{E} \tag{2.2.2}$$

Equation 2.2.2 is known as Hooke law, it indicates that the strain is proportional to stress and inversely proportional to the area.

As it is shown in Figure 2.6, when an axial deformation appears, a transverse deformation also occurs in the bar. It is evident that there are an area variation due to the diameter variation. The ratio of the strain in the transverse direction between the strain in the axial direction is shown Equation 2.2.3 and it is known as Poisson's ratio.

$$\nu = -\frac{\varepsilon_t}{\varepsilon_a} \tag{2.2.3}$$

Knowing:

$$\varepsilon_t = -\frac{D_2 - D_1}{D_1} = \frac{\Delta D}{D} \quad ; \quad D_1 = D \tag{2.2.4}$$

The minus sign in Equation 2.2.3 indicates that ε_t and ε_a have opposite signs. The value of Poisson ratio should be experimentally determined and it is typically 0.3 for many materials.

As we said, the most used kind of strain gauges are the resistive one. We are able to know its deformation by measuring its resistance variation. Considering a conductor with L length, A section and ρ resistivity, then its resistance according to ohm's law is:

$$R = \rho \frac{L}{A} \tag{2.2.5}$$

Considering a wire with circular section and diameter D, the area A will be mD^2 with $m = \pi/4$, whereas if the section is squared, $m = 1$. Then Equation 2.2.5 converts to:

$$R = \rho \frac{L}{mD^2} \tag{2.2.6}$$

If we derive the Equation 2.2.6:

$$dR = \frac{mD^2\ (L\ d\rho + \rho\ dL) - 2m\ \rho LD\ dD}{(mD^2)^2} \tag{2.2.7}$$

$$dR = \frac{1}{mD^2}\left[(Ld\rho + \rho dL) - 2\rho L\frac{dD}{D}\right] \tag{2.2.8}$$

It corresponds with a unitary variation:

$$\frac{dR}{R} = \frac{dL}{L} - 2\frac{dD}{D} + \frac{d\rho}{\rho} \tag{2.2.9}$$

According with ε_a and ε_t definitions in Equation 2.2.3, it results:

$$\varepsilon_a = \frac{dL}{L} \quad ; \quad \varepsilon_t = \frac{dD}{D} \quad ; \quad \nu = -\frac{dD/D}{dL/L} \tag{2.2.10}$$

Then equation will be:

$$\frac{dR}{R} = \frac{dL}{L} + 2\nu\,\frac{dL}{L} + \frac{d\rho}{\rho} \tag{2.2.11}$$

2

The resistivity variation, dρ, due to volume variation dV of the material, is known as piezoresistive effect, which have the following expression:

$$\frac{d\rho}{\rho} = C\frac{dV}{V} \tag{2.2.12}$$

Where C is the Bridgman constant. Knowing that $V = AL = mD^2L$, then:

$$dV = (2mDL)\,dD + \left(mD^2\right)\,dL \tag{2.2.13}$$

From Equation 2.2.10 the unitary volume variation will be:

$$\frac{dV}{V} = s\frac{dD}{D} + \frac{dL}{L} = (1-2\nu)\,\frac{dL}{L} \tag{2.2.14}$$

Keeping in mind Equation 2.2.12 and Equation 2.2.14, then the Equation 2.2.11 will be :

$$\frac{dR}{R} = \frac{dL}{L} + 2\nu\,\frac{dL}{L} + C\,(1-2\nu)\,\frac{dL}{L} \tag{2.2.15}$$

The ratio $\frac{dR/R}{dL/L}$ is known as gauge factor, K, its value is:

$$K = \frac{dR/R}{dL/L} = 1 + 2\nu + C\,(1-2\nu) \tag{2.2.16}$$

In case of metallic materials, in elastic domain the Poisson's ratio is 0.3 more or less while Bridgman constant is practically equal to one, that's why the gauge factor is about 2. Table 2.2 shows the characteristics of the principal materials used in resistive gauges. The most used one is the constantan.

MATERIAL	COMPOSITION	K FACTOR	RESISTIVITY (Ω/cm)
Nichrome V	80% Ni; 20% Cr	2.0	108
Constantan	45% Ni; 55% Cu	2.0	49
Karma	74% Ni; 20% Cr; 3% Al; 3% Fe	2.4	130
Platinum	100% Pt	4.8	10
Nickel	100% Ni	-12	7.8

Table 2.2 – *Materials for Strain Gauge*

For measuring the deformation of a structure the gauge will be glued to it surface, then measuring the resistance variation and using Equation 2.2.16, it will be possible to calculate the strain:

$$\varepsilon_a = \frac{1}{K} \cdot \frac{\Delta R}{R} \tag{2.2.17}$$

The value of K and R is provided by the manufacturer, then we only need to measure ΔR for knowing the strain ε_a and also structure deformation.

It is possible to measure deformations with a single wire, however, it will be necessary a lot of current for feeding it. That's why we need 100 Ω resistance at least and usually it is folded into a grid for measuring only in one place. In addition, the gauges are manufactured with printed circuit methods reducing the thickness of the mesh to eliminate the dependence on transverse deformations.

2.2.2 Temperature effects

In some cases during the measurement process we have to deal with temperature variations and their effect on the circuit and therefore on the measurements taken. This is an issue to take into account because our device will be located outdoors for long periods of time. There is four effects of temperature changes on our circuit:

- Variation on gauge factor, K.
- The metallic grid lengthens or contracts.
- The support material of the metallic grid lengthens or contracts with a different coefficient than the grid.
- Change the resistivity of the grid wire, then also the resistance of the gauge.

One way for eliminate this effects is using self-compensated strain gauges, they are special gauges designed with appropriate materials for the grid and the support, that can reduce the effect. Another way to get it is with the electrical measurement system, making use of passive gauges, called *dummy*, this method is analyzed in Subsection 2.2.3.

The Figure 2.7 shows a graph taken from the datasheet which represents the apparent strain perceived in the gauge as a function of temperature, from $-10\,^oC$ to $120\,^oC$. We can see that this gauges are calibrated at $20\,^oC$ where the apparent strain is 0 and it can arrive to $-130\mu m/m$.

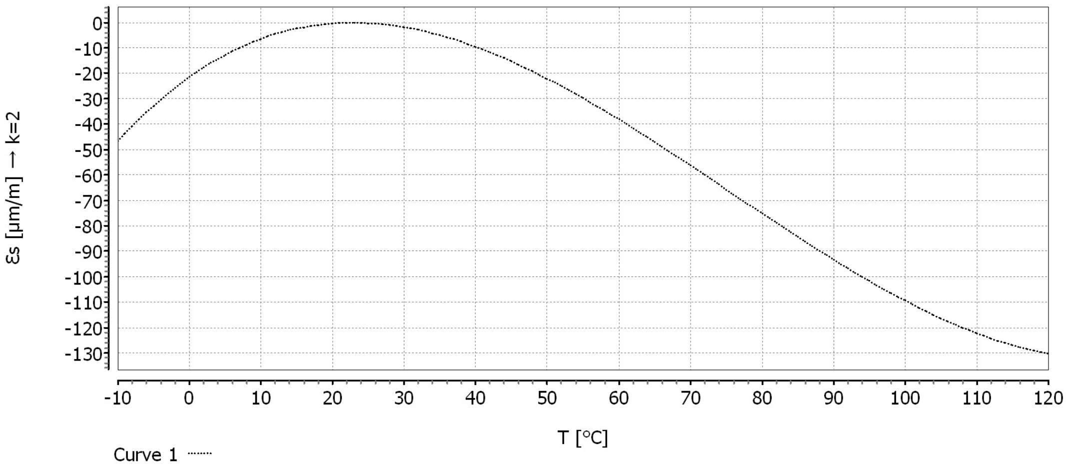


Figure 2.7 – *Apparent Strain in a gauge*

To characterize this curve, the manufacturer provide us with a third-order interpolated function. Equation 2.2.18 belongs to the curve in Figure 2.7.

$$\varepsilon_s\,(T) = -21.44 + 1.99 \cdot T - 5.15 \cdot 10^{-2} \cdot T^2 + 2.28 \cdot 10^{-4} \cdot T^3\,[\mu\text{m/m}] \tag{2.2.18}$$

Taking into account this apparent elongation, the Equation 2.2.17 becomes:

$$\frac{\Delta R}{R} = K\,(\varepsilon + \varepsilon_s) \tag{2.2.19}$$

2.2.3 Measuring circuits

According with Equation 2.2.17, if $K = 2$, $R = 120\ \Omega$ and we want to measure a strain of $10^{-4} = 100\mu\varepsilon$ then the resistance will be:

$$\Delta R = KR\varepsilon = 0.024$$

That supposes resistance variations about 0.02%. Then we need a very sensitive circuits to measure such small variations. The most used one is the Wheatstone Bridge which is represented in Figure 2.8, it is formed by two voltage dividers getting a differential output, $V_{out} = V_A - V_B$, that reduces the output noise in case of fluctuates the source. The expression of the output voltage is in Equation 2.2.20

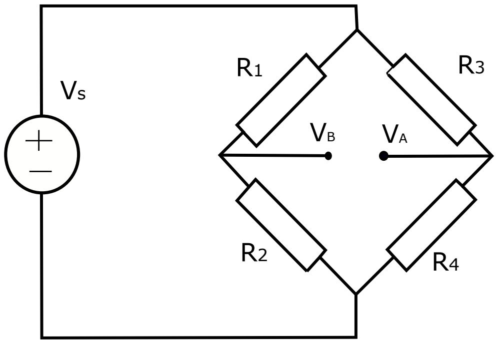


Figure 2.8 – *Wheatstone Bridge*

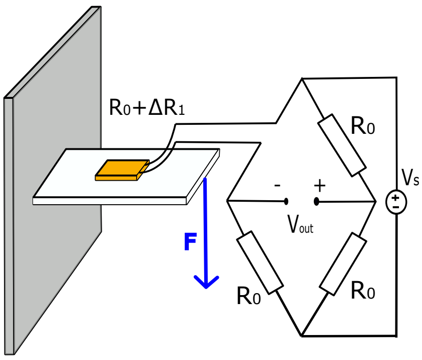
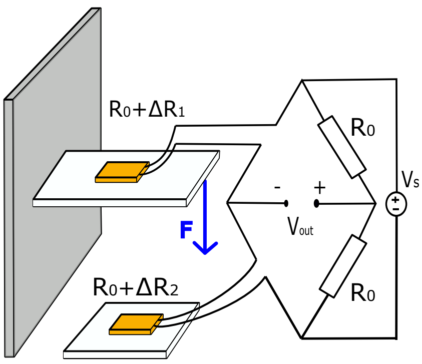
$$\begin{aligned} V_{out} = V_{AB} = V_A - V_B = V_s \left(\frac{R_4}{R_3 + R_4} - \frac{R_2}{R_1 + R_2} \right) = \\ = V_s \frac{R_4 (R_1 + R_2) - R_2 (R_3 + R_4)}{(R_1 + R_2)(R_3 + R_4)} = \\ = V_s \frac{R_1 R_4 - R_2 R_3}{(R_1 + R_2)(R_3 + R_4)} \end{aligned} \tag{2.2.20}$$

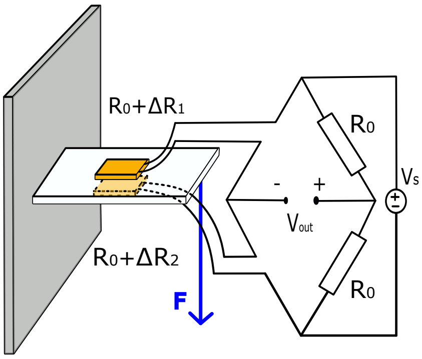
From Equation 2.2.20 we can say the bridge is balanced if $R_1 R_4 = R_2 R_3$ or equivalently $\frac{R_1}{R_2} = \frac{R_3}{R_4}$. For simplifying we will use the same value for all resistance, $R_1 = R_2 = R_3 = R_4 = R_0$. Assuming a general case in which the four arms of the bridge have some transducer and therefore can present variations around the value of R_0, considering the resistance of each gauge is $R_i = R_0 + \Delta R_i$, then Equation 2.2.20 could be simplified as:

$$V_{out} = \frac{V_s}{4} \frac{\Delta R_1 - \Delta R_2 - \Delta R_3 + \Delta R_4}{R_0} \tag{2.2.21}$$

Generally there is no fixed rule for designating the components of the Wheatstone bridge, in theory there are all kind of designations, this affects the measurements and it all depends on the required application. With the different configurations of the bridge it is possible to compensate the effects of temperature, obtain measurements of higher resolution or measure deformations due only to bending or only to tension or compression. The Table 2.3 collects some interesting examples.

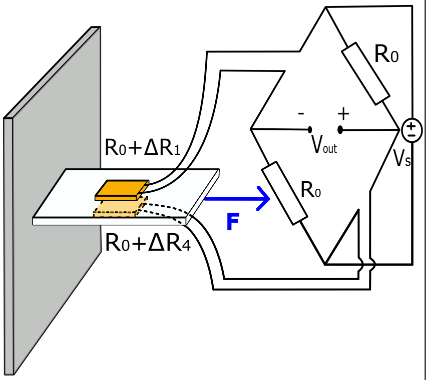
Table 2.3 – *Wheatstone Bridge Configurations*

Configuration		Expression	Description
Quarter Bridge		$\Delta R_2 = \Delta R_3 = \Delta R_4 = 0$ $\frac{\Delta R_1}{R_0} = K(\varepsilon + \varepsilon_s)$ $V_{out} = \frac{V_s}{4}\frac{\Delta R_1}{R_0} =$ $= \frac{V_s}{4} K(\varepsilon + \varepsilon_s)$ $\varepsilon = \frac{V_{out}}{V_s}\frac{4}{K} - \varepsilon_s$	Simple quarter bridge circuit with only one active gauge. It is not possible to measure the strain due to bending or tension/compression separately (superimposed bending). Measurement depends on the temperature.
Quarter Bridge with dummy strain gauge		$\frac{\Delta R_1}{R_0} = K(\varepsilon + \varepsilon_s)$ $\Delta R_3 = \Delta R_4 = 0$ $\frac{\Delta R_2}{R_0} = K\varepsilon_s$ $V_{out} = \frac{V_s}{4}\frac{\Delta R_1 - \Delta R_2}{R_0} =$ $= \frac{V_s}{4} K\varepsilon$ $\varepsilon = \frac{V_{out}}{V_s}\frac{4}{K}$	Bridge with two strain gauges, one actively measures strain, the other is mounted on a passive component made of the same material but not strained (dummy gauge). Temperature effect is well compensated. Normal and bending strain cannot be separated (superimposed bending)

<table>
<tr>
<td>Half Bridge</td>
<td>
</td>
<td>$$\frac{\Delta R_1}{R_0} = K\left(\varepsilon_1 + \varepsilon_s\right)$$
$$\frac{\Delta R_2}{R_0} = K\left(\varepsilon_2 + \varepsilon_s\right)$$
$$\Delta R_3 = \Delta R_4 = 0$$
$$\varepsilon_1 = -\varepsilon_2 = \varepsilon$$
$$V_{out} = \frac{V_s}{4}\frac{\Delta R_1 - \Delta R_2}{R_0} =$$
$$= \frac{V_s}{4} 2K\varepsilon$$
$$\varepsilon = \frac{V_{out}}{V_s}\frac{2}{K}$$</td>
<td>Two strain gauges are installed on opposite sides of the structure.

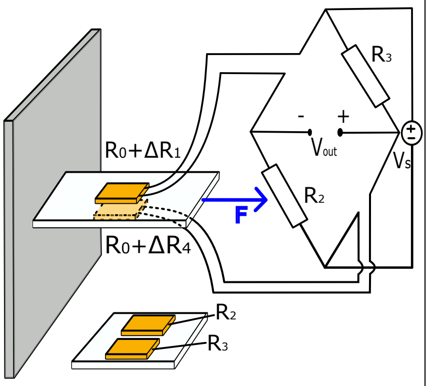
Temperature effect is well compensated.

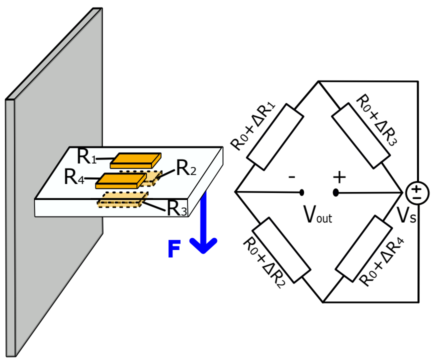
Separation of normal and bending strain, only the bending effect is measured.

More resolution than quarter bridge.</td>
</tr>
<tr>
<td>Diagonal Bridge</td>
<td>
</td>
<td>$$\frac{\Delta R_1}{R_0} = K\left(\varepsilon_1 + \varepsilon_s\right)$$
$$\frac{\Delta R_4}{R_0} = K\left(\varepsilon_4 + \varepsilon_s\right)$$
$$\Delta R_2 = \Delta R_3 = 0$$
$$\varepsilon_1 = \varepsilon_4 = \varepsilon$$
$$V_{out} = \frac{V_s}{4}\frac{\Delta R_1 + \Delta R_4}{R_0} =$$
$$= \frac{V_s}{4} 2K\left(\varepsilon + \varepsilon_s\right)$$
$$\varepsilon = \frac{V_{out}}{V_s}\frac{2}{K} - \varepsilon_s$$</td>
<td>Two strain gauges are installed on opposite sides of the structure.

It is only sensitive to normal strain independently of bending.

Temperature effect is not compensated.</td>
</tr>
</table>

<table>
<tr>
<td>Diagonal Bridge with dummy gauge</td>
<td>
</td>
<td>$$\frac{\Delta R_1}{R_0} = K\left(\varepsilon + \varepsilon_s\right)$$
$$\frac{\Delta R_4}{R_0} = K\left(\varepsilon + \varepsilon_s\right)$$
$$\frac{\Delta R_2}{R_0} = K\varepsilon_s$$
$$\frac{\Delta R_3}{R_0} = K\varepsilon_s$$
$$V_{out} = \frac{V_s}{4} 2K\varepsilon$$
$$\varepsilon = \frac{V_{out}}{V_s}\frac{2}{K}$$</td>
<td>Two active gauges are installed on opposite sides of the structure and two passive gauges for compensate temperature effects.

Sensitive only to normal strain.</td>
</tr>
<tr>
<td>Full Bridge</td>
<td>
</td>
<td>$$\frac{\Delta R_1}{R_0} = K\left(\varepsilon + \varepsilon_s\right)$$
$$\frac{\Delta R_2}{R_0} = K\left(-\varepsilon + \varepsilon_s\right)$$
$$\frac{\Delta R_3}{R_0} = K\left(-\varepsilon + \varepsilon_s\right)$$
$$\frac{\Delta R_4}{R_0} = K\left(\varepsilon + \varepsilon_s\right)$$
$$V_{out} = \frac{V_s}{4} 4K\varepsilon$$
$$\varepsilon = \frac{V_{out}}{V_s}\frac{1}{K}$$</td>
<td>Four active gauges, two in each side of the structure.

Only bending effect is measured.

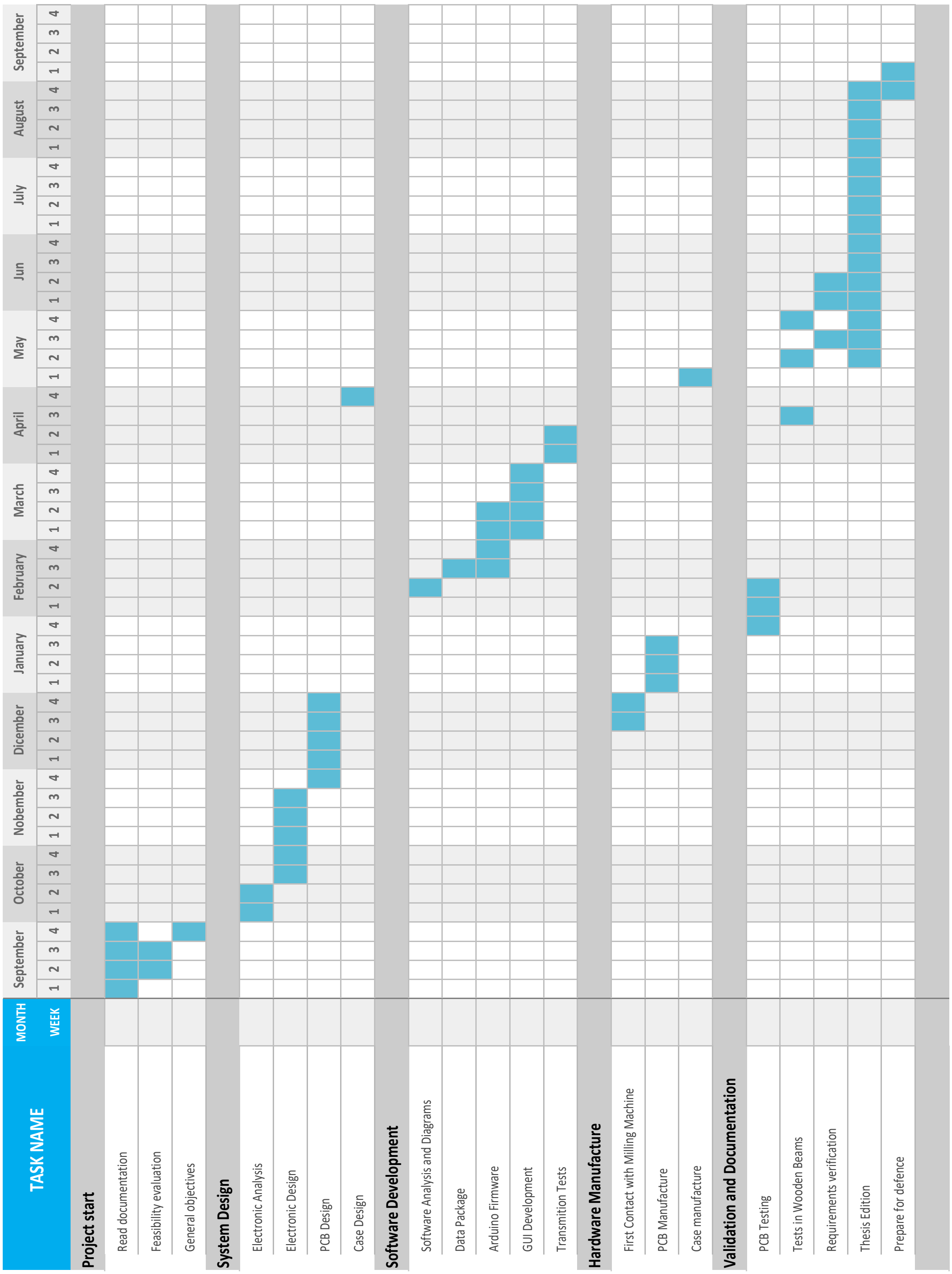
It have a higher output signal.

Temperature effects are well compensated.</td>
</tr>
</table>

Table 2.3 – *Wheatstone Bridge Configurations*

2.3 Schedule and Gantt Chart

Sharif Al-Husein Raie


TASK NAME
MONTH
WEEK
September
October
Nobember
Dicember
January
February
March
April
May
Jun
July
August
September
Project start
Read documentation
Feasibility evaluation
General objectives
System Design
Electronic Analysis
Electronic Design
PCB Design
Case Design
Software Development
Software Analysis and Diagrams
Data Package
Arduino Firmware
GUI Development
Transmition Tests
Hardware Manufacture
First Contact with Milling Machine
PCB Manufacture
Case manufacture
Validation and Documentation
PCB Testing
Tests in Wooden Beams
Requirements verification
Thesis Edition
Prepare for defence

Chapter 3

System Design

In this section we will address the entire design process of the embedded system, starting with the design of the PCB, for that we used Altium Designer® 21 program, that is one of the most used tools for designing printed circuit boards. Then we will continue with the design of the case that will protect the device, for this we will use the SolidWorks® program. The whole manufacture process of the PCB and the case according to the design criteria will be explain in Chapter 4.

3.1 Hardware design

The main Hardware elements will be included inside a PCB to deal with them and test them in an easy way. Inside this PCB we should include all the required electronics for measure the resistance variation of the strain gauge and thereby calculate the deformation. Therefore, we should be careful with this tiny analog signal and avoid interference from different noise sources, such as the power supply or the wireless communication, to get measurements as clear as possible.

This first version of the PCB aims to check the operation of the signal conditioner circuit together with the module **HX711** which is formed by a PGA (Programmable Gain Amplifier) and an ADC (Analog to Digital Converter). Moreover, the noise level in the Wheatstone bridge will be tested when the circuit is fed by different sources. Therefore, its professional finish will not be cared for and that is why we will use a simple Arduino UNO. This will allow us to have the PCB working in the shortest possible time. After testing this board, it is intended to design another version with a more professional finish, correcting the errors that the first one may have and using a microcontroller soldered on the PCB instead of the Arduino.

Starting with the general diagram shown in Figure 3.1, the board will consist of an

Arduino UNO (a) in charge of reading the analog signals through the ADC and sending them to the PC (b) for analyze them in real time. The module **HX711** (c) is formed by a PGA that allows us to select different value of gain, an 24 bit ADC and also it has got on-chip power supply regulator for feeding the Wheatstone bridge reducing electric noise as much as possible, in addition it has got two channels to measure simultaneously different signals, all that in a single 16 pin SOP-16 package, that make it a very versatile device.

The board will also have the Wheatstone bridge resistors embedded, Figure 3.1 (d) and connected with the strain gauges though a connector (g). Finally we will include a small display (f), controlled by I2C for debugging purposes. Every single block will be explained with more details in the following sections.

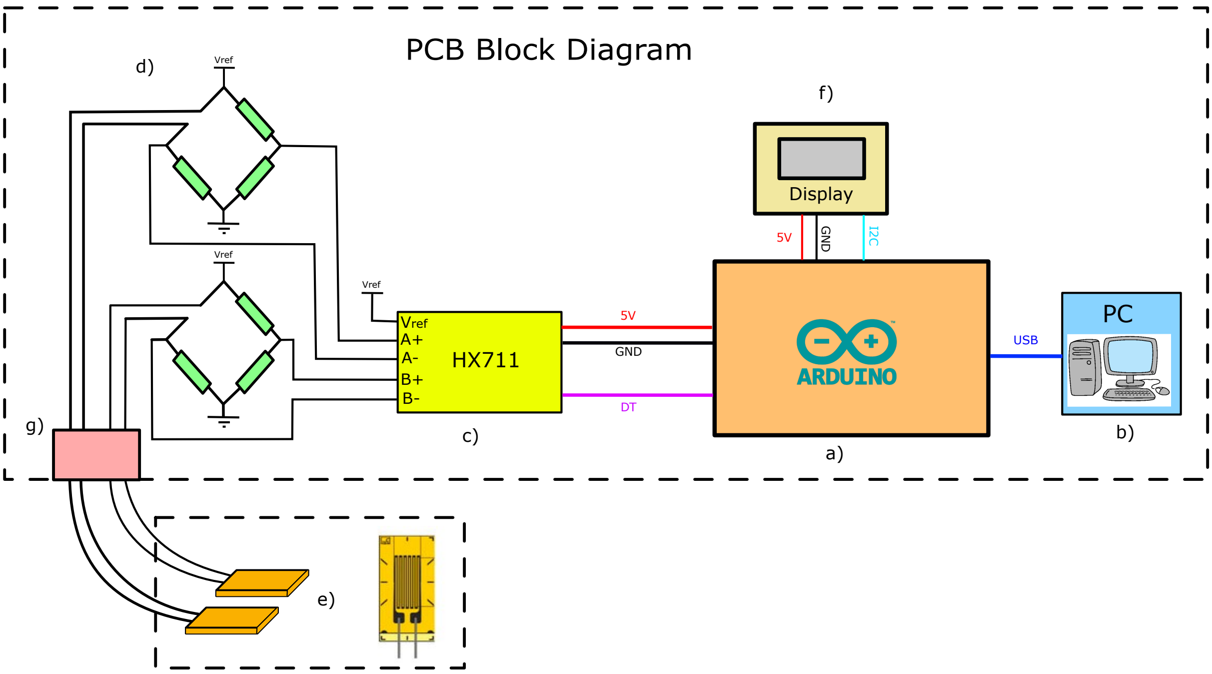


Figure 3.1 – *PCB Diagram*

3.1.1 HX711 Module

This module is designed for weigh scales and industrial control applications to interface directly with a bridge sensor. Figure 3.2 represents the block diagram of the module, we can differentiate the following main blocks:

- A input multiplexer to select a differential input from Channel A or B and carries the signal to the (PGA)
- The PGA is a low noise programmable gain amplifier. It can be programmed with a gain of 128 or 64 for Channel A, then when the AVDD pin (analog power) is connected to 5 V, the full-scale differential input voltage will correspond to ±20 mV or ±40 mV respectively. Whereas channel B has a fixed gain of 32 and the input full-scale will be ±80 mV, when the analog power is 5 V.
- 24 bit ADC
- *On-chip Power Supply Regulator*. It eliminates the need for an external supply regulator to provide analog power for the ADC and the Wheatstone Bridge.
- A flexible input clock, it can be from an external clock source, a crystal, or the on-chip oscillator that does not require any external component.
- Selectable frequency of sampling between 10 SPS or 80 SPS.

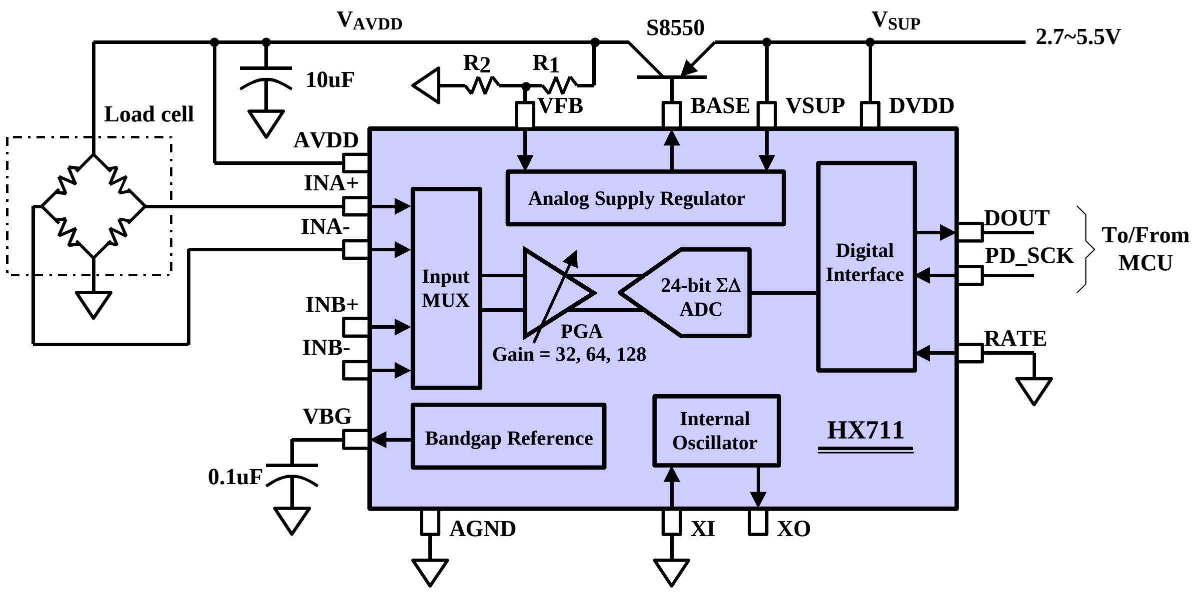


Figure 3.2 – *HX711 Block Diagram* [20]

Using this module allow us to simplify the signal conditioner circuit, the hold and sampling circuit and the voltage regulator for feeding the bridge with a only device with SOP-16 package. This fact simplify the entire design process and reduce the costs.

However later we should test the device performance and validate if the level of noise are under the project requirements.

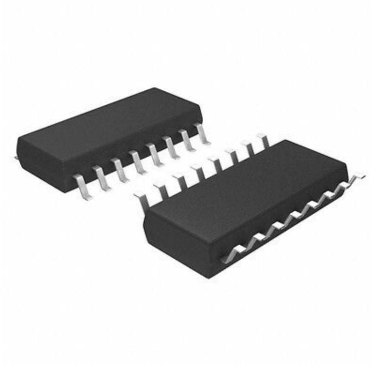


Figure 3.3 – *HX711 SOP-16 Package*

In this first PCB version we will use the HX711 integrate soldered in a shield as Figure 3.4 shows. That will simplify the design process and will allow us to have the PCB ready to test earlier. For use this shield we only need a 5 V power supply and connect it to the MCU by two digital pins, *CLK* and *DT*.

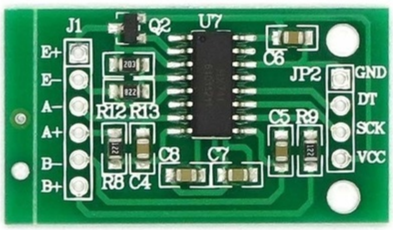


Figure 3.4 – *Shield with HX711*

In Figure 3.5 it is shown the connection of the HX711 shield to the resistor bridge from one side and to the MCU from the other side. In addition there are a couple of bypass capacitors, a ceramic one with 100 nF and the other is electrolytic with 100 μF.

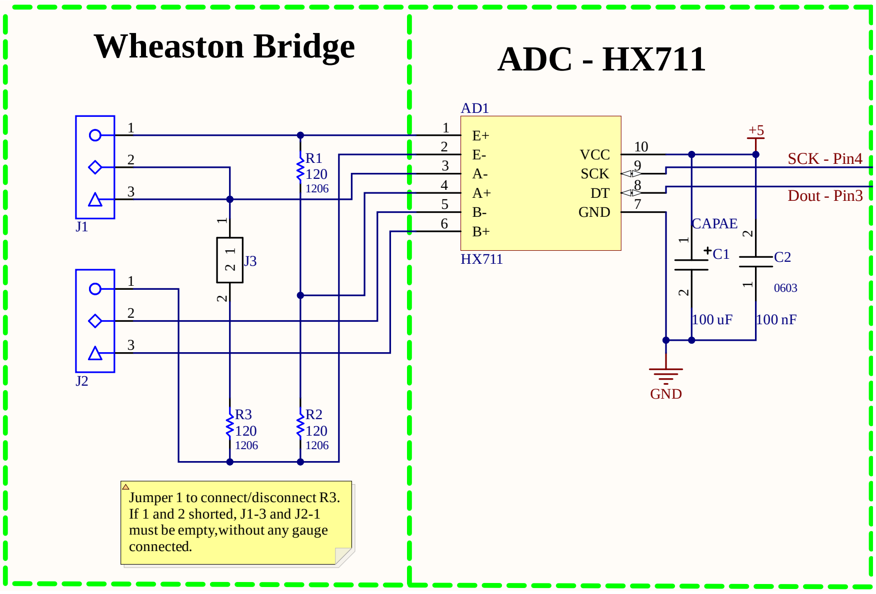


Figure 3.5 – *HX711 module connections*

3.1.2 Wheatstone Bridge Configuration

Firstly, for our application, where the device will be located outside under frequent temperature changes, we need a circuit that allows us compensate this temperature effects. The simplest one of all those exposed in Table 2.3 is the *Quarter Bridge with a dummy gauge*. For this configuration we need two strain gauges connected to the *Wheatstone Bridge* and as can be seen in Table 2.3, this circuit is quite similar to the *Half-Bridge* circuit, with the only difference that one have a dummy gauge while in the second both gauges support tension. Then we can change from one configuration to another only changing the position of the second gauge. That will allow us to test both configurations, but we should remember that the output signal of the *Half-Bridge* circuit has twice the amplitude of the *Quarter Bridge* one, that could saturate the input of the next stage, the amplifier.

Furthermore, this purpose of this board is to test the different elements of the circuit. Then it would be interesting to compare this bridges configuration with the simple *Quarter Bridge*

Then to combine all these configurations according to our needs we will implement a circuit like the one shown in the Figure 3.6, where the element *"J1"* is a jumper. When the jumper is close we can measure from a *Quarter Bridge* and when it is open from a *Half Bridge* or a *Quarter Bridge* with *dummy* gauge. Also we will use some connectors to connect one or two gauges depending on each case.

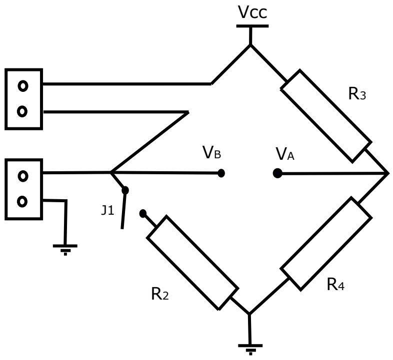


Figure 3.6 – *Multi-Configuration Wheatstone Bridge*

3

The output of the Bridge in Figure 3.6 goes to the channel 1 of the module HX711, which has another channel. For this second one we need another Bridge so we will implement another simple *Quarter Bridge* with its connector for the gauge as we can see in Figure 3.7 and then we can also compare the differences between both channels.

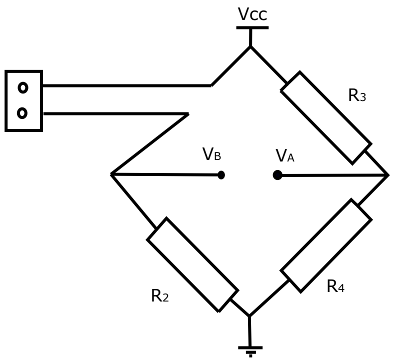


Figure 3.7 – *Quarter Bridge for HX711 second channel*

3.1.3 Strain Gauge Connectors

Depending on the circuit configuration used, we can have one, two or even three strain gauges connected at the same time. In this case we will use the connectors that are shown in Figure 3.8 which have three terminals, several of them can be joined to form a more solid set thanks to the tabs on their sides. In this case, each strain gauge will be connected throw two wires, then we will need two connectors like this.

Figure 3.8 – *Strain Gauge Connector [19]*

3.1.4 Display OLED

In this version we will include a small Display for different purposes such as showing different debugging messages or display the data measured. The display used is a 0.96” OLED controlled by I2C interface, the Figure 3.9 shows the display.

Figure 3.9 – *Display OLED [23]*

3.1.4.1 I2C communication

I2C communication standard is the mostly widely used inter-chip communication standard in today's electronic systems. It is an open-drain/open-collector communication standard which implies integrated circuits (IC) with possibility to connect different voltage supply rails for communication.

Pullup resistors need to be connected from the I2C lines to the supply to enable communication. The pull up resistors pull the line high when it is not driven low by the open-drain interface. For enabling I2C communication without troubles we should design properly the values of this resistor, a wrong value of pullup resistors can lead to signal loss.

3

The minimum value of the pullup resistor is determined by the maximum voltage level that can be read as a valid logical low by the input buffers of an IC. While the maximum value for this resistors is limited by the capacitance of the bus (C_b) and the standard rise time specifications. Keeping in mind that the rise time of an RC circuit is characterized by the constant RC, then if the value of R is too high, the I2C line may not rise to a logical high before it is pulled low.

Figure 3.10 is taken from the Arduino UNO micro-controller data-sheet (ATmega328P) where the I2C requirements are specified and the limit value for pullup resistor is:

$$R_{min} = \frac{V_{cc} - 0.4\ \text{V}}{3\ \text{mA}} \tag{3.1.1}$$

$$R_{max} = \frac{300\ \text{ns}}{C_b} \tag{3.1.2}$$

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage		-0.5	$0.3V_{CC}$	V
V_{IH}	Input High-voltage		$0.7V_{CC}$	$V_{CC} + 0.5$	
V_{hys}(1)	Hysteresis of Schmitt Trigger Inputs		$0.05V_{CC}$(2)	–	
V_{OL}(1)	Output Low-voltage	3mA sink Current	0	0.4	
t_r(1)	Rise Time for both SDA and SCL		$20 + 0.1C_b$(3)(2)	300	ns
t_{of}(1)	Output Fall Time from V_{IHmin} to V_{ILmax}	$10pF < C_b < 400pF$(3)	$20 + 0.1C_b$(3)(2)	250	
t_{SP}(1)	Spikes Suppressed by Input Filter		0	50(2)	
I_i	Input Current each I/O Pin	$0.1V_{CC} < V_i < 0.9V_{CC}$	-10	10	µA
C_i(1)	Capacitance for each I/O Pin		–	10	pF
f_{SCL}	SCL Clock Frequency	f_{CK}(4) $> \max(16f_{SCL}, 250kHz)$(5)	0	400	kHz
Rp	Value of Pull-up resistor	$f_{SCL} \leq 100kHz$	$\frac{V_{CC} - 0.4V}{3mA}$	$\frac{1000ns}{C_b}$	Ω
		$f_{SCL} > 100kHz$	$\frac{V_{CC} - 0.4V}{3mA}$	$\frac{300ns}{C_b}$	
$t_{HD;STA}$	Hold Time (repeated) START Condition	$f_{SCL} \leq 100kHz$	4.0	–	µs
		$f_{SCL} > 100kHz$	0.6	–	
t_{LOW}	Low Period of the SCL Clock	$f_{SCL} \leq 100kHz$(6)	4.7	–	
		$f_{SCL} > 100kHz$(7)	1.3	–	
t_{HIGH}	High period of the SCL clock	$f_{SCL} \leq 100kHz$	4.0	–	
		$f_{SCL} > 100kHz$	0.6	–	
$t_{SU;STA}$	Set-up time for a repeated START condition	$f_{SCL} \leq 100kHz$	4.7	–	
		$f_{SCL} > 100kHz$	0.6	–	
$t_{HD;DAT}$	Data hold time	$f_{SCL} \leq 100kHz$	0	3.45	
		$f_{SCL} > 100kHz$	0	0.9	
$t_{SU;DAT}$	Data setup time	$f_{SCL} \leq 100kHz$	250	–	ns
		$f_{SCL} > 100kHz$	100	–	
$t_{SU;STO}$	Setup time for STOP condition	$f_{SCL} \leq 100kHz$	4.0	–	µs
		$f_{SCL} > 100kHz$	0.6	–	
t_{BUF}	Bus free time between a STOP and START condition	$f_{SCL} \leq 100kHz$	4.7	–	
		$f_{SCL} > 100kHz$	1.3	–	

Figure 3.10 – *I2C Requirements for ATmega328P [21]*

Then we need to estimate the capacitance of the tracks to calculate the maximum value for pullup resistor. For that we will use a program called Saturn PCB, which is a very useful tool for make microstrip calculations. In Figure 3.11 we can see the main parameters of a microstrip line with 0.8 mm conductor width, 1.6 mm insulate height and 400 KHz frequency. In this conditions, the capacitance of the track will be C_o = 0.6362 pF/cm and if our track length is about 7 cm, then C_b = 4.45 pF.

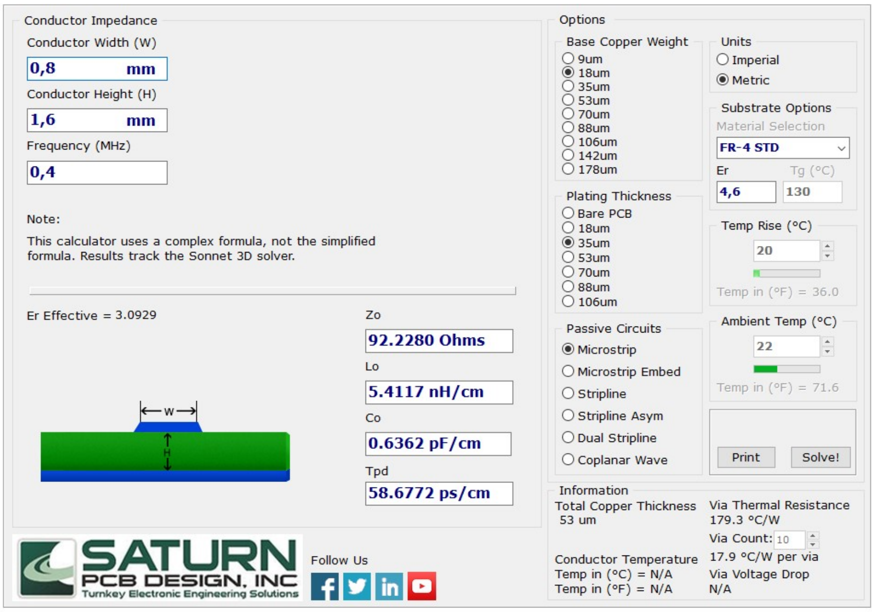


Figure 3.11 – *Track characteristics with Saturn PCB [27]*

Therefore, from Equation 3.1.1 and Equation 3.1.2 the limit values for pullup resistors are:

- $R_{min} = 1.53\ K\Omega$
- $R_{max} = 67.42\ K\Omega$

In our case we will use 4.7 KΩ pullup resistors. In Figure 3.12 is represented the schematic of the display with I2C buses pulled up with a couple of resistors and a 100 nF bypass capacitor.

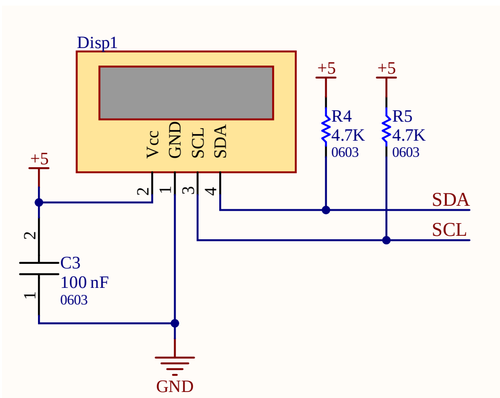


Figure 3.12 – *Display connection schematic*

3.1.5 Micro-Controller Unity

The Figure 3.13 shows the connection of the PCB MCU, in this case an Arduino UNO, with the different components of the PCB. We have included three LEDs and three switches, one switch is for reset the Arduino while the other LEDs and switches can be used for different purposes.

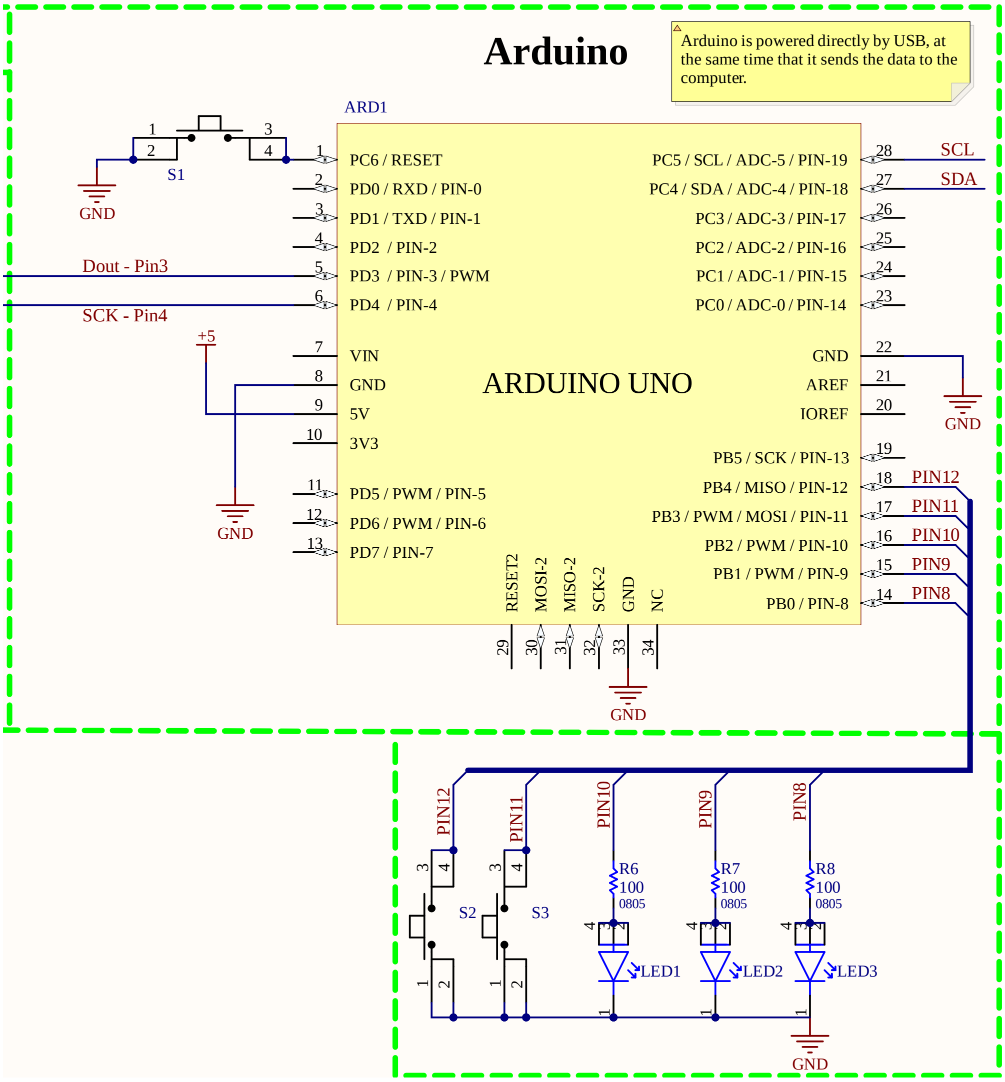


Figure 3.13 – *Arduino schematic connection*

3.2 Firmware design

This section will explain the main program of the Arduino Uno, whose basic function is read from the ADC and send measurement by serial port to the PC. Figure 3.17 represents the flowchart of the Firmware.

It begins when the Arduino is switched on, firstly the different used pins are assigned as input/output as required. Then, the micro-controller starts serial communication, initiates the HX711 module and keeps waiting until receiving the measure configuration selected by the user from the PC.

When the Arduino receives all the parameters needed from the computer and receives the start command, it begins measuring data with a rate of 10 SPS and send them by serial port.

3

Before sending data to the PC we should package them to avoid losses. In Figure 3.14 is shown the package format. Every data packet has 13 Bytes length, the first 4 Bytes represent the header, the following 9 Bytes are the body of the packet, the first Byte in the body bellongs to the gain then 4 Bytes for measured data and another 4 contain the time when the data was acquired.

Moreover we need another package format to contain other type of messages and commands. This messages are shorter, 5 Bytes only, Figure 3.15 shows more details.

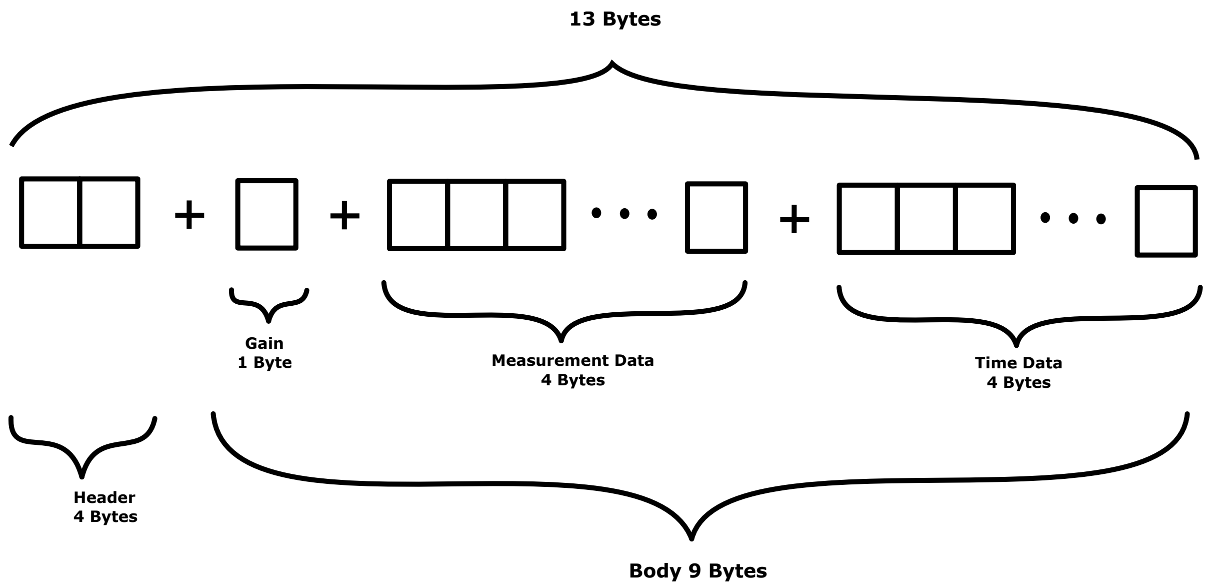


Figure 3.14 – *Data Package*

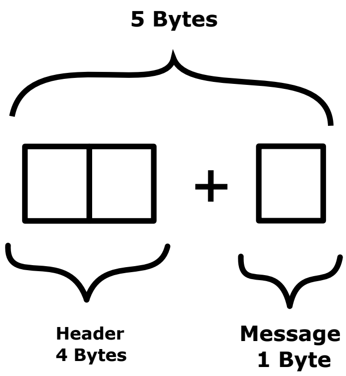


Figure 3.15 – *Message Package*

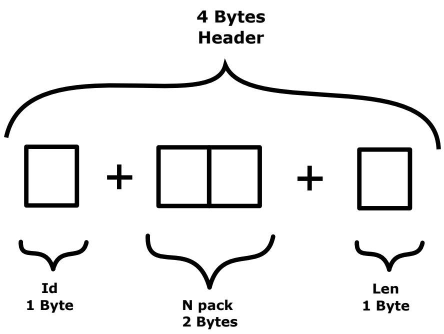


Figure 3.16 – *Packages Header*

Figure 3.16 identify the different components of each packet header. The first Byte is for the Id it can take some diferent values that are recopiled in Table 3.1.

Id	Utility
0x0 0	Info
0x0 1	Channel A data
0x0 2	Channel B data
0x0 3	Setting parameters
0x0 4	Command

Table 3.1 – *Id combinations*

Info Mess	Utility
0x0 0	Received pack (ACK)
0x0 1	All OK and ready
0x0 2	OLED disconnected
0x0 3	OLED connected

Table 3.2 – *Body info message*

Commands	Utility
0x0 0	Stop measuring
0x0 1	Start measuring
0x0 2	Restart data
0x0 3	Tare

Table 3.3 – *Command messages*

Set Parameter	Utility
0x0 0	Single channel mode
0x0 1	Dual channel mode
0x0 2	Auto-Gain selection ON
0x0 3	Auto-Gain selection OFF

Table 3.4 – *Set parameter messages*

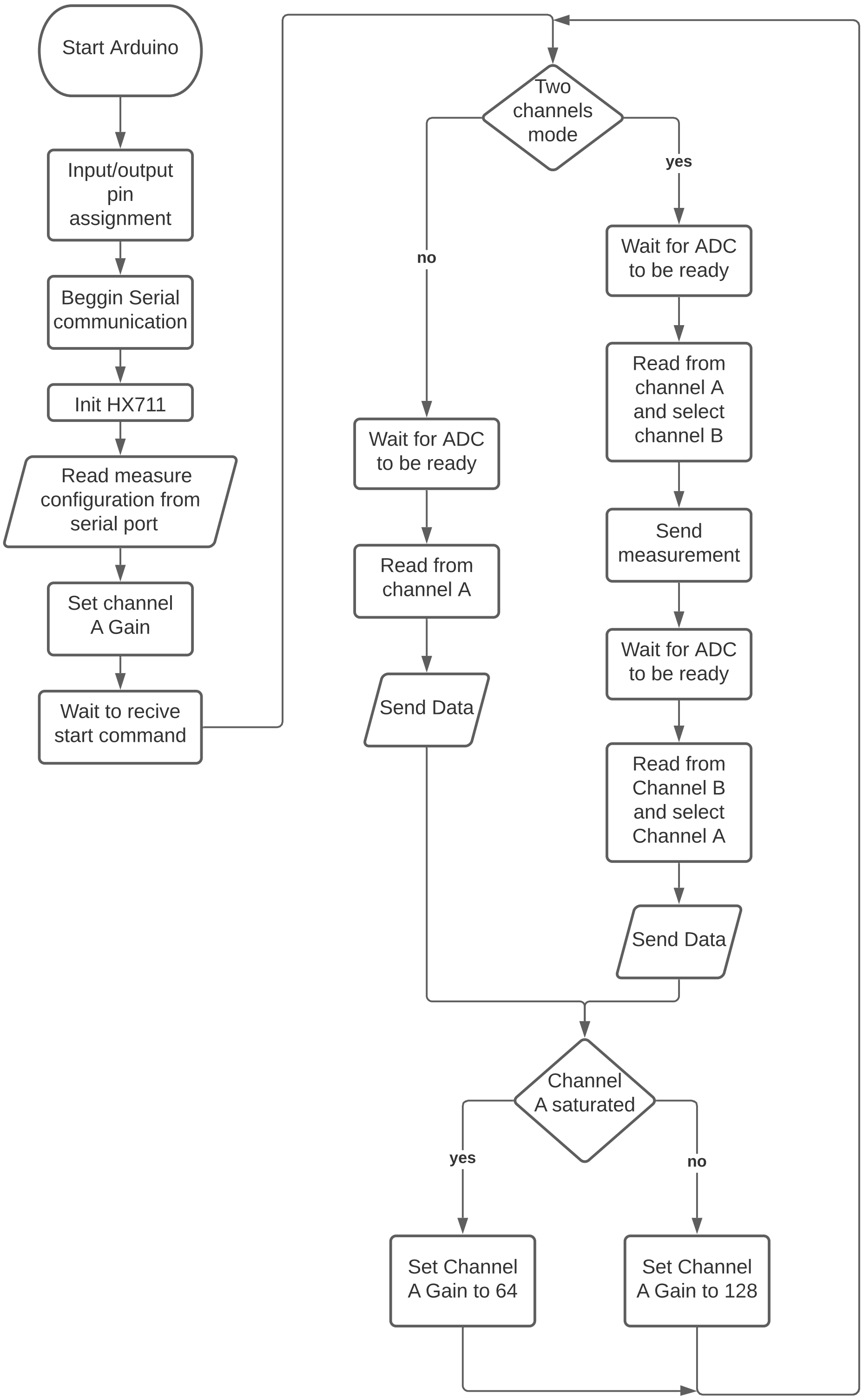


Figure 3.17 – *Firmware Flowchart*

3

3.3 Software design

In order to improve user interaction with the device, we will implement a graphical interface (GUI). For that purpose we will use Qt designer which is a very useful tool for make graphic interfaces using python as programming language. Figure 3.19 shows the Qt development interface with the elements used for the GUI.

Figure 3.18 – *Qt designer logo [40]*

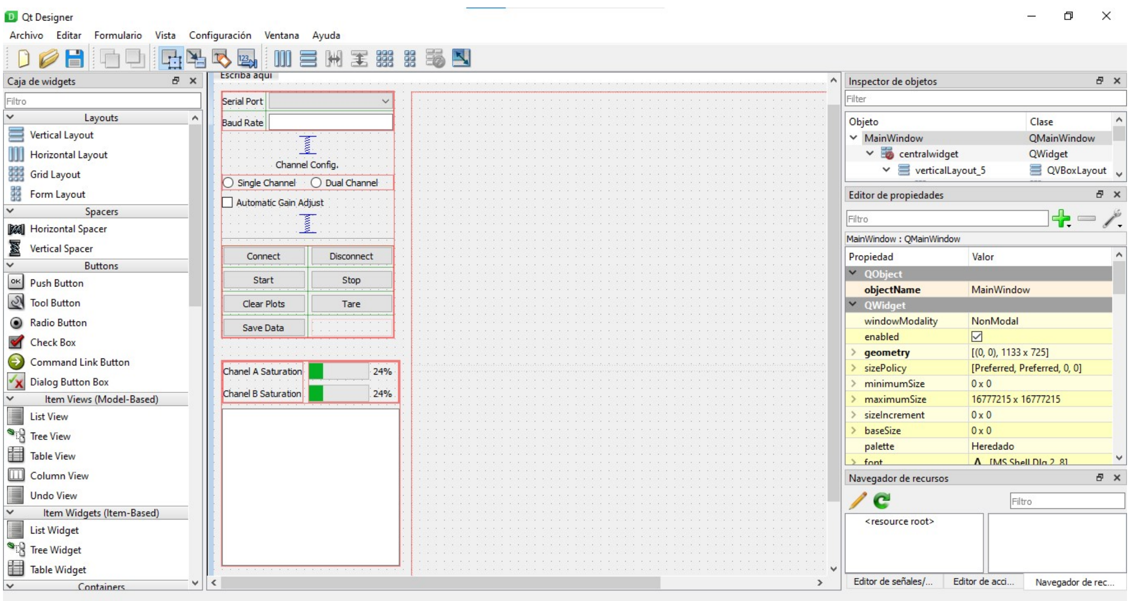


Figure 3.19 – *Qt designer development interface*

Finally, we can import this interface into a python program, then we can compile it an get something like Figure 3.20.

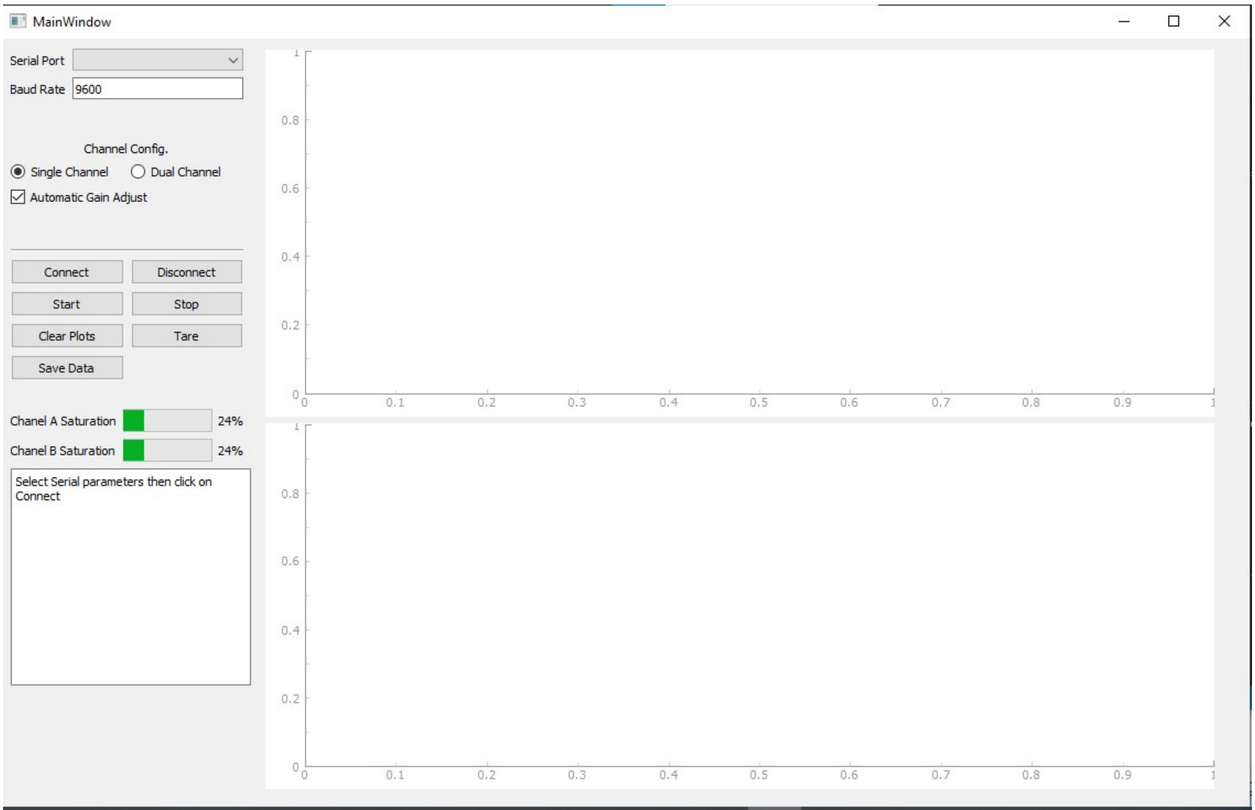


Figure 3.20 – *GUI developed with Qt*

3

Chapter 4

System Manufacture

In this chapter we are going to explain the whole manufacturing process of the different elements of the project, starting with the PCB and then the case to protect it from any damage. For that purpose we have used some CAM and CAD programs and some CNC machines, such as a milling machine or a 3D Printer.

4.1 PCB Manufacture Process

All our PCBs are manufactured in the lab using the mechanical etching technology, for that purpose we have available a PCB milling machine, we can see it in Figure 4.2. One of the advantages of using this technology is that you do not need toxic chemicals. However, we cannot make plated drills on the PCB and we do not have the needed tools to make the solder surface mask. These facts are not really a problem if we consider some parameters at design time.

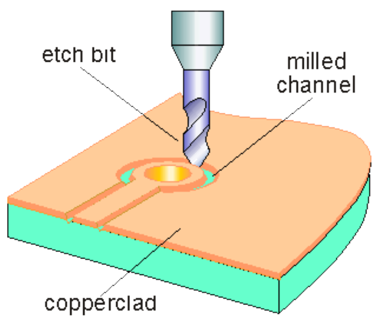


Figure 4.1 – *Mechanical Etching [28]*

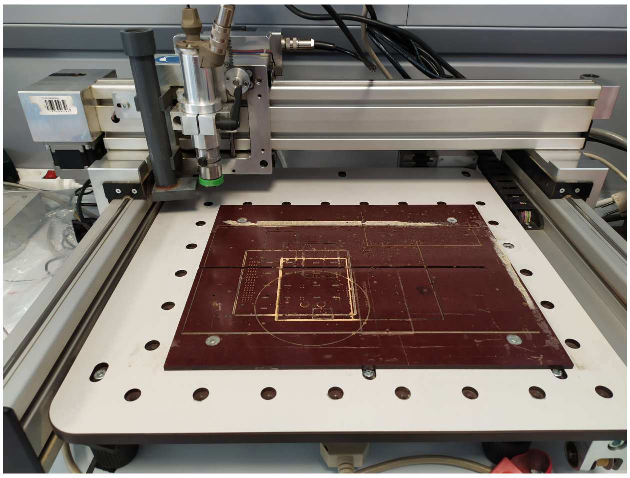


Figure 4.2 – *Milling Machine*

4.1.1 PCB Design

Using Altium Designer® 21 we should export the schematics, explained in Section 3.1, to a PCB design. After placing the components and routing the tracks, we obtain the layers shown in Figure 4.3. As it is shown the PCB has dimensions of 66 x 92 mm.

In Figure 4.4 we can see each layer separately, the *Top Layer* on the right, *Bottom Layer* in the middle and the *Mechanical Layers* at the left, this mechanical layers

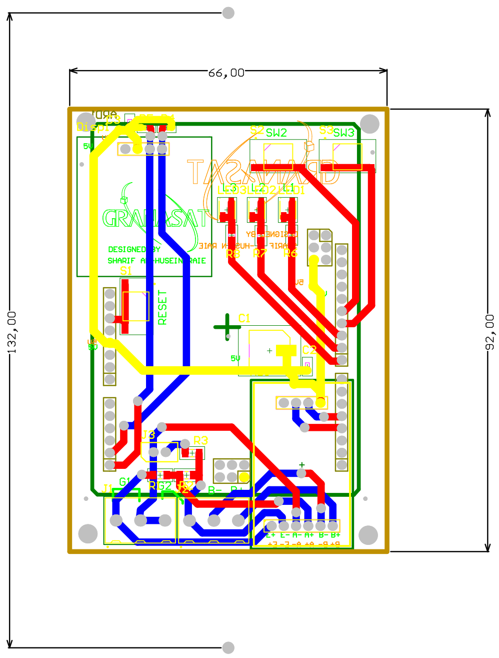


Figure 4.3 – *PCB Multi-Layer View*

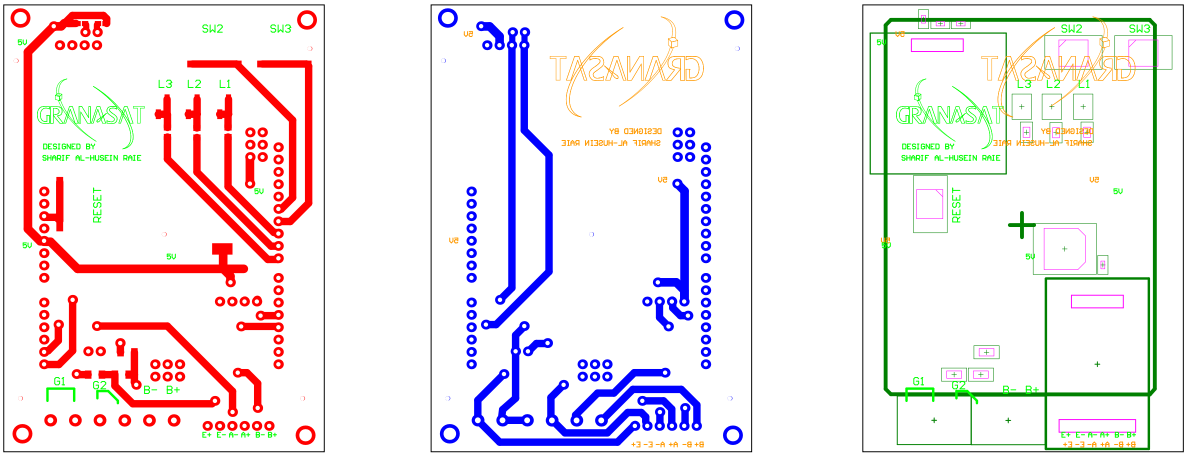


Figure 4.4 – *Top, Bottom and Mechannical Layers without ground plane*

However using this milling technology we should not retire the ground plane because it would take a long time for that. Then our PCB will look like these in Figure 4.5. Having a ground plane provides us with some advantages, such as dissipating heat from the components or reducing the electromagnetic radiation that the device could generate.

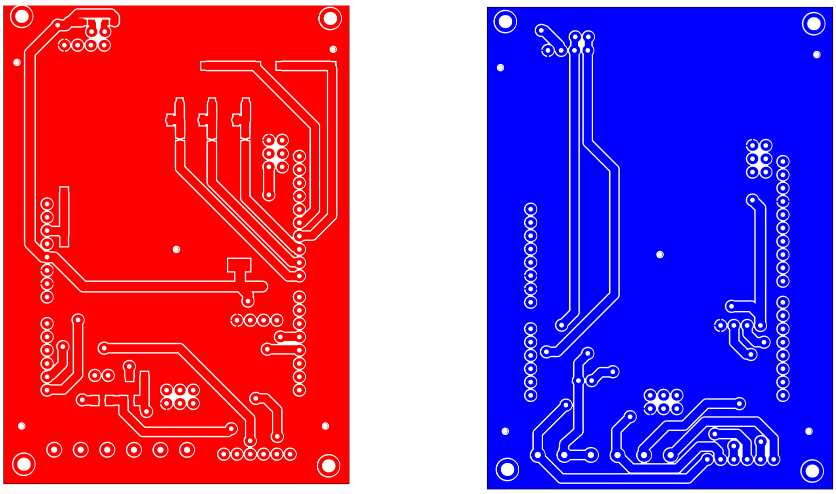


Figure 4.5 – *Top and Bottom Layers with ground plane*

Figure 4.6 shows all the drilling on the PCB and its diameters. The two holes that are seen outside the board, one above and one below, with a diameter of 2.5 mm, they are used as fiducial holes at manufacture time taking them as reference when turning the copper board.

Drill Table

Symbol	Count	Hole Size	Plated	Hole Tolerance
□	82	0.90mm	Plated	
○	6	1.20mm	Plated	
▽	6	2.50mm	Plated	
	94 Total			

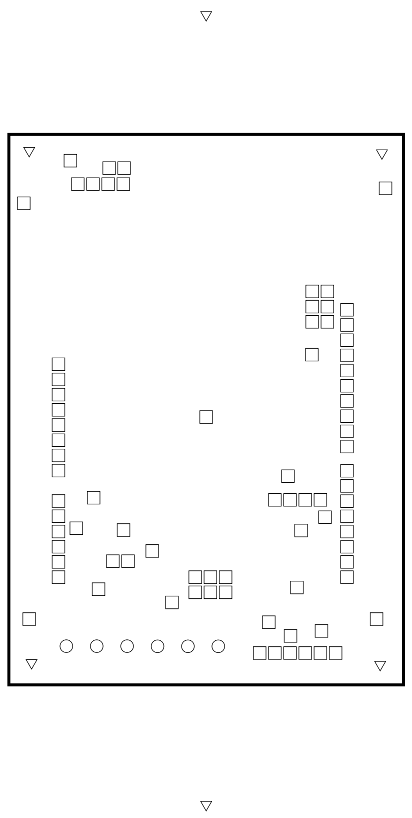


Figure 4.6 – *Drill Drawing View*

The copper boards used for manufacture the PCBs has a thickness of 1.6 mm, with 70 μm of copper on each side and a 1.46 mm of dielectric (FR4) between them. Figure 4.7 shows more details about *PCB Layer Stack*.

Figure 4.7 – *Layer Stack Legend*

Finally, Figure 4.8 and Figure 4.9 show some realistic views of the final PCB design.

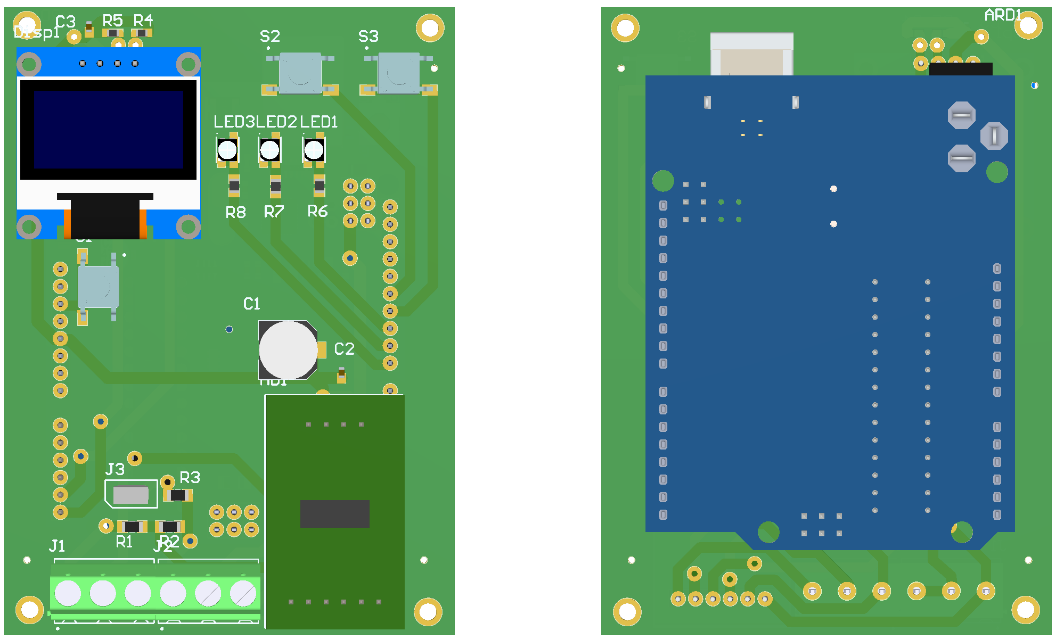


Figure 4.8 – *Top and Bottom realistic view*

4

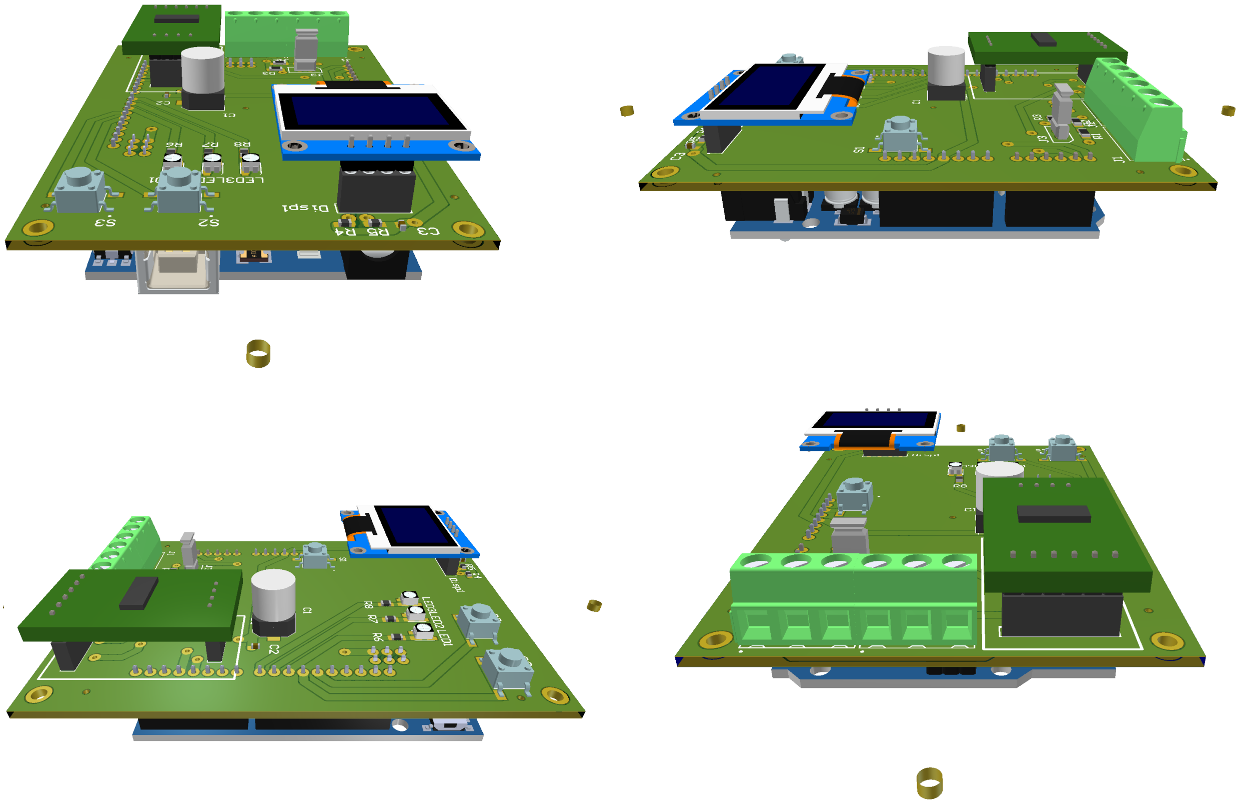


Figure 4.9 – *3D Realistic View*

4.1.2 Generating Manufacture Files

From the PCB design in Altium Designer® 21, the firs thing to do is export the Gerber files, we need to export the *top* and *bottom* layers in addition to a couple of mechanical layers used to include some text on the PCB.

For the next step we need to import and process Gerber data into useful assembly data, for that we use CircuitCAM program which is a CAM software for PCB and other related technologies from China. It is supporting technologies for both prototyping and line production.

In CircuitCAM program we can import Gerber files generated previously getting something similar to what is shown in Figure 4.10. After that we should calculate the track insulate, which are the paths where the cutter will pass. Then we can export in HP-GL which is the only format permitted by the machine program, *Route Pro 2000*, in Figure 4.11 we can see the main window of the program.

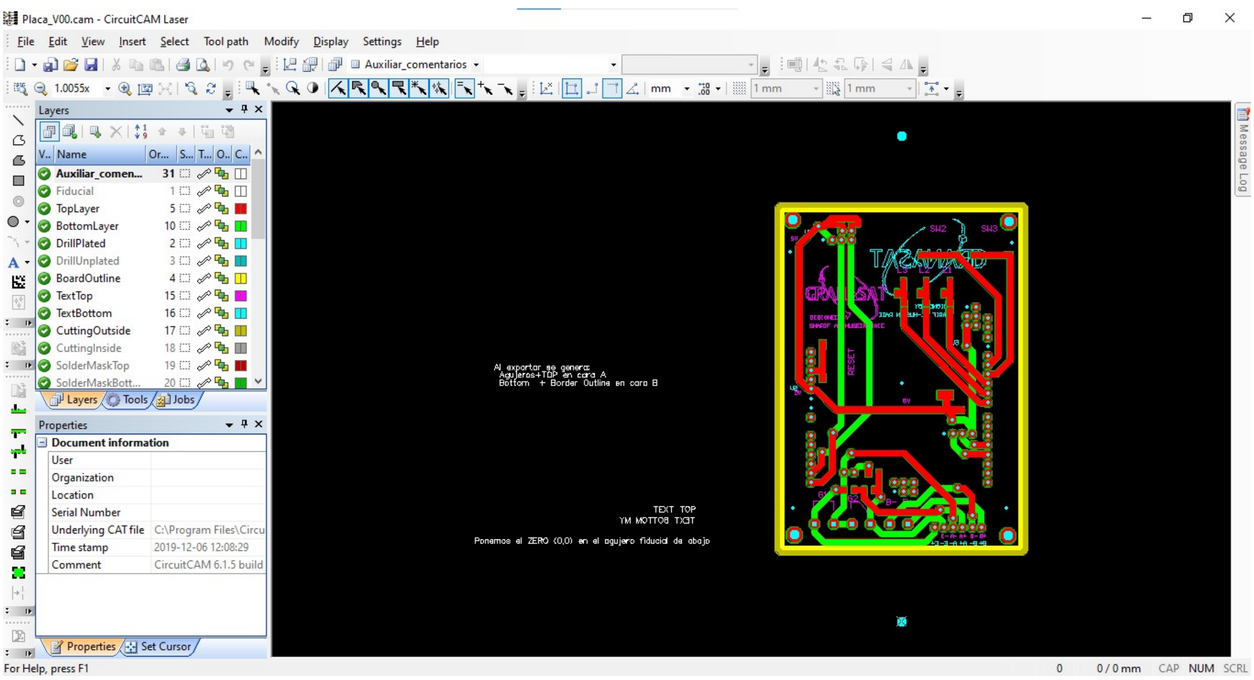


Figure 4.10 – *PCB layer in CCAM*

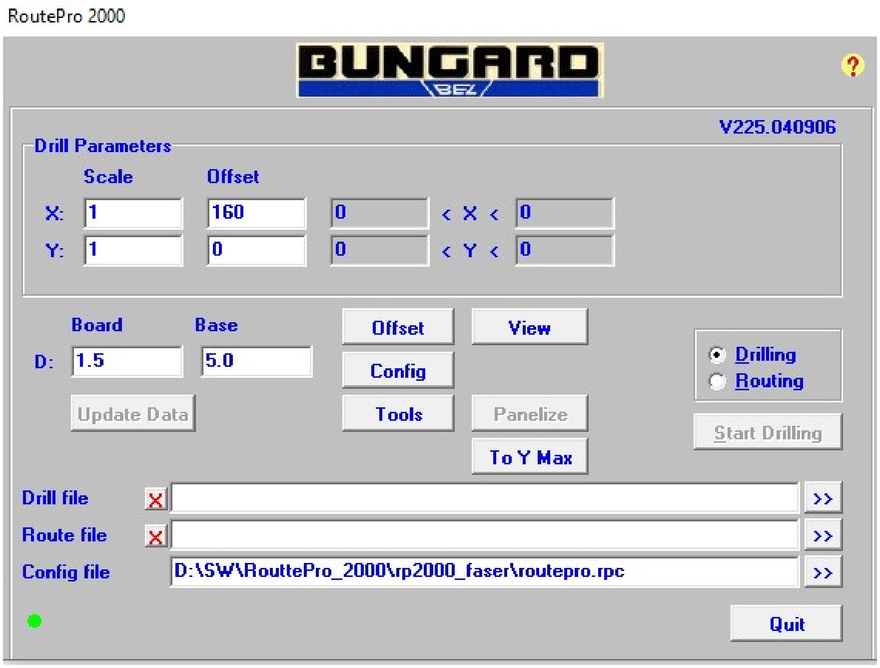


Figure 4.11 – *Route Pro 2000 Main Window*

4.1.3 Using Milling Machine

The whole fabrication process is explained in this video (Click here).

In the following figures there are some photos of the manufacture process. In Figure 4.12 there is a board with the drills done for the via and through hole component pads, while in Figure 4.13 PCB tracks are being milled.

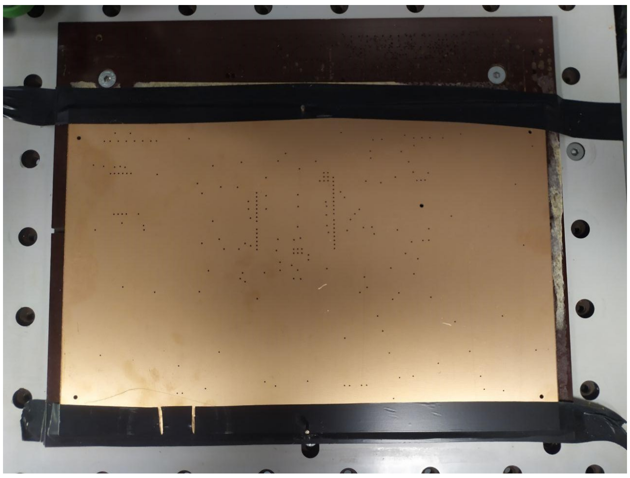


Figure 4.12 – *Drilled Copper Board*

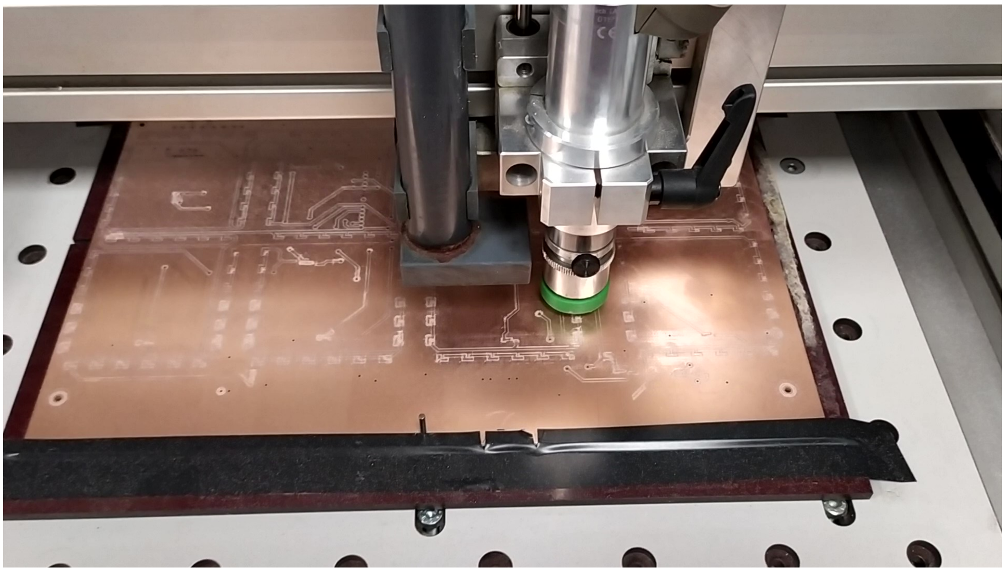


Figure 4.13 – *Track Milling*

Finally, in Figure 4.14 we can see the result of the PCB manufacturing process, from top and bottom side while in Figure 4.15, the PCB with all components soldered.

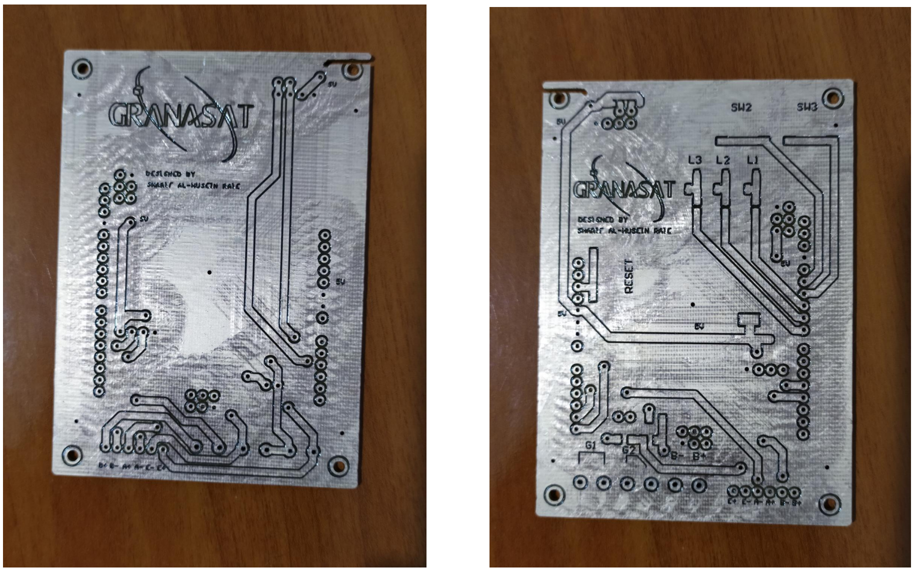


Figure 4.14 – *PCB manufactured, TOP and BOTTOM sides*

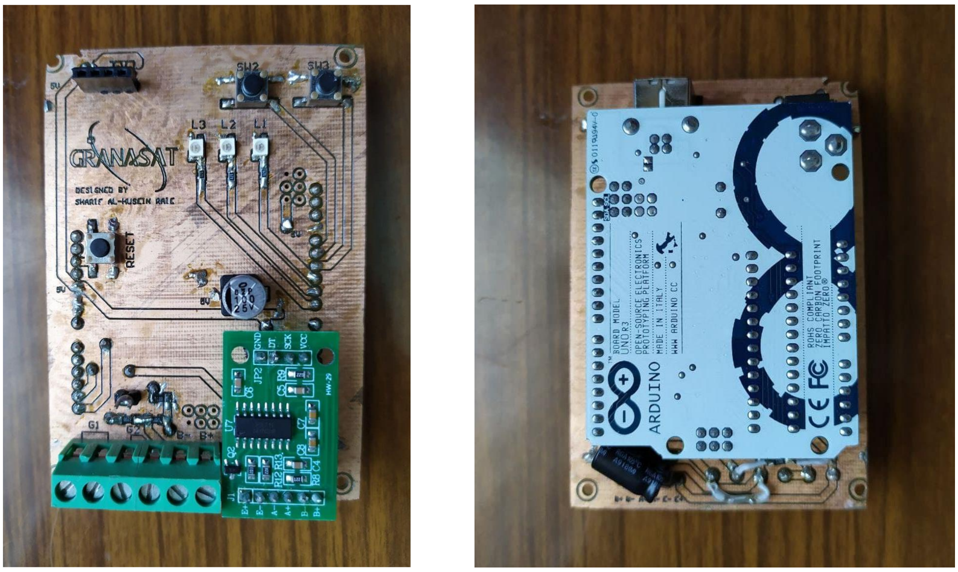


Figure 4.15 – *Soldered components on PCB*

4.2 PCB Protection Case

For design this case we have used SolidWorks® program. Figure 4.16 shows different views of the first version designed housing for the PCB. This first version is made up of two pieces, the board is inserted inside them and they fit together. In addition, on the top side, there are three buttons that allows us to push the switches on the PCB. Nevertheless, it does not have a window to view the display because at first time we thought it would be unnecessary. But, later we decided to use it for display some data and debug some errors in the firmware, then a second version was designed.

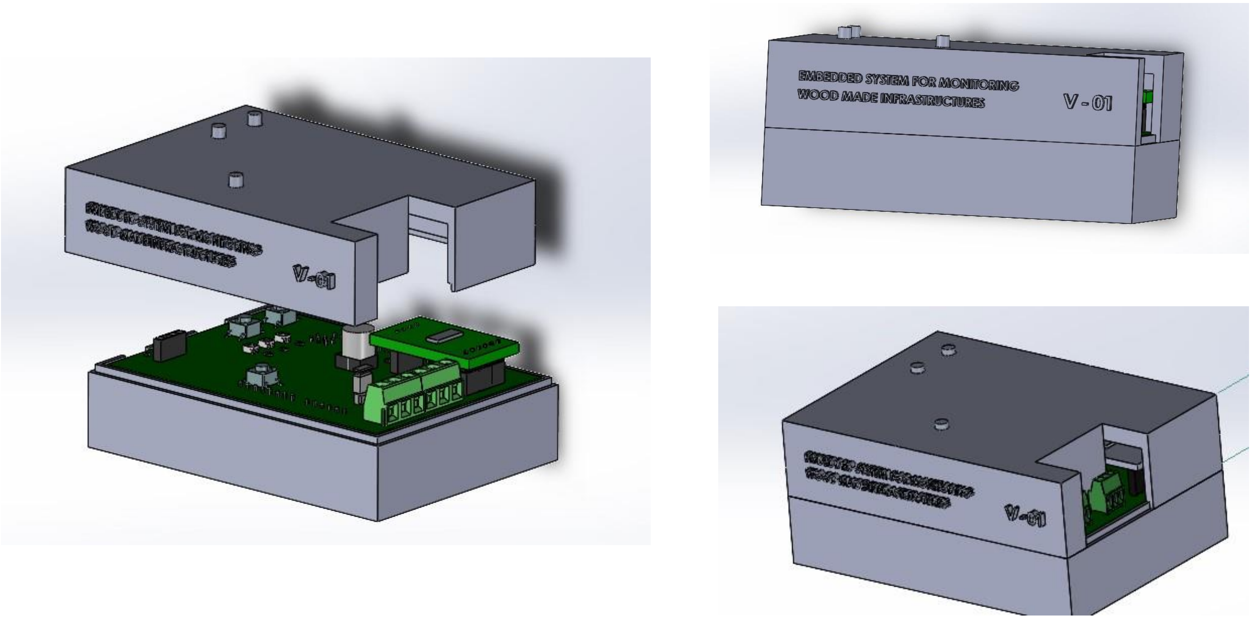


Figure 4.16 – *PCB Housing V01*

The second version is very similar to the first one with the only difference of having a window to see the display. Figure 4.17 shows the second version of the housing.

To manufacture all this parts we have used a FDM (Fused Deposition Modeling) 3D printer that we can see in Figure 4.18. While in Figure 4.19 and Figure 4.20 there are both version of the housing assembled with the PCB inside.

Figure 4.17 – *PCB Housing V02*

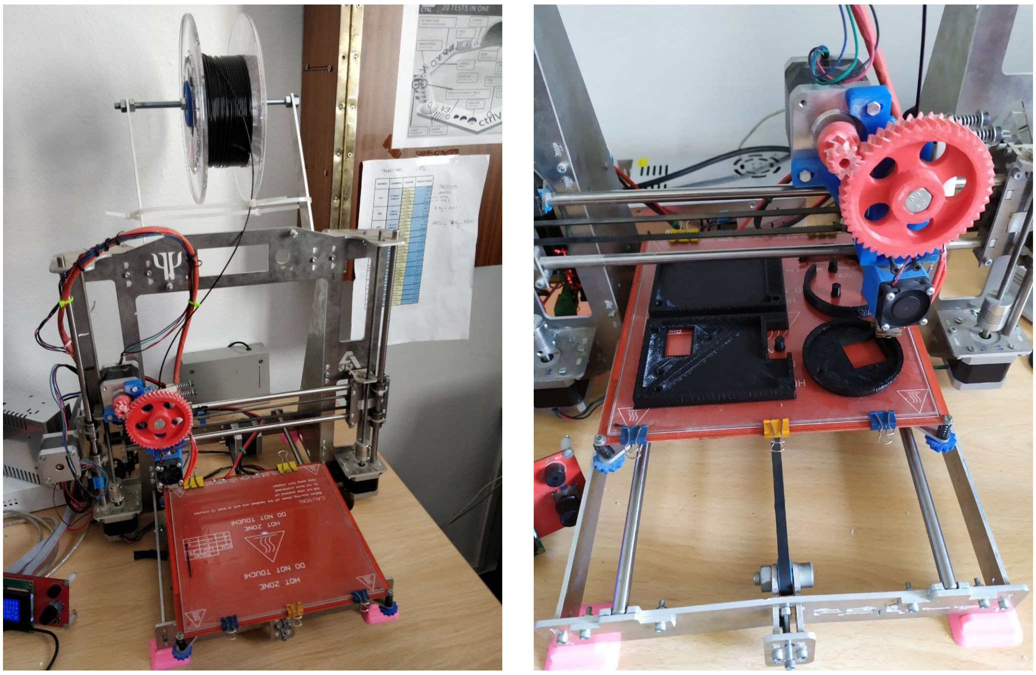


Figure 4.18 – *FDM 3D Printer*

(a) *One side*

(b) *Other side*

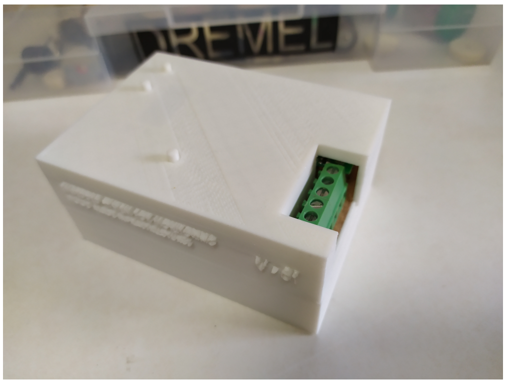


Figure 4.19 – *Assembled housing V01*

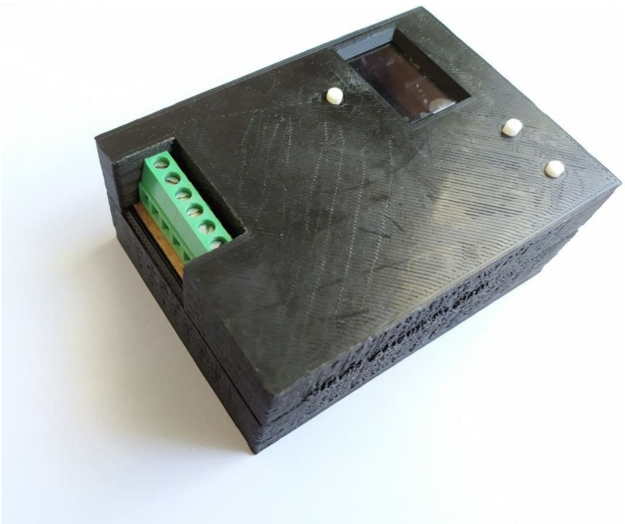


Figure 4.20 – *Assembled housing V02*

4.3 Beautifying sticker

In order to cover some imperfections of the 3D printing in addition to showing text to identify the buttons and connectors, we will design and print a sticker.

For that purpose we need some additional tools, for cutting the sticker we need a **cutting plotter**, the Silhouette Cameo 4 which has his own program, Silhouette Studio for designing and control the machine at the same time, this will make the process easier.

Figure 4.21 – *Silhouette Studio Logo*

In figure Figure 4.22 we can see the Silhouette Studio interface with our sticker design.

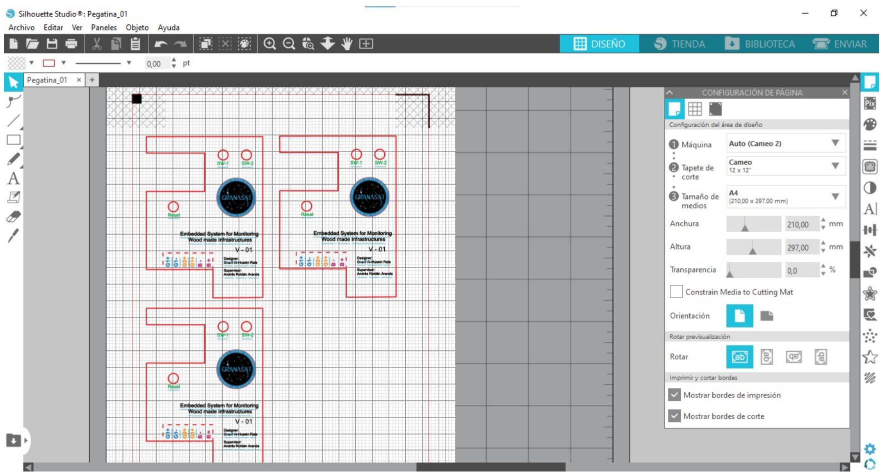


Figure 4.22 – *Silhouette Studio Interface*

After the design is done we should print it with a normal printer over an adhesive paper. Then we can cut it using the cutting plotter, Figure 4.25. It is true that it can be cut using normal scissors but with this plotter the finish is cleaner, especially in the area of the buttons where it is difficult to access with simple scissors.

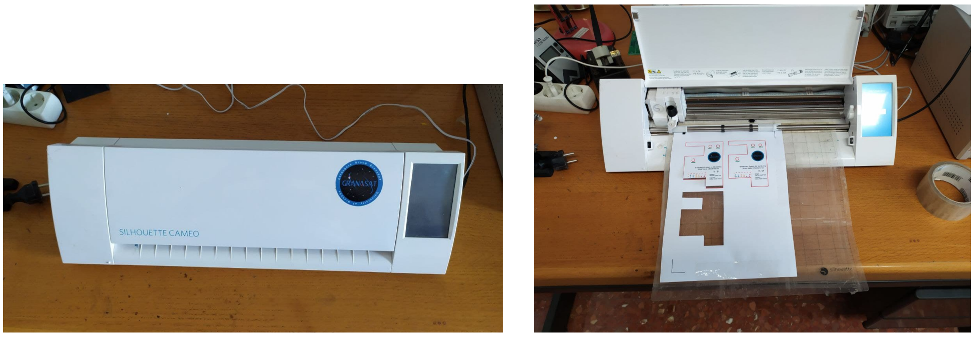


Figure 4.23 – *Cutting Plotter*

Finally, in Figure 4.24 we can see the sticker resulted and in Figure 4.25 the sticker glued over the device.

Figure 4.24 – *Sticker*

Figure 4.25 – *Sticker glued on the device*

4.4 HW/SW Manufacturing Tools

In this section, as a summary, the different tools used throughout the manufacturing process will be shown, both the design programs and the machines that intervened in the process.

In the Figure 4.26 there are some of the useful tools used in the manufacture process, like a vertical drill, a solder station, a saw or an electric sandpaper.

Figure 4.26 – *Some used tools*

Chapter 5

Validation

This fifth chapter is used to validate the solutions proposed in this Bachelor's Thesis and check the accomplishment of the goals of the project.

To test the complete product, we will use poplar wood beams with glued strain gauges, then we will subject them to destructive or non-destructive forces. For that we will use a press and then we will compare the results with a professional equipment measurements. All this will be done in the materials laboratory of the Building College.

5.1 Manufacture of Wooden Beams

Before doing any test we should prepare the needed wood beams. For this we will use Glulam technology, shown in Figure 5.1, that is a product formed by gluing some planks, with thicknesses between 6 and 45 mm. This product can eliminate defects such as knots, allow efficient use of forest and have more homogeneous characteristics.

In Figure 5.2 there are some beams glued and they are keep to dry.

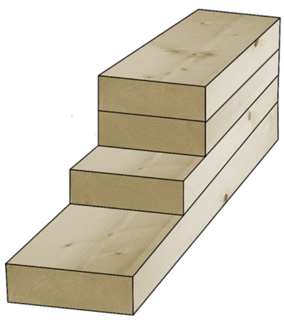


Figure 5.1 – *Glued Laminated Timber [13]*

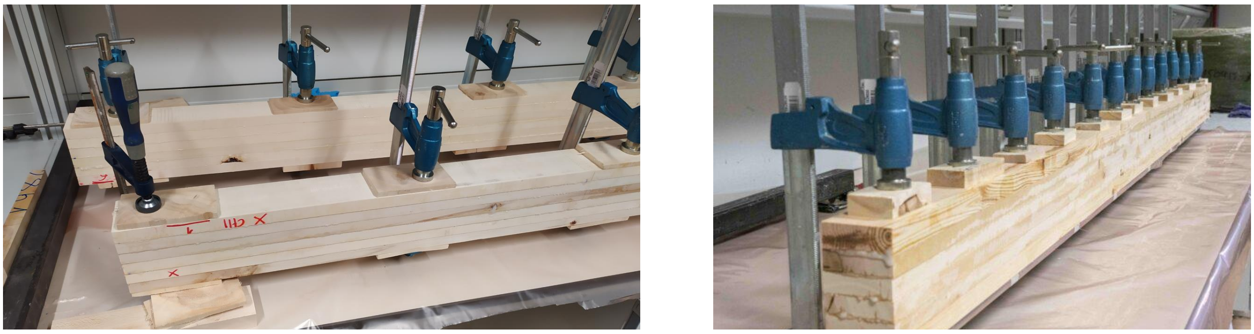


Figure 5.2 – *Glulam Beams manufacture*

If we want longer beams, we can join two wooden planks to get a longer one but, any joint that butts end grain to end grain will be weak because we are gluing wood fibers at their porous ends instead of along their sides. Then a solution for that is use finger joints. In Figure 5.3 we can see the machine used for make the finger joint cuts. While in Figure 5.4 there are some wooden boards ready to join.

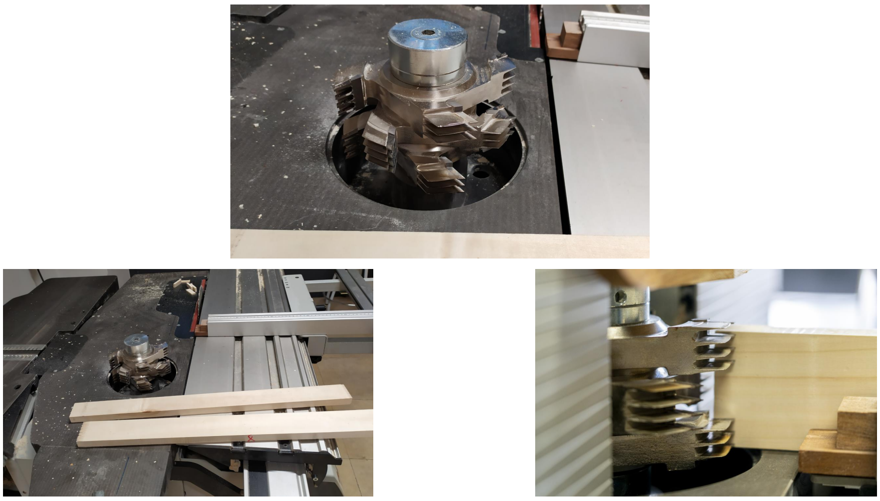


Figure 5.3 – *Finger joints timber Machine*

Figure 5.4 – *Finger joints timber*

5.1.1 Strain Gauges Preparation

After the beams dry we can place the strain gauges in the center of each beam and to ensure that all the tension subjected to the beam will be transmitted to the strain gauge, we must attach them with a special glue, like the one shown in the Figure 5.5.

Figure 5.5 – *Special Glue for Gauges*

To stick the gauge, a little glue is applied to the surface, the gauge is placed on top then we apply pressure on it letting it to dry for a few minutes. All this process is summarized with the Figure 5.6.

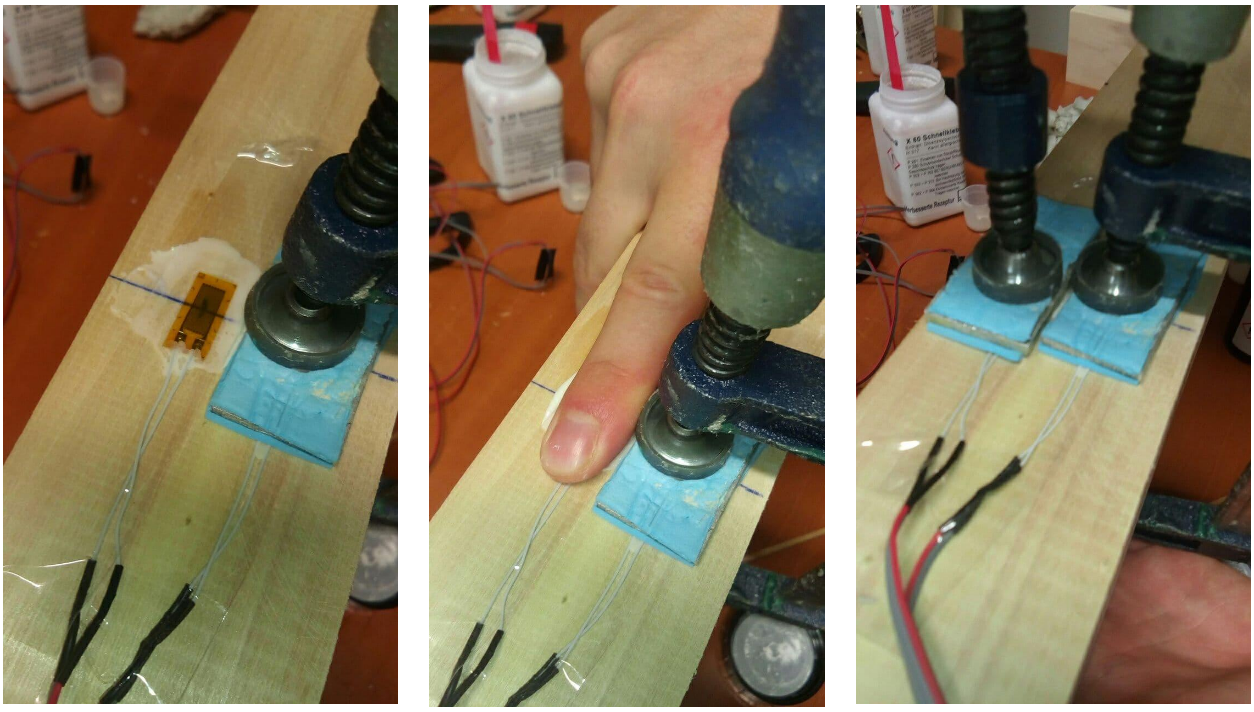


Figure 5.6 – *Gluing Strain Gauges to Beams*

5.2 First Test

In this first test we place two gauges, one next to the other and measure at the same time from our device and from a professional device from the manufacturer HBM. Our device was measuring from channel A with a gain of 128.

This first test was non-destructive one, its purpose is apply strength to the beam making it flex but without reaching the breaking point. After that we retired the strength applied previously. To apply strength to the structure in a controlled way we used a special press which we can see it in Figure 5.7.

Figure 5.7 – *Press*

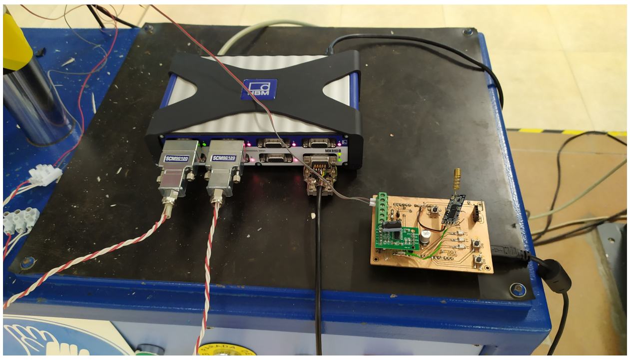


Figure 5.8 – *HBM and our device*

Figure 5.9 shows the data acquired in this first test. We can see that the curve of the graphs of our device follow quite well those of the HBM equipment. However, it may need calibration, although with this test this fact cannot be assured because it was done on an anisotropic material like the wood, this means that the performance of the material will change from one area to another.

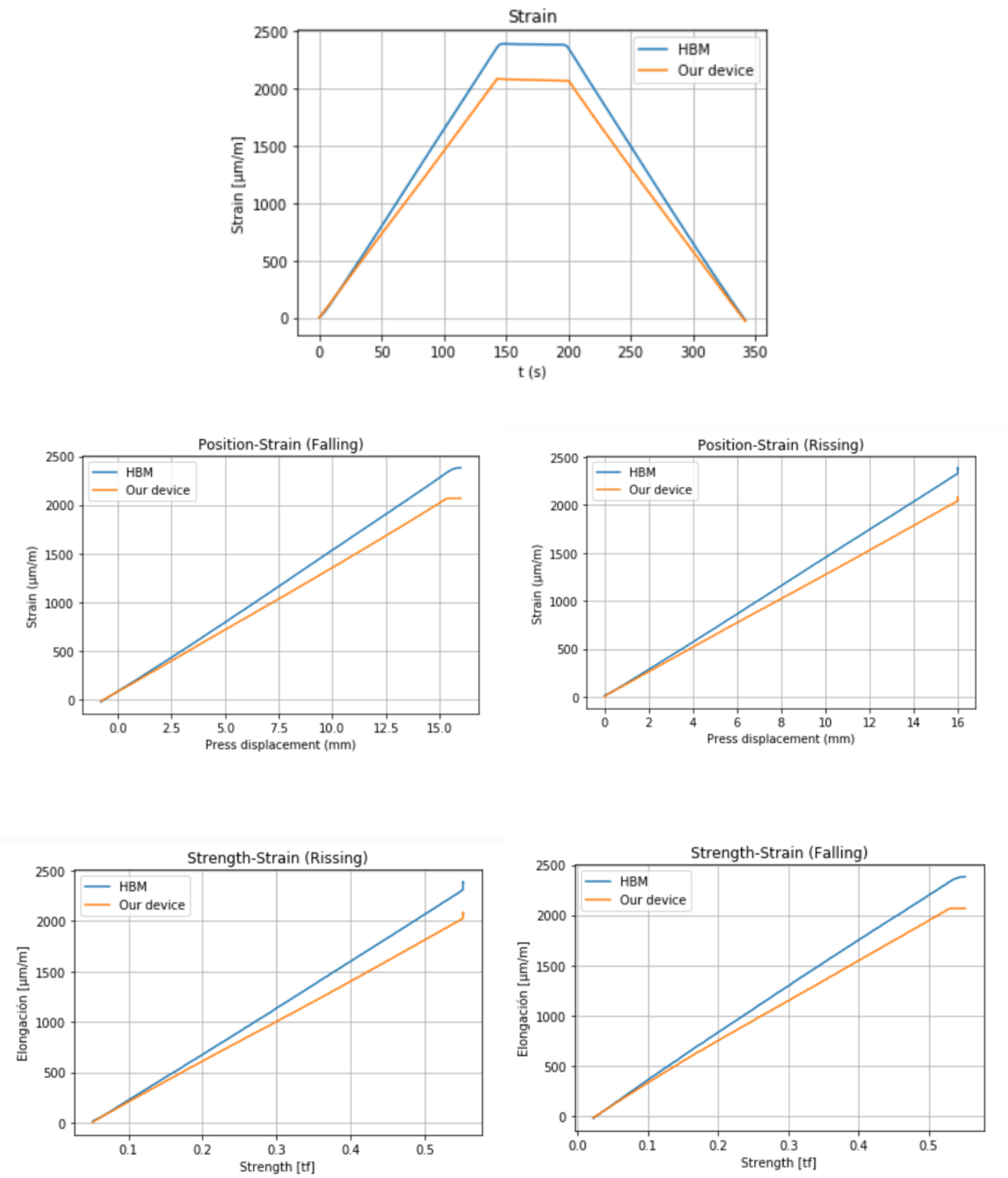


Figure 5.9 – *Test 1 Non-Destructive*

5.3 Second Test

The purpose of this test is compare the performance of both channel of the ADC. Kipping in mind that each channel have different gain, channel A have a selectable gain of 128 or 64, while channel B is fixed to 32. Therefore each channel will have a different offset and then each one should be calibrated separately.

Figure 5.10 shows the measurements result, as we said previously the gap between both channel graphics is clearly appreciated.

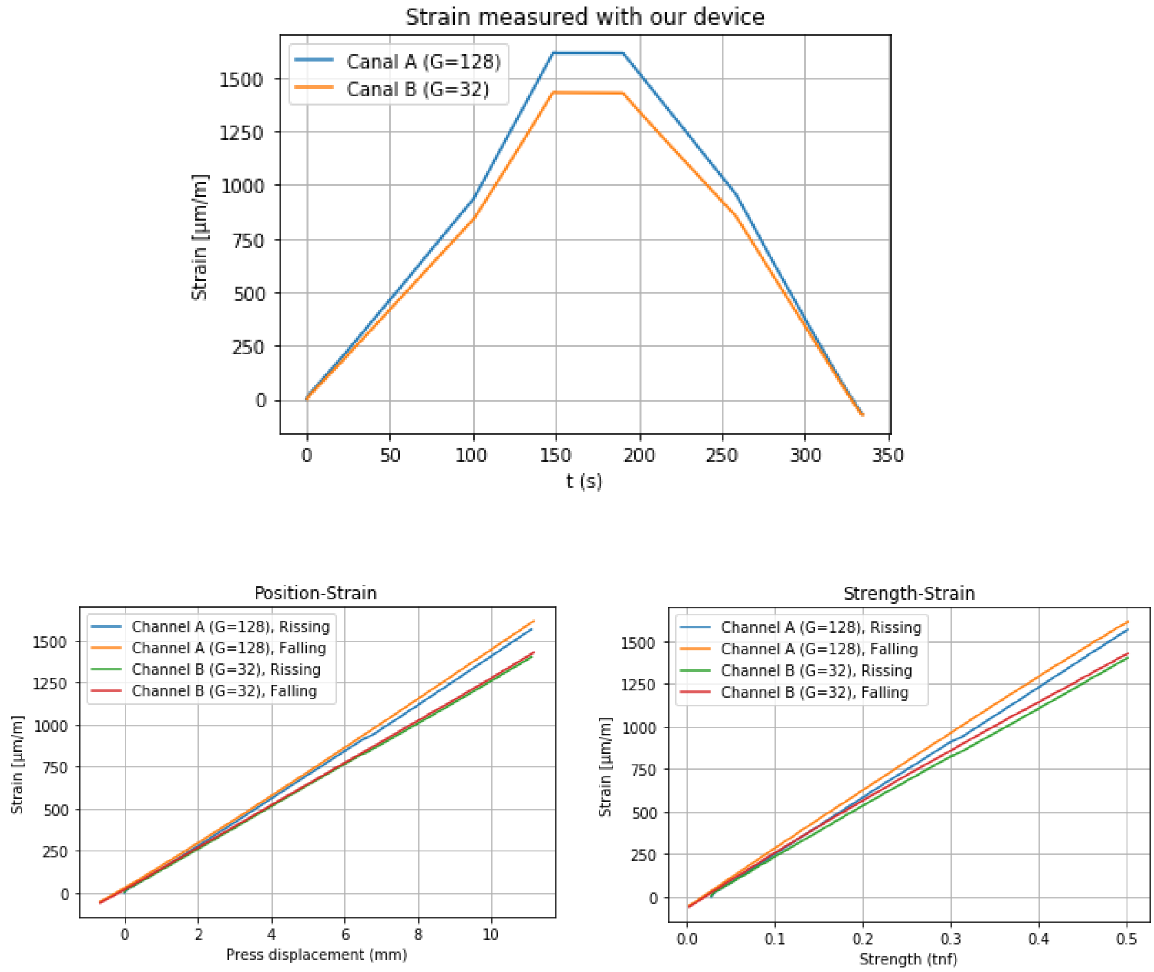


Figure 5.10 – *Test 2, both channels measurement*

5.4 Third Test, break test

In this test we get the poplar beam to the limit exceeding its breaking limits and compare results with HBM device.

We can see in Figure 5.11 the press displacement reached to 35 mm to be able to reach the breaking limit, in previous tests the displacement barely exceeded 15 mm and the process toke 250 s, around 4 min.

In this case, as in the previous ones, there is a small gap between the different graphs.

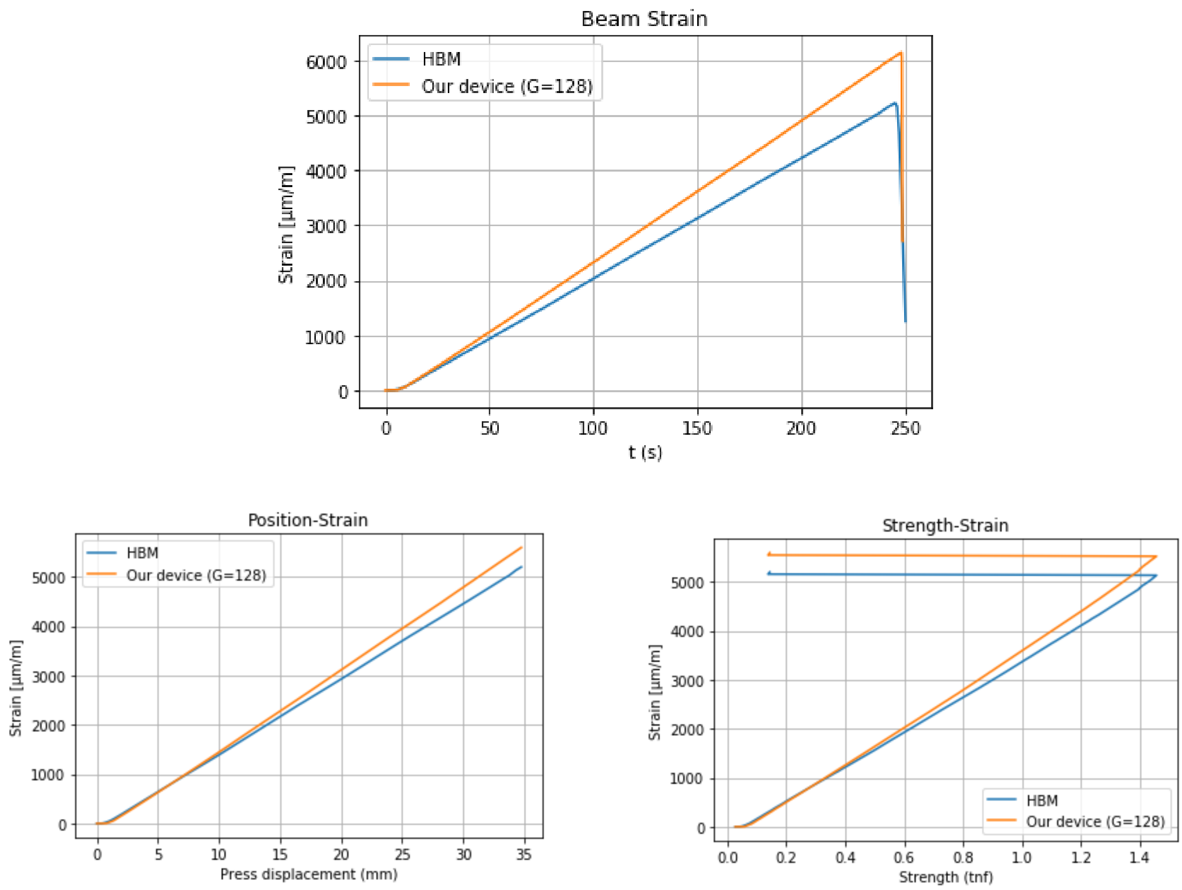


Figure 5.11 – *Test 3, break test*

5.5 Fourth Test, break test

This one is similar to the previous, Section 5.4 and we make it to get more data measurements to characterize the device properly.

However, in this test, something different happened, the ADC of our device get saturated before the beam broke. The channel was configured with gain 128, therefore if we configure it with gain 64 we will be safe from saturation, in return we would be losing signal sensitivity.

Then the optimal solution will be select the maximum gain and change it when the ADC gets close to saturation.

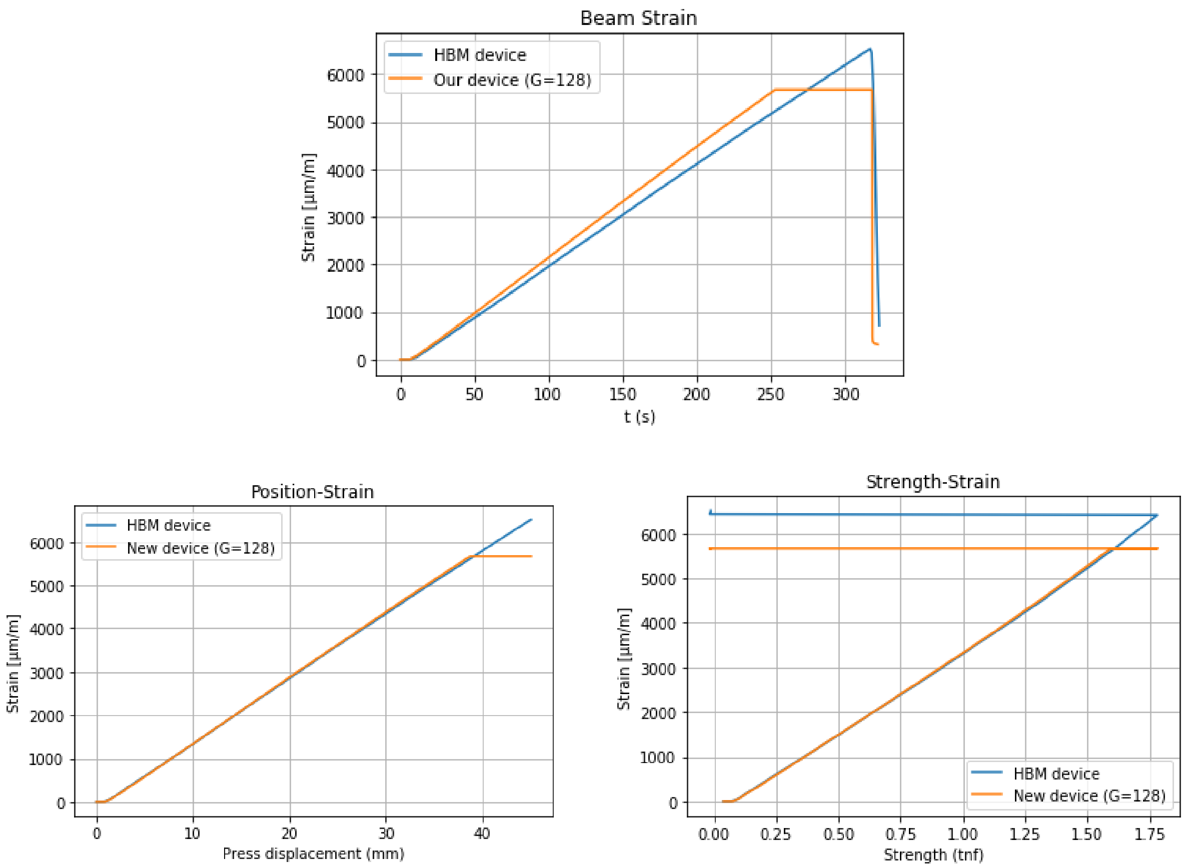


Figure 5.12 – *Test 4, break test*

5.6 Fifth Test

After doing all previous tests with wood beams we conclude that the device should be calibrated, however, for that purpose we should not use anisotropic materials, such as wood. We need a beam made with different material to ensure that the measurements taken at two different points are identical. So we decided to use a steel plate, Figure 5.13.

Firstly we decided begin with channel A and calibrate measurements with both gains. In Figure 5.14 the first section of the curve is performed with the maximum allowed gain and when it is close to saturation the gain is reduced, then we can see the step produced around 200 s.

Figure 5.13 – *Steel plate with strain gauge glued*

After this last test we can proceed with the calibration of the device, however due to time constraints it has not been possible to perform other measurements to check its behavior after calibration.

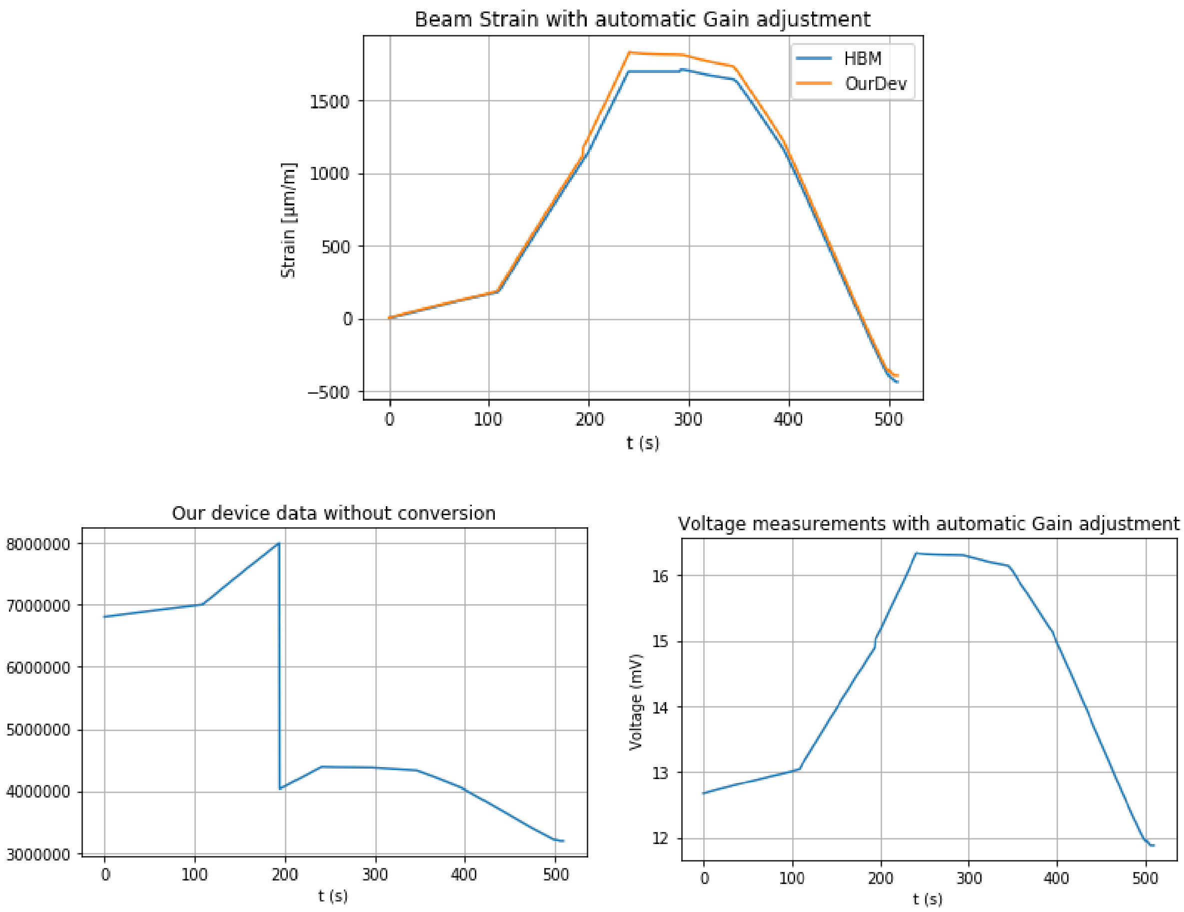


Figure 5.14 – *Test 5, with steel plate*

5

Chapter 6

Conclusions and Future Lines

References

[1] *Instrumrntación amplicada a la ingeniería*. Universidad Politecnica de Madrid, Cátedra de Electrotecnia, 1987.

[2] Pallás Areny, R. *Sensores y Acondicionadores de Señal*. Marcombo-Boixareu, 1994.

[3] Pérez García, M.A. *Instrumentación Electronica*. Thomson, 2004.

[4] Varteresian, J. *Fabricating Printed Circuit Boards*. Newnes, 2002.

[5] Roldán Viloria, J. *Máquinas y herramientas, procesos y cálculos mecánicos*. Paraninfo , 2019.

[6] Planchard, David C. *Engineering graphics with SolidWorks 2018 and video instruction, a step-by-step project based approach*. SDC , 2018.

[7] Castillo Martín A. Chacón Linares Eva Javier de Teresa Galván *Beneficios de la bioeconomía del chopo*. http://dialogosconlasociedad.es/wp-content/uploads/publicaciones/2019_bioeconomia_chopo/6/index.html.

[8] Durillo, J. C. M. Design of a multidisciplinary 1U CubeSat Simulation Platform. Master's thesis, University of Granada, Granada, September 2017.

[9] Sanchez, P. G. Diseño e implementacion de un limitador electroacustico mediante procesadores de audio de altas prestaciones. PhD. Thesis, University of Granada, Granada, September 2019.

[10] Strain Gauge fundamentals. https://www.hbm.com/en/7074/strain-gauge-fundamentals/.

[11] HBM. https://www.hbm.com/en/.

[12] Acústica y Diagnóstico de Materiales y Estructuras. https://adime.ugr.es/.

[13] COMPOP Project. http://compop.ugr.es/index.html.

[14] SelviAgroTec Project. http://compop.ugr.es/selviagrotec/.

[15] PROPOLUS. https://propopulus.eu/en/.

[16] Poplar Cultivation. https://www.antestodoestoeracampo.net/chopo/.

[17] Colegio Oficial De Ingenieros Tecnicos De Telecomunicacion. https://telecos.zone/.

[18] Escuela Tecnica Superior de Ingenierias Informatica y de Telecomunicacion. UGR. https://etsiit.ugr.es/.

[19] 3 terminal connector. https://sumador.com/collections/new-items/products/bornera-kf128-3-pines.

[20] HX711 Datasheet. https://pdf1.alldatasheet.com/datasheet-pdf/view/1132222/AVIA/HX711.html.

[21] ATmega328P datasheet. https://pdf1.alldatasheet.es/datasheet-pdf/view/1132281/ATMEL/ATMEGA328P.html.

[22] Calculation of pullup resistors. https://www.ti.com/lit/an/slva689/slva689.pdf?ts=1625209060941&ref_url=https%253A%252F%252Fwww.google.com%252F.

[23] OLED Display. https://www.mcielectronics.cl/shop/product/display-oled-de-128-x-64-pixeles-controlable-por-i2c-29546.

[24] Arduino UNO. https://www.iberobotics.com/producto/arduino-uno-rev3-original/.

[25] ESP-32. https://www.sigmaelectronica.net/producto/esp-32/.

[26] PIC-18F4550. https://www.bilbaoelectronics.com/46/pic18f4550-dip.

[27] Saturn PCB Design Toolkit. http://www.saturnpcb.com/pcb_toolkit/.

[28] Mechanical Etching. https://www.thinktink.com/stack/volumes/volvi/mechetch.htm.

[29] Espressif. https://www.espressif.com/.

[30] GranaSAT website - Bexus experience. https://granasat.ugr.es/category/rexus/.

[31] How a Solar Cell Works. https://www.acs.org/content/acs/en/education/resources/highschool/chemmatters/past-issues/archive-2013-2014/how-a-solar-cell-works.html.

[32] Keysight. https://www.keysight.com/.

[33] NXP PCF8523. https://www.nxp.com/products/analog/interfaces/ic-bus/ic-real-time-clocks-rtc/real-time-clock-rtc-and-calendar:PCF8523.

[34] SIGLENT SDM3065X. https://www.siglent.eu/siglent-sdm3065x.html.

[35] Siglent spd3303x. Siglent Website. https://www.siglent.eu/spd3303x-e-power-supply.html.

[36] Thorlabs. https://www.thorlabs.com/.

[37] TI ADS1015. http://www.ti.com/lit/ds/symlink/ads1015.pdf.

[38] Ti msp430. http://www.ti.com/microcontrollers/msp430-ultra-low-power-mcus/overview.html.

[39] VxWorks by Wind River. https://www.windriver.com/products/vxworks/.

[40] Wikipedia. https://en.wikipedia.org.

Appendix A

Project Budget

A.1 Materials and hardware

In this section, project cost regarding materials and hardware sections will be detailed. Each one of the different hardware subsections is differentiated. Human resources area also included.

A.1.1 Device Housing

According to the System Manufacture, Chapter 4, the material related with the housing is mainly PLA for 3D printing. The amount necessary of PLA costs **4.50 €**. While the sticker to cover the housing top surface costs **0.75 €**. That makes **5.25 €** without manufacturing costs, detailed in Section A.2.

A.1.2 PCB Costs

The cost of the PCB is breakdown into the electronic components, the copper board and the manufacturing, detailed in Table A.1, as well as design and manufacture costs, explained in Section A.2.

A.1.2.0.1 Bill of Materials

Item	Units	Cost/u. (€)	Cost (€)
HX711 ADC	1	0.73	0.73
Electrolitic cap.	2	0.10	0.20
Ceramic cap.	3	0.012	0.036
Precision Res.	6	0.05	0.30
Resistors SMD	2	0.01	0.02
LED SMD	3	0.09	0.27
OLED display	1	2.00	2.00
Connectors	2	1.91	3.82
Switch	3	0.20	0.60
Male pins	3	0.70	2.10
Female pins	2	0.73	1.46
Arduino UNO	1	22.50	22.50
Copper Board	1	5.75	22.50
Subtotal (before VAT):			**39.79**
Total (VAT included):			**48.15**

Table A.1 – *PCB components*

A.2 Human Resources Cost

In the first place, the testing procedures required the collaboration of the **Material Department** at Building College of the University of Granada. The cost for an hour of machinery and technical assistance was **25 €**.

Additionally, the development of this Master's Thesis has required hiring two people. The first one is a **junior engineer** (10 €/h), hired as a full-time worker during twelve months. Secondly, as Project Supervisor a **senior engineer** is hired (50 €/h), computing 5 hours per week. Then, Human Resources amounts to **31200 €**, as detailed in table A.2.

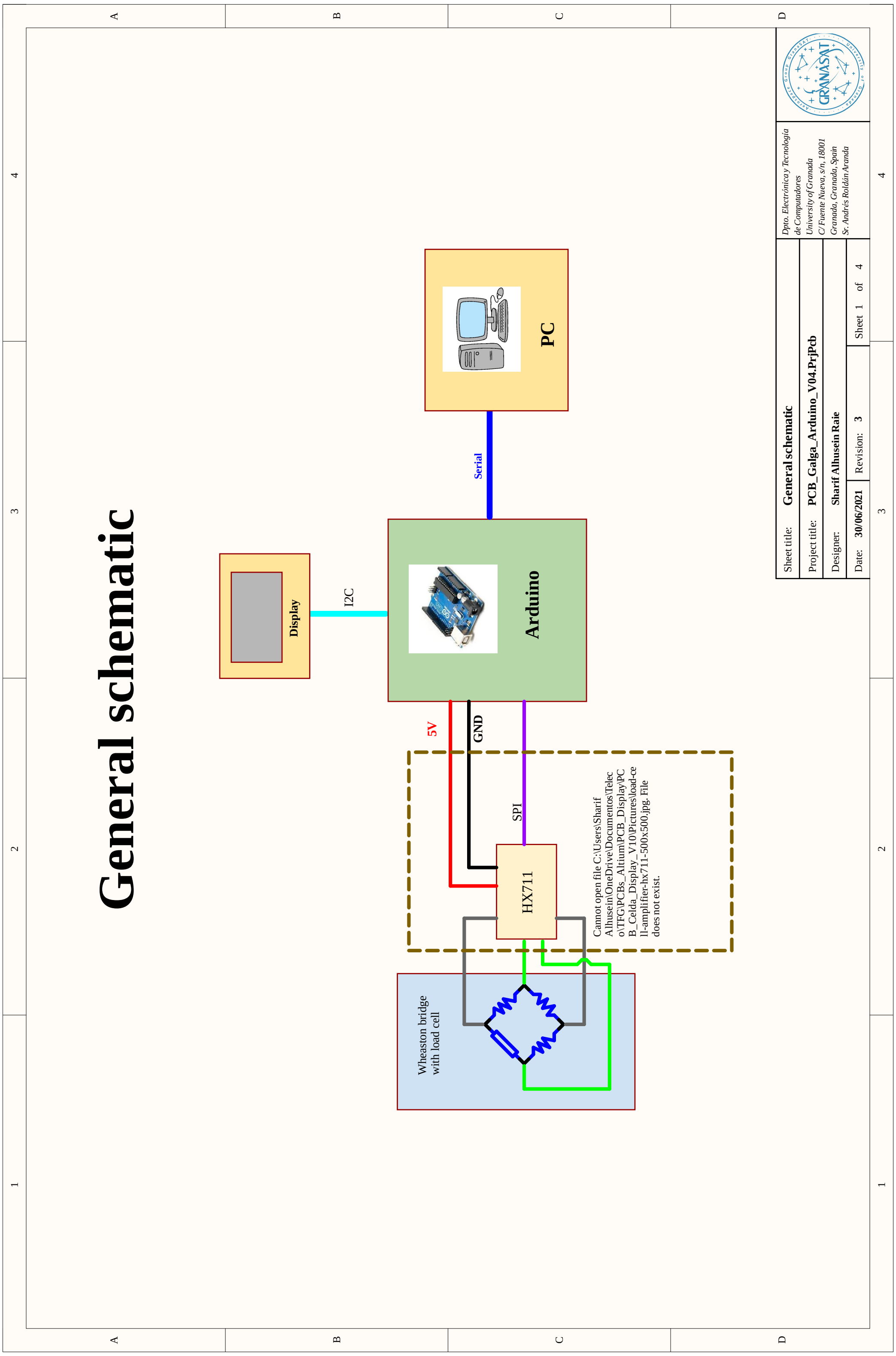
Post	Time (Hours)	Cost (€)
Junior Engineer	1830	18300.00
Senior Engineer	300	15000.00
	TOTAL	**33300.00 €**

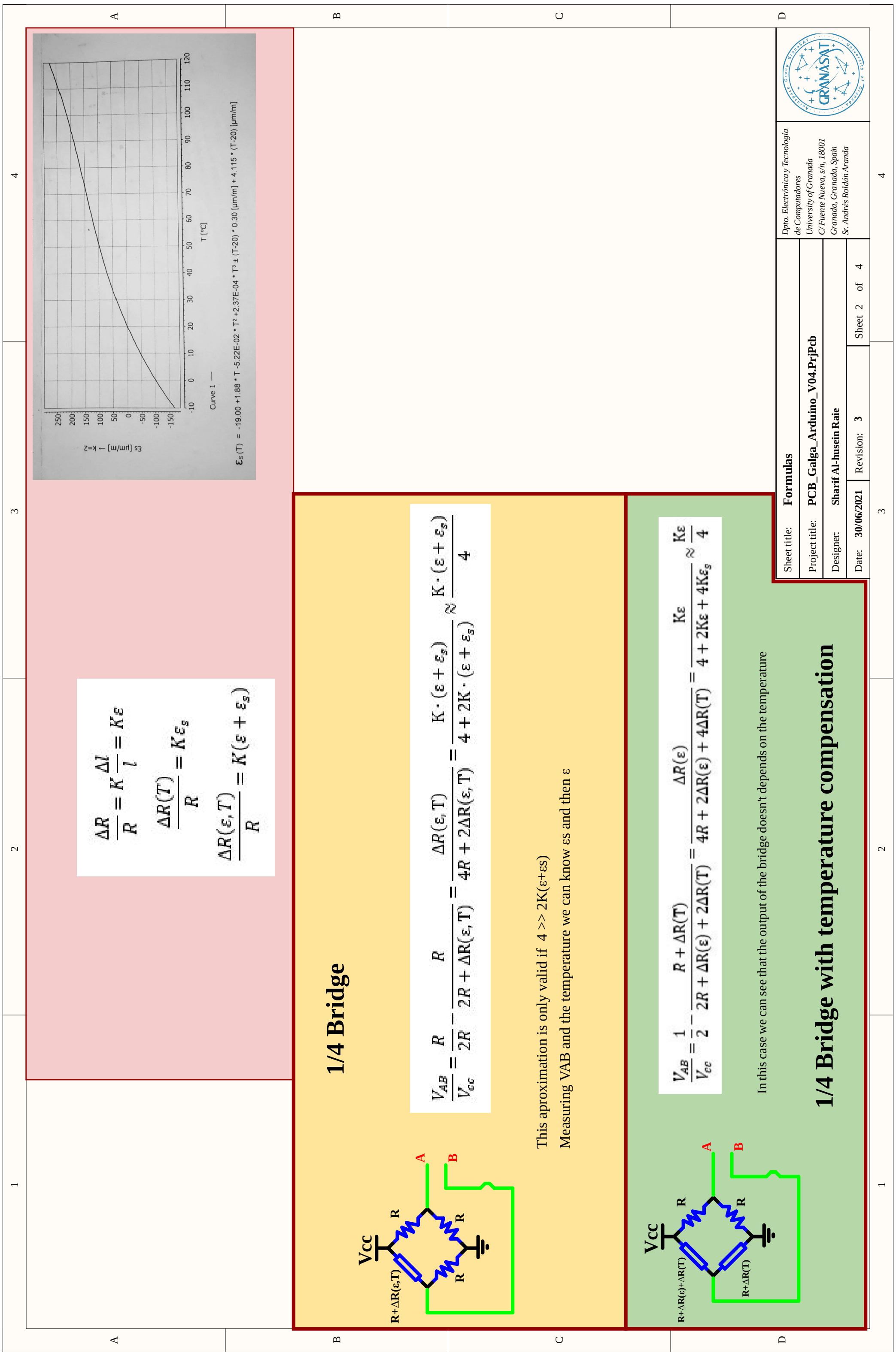
Table A.2 – *Human Resources Cost*

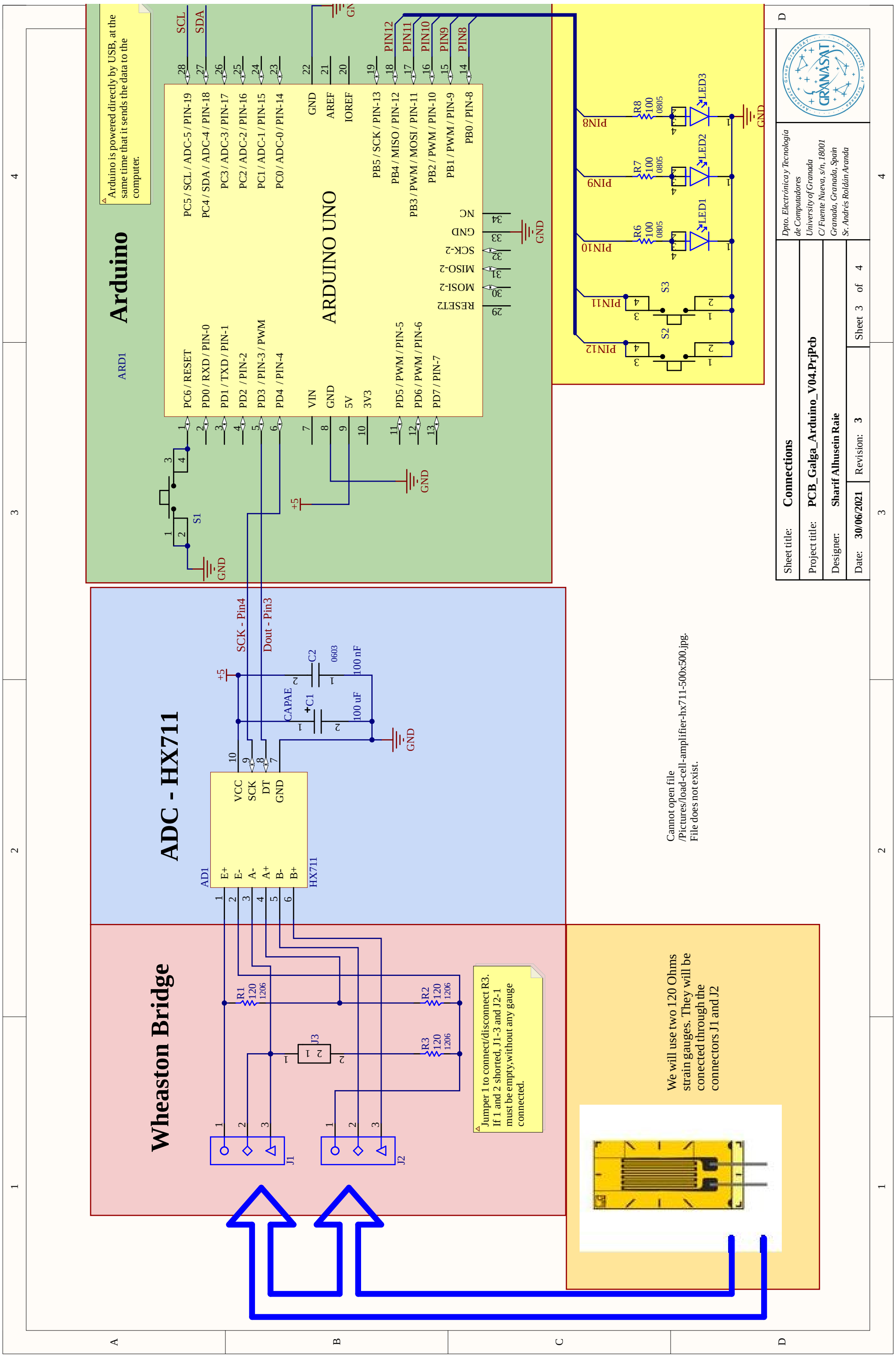
Appendix B

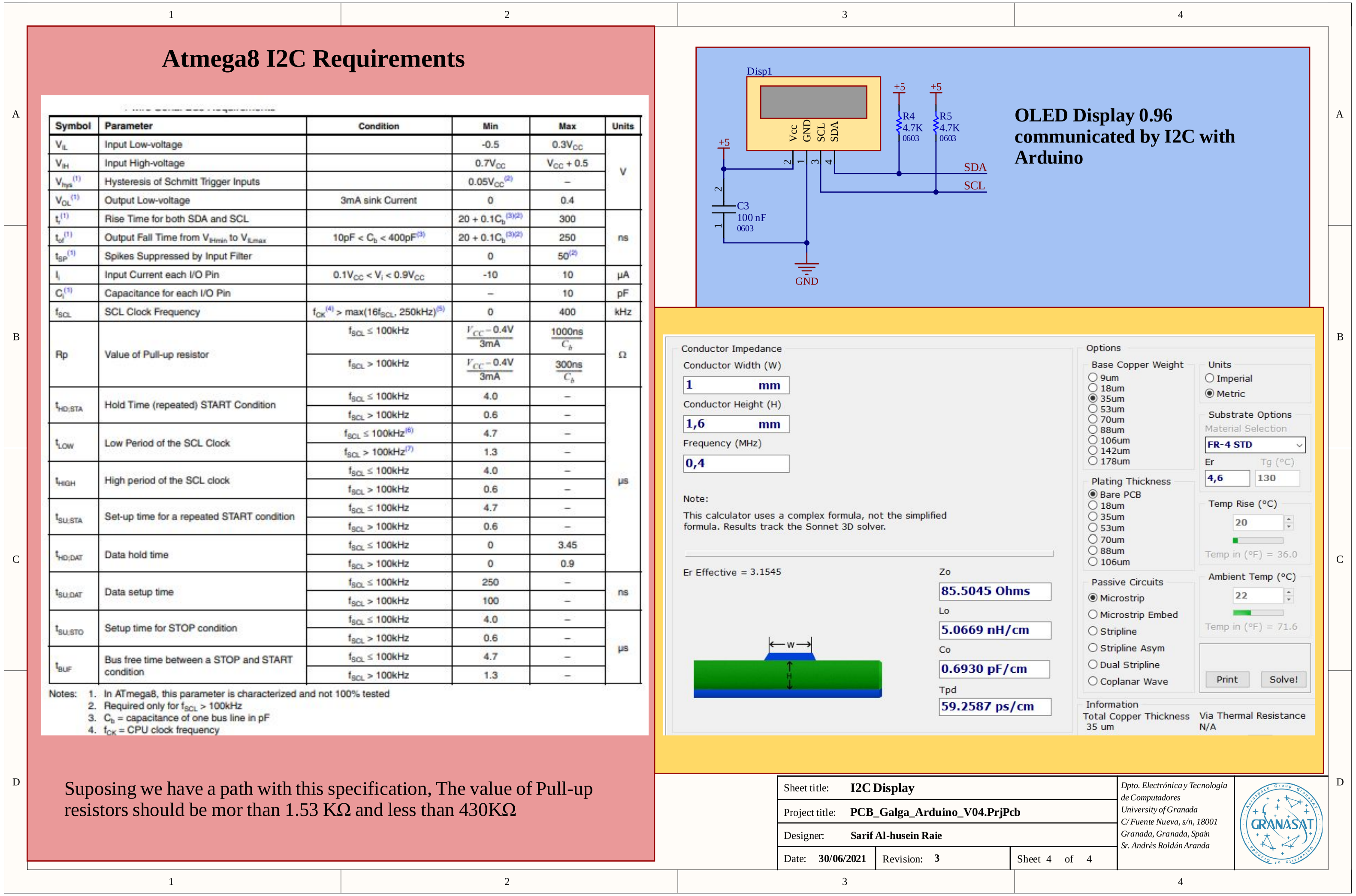
Altium Files

B.1 Main PCB SCH



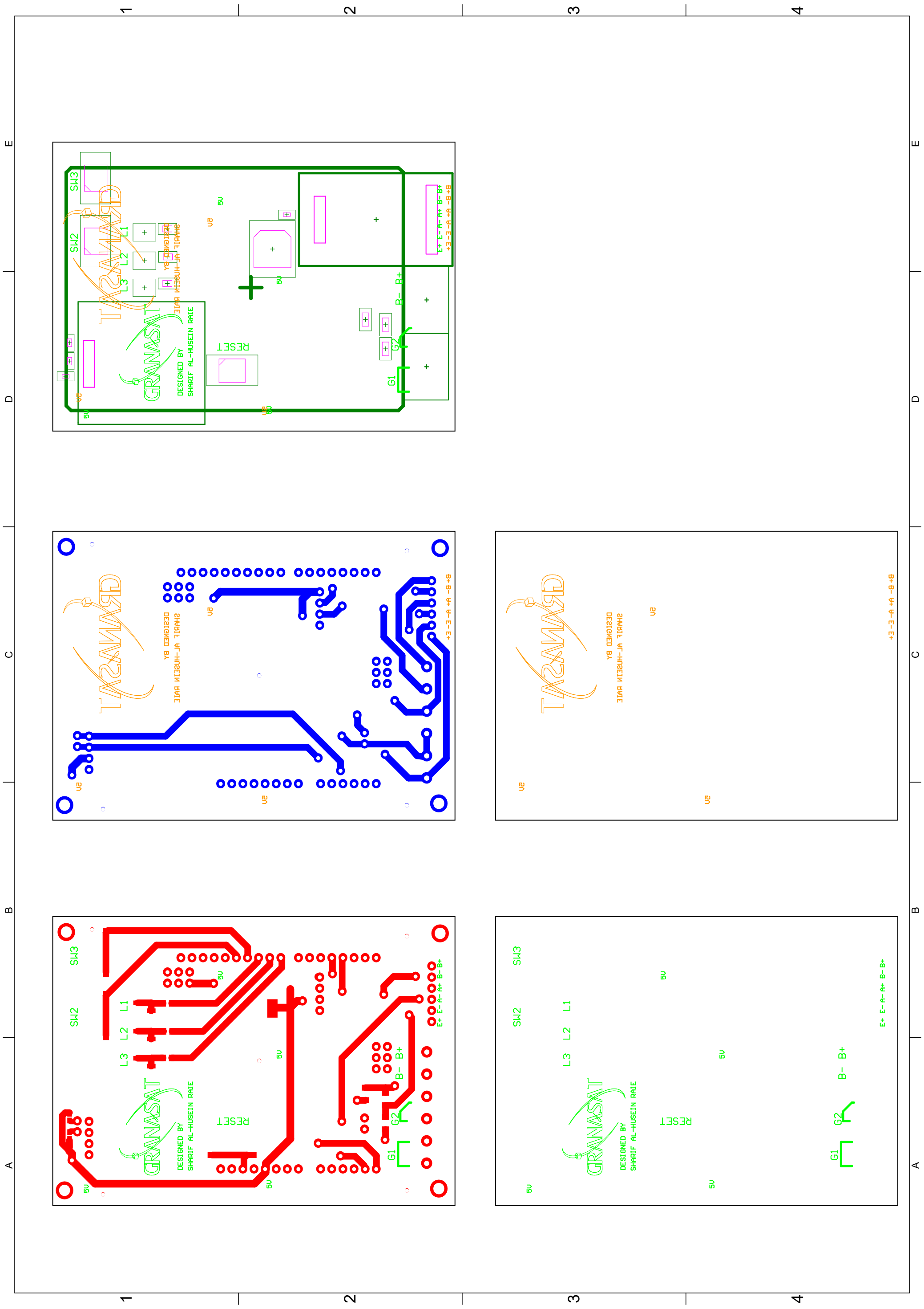

$\frac{\Delta R}{R} = K\frac{\Delta l}{l} = K\varepsilon$
$\frac{\Delta R(T)}{R} = K\varepsilon_s$
$\frac{\Delta R(\varepsilon,T)}{R} = K(\varepsilon + \varepsilon_s)$
$\varepsilon_s(T) = -19.00 + 1.88 * T - 5.22E-02 * T^2 + 2.37E-04 * T^3 \pm (T-20) * 0.30$ [μm/m] + 4.115 * (T-20) [μm/m]
T [°C]
Curve 1
εs [μm/m] → k=2
1/4 Bridge
Vcc
R+ΔR(ε,T)
R
A
B
$\frac{V_{AB}}{V_{cc}} = \frac{R}{2R} - \frac{R}{2R+\Delta R(\varepsilon,T)} = \frac{\Delta R(\varepsilon,T)}{4R+2\Delta R(\varepsilon,T)} = \frac{K\cdot(\varepsilon+\varepsilon_s)}{4+2K\cdot(\varepsilon+\varepsilon_s)} \approx \frac{K\cdot(\varepsilon+\varepsilon_s)}{4}$
This aproximation is only valid if 4 >> 2K(ε+εs)
Measuring VAB and the temperature we can know εs and then ε
R+ΔR(ε)+ΔR(T)
R+ΔR(T)
$\frac{V_{AB}}{V_{cc}} = \frac{1}{2} - \frac{R+\Delta R(T)}{2R+\Delta R(\varepsilon)+2\Delta R(T)} = \frac{\Delta R(\varepsilon)}{4R+2\Delta R(\varepsilon)+4\Delta R(T)} = \frac{K\varepsilon}{4+2K\varepsilon+4K\varepsilon_s} \approx \frac{K\varepsilon}{4}$
In this case we can see that the output of the bridge doesn't depends on the temperature
1/4 Bridge with temperature compensation
Sheet title: Formulas
Project title: PCB_Galga_Arduino_V04.PrjPcb
Designer: Sharif Al-husein Raie
Date: 30/06/2021
Revision: 3
Sheet 2 of 4
Dpto. Electrónica y Tecnología de Computadores
University of Granada
C/ Fuente Nueva, s/n, 18001
Granada, Granada, Spain
Sr. Andrés Roldán Aranda
GRANASAT

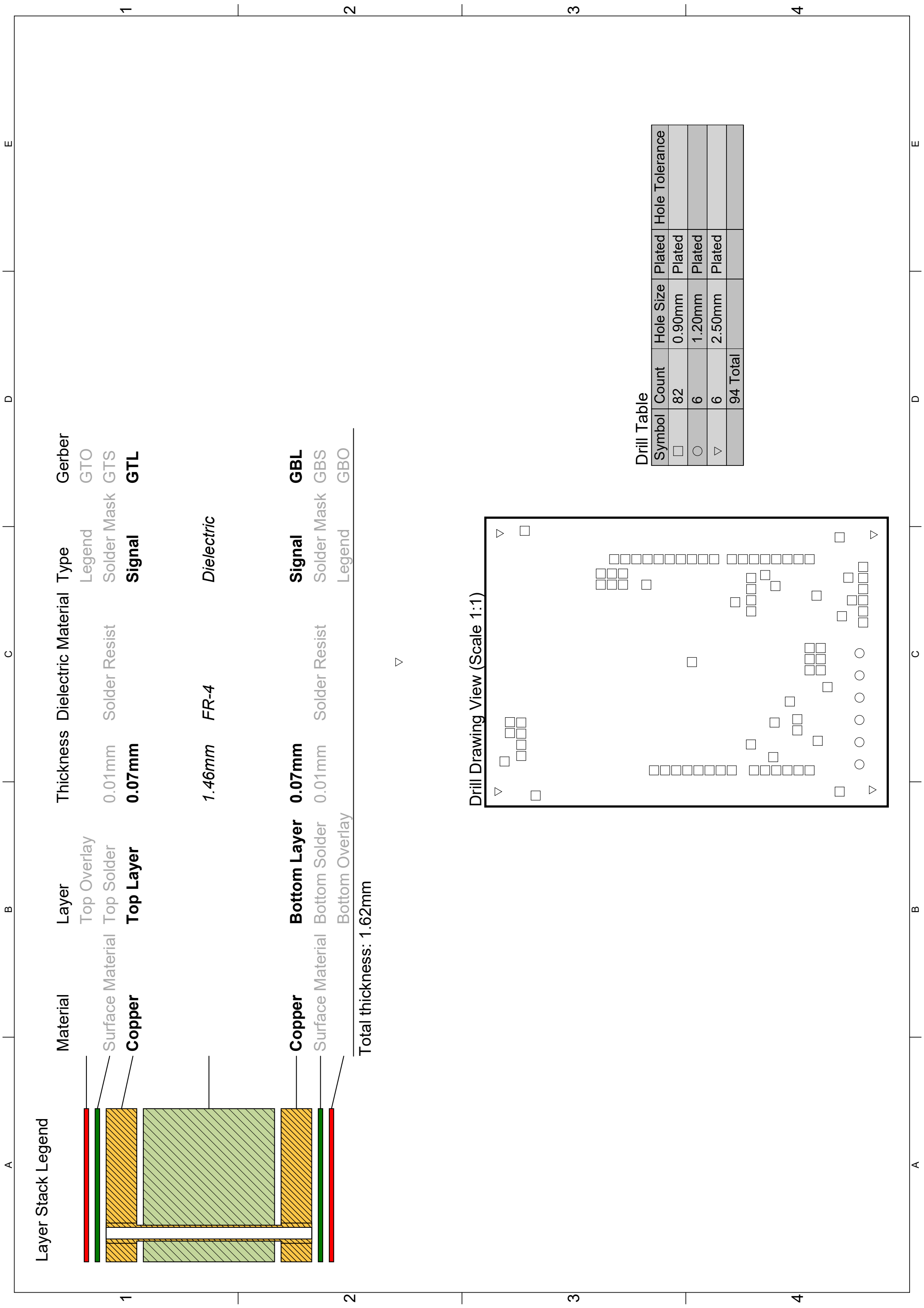

Arduino
ARDUINO UNO
ARD1
Arduino is powered directly by USB, at the same time that it sends the data to the computer.
ADC - HX711
AD1
HX711
SCK - Pin4
Dout - Pin3
Wheaston Bridge
Jumper 1 to connect/disconnect R3. If 1 and 2 shorted, J1-3 and J2-1 must be empty,without any gauge connected.
We will use two 120 Ohms strain gauges. They will be conected through the connectors J1 and J2
Cannot open file /Pictures/load-cell-amplifier-hx711-500x500.jpg. File does not exist.
Sheet title: Connections
Project title: PCB_Galga_Arduino_V04.PrjPcb
Designer: Sharif Alhusein Raie
Date: 30/06/2021
Revision: 3
Sheet 3 of 4
Dpto. Electrónica y Tecnología de Computadores
University of Granada
C/ Fuente Nueva, s/n. 18001
Granada, Granada, Spain
Sr. Andrés Roldán Aranda
GRANASAT



Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage		-0.5	$0.3V_{CC}$	V
V_{IH}	Input High-voltage		$0.7V_{CC}$	$V_{CC} + 0.5$	
V_{hys} (1)	Hysteresis of Schmitt Trigger Inputs		$0.05V_{CC}$ (2)	–	
V_{OL} (1)	Output Low-voltage	3mA sink Current	0	0.4	
t_r (1)	Rise Time for both SDA and SCL		$20 + 0.1C_b$ (3)(2)	300	ns
t_{of} (1)	Output Fall Time from V_{IHmin} to V_{ILmax}	$10pF < C_b < 400pF$ (3)	$20 + 0.1C_b$ (3)(2)	250	
t_{SP} (1)	Spikes Suppressed by Input Filter		0	50 (2)	
I_i	Input Current each I/O Pin	$0.1V_{CC} < V_i < 0.9V_{CC}$	-10	10	µA
C_i (1)	Capacitance for each I/O Pin		–	10	pF
f_{SCL}	SCL Clock Frequency	f_{CK} (4) > max(16f_{SCL}, 250kHz) (5)	0	400	kHz
Rp	Value of Pull-up resistor	$f_{SCL} \le 100kHz$	$\frac{V_{CC} - 0.4V}{3mA}$	$\frac{1000ns}{C_b}$	Ω
		$f_{SCL} > 100kHz$	$\frac{V_{CC} - 0.4V}{3mA}$	$\frac{300ns}{C_b}$	
$t_{HD;STA}$	Hold Time (repeated) START Condition	$f_{SCL} \le 100kHz$	4.0	–	µs
		$f_{SCL} > 100kHz$	0.6	–	
t_{LOW}	Low Period of the SCL Clock	$f_{SCL} \le 100kHz$ (6)	4.7	–	
		$f_{SCL} > 100kHz$ (7)	1.3	–	
t_{HIGH}	High period of the SCL clock	$f_{SCL} \le 100kHz$	4.0	–	
		$f_{SCL} > 100kHz$	0.6	–	
$t_{SU;STA}$	Set-up time for a repeated START condition	$f_{SCL} \le 100kHz$	4.7	–	
		$f_{SCL} > 100kHz$	0.6	–	
$t_{HD;DAT}$	Data hold time	$f_{SCL} \le 100kHz$	0	3.45	
		$f_{SCL} > 100kHz$	0	0.9	
$t_{SU;DAT}$	Data setup time	$f_{SCL} \le 100kHz$	250	–	ns
		$f_{SCL} > 100kHz$	100	–	
$t_{SU;STO}$	Setup time for STOP condition	$f_{SCL} \le 100kHz$	4.0	–	µs
		$f_{SCL} > 100kHz$	0.6	–	
t_{BUF}	Bus free time between a STOP and START condition	$f_{SCL} \le 100kHz$	4.7	–	
		$f_{SCL} > 100kHz$	1.3	–	

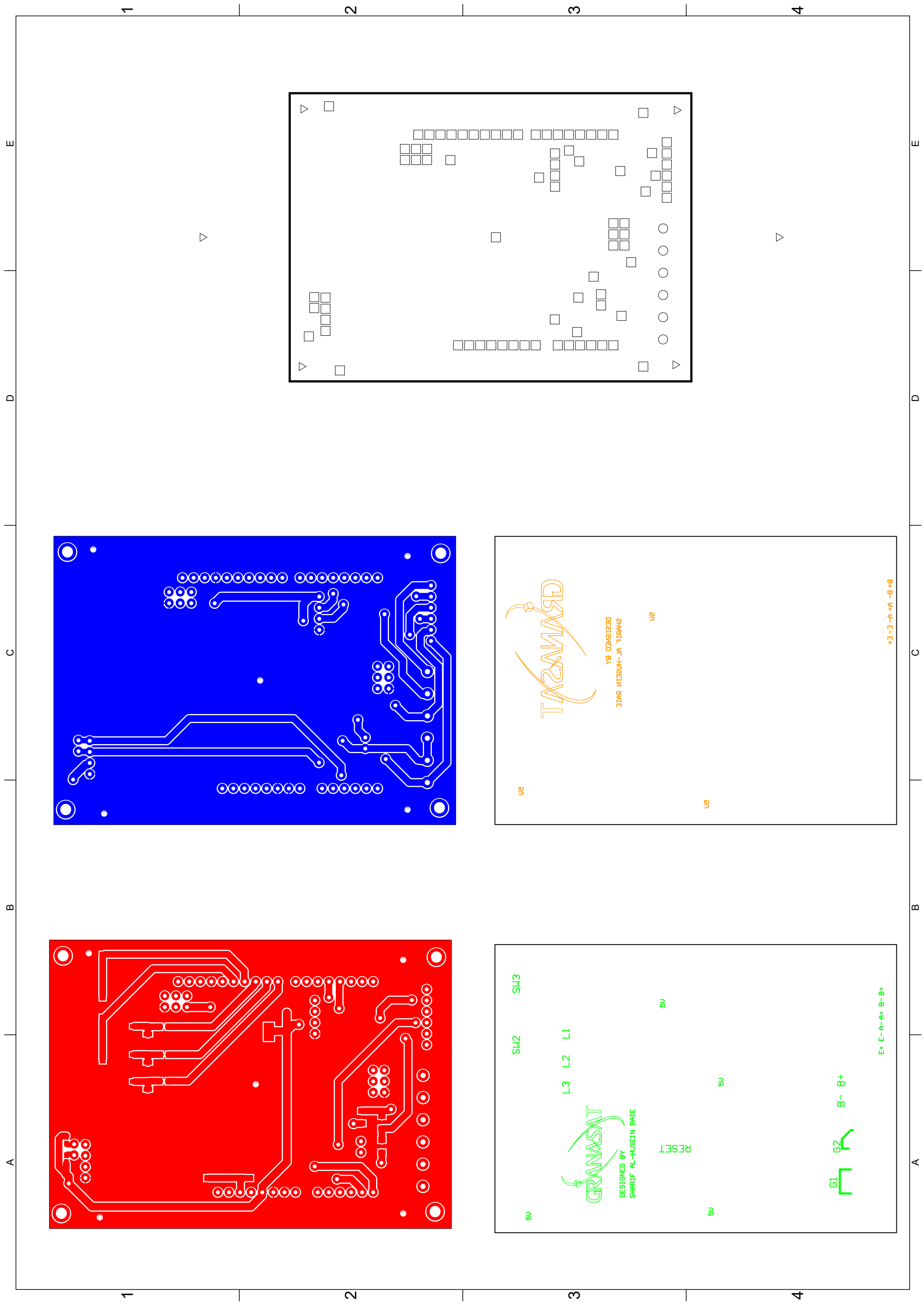
B.2 Manufacture Views

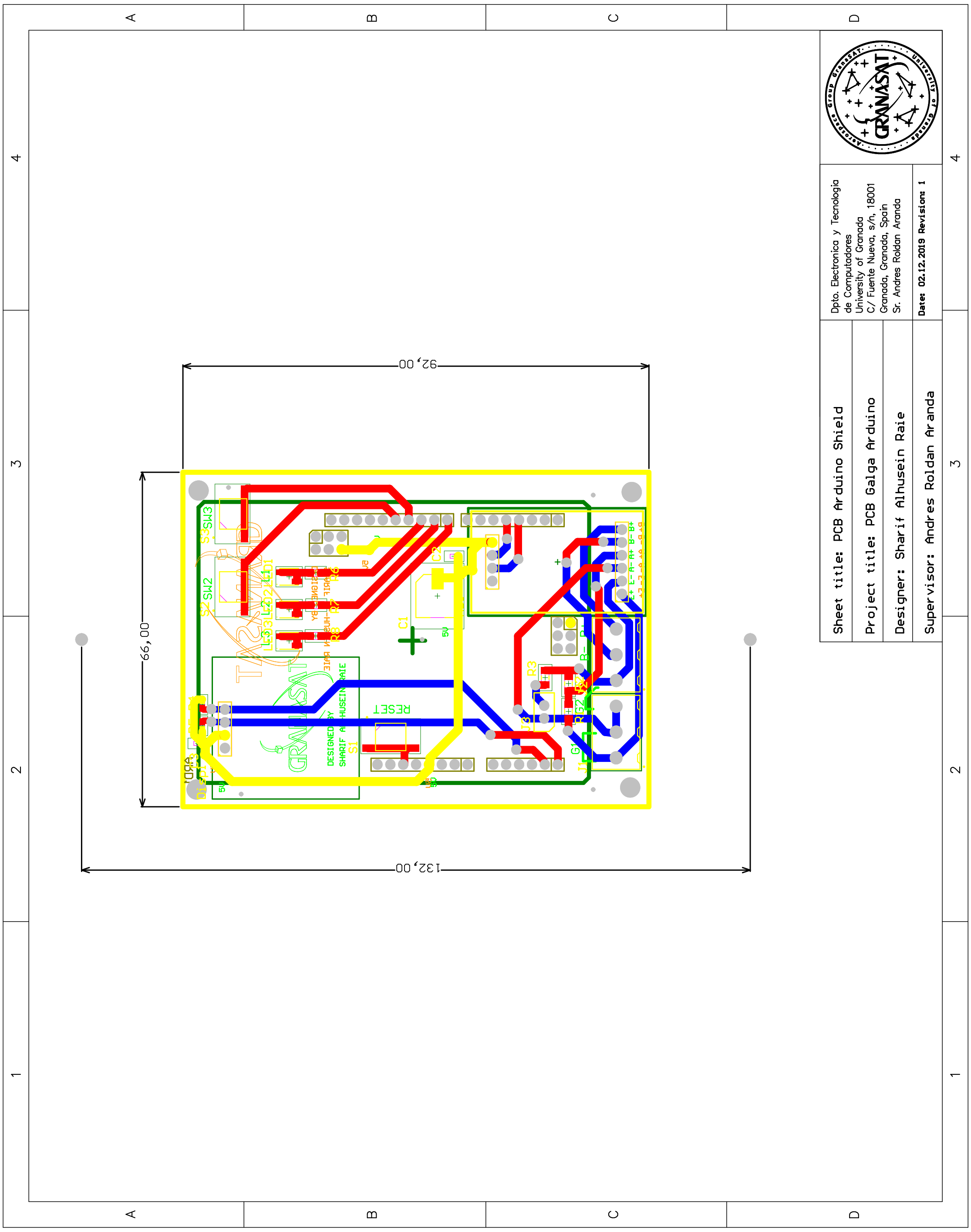


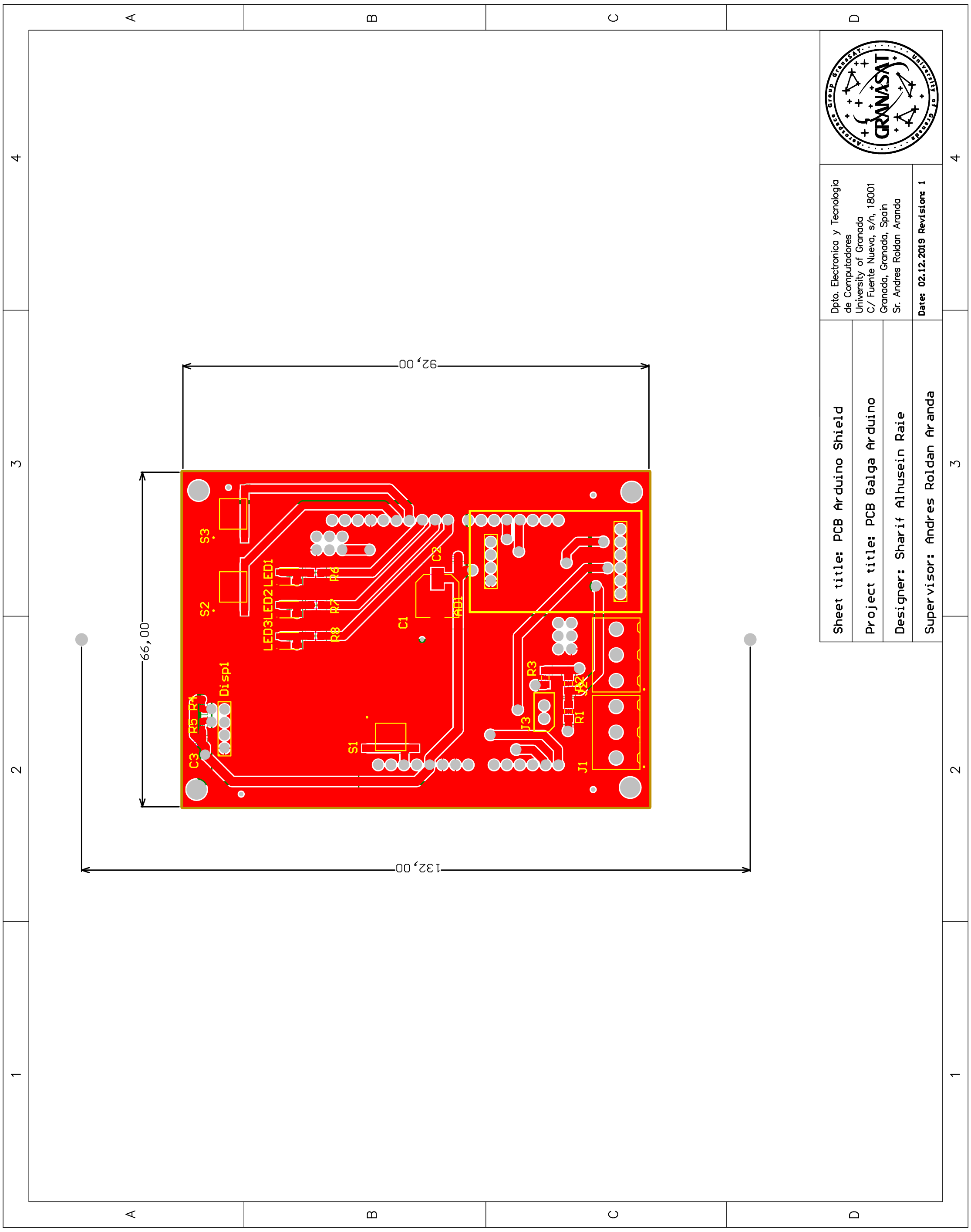

Layer Stack Legend
Material
Layer
Thickness
Dielectric Material
Type
Gerber
Top Overlay
Legend
GTO
Surface Material
Top Solder
0.01mm
Solder Resist
Solder Mask
GTS
Copper
Top Layer
0.07mm
Signal
GTL
1.46mm
FR-4
Dielectric
Copper
Bottom Layer
0.07mm
Signal
GBL
Surface Material
Bottom Solder
0.01mm
Solder Resist
Solder Mask
GBS
Bottom Overlay
Legend
GBO
Total thickness: 1.62mm
Drill Drawing View (Scale 1:1)
Drill Table
Symbol
Count
Hole Size
Plated
Hole Tolerance
82
0.90mm
Plated
6
1.20mm
Plated
6
2.50mm
Plated
94 Total
A
B
C
D
E
1
2
3
4

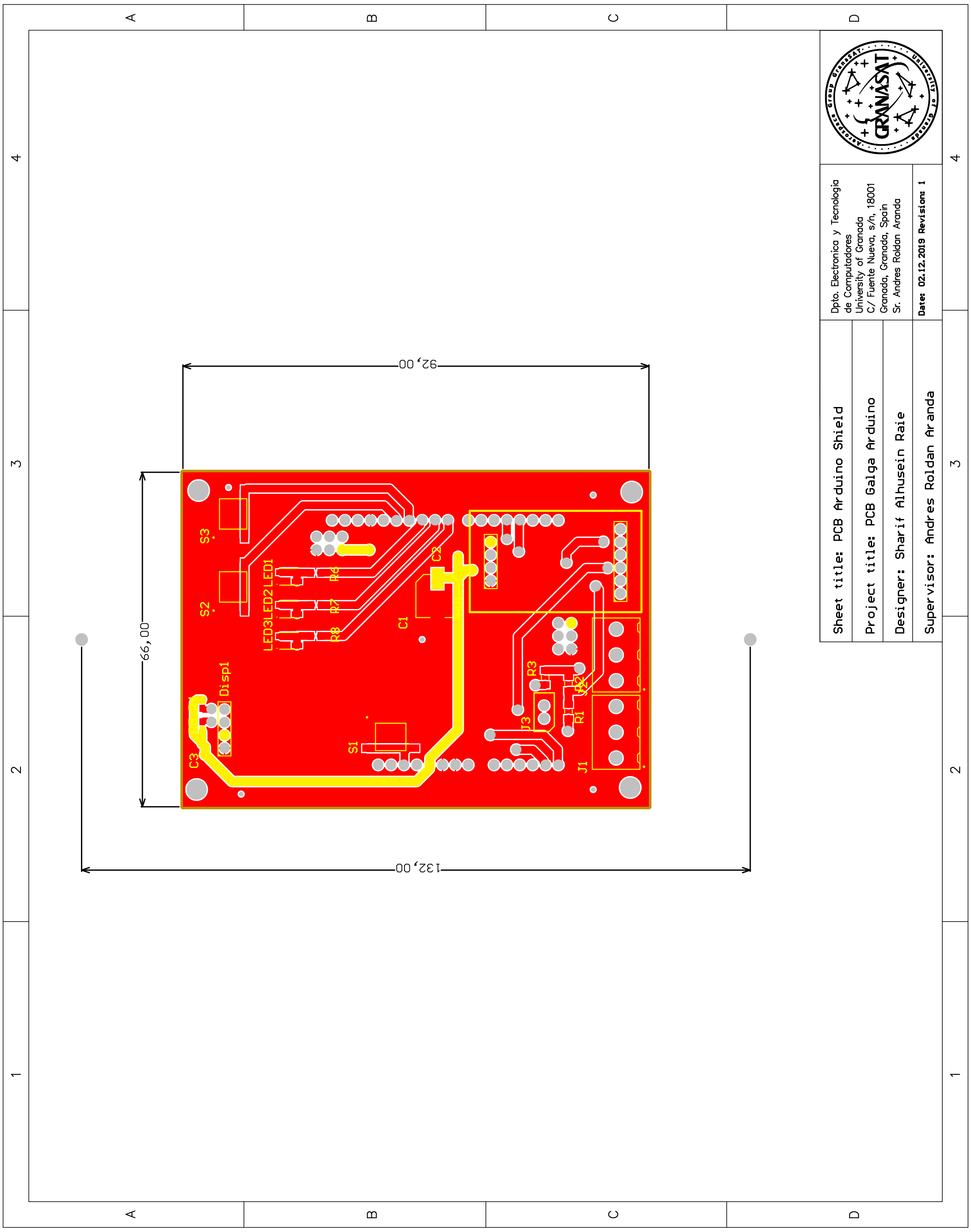
Bill Of Materials

Line #	Designator	Comment	Quantity
1	AD1	HX711	1
2	ARD1	Arduino UNO	1
3	C1	100 uF	1
4	C2, C3	100 nF	2
5	Disp1	display	1
6	J1, J2	dg500-5.08	2
7	J3	Jump-2	1
8	LED1, LED2, LED3	LAE6SF-AABA	3
9	R1, R2, R3	120	3
10	R4, R5	4.7K	2
11	R6, R7, R8	100	3
12	S1, S2, S3	ULV94V-0	3

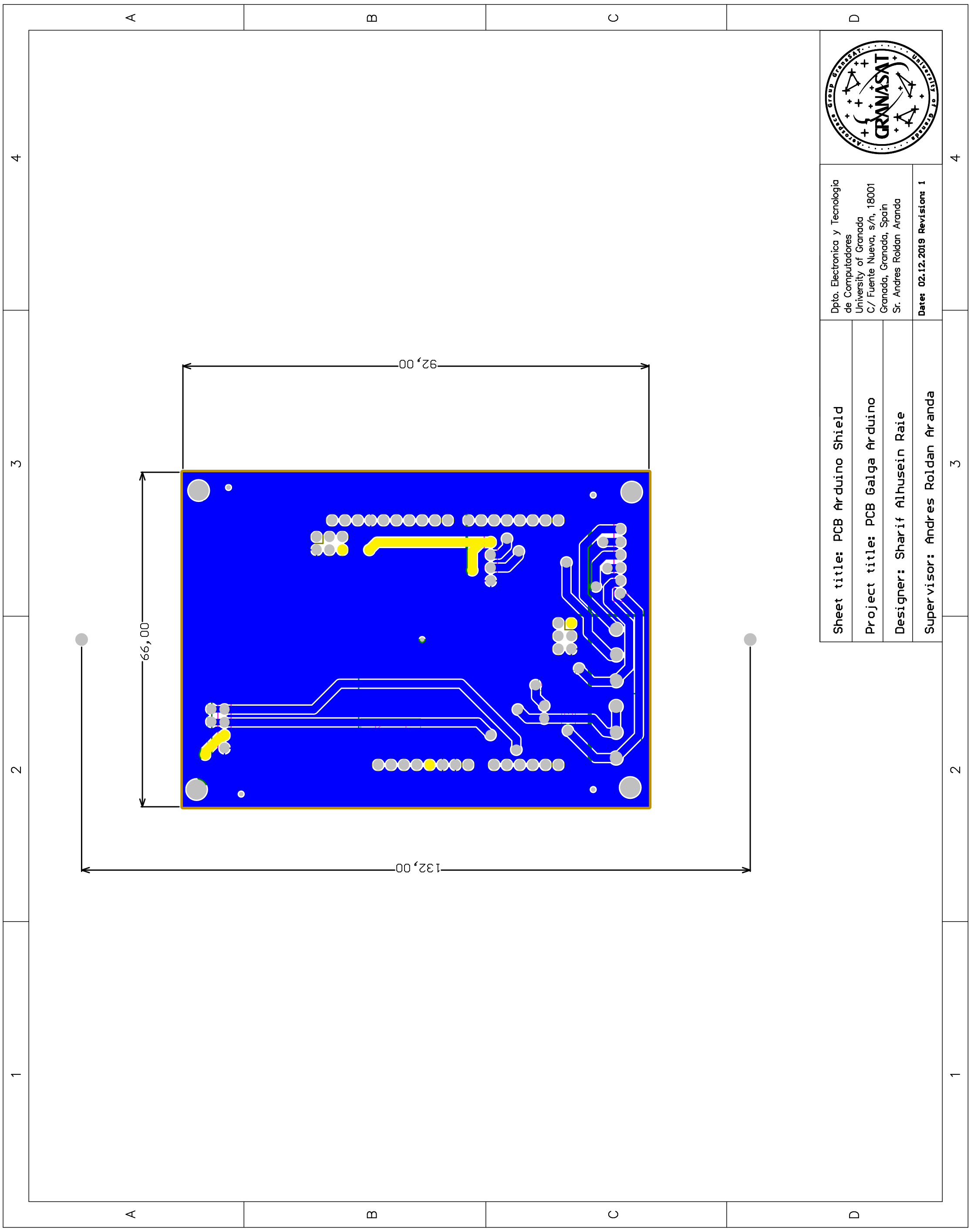


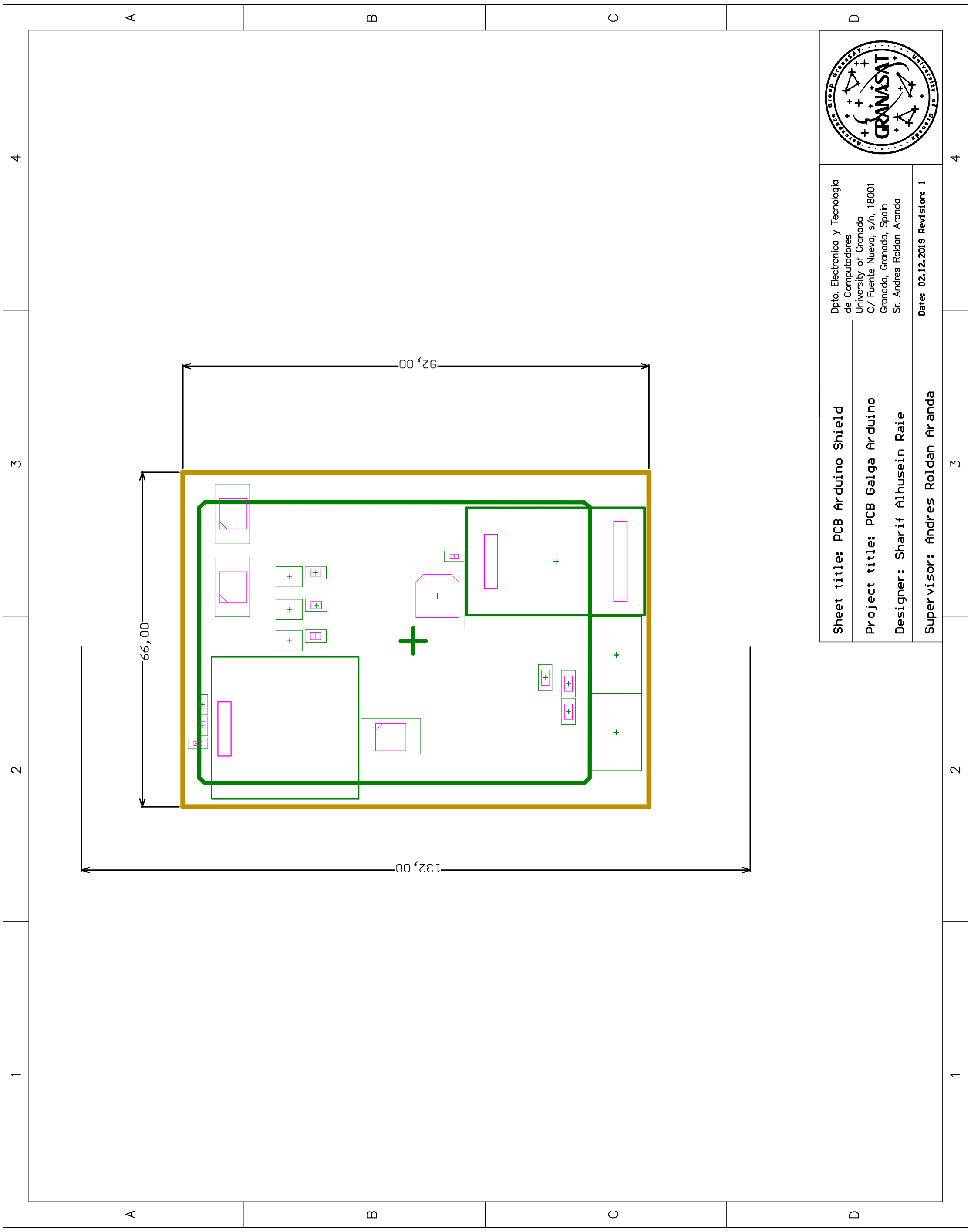

92,00
66,00
132,00
Sheet title: PCB Arduino Shield
Project title: PCB Galga Arduino
Designer: Sharif Alhusein Raie
Supervisor: Andres Roldan Aranda
Dpto. Electronica y Tecnologia de Computadores
University of Granada
C/ Fuente Nueva, s/n, 18001
Granada, Granada, Spain
Sr. Andres Roldan Aranda
Date: 02.12.2019 Revision: 1


Sheet title: PCB Arduino Shield
Project title: PCB Galga Arduino
Designer: Sharif Alhusein Raie
Supervisor: Andres Roldan Aranda
Dpto. Electronica y Tecnologia de Computadores
University of Granada
C/ Fuente Nueva, s/n, 18001
Granada, Granada, Spain
Sr. Andres Roldan Aranda
Date: 02.12.2019 Revision: 1
92,00
66,00
132,00

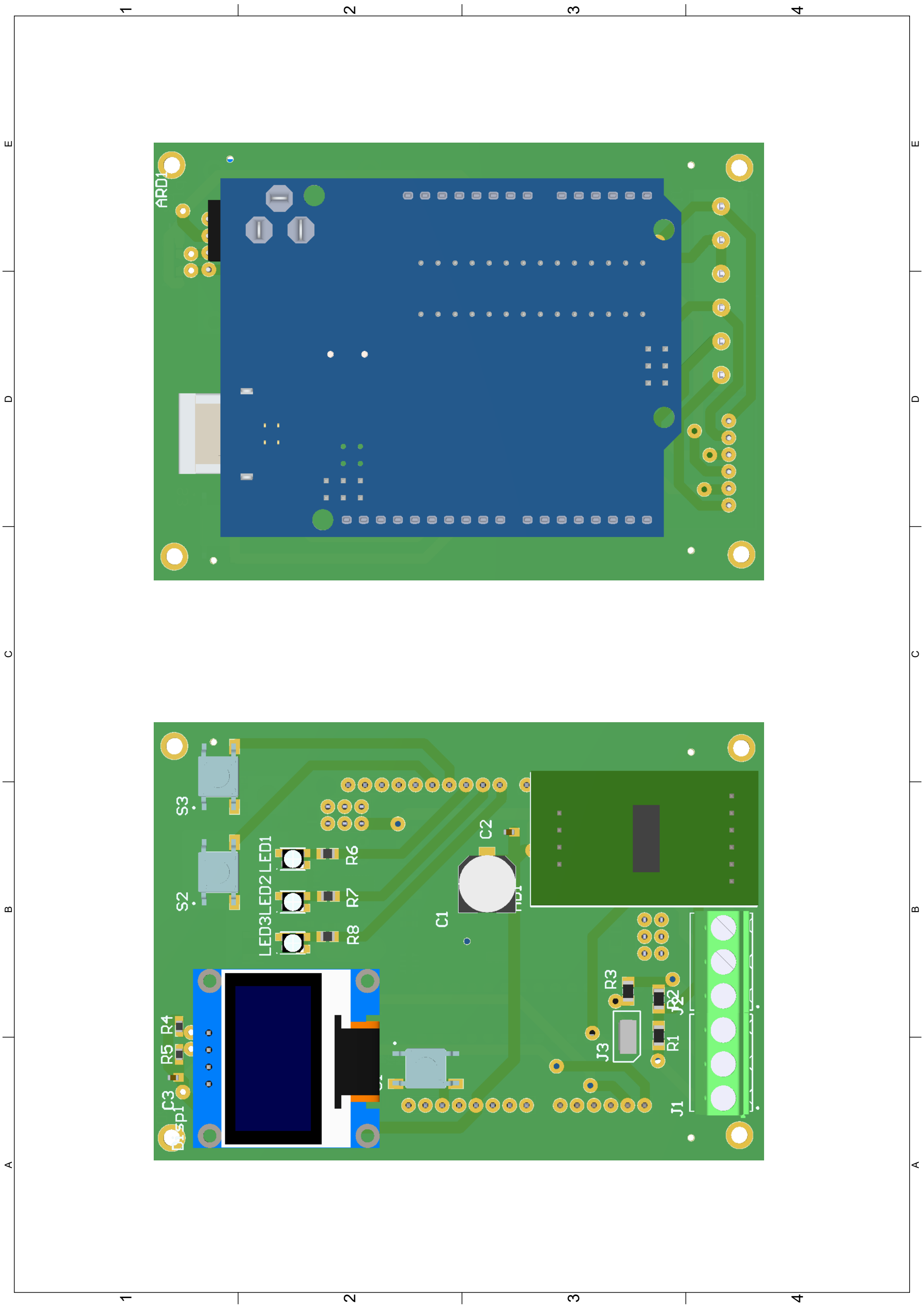


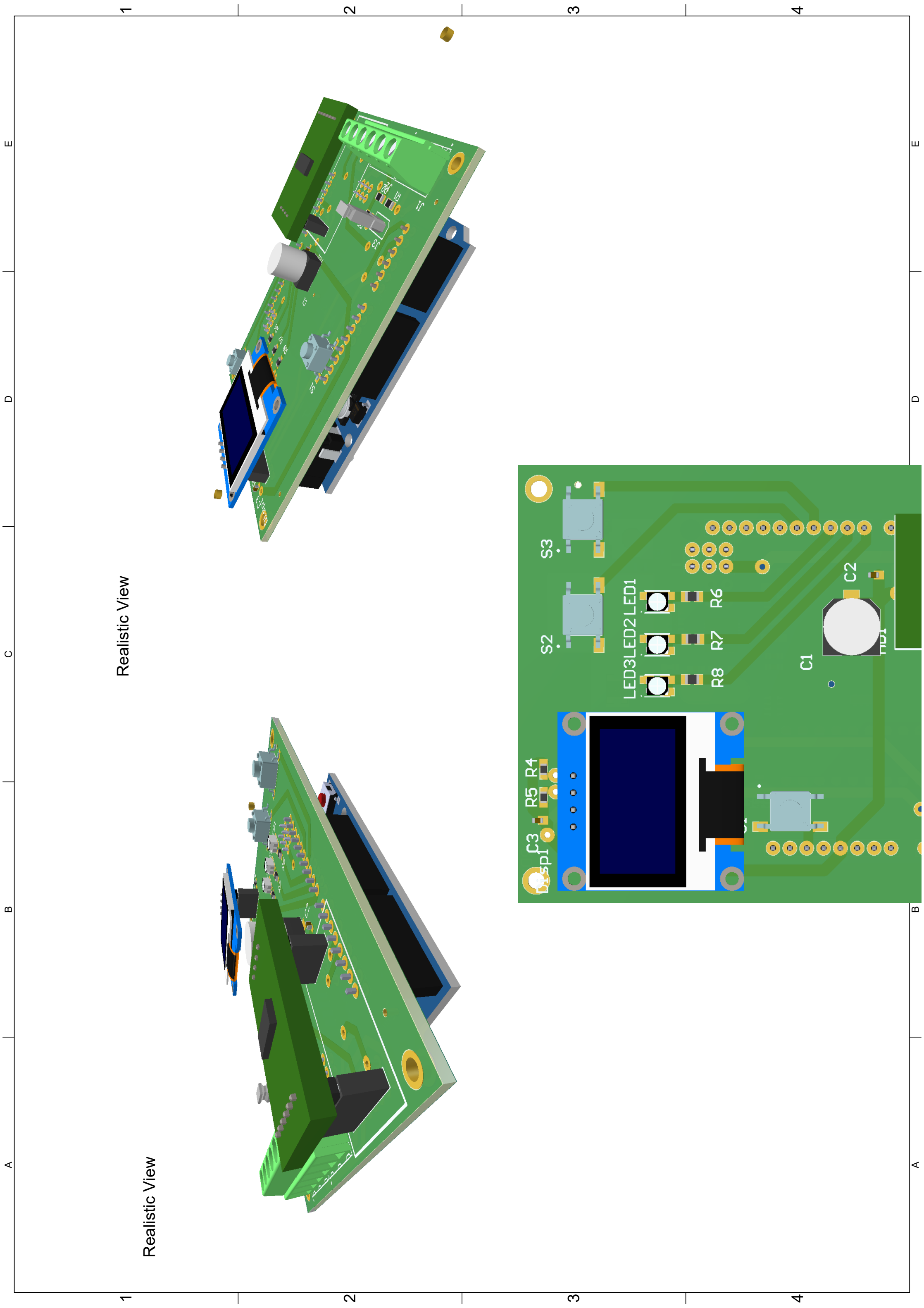
92,00
66,00
132,00
Sheet title: PCB Arduino Shield
Project title: PCB Galga Arduino
Designer: Sharif Alhusein Raie
Supervisor: Andres Roldan Aranda
Dpto. Electronica y Tecnologia de Computadores
University of Granada
C/ Fuente Nueva, s/n, 18001
Granada, Granada, Spain
Sr. Andres Roldan Aranda
Date: 02.12.2019 Revision: 1

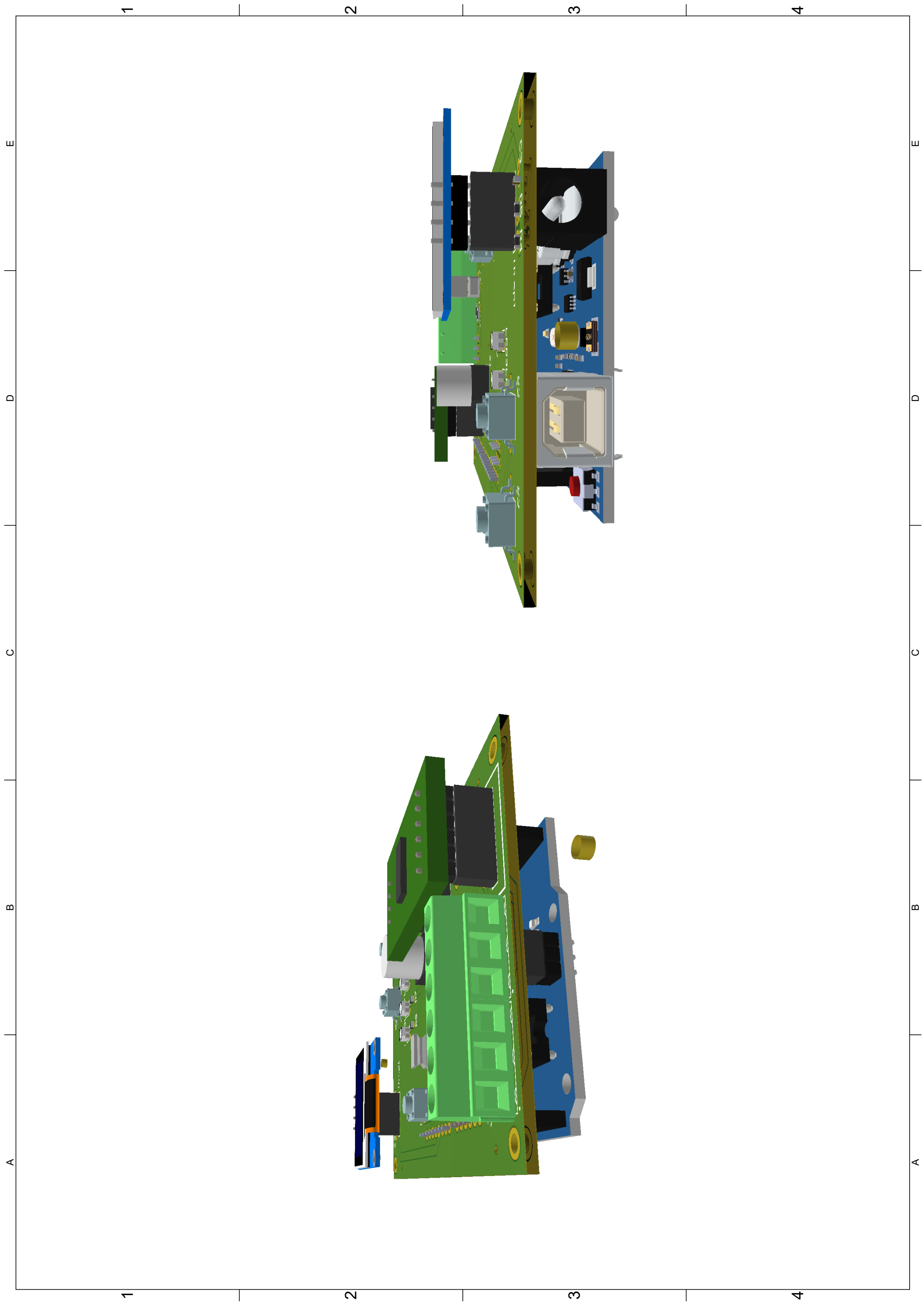

Sheet title: PCB Arduino Shield
Project title: PCB Galga Arduino
Designer: Sharif Alhusein Raie
Supervisor: Andres Roldan Aranda
Dpto. Electronica y Tecnologia de Computadores
University of Granada
C/ Fuente Nueva, s/n, 18001
Granada, Granada, Spain
Sr. Andres Roldan Aranda
Date: 02.12.2019 Revision: 1
92,00
66,00
132,00

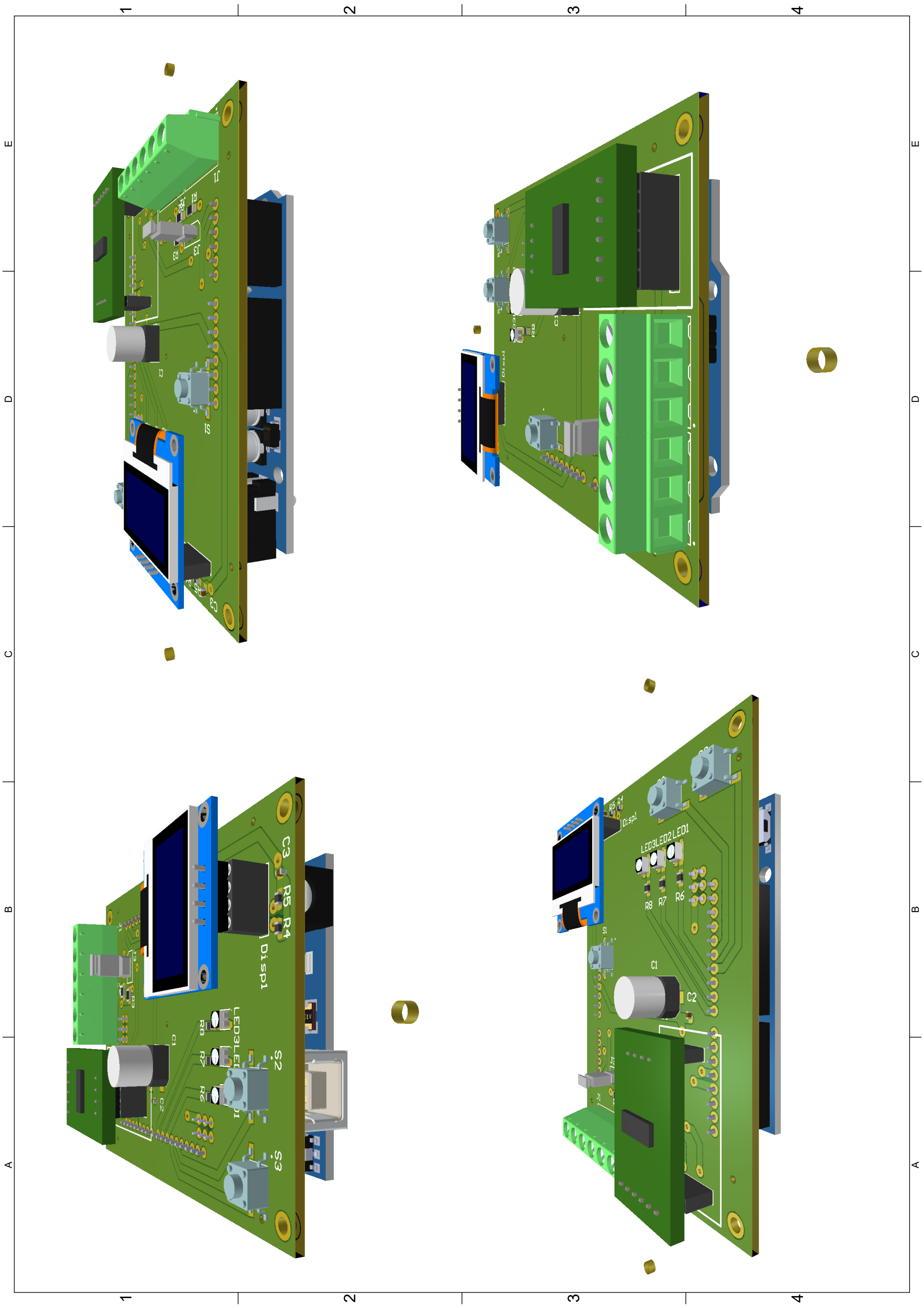

92,00
66,00
132,00
Sheet title: PCB Arduino Shield
Project title: PCB Galga Arduino
Designer: Sharif Alhusein Raie
Supervisor: Andres Roldan Aranda
Dpto. Electronica y Tecnologia de Computadores
University of Granada
C/ Fuente Nueva, s/n, 18001
Granada, Granada, Spain
Sr. Andres Roldan Aranda
Date: 02.12.2019 Revision: 1

B.3 Realistic Views




Realistic View
Realistic View
S3
S2
LED3LED2LED1
R6
R7
R8
C1
C2
R4
R5
C3


1
2
3
4
E
D
C
B
A



Appendix C

Code

C.1 GUI code with Python

```python

#   _____                       _____      _______
#  / ____|                     / ____|  /\|__   __|
# | |  __ _ __ __ _ _ __   __ _| (___   /  \  | |
# | | |_ | '__/ _` | '_ \ / _` |\___ \ / /\ \ | |
# | |__| | | | (_| | | | | (_| |____) / ____ \| |
#  \_____|_|  \__,_|_| |_|\__,_|_____/_/    \_\_|
#
#
# Python Version:  Python 3
#
# * Versión: 00 - 24/07/2021
#
#   Programa que implementa una interfaz gráfica (GUI) para facilitar la interacción del usuario con
#   el sistema embebido para monitorizar infraestructuras fabricadas con madera.
#   En la interfaz se muestran diferentes opciones para mostrar los datos recopilados por el sensor y plotearlos
#
#
# * Versión: 00
# * Autor  :  Sharif Alhusein Raie  -  shalhusein@gmail.com  - 24/07/2021
#   El diseño de la interfaz se realiza con Qt designer y se importa en este programa usando
#   uic.loadUi("MainWindow.ui", self)
#




import sys
from PyQt5 import QtWidgets, uic, QtCore
import os
import sys
import time
import serial
import serial.tools.list_ports


# //****** Define DATA PACKET STRUCTURE ******
#
# // Structure of the header
#
# //_________Id_____________Value_
# //                       |       |
# //     Info              | 0x00  |
# //     Channel A data    | 0x01  |
# //     Channel B data    | 0x02  |
# //     Setting parameters | 0x03  |
# //     Command           | 0x04  |
# //_______________________|_______|
#
# //|_____Info message_______|_Value_|
# //|                       |       |
# //|     Recived (ACK)      | 0x0   |
```

```
# //|     All OK and ready     | 0x01   |
# //|     OLED connected       | 0x02   |
# //|     OLED disconnected    | 0x03   |
# //|                          | 0x04   |
# //|__________________________|________|
#
# //|___Command message_____|_Value_|
# //|                       |       |
# //|     Stop measuring     | 0x0   |
# //|     Start measuring    | 0x01  |
# //|     Restart data       | 0x02  |
# //|     Tare               | 0x03  |
# //|                        | 0x04  |
# //|________________________|_______|
#
# //|____Setting parameters____|_Value_|
# //|                          |       |
# //|  Single Channel           | 0x00  |
# //|  Dual Channel             | 0x01  |
# //|  Auto Gain selection OFF  | 0x02  |
# //|  Auto Gain selection ON   | 0x03  |
# //|                           | 0x04  |
# //|___________________________|_______|

class messagePack:
    def __init__(self):
        super().__init__()

        self.id = 0
        self.num = 0
        self.len = 0
        self.message = 0


    def __init__(self, packet):
        super().__init__()

        self.id = packet[0]
        self.num = packet[2]*256+packet[1]
        self.len = packet[3]
        self.message = packet[4]


class dataPack:
    def __init__(self):
        super().__init__()

        self.id = 0
        self.num = 0
        self.len = 0
        self.gain = 0
        self.time = 0
        self.measure = 0


    def __init__(self, packet):
        super().__init__()

        self.id = packet[0]
        self.num = packet[2]*256+packet[1]
        self.len = packet[3]
        self.gain = packet[4]
        self.time = packet[8]*65536+packet[7]*4096+packet[6]*256+packet[5]
        self.measure = packet[12]*65536+packet[11]*4096+packet[10]*256+packet[9]




default_BaudRate = 9600

class MainWindow(QtWidgets.QMainWindow):

    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)

        #Load the UI Page
        uic.loadUi('MainWindow.ui', self)

        # Search available serial porsts and add them to comboBox
        self.list_SerialPorts()
        # Write default Baud Rate in Qline box
        self.lineEdit_BRate.setText(str(default_BaudRate))
        # Select single channel option
        self.radioButton_Sgl_CH.setChecked(True)
        self.channel_conf = 1    # 1 -> Single channel
```

```
                                # 2 -> Dual Channel
        # Select automatic gain adjustment
        self.checkBox_GainAdjust.setChecked(True)


        # En caso de evento pulsar connect_button se llama a la función connect() lo mismo con todos los botones
        self.connect_button.clicked.connect(lambda: self.connect())
        self.disconnect_button.clicked.connect(lambda: self.disconnect())
        self.start_button.clicked.connect(lambda: self.startSampling())
        self.stop_button.clicked.connect(lambda: self.stopSampling())
        self.clear_button.clicked.connect(lambda: self.clearPlot())
        self.tare_button.clicked.connect(lambda: self.tareDevice())
        self.save_button.clicked.connect(lambda: self.saveData())

        # Set graphics bacground white
        self.graph_1.setBackground('w')
        self.graph_2.setBackground('w')

        # Set QPlainTextEdit to read only mode, user can't modify the text
        self.plainTextEdit.setReadOnly(True)
        self.plainTextEdit.setPlainText("Select Serial parameters then click on Connect\n")

    def list_SerialPorts(self):
        # Add in comboBox an item for each COM port detected to allow user to select one

        # print('Searching serial ports...')
        # Available ports
        ports = serial.tools.list_ports.comports(include_links=False)
        if len(ports)>0:
            self.serialPort = ports[0].device
        else:
            self.serialPort = ""

        for port in ports :
            print('Find port '+ port.device)
            # Add items, comboBox is the name given to the item in our Qt desig
            self.comboBox.addItem(port.device)


    def connect(self):

        # Get selected COM Port
        serialPort = self.comboBox.currentText()
        # Read selected baud rate
        baud = self.lineEdit_BRate.text()

        # Serial Port
        self.ser = serial.Serial(serialPort, baud, timeout=10)

        # Select channel configuration. Reading from single chanel or dual channel
        if (self.radioButton_Sgl_CH.isChecked ()):
            self.channel_conf = 1    # 1 -> Single channel
        else:
            self.channel_conf = 2    # 2 -> Dual Channel

        # Automatic Gain adjustment
        self.auto_Gain_Adj = self.checkBox_GainAdjust.isChecked()

        while(not self.ser.inWaiting()):
            # Wait for input data
            continue

        pack = self.ser.read(5)
        msg = messagePack(pack)

        if (msg.message == 2):
            #OLED ON
            self.plainTextEdit.setPlainText("Device Connected\n")
            self.plainTextEdit.setPlainText("OLED is ON\n")
        elif (msg.message == 3):
            #OLED OFF
            self.plainTextEdit.setPlainText("Device Connected\n")
            self.plainTextEdit.setPlainText("OLED is OFF\n")

        # Enviar paquete con los parámetros de configuración, un paquete por parámetro, en total dos paquetes
        # Desactivar el boton de connect y las casillas de eleccion de parámetros





    def disconnect(self):
        self.ser.close()
        # Activar botones deconnect y los de elección de parametros

```

```


    def startSampling(self):



        # Sampling at 10 Hz
        # ... init continued ...
        # Set a timer, every 80 ms calls update_plot_data function to update the plot
        self.timer = QtCore.QTimer()
        self.timer.setInterval(80)
        self.timer.timeout.connect(self.update_plot_data)
        self.timer.start()

        self.grafics_config()


    def grafics_config(self):
        if (self.GainA == 128):
            color = (255, 0, 0)
        else:
            color = (0, 0, 255)

        penA = pg.mkPen(color=color)
        penB = pg.mkPen(color=(0, 0, 255))
        self.data_line =  self.graphicsView_1.plot(self.x2, self.y2, pen=penA)
        self.data_line_2 =  self.graphicsView_2.plot(self.x2, self.y2, pen=penB, symbol='o')


    def update_plot_data(self):
        header = self.ser.readline()
        #print(header)


        if(header==header_channelA):
            if (self.x1[-1] >= tmax1):    #
                self.x1 = self.x1[1:]     # Remove the first element.
                self.y1 = self.y1[1:]

            t = self.ser.readline()
            dt = self.ser.readline()
            self.x1.append(int(t)/1000)          # Add a new time value in seconds
            self.y1.append((int(dt) - self.offsetA) / 2**23 / self.GainA / K_factor * 4 * 10**6)       # Add new
             data in um/m

            self.data_chanelA.append(int(dt))
            self.time_chanelA.append(self.x1[-1])

            self.data_line.setData(self.x1, self.y1)  # Update the data.

            print("time " + str(self.x1[-1]) + "\t; Ch1 " + str(self.y1[-1]) )



            if (self.x2[-1] >= tmax2):    #
                self.x2 = self.x2[1:]     # Remove the first element.
                self.y2 = self.y2[1:]

            self.x2.append(self.x1[-1])          # Add a new time value in seconds
            self.y2.append(self.y1[-1])      # Add new data in um/m

            self.data_line_2.setData(self.x2, self.y2)  # Update the data.


        elif(header==header_channelB):
            if (self.x2[-1] >= tmax2):    #
                self.x2 = self.x2[1:]     # Remove the first element.
                self.y2 = self.y2[1:]

            t = self.ser.readline()
            dt = self.ser.readline()
            self.x2.append(int(t)/1000)          # Add a new time value in seconds
            self.y2.append((int(dt) - self.offsetB) / 2**23 / self.GainB / K_factor * 4 * 10**6)       # Add new
             data in um/m

            self.data_chanelB.append(int(dt))
            self.time_chanelB.append(self.x2[-1])

            self.data_line_2.setData(self.x2, self.y2)  # Update the data.


        if (self.change_G == 1):
            print("Cambio ganancia")
            #self.offsetA = self.offsetA/self.GainA
            self.GainA = self.nextGainA
```

```
                #self.offsetA = self.offsetA*self.GainA
                self.printText()
                self.change_G = 0
                #self.grafics_conf()
                self.GainChange.append([len(self.data_chanelA),self.GainA])

        if (header == gain_header):
            self.change_G = int(self.ser.readline())
            if (self.change_G==0):
                self.change_G = 1
                self.nextGainA = int(self.ser.readline())
            else:
                print("Cambio ganancia")
                #self.offsetA = self.offsetA/self.GainA
                self.GainA = int(self.ser.readline())
                #self.offsetA = self.offsetA*self.GainA
                self.printText()
                self.change_G = 0
                #self.grafics_conf()
                self.GainChange.append([len(self.data_chanelA),self.GainA])


    def stopSampling(self):
        1

    def clearPlot(self):
        1
        # self.graphWidget.clear()

    def tareDevice(self):
        1

    def saveData(self):
        1




def main():
    app = QtWidgets.QApplication(sys.argv)
    main = MainWindow()
    main.show()
    sys.exit(app.exec_())

if __name__ == '__main__':
    main()
```

Code C.1 – *GUI code*

C.2 FW code for Arduino UNO

```

/*
#   _____                       _____       _______
#  / ____|                     / ____|   /\|__   __|
# | |  __ _ __ __ _ _ __   __ _| (___    /  \  | |
# | | |_ | '__/ _` | '_ \ / _` |\___ \  / /\ \ | |
# | |__| | | | (_| | | | | (_| |____) |/ ____ \| |
#  \_____|_|  \__,_|_| |_|\__,_|_____//_/    \_\_|
#
#
#
#  Version: 0.0
#  Autor  :  Sharif Alhusein Raie  -  shalhusein@gmail.com  - 03/08/2021
#  Advisor:  Prof. Andrés Roldán Aranda  - xx/xx/xxxx
#
*/
 /****************************************************************************************************************
  ***   This program is for programe the Embeded System for Monitoring Wood Made Infraestructures
  ***   It is based on the implementation of Small_PCB_test_V06 programme
  ***
  ***   V00 (03/08/2021):
  ***       In this version the Arduino recive parameters and commands from the PC and actuate acording to them
  ***       it reads data from ADC, encapsulate it and send by serial port
  ***       This version use a diferent method to encapsulate the data (explained below)
  ***
  ****************************************************************************************************************/

// #define _debugging_ 1

#include <Arduino.h>
#include <HX711.h>
#include "../lib/GranasatLogo/GranasatLogo.h"
#include <SPI.h>
#include <Wire.h>
#include "../lib/Adafruit_SSD1306/Adafruit_SSD1306.h"
#include "../lib/Adafruit_GFX/Adafruit_GFX.h"

#define HX711_mean_delay 0  // Este parámetro está copiado en la librería HX711.cpp
#define HX711_mean_times 2  // Not used in this version

//Pines E/S
#define pinLED_1      10      // Sin funcionalidad
#define pinLED_2      9       // Indica cambio de ganancia
#define pinLED_3      8       // Sin funcionalidad
#define PinSwitch_2   12      // Para medir tara
#define PinSwitch_3   11      // Sin funcionalidad
#define gain_A_128    128
#define gain_A_64     64
#define gain_B_32     32

//****** Define data packet structure ******

// Structure of the header

//_________Id_____________Value_
//                       |      |
//     Info              | 0x00 |
//     Channel A data    | 0x01 |
//     Channel B data    | 0x02 |
//     Setting parameters | 0x03 |
//     Command           | 0x04 |
//_______________________|______|

typedef struct {
  uint8_t id;      // Identifier
  uint16_t num;    // Packet number
  uint8_t len;     // length of the packet
} pack_Header;

// Structure of the data packet
typedef struct {
  pack_Header header; // Header
  uint8_t gain;       // gain
  long time;          // data time
  long measure;       // data measurement
} dataPack;

// Structure of the command packet
typedef struct {
  pack_Header header; // Header
  uint8_t message;     // Command
} messagePack;

```

```
//|_____Info message_______|_Value_|
//|                        |       |
//|     Recived (ACK)      | 0x0   |
//|     All OK and ready   | 0x01  |
//|     OLED connected     | 0x02  |
//|     OLED disconnected  | 0x03  |
//|                        | 0x04  |
//|________________________|_______|

//|___Command message_____|_Value_|
//|                        |       |
//|     Stop measuring     | 0x0   |
//|     Start measuring    | 0x01  |
//|     Restart data       | 0x02  |
//|     Tare               | 0x03  |
//|                        | 0x04  |
//|________________________|_______|

//|____Setting parameters____|_Value_|
//|                            |       |
//|  Single Channel            | 0x00  |
//|  Dual Channel              | 0x01  |
//|  Auto Gain selection OFF   | 0x02  |
//|  Auto Gain selection ON    | 0x03  |
//|                            | 0x04  |
//|____________________________|_______|

dataPack data;
messagePack message, received_msg;
// Pointers to send data
const char *data_ptr = (char*) &data;
const char *message_ptr = (char*) &message;
char *received_ptr = (char*) &received_msg;
#define rec_pack_len 5  // lenth of recived pack in bytes


//double data;
// long data;
byte nextGain;
long t_inicial;
// long t_muestra;
int gain_A = gain_A_128;          // initial gain for channel A
uint16_t n_pack=0;                // number of packets sent
bool dual_Channel_Measure = 0;    // single/dual channel configuration
bool autoGainSelect = 1;          // Automatic gain adjust configure
bool continue_meas = 0;           // To control start and stop measurement


// HX711 circuit wiring
const int LOADCELL_DOUT_PIN = 3;
const int LOADCELL_SCK_PIN = 4;

//Strain Gauge parameters
//const int K = 2;


// OLED Parameters definition
#define screenWidth 128    // ancho pantalla OLED
#define screenHeight 64    // alto pantalla OLED
#define I2C_address 0x3C   // I2C device address
bool OLED_ON = 0;          // Global variable to indicate the display conection/disconection

// Adafruit_SSD1306 class object
Adafruit_SSD1306 display(screenWidth, screenHeight, &Wire, -1);

// HX711 class object
HX711 scale;

// Function definition
// It decodes the recived packet, the implementation is at the end
void packDecoder(messagePack msg);

void setup() {

  // Pin Mode declarations
  pinMode(PinSwitch_2, INPUT_PULLUP);
  pinMode(PinSwitch_3, INPUT_PULLUP);
  pinMode(pinLED_1, OUTPUT);
  pinMode(pinLED_2, OUTPUT);
  pinMode(pinLED_3, OUTPUT);

  Serial.begin(9600);

#ifdef _debugging_
  Serial.println("Iniciando dispositivo");
```

```
#endif

  // Initialize OLED screen at 0x3C address
  if (display.begin(SSD1306_SWITCHCAPVCC, I2C_address)) {
    // OLED display is connected
    OLED_ON = 1;

#ifdef _debugging_
    Serial.println("OLED ON");
#endif

    // Info message -> OLED ON
    message.header.id = 0x00;
    message.header.num = 26543;
    message.message = 0x02;
    message.header.len = sizeof(message);

    // Send message byte to byte by serial port
    for (int i=0; i<message.header.len; i++){
      Serial.print(message_ptr[i]);
    }
    // Serial.print("\n");

    // Clear the buffer.
    display.clearDisplay();

    // Draw bitmap on the screen
    // display.drawBitmap(X0, Y0, Image-Data-Array, Image_Width, Image_Height, Color);
    display.drawBitmap(0, 0, Image.data, Image.width, Image.height, SSD1306_WHITE);

    // Show image in display
    display.display();
    delay(500);
  }
  else{
    // OLED display is disconnected
    OLED_ON = 0;

#ifdef _debugging_
    Serial.println("OLED OFF");
#endif

    // Info message
    message.header.id = 0x00;
    message.header.num = 0;
    message.message = 0x03;   // OLED OFF
    message.header.len = sizeof(message);

    // Send message byte to byte by serial port
    for (int i=0; i<message.header.len; i++){
      Serial.print(message_ptr[i]);
    }
    // Serial.print("\n");
  }

  // Initialize ADC library with data output pin, clock input pin and gain factor.
  // Channel selection is made by passing the appropriate gain:
  // - With a gain factor of 64 or 128, channel A is selected
  // - With a gain factor of 32, channel B is selected
  // By omitting the gain factor parameter, the library
  // default "128" (Channel A) is used here.
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN, gain_A_128);

#ifdef _debugging_
    Serial.println("Scale ON");
#endif

  // Wait until recive comands from GUI
  while (!Serial.available());

  // Read parameters from serial port
  while(Serial.available()){
    for (int i=0; i<rec_pack_len; i++){
      received_ptr[i] = Serial.read();
    }
    packDecoder(received_msg);
  }




  if (OLED_ON){
    // Clear the buffer.
    display.clearDisplay();
    // Set color and size for display text
```

```
    display.setTextColor(SSD1306_WHITE);
    display.setTextSize(1);
  }

  // Select first measure gain and channel
  if (dual_Channel_Measure){
    nextGain = gain_B_32;
  }
  else{
    nextGain = gain_A;
  }

  // First sample is invalid -> discard it. Set channel A or B for next sample
  scale.read_setting_next_gain(nextGain);

#ifdef _debugging_
    Serial.println("First read discarted");
#endif
  // Initilize time auxiliar variable
  t_inicial = millis();
}


void loop() {

  if (continue_meas){
      // Se envían los datos sin pasarlos a V con un rango de +-2^23. Se pasan a voltios en el programa python

    if (OLED_ON){
      // Mostrar datos del canal B
      display.clearDisplay();
      display.setCursor(0, 56);
      display.print("-CH B -> ");
    }

    if (dual_Channel_Measure){
      data.gain = nextGain;
      nextGain = gain_A;

      // Read from channel B and set channel A for next sample
      data.measure = scale.read_setting_next_gain(nextGain);
      data.time = millis() - t_inicial;
      data.header.id = 0x02;
      data.header.num = n_pack;

      n_pack++;

      // Send data byte by byte over serial interface
      for (int i=0; i<data.header.len; i++){
        Serial.print(data_ptr[i]);
      }
      // Serial.print("\n");


      if (OLED_ON){
        display.println(data.measure);
      }
    }
    else if (OLED_ON)
    {
      // if single mode is on display ---
      display.println(" ---");
    }


    data.gain = nextGain;

    // Gain/Channel selection
    if (dual_Channel_Measure){
      nextGain = gain_B_32;
    }
    else {
      nextGain = gain_A;
    }


    // Read data from channel A and set next sample channel
    data.measure = scale.read_setting_next_gain(nextGain);
    data.time = millis() - t_inicial;
    data.header.id = 0x01;  // Channel A id
    data.header.num = n_pack;
    data.header.len = sizeof(data); // Number of bytes ocupped by data variable

    n_pack++;

```

```
    for (int i=0; i<data.header.len; i++){
      Serial.print(data_ptr[i]);
    }
    // Serial.print("\n");

    if (OLED_ON){
      display.setCursor(0, 0);
      display.print("-Time(s) -> ");
      display.print(data.time/1000);
      display.setCursor(0, 28);
      display.print("-CH A -> ");
      display.println(data.measure);
      display.print("G = ");
      display.print(data.gain);
      display.display();

    }


    // if automatic gain selection for channel A is active
    if (autoGainSelect){
      // For aboid the saturation of the device, when data from chanel A arrives to 95% of the maximum the gain is
        changed to 64
      if(gain_A==128 && (data.measure>=8000000 || data.measure<=-8000000)){
        digitalWrite(pinLED_2,HIGH);

        gain_A = gain_A_64;

        delay(50);
        digitalWrite(pinLED_2,LOW);
      }

      // When data drops to 20% of the maximum range, the gain is changed to 128
      else if(gain_A==64 && data.measure<=3500000 && data.measure>=-2000000){
        digitalWrite(pinLED_2,HIGH);

        gain_A = gain_A_128;

        delay(60);
        digitalWrite(pinLED_2,LOW);
      }
    }
  }

  // Read parameters from serial port
  while(Serial.available()){
    for (int i=0; i<rec_pack_len; i++){
      received_ptr[i] = Serial.read();
    }
    packDecoder(received_msg);
  }


//       //Si se pulsa el PinSwitch_2 se realiza una TARA con el valor medido en ese momento -> cambiar a
  interrupcion
//       if(digitalRead(PinSwitch_2)==false)
//       {
//         scale.tare(50);
//         //Serial.print("Offset = ");
//         Serial.println(scale.get_offset());
//       }

}


//|____Setting parameters____|_Value_|
//|                          |       |
//|  Single Channel          | 0x00  |
//|  Dual Channel            | 0x01  |
//|  Auto Gain selection OFF | 0x02  |
//|  Auto Gain selection ON  | 0x03  |
//|                          | 0x04  |
//|__________________________|_______|

//|___Command message_____|_Value_|
//|                       |       |
//|    Stop measuring     | 0x00  |
//|    Start measuring    | 0x01  |
//|    Restart data       | 0x02  |
//|    Tare               | 0x03  |
//|                       | 0x04  |
//|_______________________|_______|

void packDecoder(messagePack msg){

```

```
    // If setting parameters message
    if (msg.header.id == 0x03){

      switch (msg.message)
      {
      case 0x00:
        dual_Channel_Measure = 0;
        break;

      case 0x01:
        dual_Channel_Measure = 1;
        break;

      case 0x02:
        autoGainSelect = 0;
        break;

      case 0x03:
        autoGainSelect = 1;
        break;

      default:
        break;
      }
    }
    else if (msg.header.id == 0x04)
    {
      // Comand
      switch (msg.message)
      {
      case 0x00:
        continue_meas = 0;
        break;

      case 0x01:
        continue_meas = 1;
        break;

      case 0x02:
        continue_meas = 0;
        n_pack = 0;
        break;

      case 0x03:
        // Tare
        break;

      default:
        break;
      }
    }

  }
```

Code C.2 – *FW code for Arduino*

Appendix D

Device 3D models

D.1 PCB 3D model

D.2 Full device 3D model