Three-dimensional mapping of an unknown environment based on 2D laser measurement

Edilson Santos Ferreira

Dissertation presented to the School of Technology and Management of Bragança to obtain the Master Degree in Industrial Engineering. Work developed under the Dual Degree Program between the Polytechnic Institute of Bragança (IPB) and the Federal Center for Technological Education of Minas Gerais (CEFET-MG).

Work oriented by:

Prof. PhD José Luís Sousa de Magalhães Lima

Prof. PhD Murillo Ferreira dos Santos

Prof. M.Sc. João Afonso Braun Neto

Bragança

2023

Three-dimensional mapping of an unknown environment based on 2D laser measurement

Edilson Santos Ferreira

Dissertation presented to the School of Technology and Management of Bragança to obtain the Master Degree in Industrial Engineering. Work developed under the Dual Degree Program between the Polytechnic Institute of Bragança (IPB) and the Federal Center for Technological Education of Minas Gerais (CEFET-MG).

Work oriented by:

Prof. PhD José Luís Sousa de Magalhães Lima

Prof. PhD Murillo Ferreira dos Santos

Prof. M.Sc. João Afonso Braun Neto

Bragança

2023

Dedication

To my parents Horaci and Valtanir, my sister Elaine and my love Elidiane.

Acknowledgement

A master's work is an arduous journey, full of uncertainties and many obstacles to be overcome. Although it is a lonely process for any researcher, it is enriched by the contributions of several people who are essential to help find the best path at each stage of the journey. I was only able to walk this path thanks to the support, energy and strength of several people, whom I especially thank:

To God, for life, strength, courage and inspiration to complete this work.

To my advisors, José Lima and Murillo Ferreira, for guidance and support along the way, and especially to João Braun, who was not only an advisor, but also a friend, providing invaluable help and sharing valuable advice on life.

To my parents, Horaci and Valtanir, and my sister Elaine, for their unconditional love, support and encouragement that helped me overcome the challenges and persist in my goal. You are my base. And also to Grandma Dejanira for her prayers and love.

To my friend and love Elidiane, who was by my side at all times, supporting and encouraging me throughout this journey. Her love, patience and understanding during the ups and downs were invaluable.

To CEFET-MG and IPB for the opportunity to participate in the double degree program and to CeDRI for all the structure available during my research.

To my friends Fernando, L. Schuenck and Jhonatan for their support, and to Ana, Bruno, Éric, Leo, Melo, Paloma, Vinícius and William, who made this exchange unique.

Finally, I would like to express my deep gratitude to all the people who in one way or another contributed to the completion of this work.

"Believe you can and you're halfway there."

(Theodore Roosevelt)

Abstract

3D mapping technology plays a key role in the field of robotics and autonomous navigation by enabling accurate perception and understanding of a robot's surrounding environment. However, traditional 3D mapping systems can be expensive and inaccessible, limiting their application in low-cost scenarios.

This work presents the development of a low-cost 3D mapping technology for trajectory planning using the A* algorithm. The main objective of this study is to provide a feasible and cost-effective solution for 3D mapping, enabling accurate trajectory planning in robotic environments.

The proposed approach combines a 2D LiDAR, a stepper motor, and the A* algorithm for trajectory planning. In addition, a circuit board for connecting and controlling the equipment and a 3D-printed support for attaching the LiDAR to the motor shaft are developed. The system is able to acquire data, perform the processing and generate a point cloud for use in trajectory planning with the A* algorithm.

The experimental results demonstrate the effectiveness and feasibility of the proposed technology, opening doors to low-cost robotics applications, such as small autonomous vehicles. Furthermore, this approach offers an affordable alternative for research and development in the field of 3D mapping and trajectory planning. Finally, possible improvements to be implemented in future work are highlighted.

Keywords: Path planning algorithm, A*, LiDAR, Low-cost 3D mapping.

Resumo

A tecnologia de mapeamento em 3D desempenha um papel fundamental no campo da robótica e da navegação autônoma, pois permite a percepção e a compreensão precisas do ambiente ao redor de um robô. Entretanto, os sistemas tradicionais de mapeamento em 3D podem ser caros e inacessíveis, limitando sua aplicação em cenários de baixo custo.

Este trabalho apresenta o desenvolvimento de uma tecnologia de mapeamento em 3D de baixo custo para o planejamento de trajetórias usando o algoritmo A*. O principal objetivo deste estudo é fornecer uma solução viável e econômica para o mapeamento em 3D, permitindo o planejamento preciso da trajetória em ambientes robóticos.

A abordagem proposta combina um LiDAR 2D, um motor de passo e o algoritmo A* para planejamento de trajetória. Além disso, foram desenvolvidos uma placa de circuito para conexão e controle do equipamento e um suporte impresso em 3D para fixar o LiDAR no eixo do motor. O sistema é capaz de adquirir dados, realizar o processamento e gerar uma nuvem de pontos para uso no planejamento de trajetória com o algoritmo A*.

Os resultados experimentais demonstram a eficácia e a viabilidade da tecnologia proposta, abrindo portas para aplicações de robótica de baixo custo, como pequenos veículos autônomos. Além disso, essa abordagem oferece uma alternativa acessível para pesquisa e desenvolvimento no campo de mapeamento 3D e planejamento de trajetória. Por fim, são destacados possíveis aprimoramentos a serem implementados em trabalhos futuros.

Palavras-chave: Algoritmo de planejamento de caminho, A*, LiDAR, mapeamento 3D de baixo custo.

Contents

List of Tables

List of Figures

Acronyms

1D One-dimensional

2D Two-dimensional

3D Three-dimensional

ADAS Advanced Driver-Assistance System

AGV Automated Guided Vehicle

bps bits per second

CLI Command-Line Interface

CNC Computer Numerical Control

DC Direct Current

DWA Dynamic Window Approach

ESP-IDF Espressif IoT Development Framework

GPIO General-Purpose Input/Output

GUI Graphical User Interface

I2C Inter-Integrated Circuit

IC Integrated Circuit

IDE Integrated Development Environment

IoT Internet of Things

LDS Laser Distance Sensor

LiDAR Light Detection And Ranging

LRF Laser Rangefinder

PCB Printed Circuit Board

PLA Polylactic Acid

PWM Pulse-Width Modulation

RADAR Radio Detection and Ranging

RF Radio Frequency

SLAM Simultaneous Localization and Mapping

SoC System-on-Chip

SONAR Sound Navigation and Ranging

ToF Time of Flight

TTL Transistor-Transistor Logic

UAV Unmanned Aerial Vehicle

UDP User Datagram Protocol

USB Universal Serial Bus

VS Code Visual Studio Code

Chapter 1

Introduction

Modern industry is a complex and dynamic sector that involves the large-scale production of a variety of products, from consumer goods to industrial components. For this production to take place efficiently and economically, it is crucial to have an effective organizational system, especially with regard to stock and the movement of raw materials and goods [1]. In this context, automation plays a key role, and robots known as Automated Guided Vehicles (AGVs) are key pieces of this puzzle [2].

Organizing stock and moving materials in a traditional industry can be a challenging task. Traditionally, forklift or trolley operators transport materials from one place to another, which can be inefficient and susceptible to human error. This is where AGVs come in, which are autonomous robots designed to transport materials efficiently and safely within an industrial facility [3].

AGVs are programmed to follow an optimized path, which takes into account the location of materials and space constraints. This can significantly increase production, as it eliminates the need for downtime caused by inappropriate movements or delays in the delivery of raw materials. In addition, AGVs can operate 24 hours a day, seven days a week, without the need for rest, which further increases the industry's efficiency [3].

Path planning for AGVs involves optimizing routes to ensure that they reach their destinations as quickly and efficiently as possible. This is done by taking into account factors such as the current location of materials, space availability, delivery priority and

collision minimization. Advanced algorithms are used to calculate optimal routes in real time, ensuring that AGVs can adapt to changing environmental conditions [4], [5].

In the context of Industry 4.0, 3D scanning of the environment plays a crucial role in path planning. It allows AGVs to have a more accurate understanding of the environment in which they operate, including the precise location of machines, stock and obstacles. This enables more precise and flexible path planning, reducing the risk of collisions and improving operational efficiency [6].

3D scanning of the environment is also a key part of the smart factory trend, where automation and connectivity are maximized. However, the cost associated with implementing 3D scanning systems can be a significant obstacle for many companies [7]. Therefore, the development of low-cost systems for 3D scanning is essential to ensure that companies of all sizes can benefit from this technology and remain competitive in the age of Industry 4.0, which reflects the approach presented in the Original Project Proposal (see Appendix A).

1.1 Objectives

The objective of the proposed work is to develop a low-cost autonomous system for 3D scanning of indoor environments, in order to carry out path planning from a point cloud generated by a 2D LiDAR sensor.

This being the general objective, the work will be divided into the following specific tasks:

- Study of the hardware components: ESP32, stepper motor and 2D LiDAR, followed by their integration;
- Design of the prototype's physical structure, to accommodate and interconnect the hardware components;
- Development and testing of the electronic circuit on a protoboard and design of the circuit board;

- Study of the C++ and Python programming languages and the necessary libraries to develop the control and data processing algorithm;
- Development of the Graphical User Interface (GUI);
- Carrying out tests and validating the system.

1.2 Work Structure

The present work is structured in five chapters, the first being an introduction to the theme in general, the presentation of the justification that motivated the development of this work and the objectives that are expected to be achieved.

The second chapter describes some fundamental concepts to help understand this work, as well as discussing the tools, devices and techniques used in the development of this work and presenting some related work.

The third chapter presents a detailed description of all the stages of the project development, which includes illustrative schemes of the system operation and the tasks performed by it.

In the fourth chapter, the methodology developed will be applied and the respective results will be presented and discussed.

Finally, in the fifth and last chapter, the final considerations are presented, containing the conclusions obtained from this work, as well as proposals for improvements and ideas for future work.

Chapter 2

State of Art and Study of Tools

This chapter contextualizes, through a literature review, the main concepts covered in this work. It also presents some of the projects in which the LiDAR sensor and 3D mapping have been applied. Additionally, the tools, devices and techniques used in the development of this work will be described, highlighting the main features and functionalities.

2.1 Theoretical Background

In this section, an overview of the relevant theoretical foundations related to the topic of the study will be presented. Fundamental concepts and theories will be covered, establishing a solid basis for understanding and developing the research in question.

2.1.1 LiDAR Technology

Light Detection And Ranging (LiDAR) is an optical scanning technology that uses laser pulses to measure the distance between a sensor and an object and create representations of the environment [8]–[11].

A typical LiDAR system operates by scanning its field of view with one or several laser beams, which are reflected by the environment back to the scanner. The returned signal is received by a photodetector, and the range is estimated from the sensor model

based on the difference between the transmitted and received signals [10].

2.1.2 Principle of Triangulation

One of the methodologies used to calculate the distance to the object of interest is measurement using the triangulation principle. To do this, the sensor emits a laser beam which is reflected and captured by a photosensitive sensor present in the equipment. The working principle of a triangulation-based ranging system is shown in Figure 2.1.

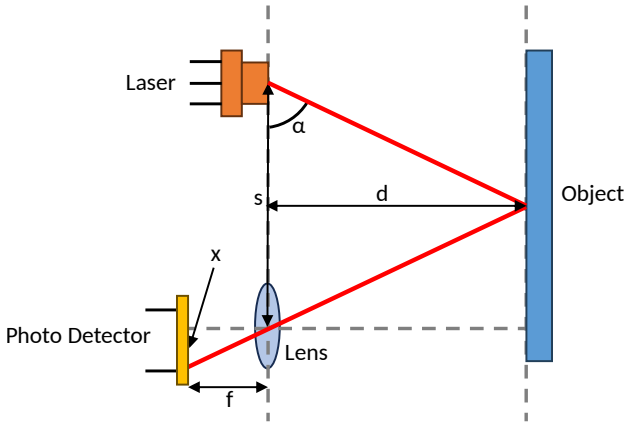


Figure 2.1: Working principle of a triangulation-based LiDAR sensor.
Source: Adapted from [12].

In this way, when the object approaches or moves away from the emitter, the reflected light beam touches the photosensitive sensor at another position, where the position of the object is estimated using geometric triangle theorem [12], [13]:

$$d = \frac{fs}{x} \tag{2.1}$$

where d is the distance to be measured, s is the distance between the lens and the laser, f is the focal length and x is the offset of the point on the lens.

2.1.3 Time of Flight

Another method used for distance calculations in LiDAR sensors is Time of Flight (ToF), which measures distance based on the time it takes for a signal to travel from a source to a target and vice versa [14].

An important feature of ToF systems is that their range resolution is based on the shortest measurable time interval and not on the distance to the object. As a result, measurement errors remain constant regardless of the proximity of the object, as measurement accuracy is determined by the sensor's ability to measure the difference between the moment the reflected signal is emitted and received. And this makes ToF the most abundant type of Laser Rangefinder (LRF) on the market due to its cost-effectiveness.

The basic principle of an ToF ranging system is shown in Figure 2.2.

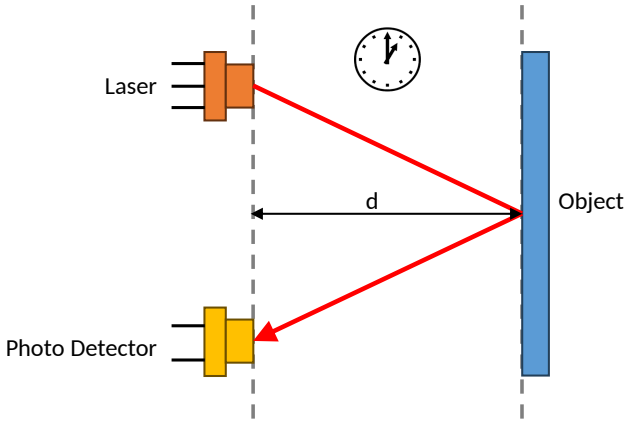


Figure 2.2: Working principle of a ToF LiDAR sensor.
Source: Adapted from [11].

In this method, the distance traveled by the signal round trip is given by Equation (2.2) [15]:

$$d = v \times \Delta t \tag{2.2}$$

where d is the distance traveled by the signal round trip, Δt is the measured time and v is the speed of the signal.

The signal used in ToF systems can be in the form of light frequencies, radio frequencies or sound energy, depending on the specific application, and these devices go by names such as LiDAR, RADAR and SONAR, respectively [16], [17].

In the case of LiDAR, which uses laser light pulses to measure distances, the speed of the signal is the speed of light. Therefore, the distance can be estimated by Equation 2.3.

$$d = \frac{c \times \Delta t}{2} \tag{2.3}$$

where d is the distance from the object found, Δt is the measured time and c is the speed of light, which is around $3.00 \times 10^8 m/s$ in vacuum.

2.1.3.1 LiDAR sensors as 1D, 2D or 3D variant

LiDAR sensors can operate in one, two or three dimensions, depending on their configuration and movement [18]. Initially, simple LiDAR sensors operate in a single dimension (1D), measuring the distance between the sensor and a target.

When the measuring beam is moved or rotated in a plane, information is obtained on both distance and angle, resulting in a 2D measurement [18]. These sensors are known as 2D laser scanners or 2D-LiDAR sensors.

For 3D measurements, LiDAR sensors can be tilted, providing information on distance, position in the x, y and z directions. In addition, systems with emitters or receivers moved at different horizontal angles on a scanning sensor can collect detailed information about space, and are referred to as multi-position measurement scanners [18].

2.1.4 Point Cloud

A point cloud is a collection of data points in a three-dimensional coordinate system that can be generated using a variety of methods, including 3D scanning technologies such as LiDAR, structured light scanning, 3D image capture or photogrammetry [19].

Each point in the point cloud has information about its spatial position (x, y and z

coordinates) and, in some cases, additional information such as color, intensity or texture [20], [21].

Point clouds are a rapidly growing technology that can generate virtual replicas of existing objects, offering numerous applications in education, entertainment, preservation, restoration, safety inspections, and autonomous vehicles. They can be used to scan real-world locations, create detailed digital models, and assist in safety inspections during construction projects. As an example, Figure 2.3 shows the reconstruction of the Colosseum in a point cloud.

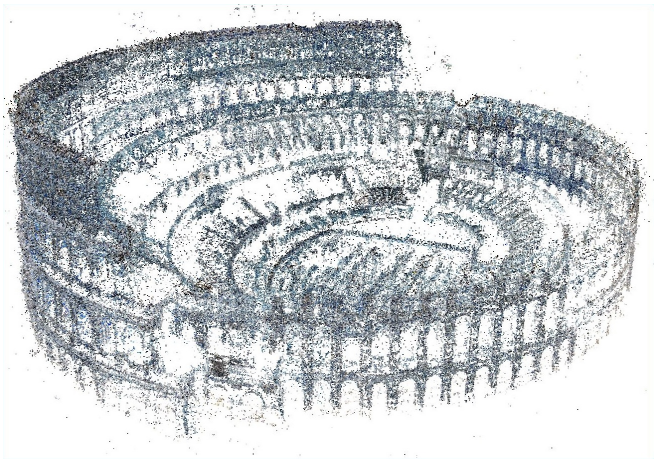


Figure 2.3: Point cloud reconstruction of the Colosseum.
Source: Adapted from [22]

Additionally, point clouds can be used to collect data for future self-driving cars, enabling the development of machine learning and deep learning algorithms, ultimately leading to the advancement of Advanced Driver-Assistance System (ADAS) technology.

2.2 Related Work

This section will present a general review of the work related to this research. Understanding previous advances and findings is essential to contextualize the contribution of this work.

Laser scanning technology is proving to be a promising option for a wide range of mapping and modeling applications, making it possible to very quickly acquire a huge amount of 3D data that offers a lot of information about the environment [23].

For this, many works are developed using 3D LiDAR or sensors with more than one sensing array to capture more information and obtain a 3D point cloud, such as the work developed in [24], which describes the construction of a point cloud database containing 389 stereo and optical flow image pairs, 39.2 km long stereo visual odometry sequences and more than 200,000 annotations of 3D objects captured in cluttered street scenes and their specificities in the city of Karlsruhe, Germany. The authors used their autonomous driving platform equipped with four high-resolution cameras, a laser scanner and a state-of-the-art localization system to create benchmarks for various tasks, including stereoscopy, optical flow, visual odometry/Simultaneous Localization and Mapping (SLAM) and 3D object detection. The laser scanner used was the Velodyne HDL-64E, a sensor with 64 lasers, a range of up to 120 meters, and which returns up to 2,200,000 points per second. Other works have been developed using the database cited, such as [25]–[30].

In [31], the authors address the limitations that still exist when it comes to acquiring high-precision 3D LiDARs, which remain expensive and difficult to access. The focus is directed towards the Robosense RS-LiDAR, a low-cost LiDAR model with sufficient supplies, compared to the Velodyne VLP-16. In addition, the study presents a series of evaluations aimed at analyzing the characteristics and performance of LiDAR sensors. This work examines various properties, such as drift effects, distance effects, color effects and sensor orientation effects, in the context of 3D perception. The comparison with Velodyne's LiDAR reveals the RS-LiDAR as a more economical and accessible alternative to the VLP-16, while maintaining similar efficiency. However, despite being a cheaper model compared to the VLP-16, the HDL-64E used in [24] and other top-of-the-line models, it is still not feasible to use it in low-cost applications.

The use of 3D sensors has advantages that go beyond adding a dimension in the aspect of the amount of information that is captured. However, for the moment, 3D LiDARs are much more expensive than two-dimensional sensors and become impractical for low-cost

applications. To get around this problem, there is an effort to transform a 2D LiDAR into a three-dimensional sensor.

Thus, the authors in [32] discuss the possibility of creating three-dimensional maps of the environment using a commercial 2D LiDAR in motion, based on the data from the 2D LiDAR and its movements. Compared to a commercial 3D LiDAR, using a moving 2D LiDAR proves to be a more cost-effective option. However, a number of challenges need to be overcome if the performance of the moving 2D LiDAR is to be optimized. To address these challenges, it is necessary to address the estimation of 2D LiDAR movements and the identification and removal of moving objects in the environment at lower scanning frequencies. Furthermore, how to deal with moving objects in a dynamic environment using a moving 2D LiDAR still represents an unsolved challenge in previous research.

In [33] a mechanism was developed to be mounted on the rear of a tractor that allows the attachment of a 2D LiDAR model SICK LMS200 oriented to the side of the vehicle. This mechanism also allows the vertical linear movement of the sensor with a known constant speed. Although the sensor used is 2D, i.e., it obtains a cloud of points corresponding to a plane or section of the object of interest, the linear movement mechanism adds one more degree of freedom and allows the registration of measurement results corresponding to different planes or cross-sections of the object, generating a 3D point cloud.

One way to improve mapping with LiDAR is to combine it with other sensors, such as the work developed by [34], proposes an approach for navigating mobile robots in complex environments by combining measurements from a Bumblebee stereo vision camera and SICK LD-OEM 2D LiDAR sensor mounted on an iRobot's Packbot. The fusion of this information is used to dynamically plan the robot's route and navigation in congested environments.

Autonomous navigation requires an accurate and robust mapping and localization solution. In these areas, several technologies are well studied and improvements are regularly proposed in the literature, such as SLAM which is widely used in various applications,

including mobile robotics, autonomous cars, Unmanned Aerial Vehicles (UAVs) and autonomous underwater vehicles. However, the techniques used have undergone little change over the years, as depicted in [35].

The authors of [36] study mapping and path planning for mobile robots in the context of the RoboCup competition for indoor rescue. Three SLAM algorithms were applied (GMapping, Hector-SLAM and Cartographer) and the data was obtained using an RGB-D camera, to obtain an RGB image and a depth image; and LiDAR to obtain a 2D point cloud. Path planning was done by combining the A* algorithm for global path planning and the Dynamic Window Approach (DWA) algorithm for local path planning.

Another application of path planning with LiDAR is presented in [37], where a 3D path planning method for bridge inspection using UAVs is proposed, integrating a Genetic Algorithm and the A* algorithm to solve the Traveling Salesman Problem, taking into account the identification of possible defects on the bridge surface, such as cracks.

The project developed in [38] was a precursor to the present work, and proposes the simulation of a 3D scanning system to estimate the volume of objects in indoor spaces, with useful applications for inventory monitoring.

2.3 Used Tools

In this section, a description of the software used in the development of the system is presented. The choice of these software was based on previous knowledge about them, as well as their ease of use. In addition, the fact that they are free tools or have free versions for students was considered.

2.3.1 Autodesk Inventor

Autodesk Inventor is a powerful 3D parametric modeling and engineering design software developed by Autodesk [39]. Its logo is shown in Figure 2.4.

Figure 2.4: Autodesk Inventor Logo.

Source: [39].

Widely used by engineers, designers and industry professionals, Inventor offers a comprehensive approach to product development from concept to production. With advanced capabilities for creating and editing parts, assemblies and technical drawings, the software allows users to visualize and simulate projects in 3D, optimizing the design process and ensuring greater accuracy and efficiency [40]. In addition, Inventor's integration with other Autodesk tools, such as AutoCAD, enables effective collaboration between multidisciplinary teams [39].

The screenshot in Figure 2.5 shows the Autodesk Inventor 2023 workspace.

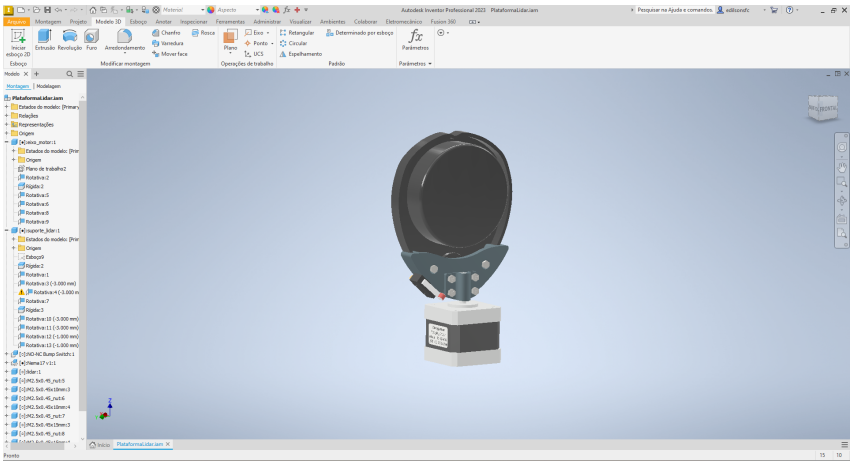


Figure 2.5: Autodesk Inventor 2023 workspace.

2.3.2 Autodesk Eagle

Autodesk Eagle is a Printed Circuit Board (PCB) design software widely used by engineers and designers to create circuit board schematics and layouts [41]. It offers a variety

of advanced tools that allow users to schematic electronic circuits, define components, route tracks and perform signal analysis [42]. Its logo is shown in Figure 2.6.

Figure 2.6: Autodesk Eagle Logo.

Source: [41].

With an intuitive interface, Eagle offers the ability to check design rules, such as minimum distances between tracks and layout constraints, to ensure the functionality and reliability of the circuits created. It's able to generate production-ready files and its integration with PCB fabrication tools make it a popular choice for electronic projects, from simple devices to more complex systems [43].

The screenshot in Figure 2.7 shows the Autodesk Eagle workspace: on the left, the schematic circuit editing window and on the right the board editing window.

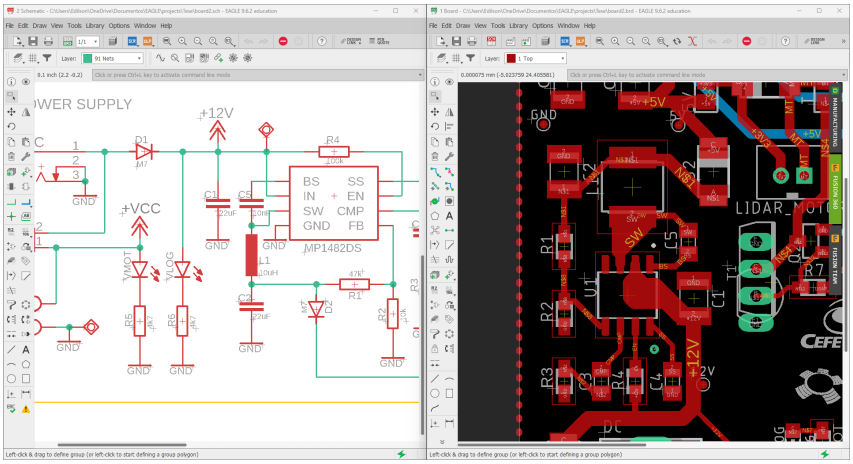


Figure 2.7: Autodesk Eagle workspace.

2.3.3 Visual Studio Code

Visual Studio Code (VS Code) [44] is a free, open-source, cross-platform code editor developed by Microsoft. It is designed to be lightweight and customizable, with support for a wide range of programming languages and extensions. Its logo is shown in Figure 2.8.

Figure 2.8: Visual Studio Code Logo.
Source: [44].

According to [44], VS Code provides features such as syntax highlighting, code completion, debugging, version control, and integrated terminal, among others. It also includes a wide range of extensions, which can add functionality such as language support, code snippets, themes, and integration with other tools and services.

One of the key benefits of VS Code is its cross-platform support, which allows developers to use the same tool across different operating systems, such as Windows, macOS, and Linux. It also has a large and active community of developers, who contribute to the development of the editor and its extensions [44].

The screenshot in Figure 2.9 shows the Visual Studio Code workspace.

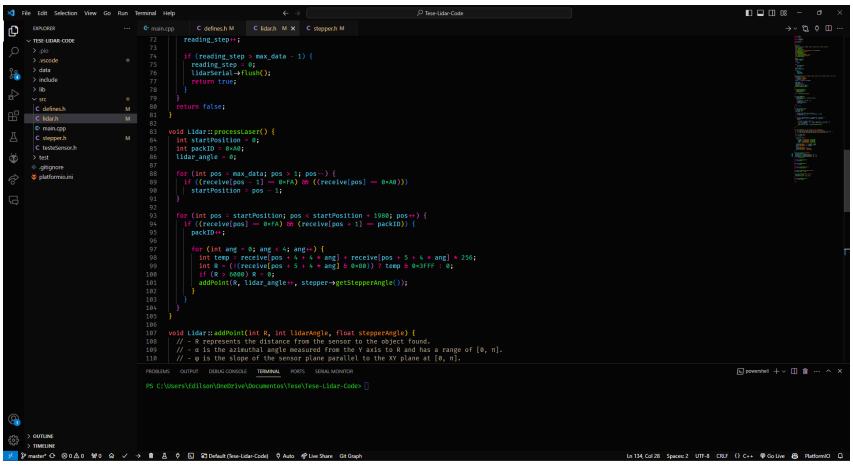


Figure 2.9: Visual Studio Code workspace.

2.3.4 PlatformIO

PlatformIO [45] is an open source ecosystem for Internet of Things (IoT) development and provides a unified development environment for a wide range of microcontroller platforms, including Arduino, ESP8266, ESP32, STMicroelectronics and many others. Its logo is shown in Figure 2.10.

Figure 2.10: PlatformIO Logo.

Source: [45].

According to [45], PlatformIO includes a cross-platform Integrated Development Environment (IDE) based on VS Code, as well as a Command-Line Interface (CLI) and a library manager, and also supports debugging, testing and firmware updates, among other features.

One of the key benefits of PlatformIO is its support for a wide range of microcontroller platforms, which makes it easier for developers to switch between different platforms and maintain a consistent development environment. PlatformIO also provides a unified library manager, which makes it easier to manage dependencies and reuse code across different projects.

PlatformIO is licensed under the permissive Apache 2.0 license and is developed by PlatformIO Labs, a company focused on providing tools and services for embedded systems development [45].

The screenshot in Figure 2.11 shows the PlatformIO extension in the Visual Studio Code workspace.

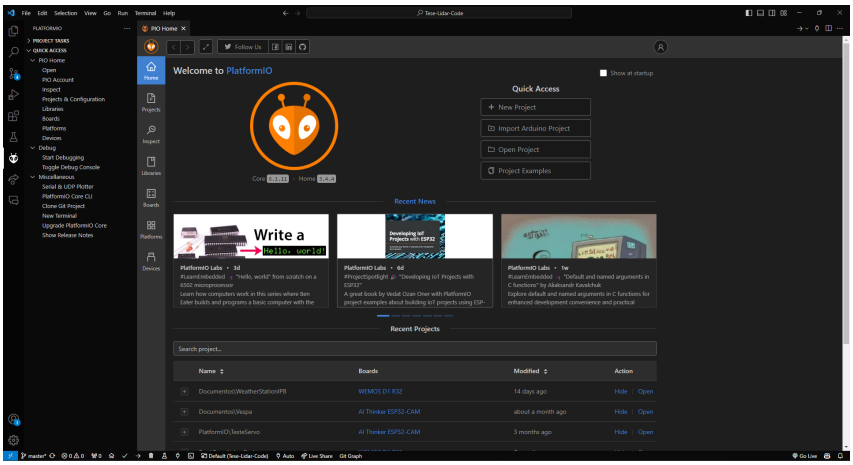


Figure 2.11: PlatformIO extension in the VS Code workspace.

2.4 Equipment and Components Used

In this section, a description of the system hardware is provided, consisting of the laser sensor, the drive motor and the microcontroller. These components play a key role in measurement and data acquisition.

2.4.1 ESP32

ESP32 is a low-cost, low-power System-on-Chip (SoC) microcontroller designed for IoT applications which is developed by Espressif Systems, a Chinese company that specializes in IoT solutions [46].

The Figure 2.12 shows an ESP32-WROOM module, that contains the ESP32 SoC, flash memory and a PCB antenna which provides outstanding Radio Frequency (RF) performance in space-constrained applications.

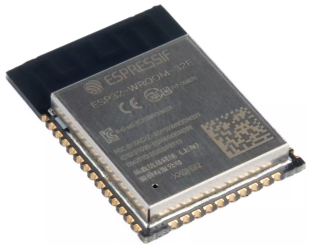


Figure 2.12: ESP32 WROOM-32E Module.
Source: [46].

In [47] and [48] are shown the technical specification of the ESP32 WROOM-32E Module, which include a 32-bit dual-core Tensilica Xtensa LX6 microprocessor running at up to 240MHz. Some of its technical specifications that make it suitable for this project are presented in Table 2.1.

Parameter	Value
Processor	2-Core Tensilica Xtensa LX6, 32-bit
Frequency	240MHz
SRAM	520kB
Flash	4MB
Wi-Fi	2.4GHz (802.11 b/g/n)
Bluetooth	v4.2 BR/ERD + BLE

Table 2.1: Some technical specification of the ESP32 WROOM-32E Module. Source: Adapted from [48].

ESP32 is supported by a range of development tools, including the Arduino IDE, PlatformIO, and the ESP-IDF, which is Espressif's official development framework for the ESP32 and ESP8266 microcontrollers.

2.4.2 Neato XV-11 Laser Scanner

The Neato XV-11 laser scanner, shown in Figure 2.13, is a small, low-cost, low-power LRF that is usually taken from Neato XV vacuum robots and is what started this whole project.

Figure 2.13: The Neato XV-11 laser scanner.

The actual name of the sensor is Piccolo Laser Distance Sensor (LDS), as can be seen

on the silkscreen of the sensor PCB (shown in Figure 2.14), but many different names are used to refer to it, such as Neato LDS, Neato LiDAR, XV-11 LiDAR, XV-11 sensor, among others.

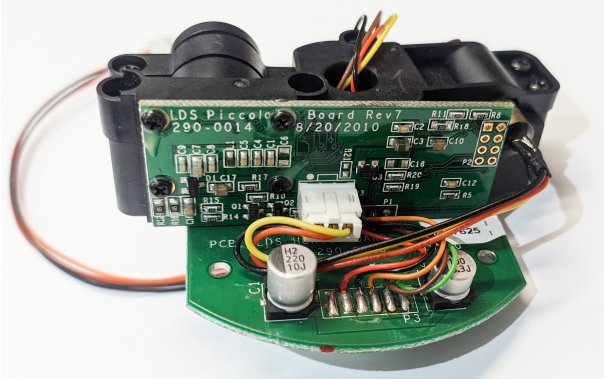


Figure 2.14: The Piccolo LDS PCB

The sensor communicates via a serial connection at a rate of 115200 bits per second (bps), transmitting data at a frequency of about 5 Hz [49]. A full rotation provides a measurement range of 360°, with an angular resolution of 1° and a distance range from 0.2 m to 6 m [50]. Some of its technical specifications are presented in Table 2.2.

Parameter	**Value**
Sensor	Piccolo LDS
Supply Voltage	5V DC
Interface	3.3V UART (8N1 @ 115200 bps)
Sampling Rate	$\sim$ 5 Hz
Scan Rate	200 $\sim$ 300 rpm
Distance Range	200 $\sim$ 6000 mm
Angular Range	360°
Angular Resolution	1°

Table 2.2: Some technical specification of the Neato XV-11 laser scanner. Source: Adapted from [49], [50].

In Figure 2.13 you can see the 2 connectors of the sensor. The first, from left to right, is a 4-pin connector responsible for the power supply and communication of the LiDAR sensor. The second is a 2-pin connector responsible for the power supply of the motor. The pinout of the 2 connectors is presented in Table 2.3.

Connector	Color	Function
LDS 4-pin	Red	5V DC
	Brown	RX
	Orange	TX
	Black	GND
Motor 2-pin	Red	VCC
	Black	GND

Table 2.3: Pinout of the Neato XV-11 laser scanner.
Source: Adapted from [49].

2.4.3 Stepper Motor

A stepper motor is a brushless DC motor whose rotor rotates in discrete angular increments when its stator windings are energised in a programmed manner [51]. Rotation occurs because of magnetic interaction between rotor poles and poles of the sequentially energised stator windings. The rotor has no electrical windings, but has salient and/or magnetised poles [52]. These motors are recognized for their positioning and control accuracy, being used in applications such as 3D printers, Computer Numerical Control (CNC) machines and industrial robots, where controlled and precise movements are essential [53].

The stepper motor used in this work is a Wantai NEMA 17 model 42BYGHW811, shown in Figure 2.15. According to [54], it's a bipolar stepper motor with a torque of 280 g.cm, 2.5A of current per phase, weight of 0.34kg and step angle of 1.8 degrees, which results in 200 pulses per revolution.

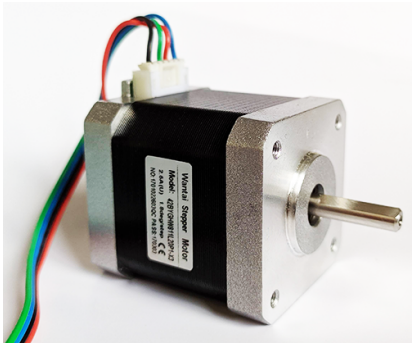


Figure 2.15: The Wantai NEMA 17 stepper motor model 42BYGHW811.

2.4.3.1 DRV8825 Stepper Motor Driver

The stepper motor needs to be controlled by the microcontroller, but it demands a higher current than the microcontroller General-Purpose Input/Outputs (GPIOs) pins can supply. To solve this issue, an intermediate circuit such as the DRV8825 is used. The microcontroller sends control signals to the DRV8825, indicating the direction and steps that the stepper motor should perform, and the DRV8825 translates these signals into suitable currents for the stepper motor coils. This allows the stepper motor to perform precise movements according to the instructions received from the microcontroller.

The DRV8825 is used in various applications such as 3D printers, CNC and robotics. It has features such as micro-step adjustment, overcurrent and thermal protection [55]. Some of its technical specifications are presented in Table 2.4.

Parameter	Value
Supply Voltage	$8.2 \sim 42$V DC
Output Current	Up to 2.5A
Min Pulse Duration	1.9 μs
Max Step Frequency	250 kHz
Microstepping	$1 \sim 1/32$
Protection	Overcurrent, Undervoltage, Thermal

Table 2.4: Some technical specification of the DRV8825 Stepper Motor Driver. Source: Adapted from [55].

2.4.4 Buck Converter

According to [56], a buck DC-DC converter, also called a step-down regulator, is an electronic circuit that lowers the input DC voltage to a lower output level. It works through a controlled switch, typically a transistor, rapidly switching between on and off. An inductor stores energy when the transistor is on, transferring it to the output when the transistor is turned off. Buck converters are widely employed in power supplies, solar energy and portable devices for efficient voltage regulation.

In this work, a buck converter was implemented using the MP1482 IC [57] as a controlled switch. Some of its technical specifications are presented in Table 2.5.

Parameter	Value
Input Voltage	4.75 ∼ 18V DC
Output Voltage	0.923 ∼ 15V DC
Output Current	2A
Efficiency	up to 93%
Frequency	340 kHz
Protection	Overcurrent, Undervoltage

Table 2.5: Some technical specification of the MP1482 Buck Converter. Source: Adapted from [57].

2.5 Methods and Techniques Used

In this section the methods and techniques used in data processing will be described. They are of great importance for the accurate functioning of the path planning algorithm.

2.5.1 Coordinate System Conversion

The conversion from spherical to Cartesian coordinates is performed using the system of Equations (2.4) [58].

$$\begin{cases} x = \rho\, sin(\phi)\, cos(\theta) \\ y = \rho\, sin(\phi)\, sin(\theta) \\ z = \rho\, cos(\phi) \end{cases} \tag{2.4}$$

As shown in Figure 2.16, ρ represents the distance, from the origin, of each point in the coordinate system. θ is the angle formed between the positive semi-axis of X and r, which is the projection of ρ on the XY plane and ϕ is the angle between the positive semi-axis of Z and the vector formed by the distance of ρ.

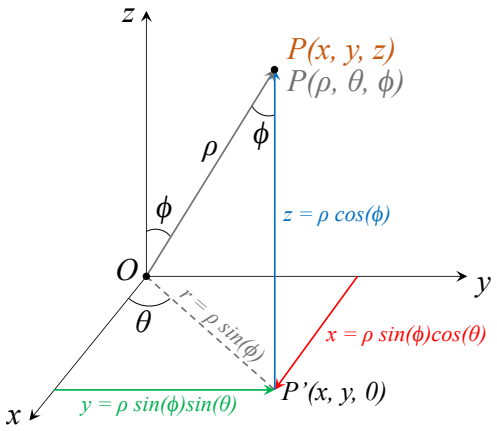


Figure 2.16: Coordinates system.

Source: Adapted from [58].

2.5.2 Morphological Transformations

Morphology, in the context of image processing, is a technique that focuses on the shape and form of objects in an image and uses a structuring element to compare the pixels of the input image with their neighbors and generate an output image of the same size [59]. By adjusting the size and shape of the structuring element, it is possible to create morphological operations that are sensitive to specific shapes in the image.

The fundamental operators of mathematical morphology are erosion and dilation, and all other morphological operators can be expressed in terms of these two [60].

2.5.2.1 Erosion

Erosion is an operation that shrinks or thins the objects in a binary image [61]. It works by going through the image with the structuring element and checking whether all the pixels underneath it are marked as object (value 1). If all the pixels are marked as object, then the corresponding pixel in the output image remains as object. Otherwise, the pixel in the output image is marked as background (value 0) [59].

In essence, the erosion operation removes the edges of objects in the image by shrinking them. It is often used as a linear filter, where objects smaller than the structuring element

are filtered out, thus allowing the removal of small noises or the separation of objects that are too close together [61].

2.5.2.2 Dilation

Dilation has the opposite effect to erosion. While erosion reduces the size of objects in the image, dilation increases the size of objects and fills in small voids or gaps in them [61]. It works by traversing the image with the structuring element and checking whether at least one pixel under the structuring element in the input image is marked as an object (value 1) and marks the corresponding pixel in the output image as an object. Otherwise, the pixel in the output image remains as background (value 0) [59].

In essence, the dilation operation expands objects in the image, increasing the size of their edges and filling in small gaps within them [61].

2.5.2.3 Opening and Closing

Opening and closing operations in morphological transformations are combinations of erosion and dilation operations, where opening is an erosion followed by dilation, while closing is a dilation followed by erosion [61].

These operations are used to remove noise and small objects from an image: opening removes small objects and smooths the edges of larger objects, while closing fills in small holes and gaps in larger objects, and both operations use a structuring element to determine the size and shape of the objects to be removed or filled [59].

2.5.3 A* Search Algorithm

The A* algorithm [62] is a heuristic search algorithm [63] that finds the shortest path in graphs or grids and is probably the most popular path planning algorithm in artificial intelligence and widely used in various applications, such as route planning in maps, robotics and video games [64]–[66].

This algorithm continuously expands the adjacent nodes from the start point until

it reaches the target point, thus searching for a path around the obstacle from the start point to the target point [67]. It uses two lists, *open* and *closed*, which will store the nodes to be explored and the nodes that have already been explored, respectively. The cost of each node in the *open* list is evaluated by the function shown in Equation (2.5) [62]

$$f(n) = g(n) + h(n) \tag{2.5}$$

where $g(n)$ is the actual cost from the starting node to the node n and $h(n)$ is the estimate of the cost of optimal path from node n to the target node, which depends on the heuristic information of the problem areas [66].

According to [62], the functioning of the A* Search Algorithm can be defined as follows:

1. Mark the starting node open and calculate the f cost;
2. Select the open node n whose f cost is smallest. Resolve ties arbitrarily, but always in favor of any node;
3. If n is the target node, mark n closed and terminate the algorithm;
4. Otherwise, mark n closed and apply the successor operator to n. Calculate f for each successor of n and mark as open each successor not already marked closed. Remark as open any closed node n_i which is a successor of n and for which $f(n_i)$ is smaller now than it was when n_i was marked closed. Go to Step 2.

If the open list becomes empty before the goal node is reached, then there is no path from the start node to the goal node, and the algorithm terminates [62]. Once the target node is reached, the shortest path can be reconstructed by following the parent pointers from the target node back to the start node.

Chapter 3

Development

This chapter presents the development of the proposed work, detailing the design of the physical structure of the system, the electronic circuitry, the user interface and the methods used for data processing and path planning.

3.1 Acquisition System

The indoor space measurement data were collected through an automated system, shown in Figure 3.1, which is consisted of the LiDAR, a stepper motor and the ESP32.

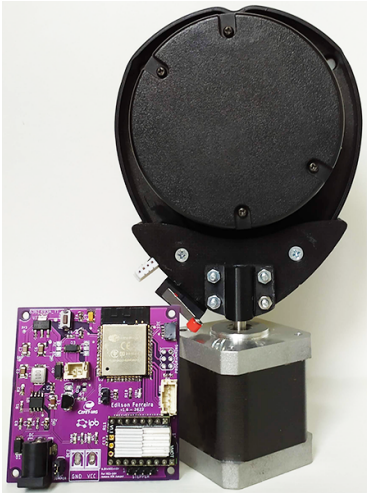


Figure 3.1: The acquisition system.

The raw data collected by the Neato XV-11 laser scanner were processed by ESP32 and sent via WiFi to a computer using User Datagram Protocol (UDP) datagrams. Data processing includes removing any outliers and converting spherical to Cartesian coordinates, which will be better described later.

3.1.1 The Structure

The hardware structure of the prototype, shown in Figure 3.1, was created using Autodesk Inventor [39] software. It consists of a mechanism that connects the LiDAR sensor to the stepper motor's shaft, ensuring stability and minimizing vibration during operation. The design ensures that the sensor remains aligned with the motor shaft, preventing any displacement in measurement when the stepper motor rotates.

The structure was printed with PLA filament on a 3D printer. For ease of printing and to avoid the use of brackets, the structure was divided into two parts and attached to each other with screws and nuts.

The first part, which accommodates the LiDAR, is shown in Figure 3.2. It was drawn from the outline of the LiDAR and all exact measurements were taken to ensure a perfect fit.

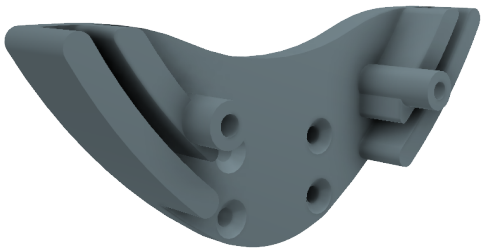


Figure 3.2: LiDAR Housing Component.

The second part is the one that fits the motor shaft and secures it in place, and is shown in Figure 3.3. It was designed so that its attachment to the motor shaft and the LiDAR housing component happened simultaneously with the tightening of the screws.

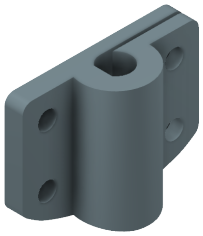


Figure 3.3: Motor Shaft Component.

The final assembly of the two parts is shown in Figure 3.4.

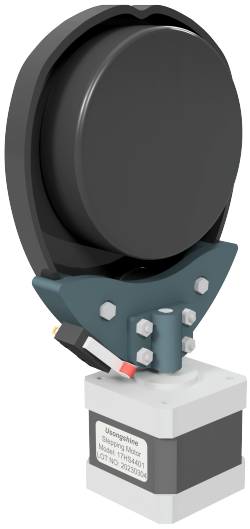


Figure 3.4: Hardware structure assembled.

3.1.2 The Electronic Circuit

The electronic circuit is an essential component of this project, interconnecting all the components. To ensure its reliability and proper functionality, it was assembled and tested on a protoboard, as shown in Figure 3.5.

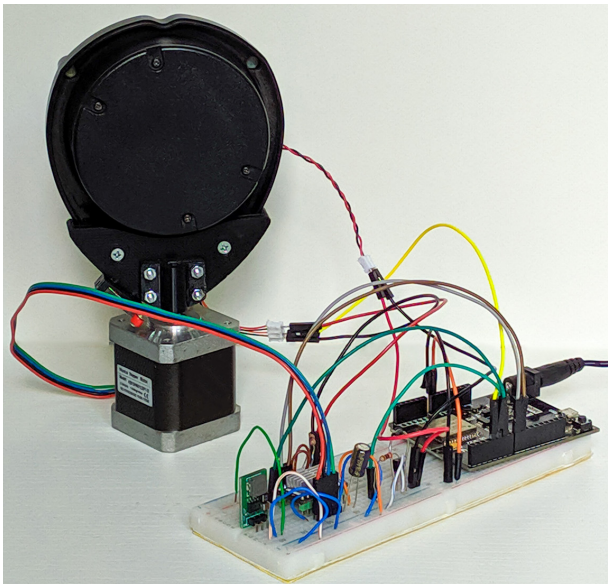


Figure 3.5: Circuit test on the protoboard.

After successful tests, the PCB was designed using Autodesk Eagle [41] software, as shown in Figure 3.6. This PCB has several essential components, which includes a voltage regulator circuit, a circuit responsible for controlling and communicating with the LiDAR, a circuit to drive the stepper motor and configure the microsteps, as well as the circuit with the microcontroller and its peripherals.

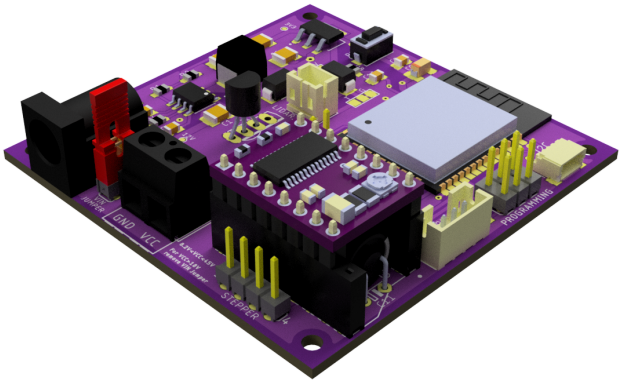


Figure 3.6: 3D perspective view of the developed PCB.

3.1.2.1 Voltage Regulator Circuit

The voltage regulator circuit, represented in Figure 3.7, is composed of a buck converter and a linear voltage regulator. It has two voltage inputs, allowing the stepper motor to be powered by a different voltage than the logic part, just by removing the VIN Jumper. Despite the functionality, in this study only one 12V supply was used to power both circuits.

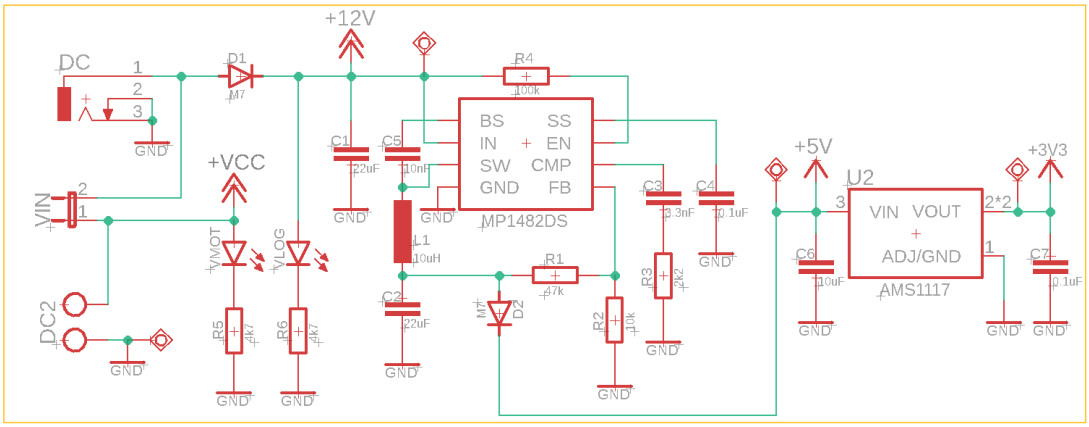


Figure 3.7: Power supply circuit.

The buck converter uses the MP1482DS IC and the developed circuit was based on the typical application example available in [57] and was configured to output 5V to power the LiDAR and other components. According to [57], the output voltage is set using a resistive divider from the output voltage to the feedback pin by the ratio:

$$V_{OUT} = 0.923 \times \frac{R_1 + R_2}{R_2} \tag{3.1}$$

Where V_{OUT} is the output voltage, and R_1 and R_2 are the adjustment resistors. Using the typical value for R_2 of $10k\Omega$, to get an output voltage of 5V, R_1 needs to be $44.1k\Omega$, but the most common commercial value to find is $47k\Omega$. Thus, this combination of R_1 and R_2 generates an output voltage around 5.26V. In practice, due to the tolerance of the resistors used, the values were $R1 = 46.5k\Omega$ and $R2 = 9.99k\Omega$. The output voltage obtained was $V_{OUT} = 5.21$V and the linear regulator reduces the final voltage to 3.3V.

3.1.2.2 Stepper Motor Driver Circuit

To drive the stepper motor, the breakout board for the DRV8825 stepper motor driver was used. According to [55], the DRV8825 allows controlling DC stepper motors using only 2 or 3 microcontroller's pins, and is capable of driving up to 2.5A of current per coil, with proper heat sinking. It also supports microstepping to control the motor in full step up to 1/32 step and has 3 pins (M0, M1 and M2) to configure them. Therefore, 3 pull-up jumpers located beneath the driver module were added, allowing the microsteps to be adjusted according to Table 3.1.

Microstep Resolution	**M2**	**M1**	**M0**
Full Step	Low	Low	Low
1/2 Step	Low	Low	High
1/4 Step	Low	High	Low
1/8 Step	Low	High	High
1/16 Step	High	Low	Low
1/32 Step	High	Low	High
1/32 Step	High	High	Low
1/32 Step	High	High	High

Table 3.1: Microstepping configuration jumpers for the DRV8825.

Source: Adapted from [55].

Although the PCB was designed to work with the DRV8825, it is also compatible with the A4988 and other similar boards. These drivers are small, cheap and widely used in CNCs and 3D printers, making them a good choice for this job.

The stepper motor control circuit, together with the jumpers for configuring the microsteps, are shown in Figure 3.8.

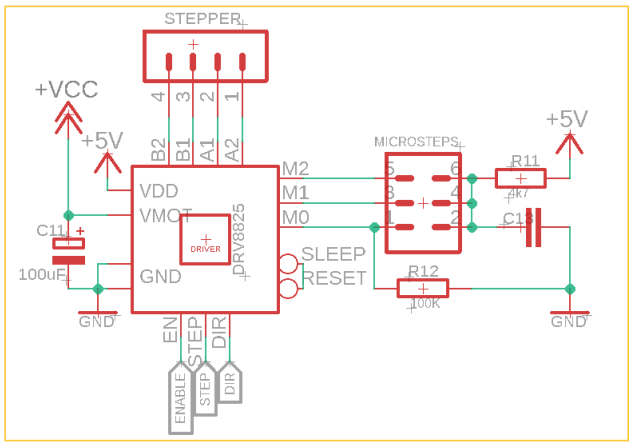


Figure 3.8: Stepper control circuit.

3.1.2.3 LiDAR Communication and Control Circuit

The Neato XV-11 LiDAR sensor is powered by 5V but communicates at 3.3V using UART, 8N1, with baud rate of 115200, and it must rotate counterclockwise at a speed between 200 and 300 rpm to measure correctly [50]. For this purpose, the PCB allows it to be driven by Pulse-Width Modulation (PWM) from a voltage of 5V, with closed-loop speed control, or it can be powered by continuous 3.3V in open-loop, which will produce a rotation rate of around 240 rpm. For this study, the second option was chosen. The interface circuit of the LiDAR is shown in Figure 3.9.

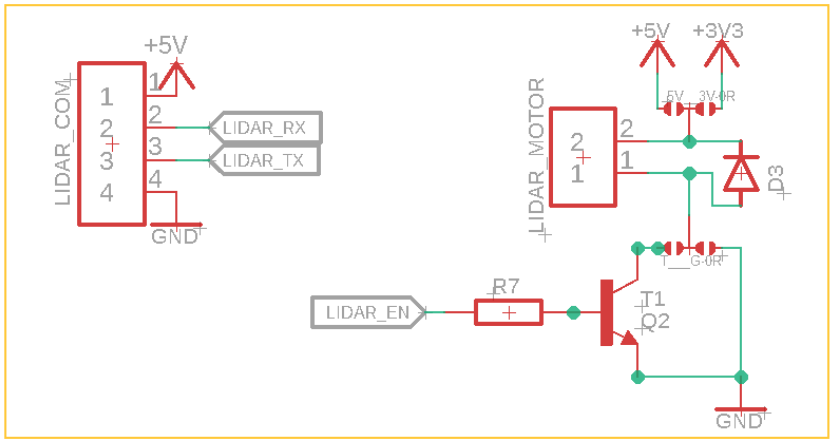


Figure 3.9: LiDAR connection circuit.

3.1.2.4 Microcontroller Circuit

Finally, the ESP32-WROOM-32E module [48] was used as the microcontroller due to its native Bluetooth and WiFi wireless connectivity, as well as meeting the necessary hardware requirements related to processing power and peripherals. A limit sensor was used to inform the microcontroller the home position of the LiDAR platform and a 4-pin connector in Adafruit's Stemma QT standard was added to allow future expansions using the I2C connection. The circuit with all components is shown in Figure 3.10. To upload the code into the microcontroller, an external USB to TTL Serial converter is required.

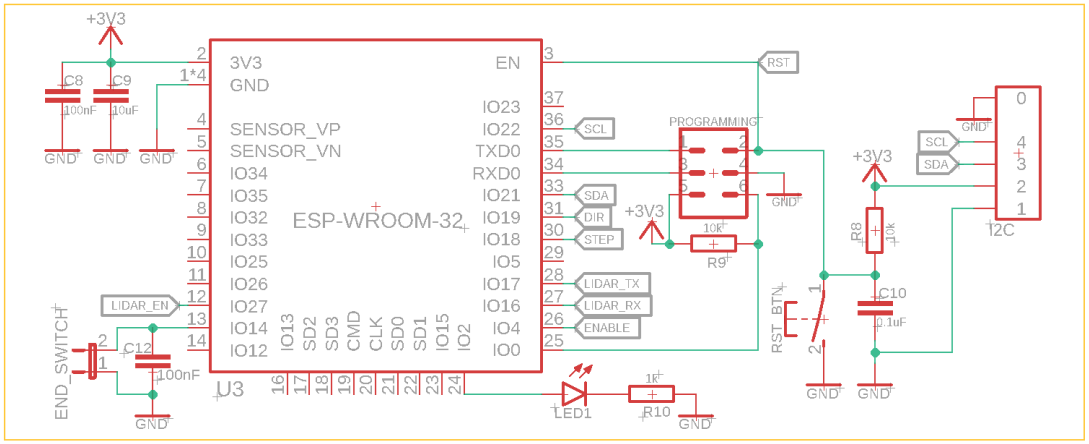


Figure 3.10: Microcontroller circuit.

3.1.2.5 PCB Design

With the circuit schematic completed, the design of the PCB was developed in two layers. Thus, the positioning of the components and connectors were idealized to allow a good final assembly. The final design of the PCB can be seen in Figure 3.11, and the 3D design in Figure 3.12. The PCB was manufactured by the company JLCPCB[1] and can be seen in Figure 3.13 and the PCB with all soldered components is shown in Figure 3.14.

[1] https://jlcpcb.com/

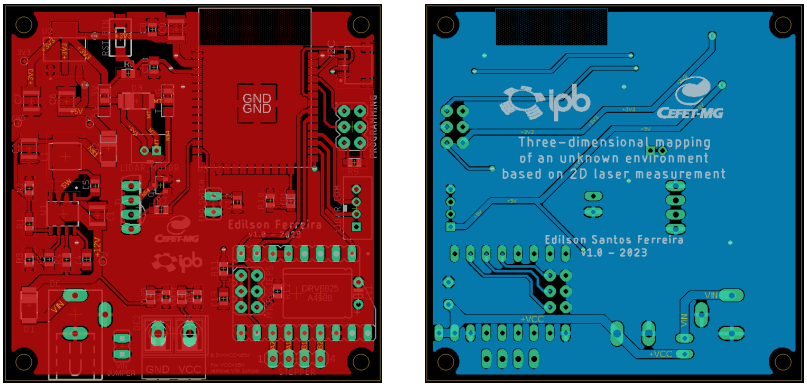


Figure 3.11: Top and bottom layers of the final PCB design.

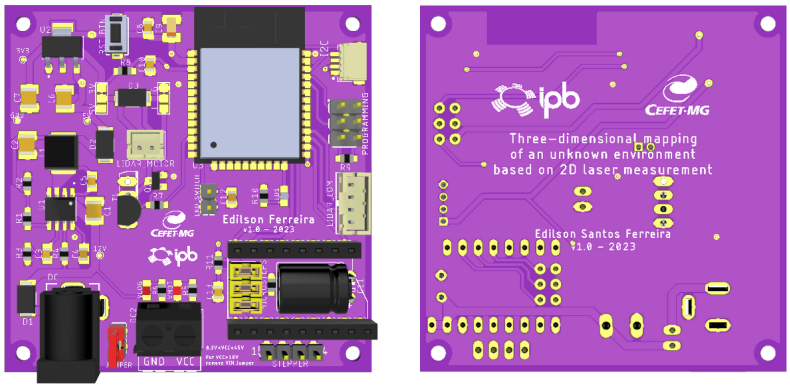


Figure 3.12: Top and bottom of the 3D visualization of the developed PCB.

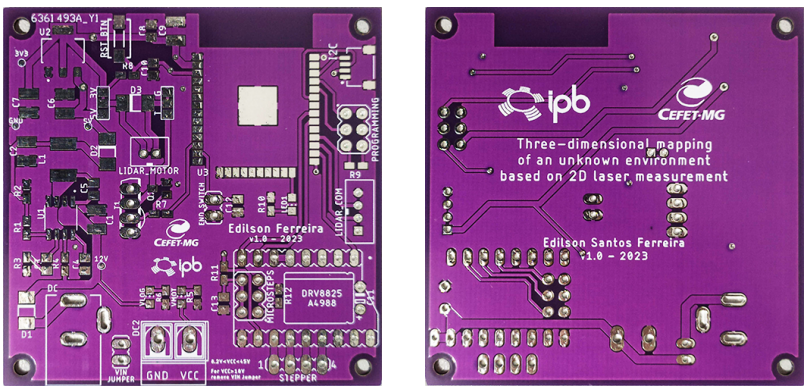


Figure 3.13: Top and bottom view of the manufactured PCB.

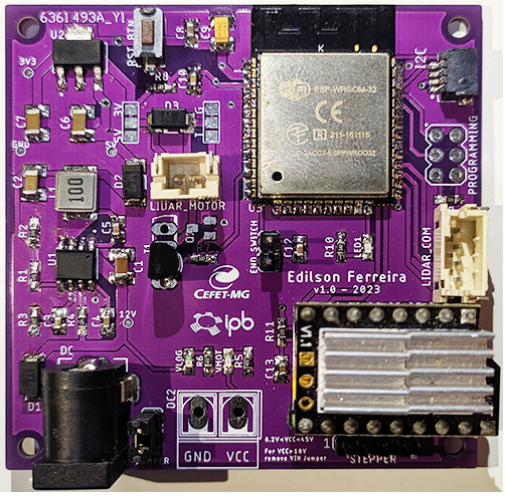


Figure 3.14: PCB with all components soldered.

The complete circuit and the full-size PCB with dimensions can be found in Appendix B.

3.2 Data Collection and Processing

The proposed system consists of a two-dimensional laser scanning subsystem that is driven by a stepper motor to achieve 3D acquisition capability. This system measures the distance along with the orientation of each beam of light in the two rotational degrees of freedom.

When the process starts, the stepper motor starts to rotate counterclockwise until it hits the limit switch. Then, the LiDAR device starts to scan the environment and, at each 360° rotation of the LiDAR, the stepper motor increases the angle in a chosen step. In this way, the device will scan new information at each rotation until the stepper motor reaches 180°. In order to understand the measuring process, we can refer to Figure 3.15, which presents a flowchart outlining the key steps involved.

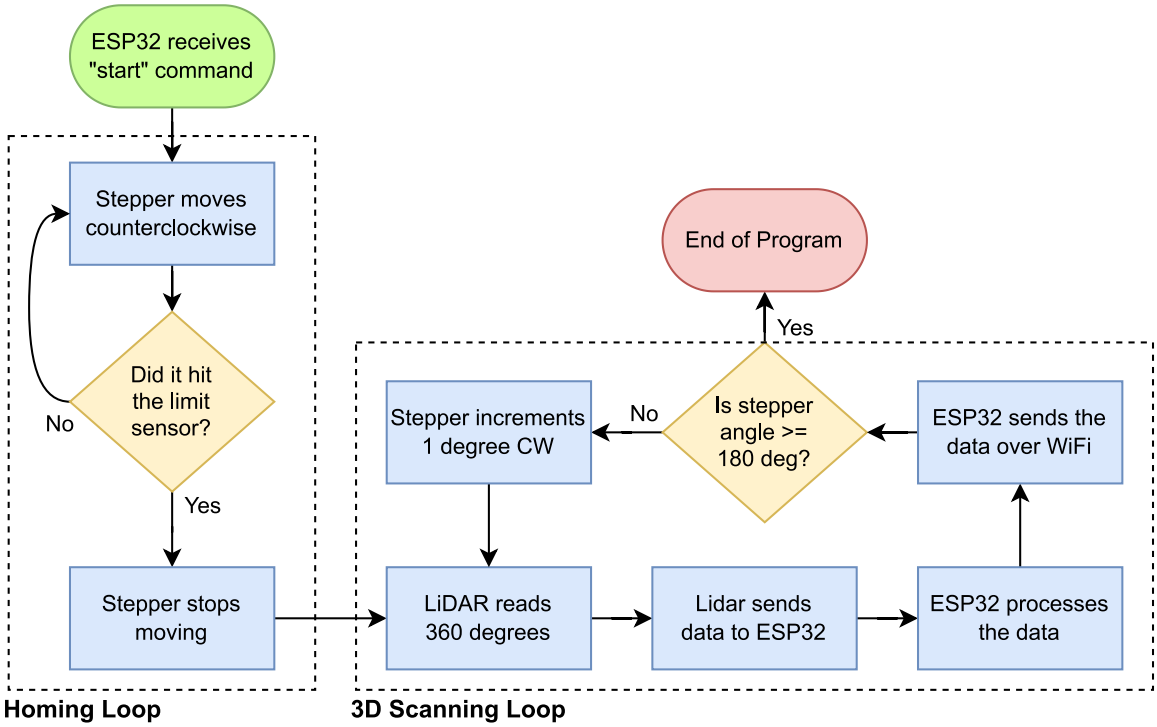


Figure 3.15: Flowchart to illustrate the program's operation.

The LiDAR sends the reading information to the microcontroller every complete revolution. It sends 90 packets with 4 consecutive readings each, for a total of 360 readings (1 reading per degree). The size of each packet is 22 bytes and its structure is as shown in Table 3.2.

`Start`	`Index`	`Speed`	`Data 0`	`Data 1`	`Data 2`	`Data 3`	`Checksum`
1 byte	1 byte	2 bytes	4 bytes	4 bytes	4 bytes	4 bytes	2 bytes

Table 3.2: Packet Structure of LiDAR Data Transmission.

Source: Adapted from [50]

where:

- `Start` is always 0xFA;

- `Index` is the index in the 90 packets, going from 0xA0 to 0xF9;
- `Speed` is a little-endian information. It represents in 64th of RPM (i.e. RPM in fixed point, with 6 bits for the decimal part);
- `Data 0` to `Data 3` are the four readings, organized as follows:
  - byte 0: distance (LSB);
  - byte 1: "invalid data" flag : "strength warning" flag : distance (MSB);
  - byte 2: signal strength (LSB);
  - byte 3: signal strength (MSB);

The microcontroller receives the LiDAR packets and stores them in an array. Then the data is processed from the 0xA0 index packet, which represents the readings from angles 0, 1, 2, and 3. Processing starts by checking whether the flag "invalid data" is set. If yes, the measurement is discarded; otherwise, the LSB of the distance is concatenated with the MSB of the distance and an AND operation (&) is performed with the value 0x3FFF, resulting in the distance value in millimeters. This step is repeated for the four measurement data for each of the 90 packages.

The azimuth and inclination angle is given by the packet index and the stepper motor angle, respectively. Thus, these three variables (distance, azimuth and inclination angle) are used to calculate the transformation from spherical to Cartesian coordinates, using the system of Equations (3.2). This relation can be proved through the variable change theorem in triple integrals available in [68].

$$\begin{cases} x = \rho\, sin(\phi)\, cos(\theta) \\ y = \rho\, sin(\phi)\, sin(\theta) \\ z = \rho\, cos(\phi) \end{cases} \tag{3.2}$$

where:

- ρ is the distance from the sensor to the object found.

- θ is the azimuth angle between the Y axis and the projection of ρ onto the XY plane and has a range [0, π].
- ϕ is the angle of the sensor plane from the Z axis with a range of [0, 2π].

In this scenario, the position of the sensor is considered to be the origin of the coordinate system for the calculations. The location of the scanned point is always described relative to that point in the system.

After the conversion, the information P(x, y, z), which represents the corresponding coordinates in millimeters of the real-world system, was initially compiled into a CSV file and stored in the ESP32's flash memory, which could be downloaded from a web interface accessed via the microcontroller's IP address. However, due to the limited size of this memory (4 MB) and the fact that it is also used to store the ESP32 program, it was barely possible to store a single 360° scan. Therefore, if a new scan were executed without downloading the data from the previous scan, the information would be lost. It was then decided to transmit the point data over WiFi via UDP datagrams to a computer, where it is compiled into a point cloud file.

This whole process, i.e. 360° scanning, processing and generating the point cloud file with almost 65,000 points, takes around 100 seconds to complete.

3.3 Data preparation for the A* Search Algorithm

To apply the A* search algorithm, it was necessary to address the following steps:

Initially, it was necessary to identify the floor plane in the point cloud. For this, a method was adopted that determines the value of Z with the highest occurrence of points, i.e. a flat surface with height Z. To obtain a more accurate answer, it is possible to inform an estimated distance between the sensor and the ground, allowing the method to find the Z values of the closest surfaces.

Then, the found floor plan is transformed into a matrix of size $M \times N$, where each cell represents a square of 1 centimeter side in the real world. Then, the whole floor plan

is traversed and the matrix is filled in the respective coordinate. The values used are 1 if there is any point at that particular coordinate and 0 otherwise.

After filling the matrix, two morphological transformations are performed using the OpenCV library [69]: Closing and Erosion. The closing operation was used to fill in the gaps between the measurement points in the point cloud, generating a ground plane where the algorithm can plot the route. The erosion operation was carried out in order to increase the size of the obstacles and ensure greater safety against possible collisions.

The main challenge here is to define the dimension of the structuring element in a way that does not eliminate small obstacles but also does not leave unfilled areas, creating false obstacles when performing the morphological transformation.

Finally, the A* search algorithm can be applied to find the shortest path between two points of interest.

3.4 User Interface

To make the interaction between the user and the LiDAR platform more user-friendly, a GUI was developed using the Python programming language and the tkinter and ttk-bootstrap packages. The communication between the GUI and the microcontroller is done by UDP datagrams and is composed of simple commands, such as starting or stopping the operation, and receiving the measurement data to generate the point cloud. In both cases, the message is encoded and sent together with the checksum in the header of the UDP datagram to ensure that the data will not be corrupted.

The GUI, shown in Figure 3.16, was designed to be as simple and intuitive as possible. It has a sidebar on the left that allows you to adjust various parameters for the data acquisition from the environment, such as the type of coordinate used, the range and resolution of the scan and some movement commands. There is also the option to visualize the acquired data in 3D and a progress bar that shows the progress of the data acquisition. In the central part of the GUI are the fields to define parameters to treat the point cloud to apply the A* Search algorithm, as well as the graphical display area of the results.

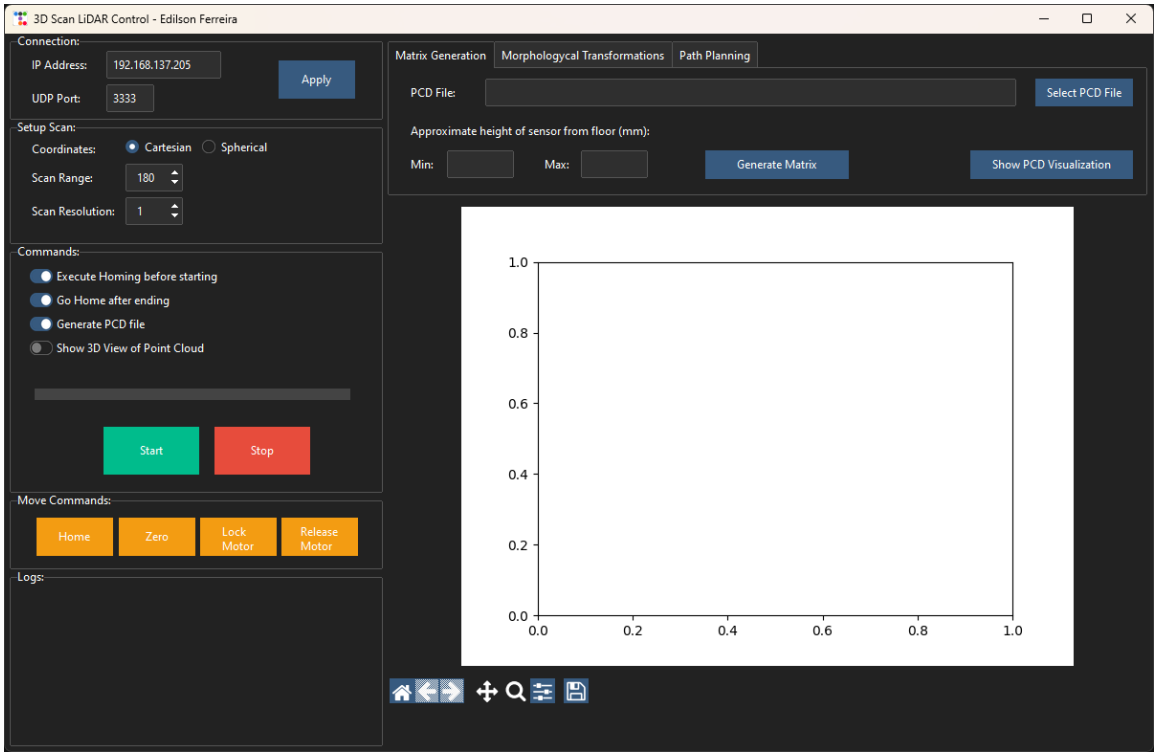


Figure 3.16: The GUI developed in Python to control the LiDAR platform.

The point cloud data processing part was divided into three tabs, being "Matrix Generation", "Morphological Transformations" and "Path Planning".

The first of these, shown in Figure 3.17, converts the point cloud floor plan into a matrix of zeros and ones. To do this, simply select an existing point cloud file. When performing a scan, the generated PCD file is automatically selected. There is also the possibility to define an estimated height of the sensor position, so that the ground plane is defined correctly.

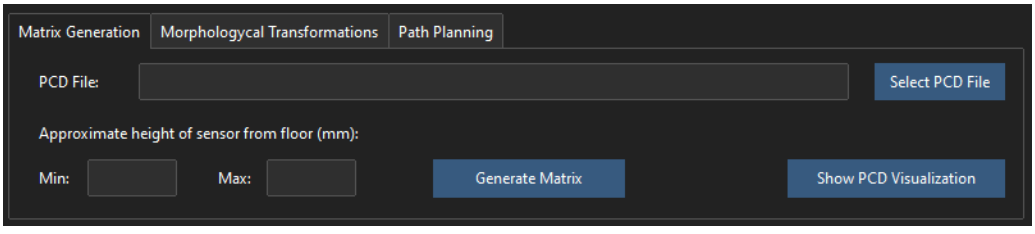


Figure 3.17: The matrix generation tab.

The second one, shown in Figure 3.18, allows to apply the operations to fill the spaces between the points of the point cloud. It is also possible to define the type of structuring element and its size, as well as the number of iterations to be performed.

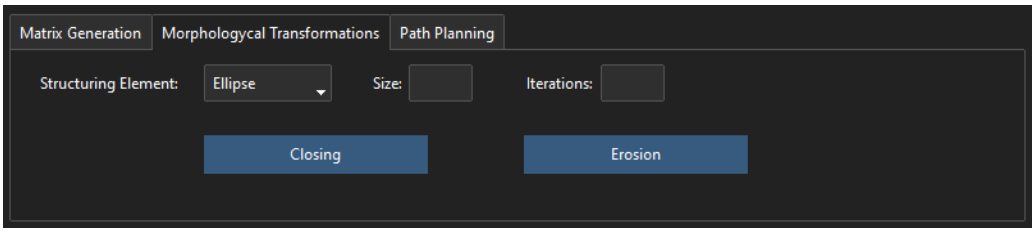


Figure 3.18: The morphological transformations tab.

And finally the third one, shown in Figure 3.19, is where the start and target points and the size of the robot are defined for the trajectory calculation. These points can also be defined by right-clicking on the generated image in the results visualization area.

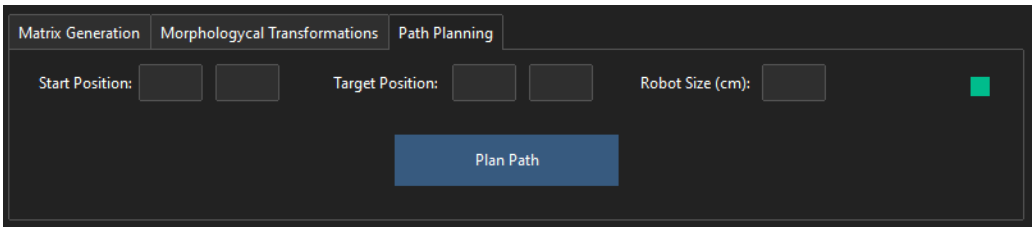


Figure 3.19: The path planning tab.

Chapter 4

Results

This chapter shows the results obtained from the tests carried out and the related analyses. Also presented are the results obtained from the measurements by the hardware prototype developed, as well as the processing, storage and visualization of this data and the path planning.

4.1 2D Measurement Test

In order to evaluate the operation of the LiDAR and verify the accuracy of the measurements taken, initial tests were performed by directly connecting the LiDAR to the computer through a USB to TTL Serial converter, as shown in Figure 4.1.

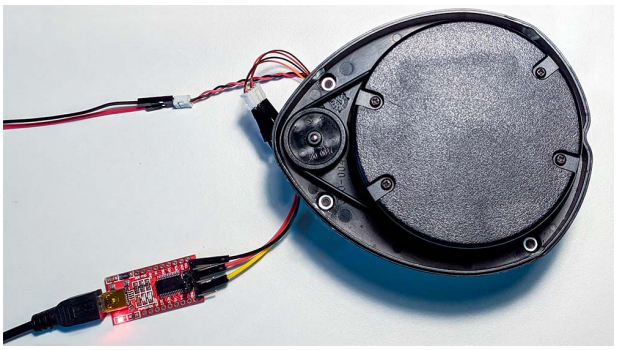


Figure 4.1: LiDAR connected to the USB to TTL Serial converter.

The Python programming language was used to receive and process the information provided by the LiDAR, as well as to plot the data on a two-dimensional graph, shown in Fig. 4.2, where it is possible to observe the distribution of the distance readings of a room, with the blue area being the area of the room measured in the sensor plane, and the red line being the current measurement taken by the sensor.

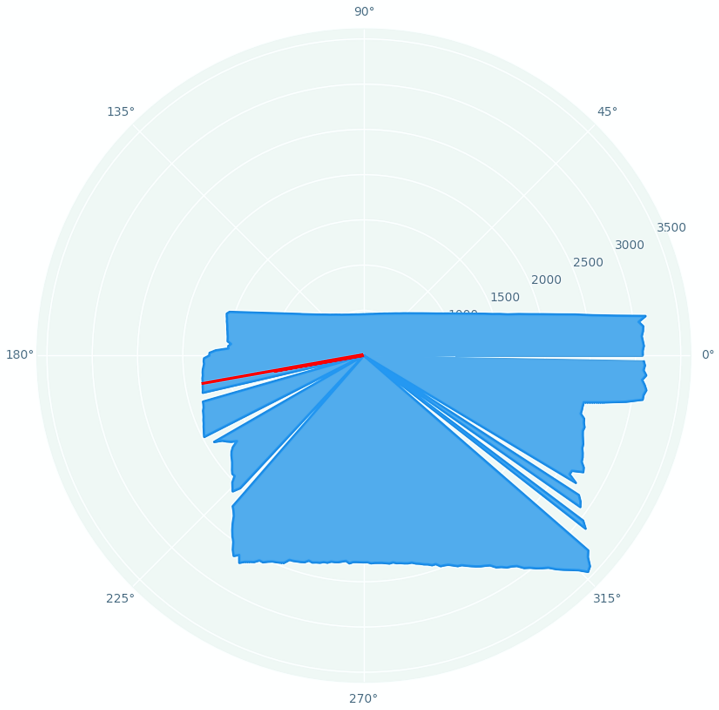


Figure 4.2: 2D visualization of a room measurement.

This initial test was crucial to verify the proper functioning of the LiDAR, as well as the correct conversion of the distances provided by the sensor. The results obtained provided important insights to proceed with the validation and further analysis of the data collected by the LiDAR.

4.2 3D Scanning and Point Cloud Generation

After the initial test was completed, the next step of the experiment was to perform a 3D scan using the LiDAR in conjunction with the stepper motor and the microcontroller reading and processing the data. This stage involved scanning a room, shown in Figure 4.3, with the sensor positioned on the floor.

Figure 4.3: Room used for measuring and generating the point cloud.

On the computer, a Python program received the UDP datagrams containing the coordinates and stored them in a point cloud file. This data structure allows the 3D representation of the points captured by the LiDAR.

After generating the point cloud file, the Open3D and PyVista libraries were used to display the points captured by the LiDAR in a 3D visualization, providing a faithful representation of the scanned environment, as shown in Figure 4.4. In this point cloud it is possible to see the bed, a bedside table and a chair.

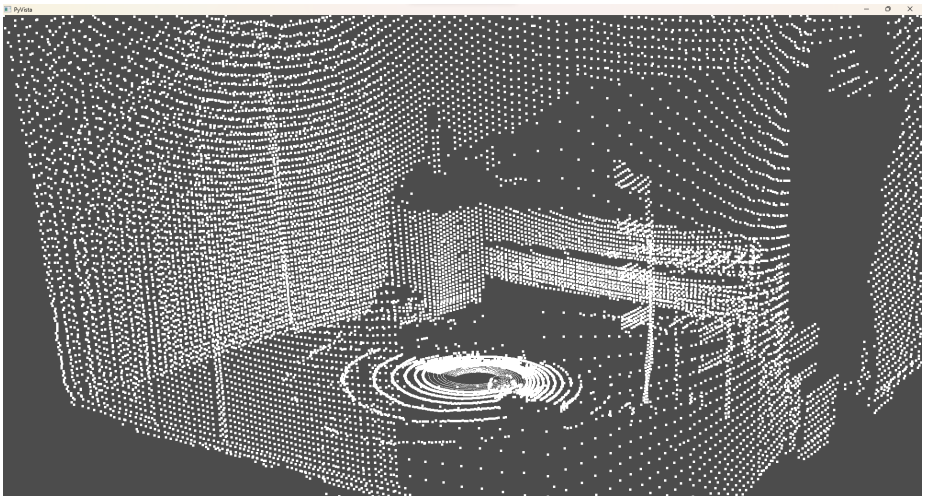


Figure 4.4: 3D visualization of a point cloud obtained from the room.

This step of the experiment demonstrated the system's ability to perform a complete 3D scan using the LiDAR, the stepper motor and the microcontroller. The conversion of the data into Cartesian coordinates and the generation of a point cloud file allowed a 3D visualization of the results, opening possibilities for more advanced analysis and several applications.

4.3 Data Preparation and Path Planning

In this stage of the experiment, a classroom was chosen as the environment to scan and generate the point cloud, which is shown in Figure 4.5.

Figure 4.5: Classroom used for measuring and generating the point cloud.

The LiDAR was installed on a projector support fixed on the ceiling, as shown in Figure 4.6, at a distance of approximately 2.4 meters from the floor, and the arrangements of tables and chairs were maintained.

Figure 4.6: Fixing the LiDAR platform to the projector stand.

The point cloud obtained from scanning the classroom is shown in Figure 4.7.

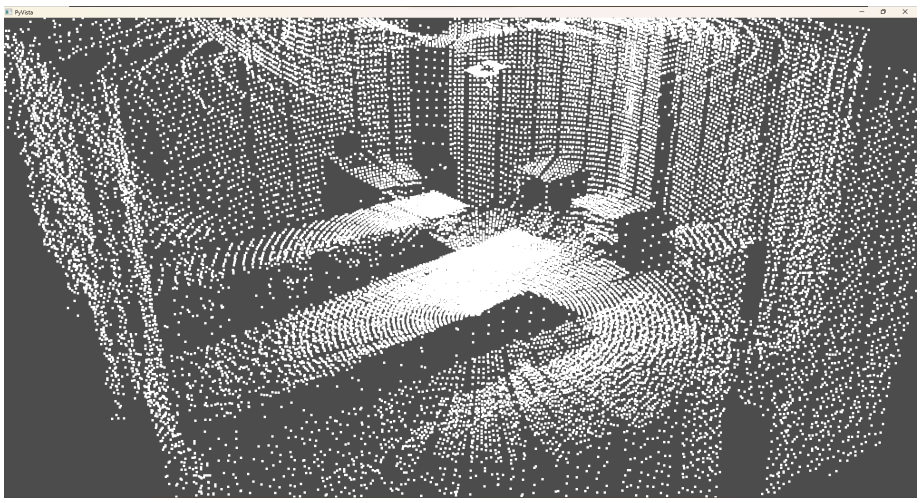


Figure 4.7: 3D visualization of the point cloud obtained from the classroom.

Once the classroom measurement data had been acquired and compiled into a point cloud, the first step in preparing the data to apply the path planning algorithm was to transform this point cloud into a matrix representing the floor plan. The point cloud was then sliced to obtain only the floor measurement points and the matrix obtained from these points is shown in Figure 4.8.

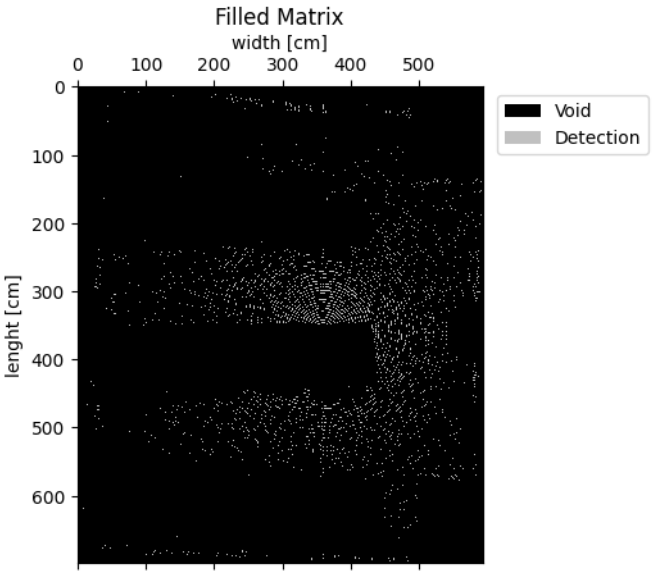
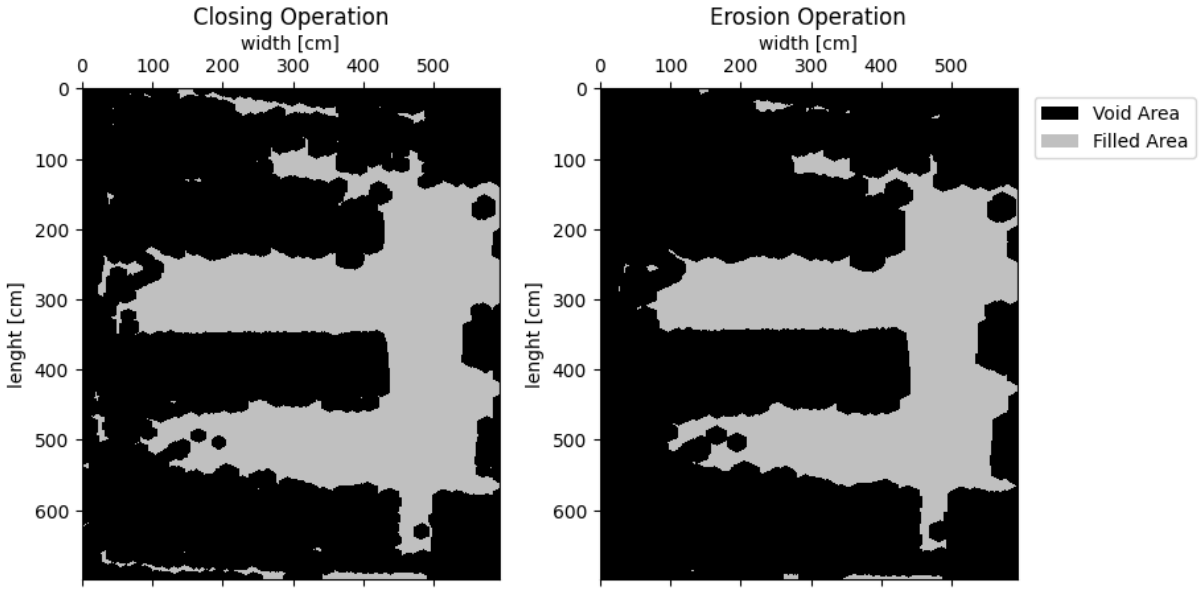


Figure 4.8: The matrix obtained from the point cloud floor plan.

The next step was to apply the morphological transformations of closing and erosion to the matrix obtained, to unify the points and create a closed surface. In this study, an ellipse of size 5×5 cm was used as a structuring element for the transformations, with 3 iterations for the closing operation and 2 iterations for the erosion operation. The result of the two transformations is shown in Figure 4.9.



(a) (b)

Figure 4.9: Results of morphological transformations: **(a)** Closing. **(b)** Erosion.

Finally, the path planning algorithm could be applied. For the search element was considered a square robot of dimensions 30×30 centimeters which its reference coordinate is located in the center of the robot. The start and target points were defined arbitrarily, without specific constraints. The objective was to evaluate the effectiveness of the A* algorithm in this search context, considering the search space and the characteristics of the search element mentioned.

Figure 4.10 shows the path found by the A* algorithm, marked in blue, as well as the start and target points defined by the green and red squares, respectively.

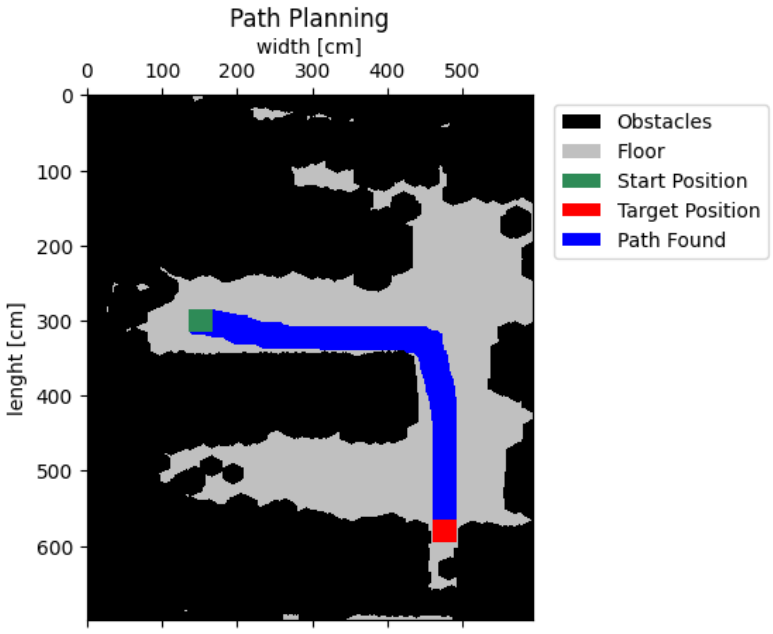


Figure 4.10: Graphical representation of the path found by the A* Algorithm.

4.4 Analysis of Discrepancies

After analyzing the result obtained, some discrepancies between the measured and actual values were identified, including the position of the tables and the distance between them. To understand what had happened, the point cloud was visually sectioned and symbolic beams were drawn to represent the sensor readings, as shown in Figure 4.11.

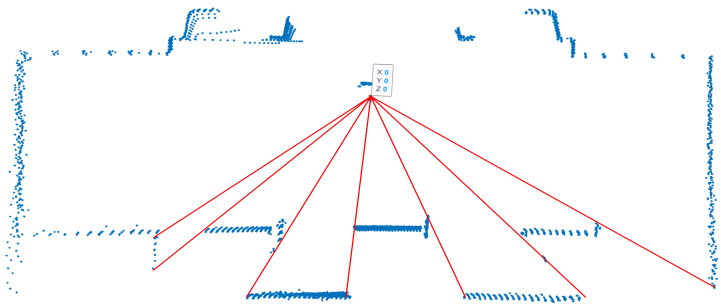


Figure 4.11: Side view of the sectioned point cloud.

The measurements made by the LiDAR do not correspond faithfully to reality. Due

to the fixed position of the equipment on the ceiling, the position and size of the tables and the distance between them were projected with larger dimensions and displaced from their actual positions on the floor.

Figure 4.12 provides a visual representation of the identified surface of the tables, in red, overlaid with the obstacle-free region on the floor, in blue, underscoring the significant disparities observed in this analysis.

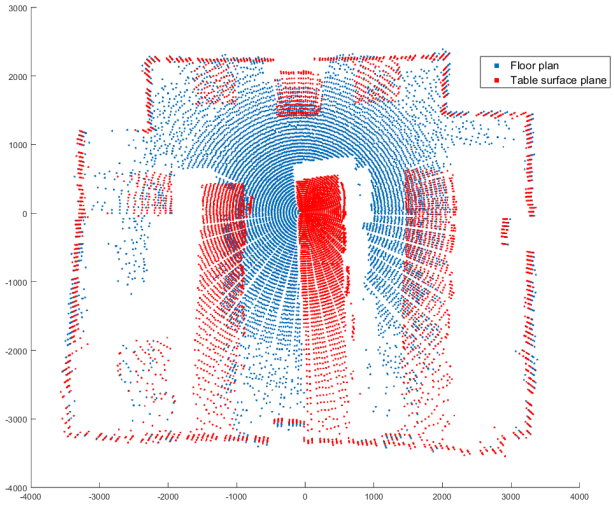


Figure 4.12: Top view of the superposition of the point cloud sectioned at the height of the tables and in the plane of the floor.

Moreover, due to the openness of the tables, the floor under them was wrongly identified as obstacle-free. Another limitation observed was the lack of detection of the space between the teacher's desk and the desk in the first row, and also at the back of the room, behind the desks in the last row.

4.5 Validation of the 3D LiDAR System

The aim of this stage is to validate the accuracy and reliability of the 3D LiDAR system developed, ensuring that the data captured accurately aligns with real measurements in a controlled environment.

For this validation, a kitchen was scanned repeatedly. A total of 10 point clouds were collected and the dimensions of a countertop with a known width of 65cm were evaluated. A visualization of one of the point clouds obtained is shown in Figure 4.13. In this scenario, the sensor was positioned about 1.4m above the countertop used as a parameter.

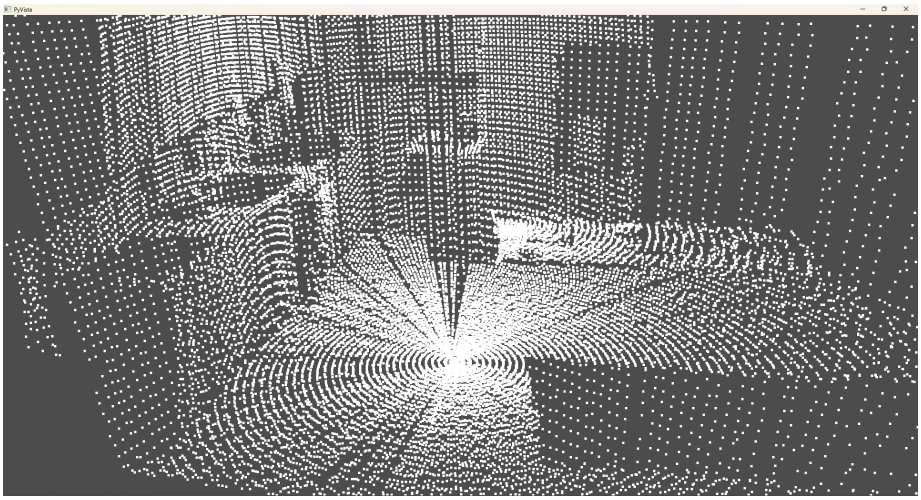


Figure 4.13: 3D visualization of the point cloud obtained from the kitchen.

From each of the kitchen's point clouds, the X, Y and Z coordinates of the countertop's corner points were obtained, as shown in Figure 4.14.

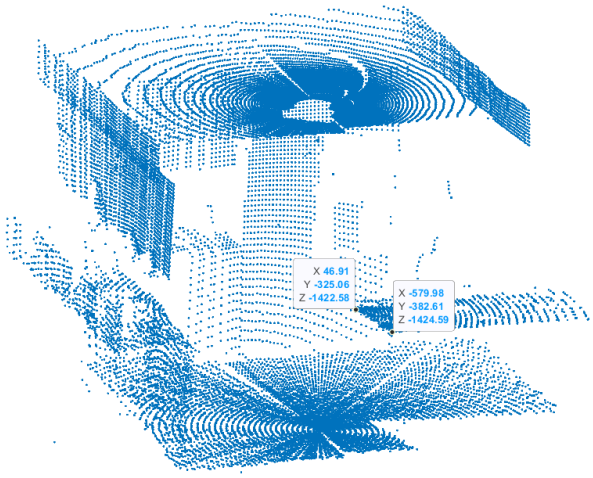


Figure 4.14: Obtaining the coordinates of the countertop corner points.

The distance between these points was calculated using Equation (4.1). For the example in Figure 4.14, the points were $A(46.91, -325.06, -1422.58)$ and $B(-579.98, -382.61, -1424.59)$ and the distance obtained was 659.18mm, or 65.918cm.

$$d_{A,B} = \sqrt{(X_B - X_A)^2 + (Y_B - Y_A)^2 + (Z_B - Z_A)^2} \tag{4.1}$$

The distances obtained from the 10 point clouds are shown in Table 4.1, along with the absolute and relative errors calculated to assess the accuracy of the measurements in relation to the real value.

Measurement	**Distance (cm)**	**Absolute Error (cm)**	**Relative Error (%)**
1	65.918	0.918	1.412
2	65.969	0.969	1.492
3	62.953	2.047	3.149
4	65.008	0.008	0.012
5	65.955	0.955	1.469
6	65.032	0.032	0.048
7	65.233	0.233	0.358
8	66.915	1.915	2.946
9	65.782	0.782	1.204
10	64.492	0.508	0.781
Average	**65.326**	**0.326**	**0.501**

Table 4.1: Measurement and associated errors.

Analysis of the errors associated with the 10 measurements reveals valuable insights into the accuracy of the device developed. The average relative error, calculated as the average of the relative errors of all the measurements, is approximately 0.5%. This indicator suggests consistent and acceptable accuracy in the measurements taken, with the measured values generally approaching the true values with minimal variation.

It is important to note that the accuracy of the measurements is directly related to the distance of the object or surface measured from the device developed. In this case, at a distance of approximately 1.4m, an average relative error of 0.5% is a good indication

that the measurements conform to the desired standards.

4.6 Cost survey for the prototype

This section will present a survey of the costs of the main components of the prototype, providing an estimate of the final cost of the project. The cost survey was carried out considering the value in national stores, with the exception of the LiDAR (which was used in stores such as eBay and AliExpress) and the PCB.

Table 4.2 shows the main components used in the prototype, together with their respective costs.

Component	**Cost (€)**
Neato XV-11 LDS	40.00
Stepper motor	15.50
ESP32	4.90
DRV8825	3.90
PCB	0.40
Others[1]	10.00
Total	**74.70**

Table 4.2: Prototype cost estimates.

The total estimated cost for the components listed above is around €75.00. This amount only covers the direct costs of the materials and physical components, and does not include expenses associated with shipping or indirect costs.

[1] includes resistors, capacitors, connectors, screws, and 3D printing.

4.7 Specifications of the device developed

Below are some of the technical specifications of the device developed:

- Data points generated: approximately 64,000 dots per full scan.
- Field of view:
  - Horizontal: 360°.
  - Vertical: 180°.
- Measuring range: 0.2 $\sim$ 6m
- Angle resolution:
  - Horizontal: 1°.
  - Vertical: 0.0562°
- Rotation speed: 3.3 $\sim$ 5 Hz
- Weight: 450g
- Communication interface: Wi-Fi (802.11 b/g/n)
- Operating voltage: 7.5 $\sim$ 18VDC

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Throughout this research, we sought to develop a low-cost 3D LiDAR system in order to overcome previously identified challenges and meet the growing demands for accessible and efficient three-dimensional mapping solutions. The results obtained corroborate the successful achievement of the proposed objectives, demonstrating not only the technical feasibility, but also the effectiveness of the device developed.

The successful implementation of the low-cost 3D LiDAR stands out not only for the quality of the three-dimensional capture, but also for its ability to adapt to different scenarios and applications. The precision achieved, combined with the simplicity of the system, opens doors to a variety of fields, from environmental monitoring to applications in autonomous vehicles and robotics.

In addition, the development process provided valuable insights, contributing to the advancement of knowledge in the field of three-dimensional remote sensing. The optimization of resources, the strategic choice of components and the approach adopted reflect not only on the efficiency of the system, but also on its practical applicability in real-world environments.

This work has not only achieved the pre-established objectives, but also offers a significant contribution to the scientific community and industry by providing an affordable and effective solution for three-dimensional mapping. The perspectives opened up by this development indicate a promising path for future research and practical applications, highlighting the potential of low-cost 3D LiDAR as a versatile and accessible tool for a variety of purposes.

Finally, the analyses conducted in this research are corroborated by the demonstrations presented in the article titled "Development of a Low-cost 3D Mapping Technology with 2D LIDAR for Path Planning Based on the A* Algorithm". This article was previously presented at the Sixth Iberian Robotics Conference 2023 (ROBOT2023).

5.2 Future Work

Based on the results and findings achieved in this research into the development of a low-cost 3D LiDAR, several opportunities for future work arise, exploring new horizons and broadening the impact of the study. Some proposals for future work include:

- To explore the feasibility and benefits of using 2D LiDAR sensors with higher spatial resolution, greater range and higher sampling rates to not only improve the accuracy of scans, but also enable data capture in dynamic and fast-moving environments;

- Use and/or development of more advanced algorithms for point cloud processing, especially in the application of path planning, taking into account the parallax inherent in the LiDAR measurement process, overcoming the limitations imposed by the current approach of extracting only the ground plane and applying morphological transformations;

- Integration of 3D LiDAR with other emerging technologies, such as artificial intelligence and machine learning, to improve the system's ability to interpret and analyze data automatically;

- Further evaluation of the costs associated with the production of the low-cost 3D LiDAR, considering the economic viability of implementation in comparison with models already on the market.

5.3 Academic Contributions

The following are the main contributions resulting from my research work during my master's degree. Five articles ([70]–[74]) were developed and sent to specialized conferences, where they were accepted and presented:

E. Ferreira, V. Grilo, J. Braun, et al., "Development of a Low-cost 3D Mapping Technology with 2D LIDAR for Path Planning Based on the A* Algorithm", *ROBOT 2023 - Sixth Iberian Robotics Conference*, 2023.

V. Grilo, **E. Ferreira**, A. Barbosa, A. A. Chellal, and J. Lima, "Design and Development of a Differential Drive Platform for Dragster Competition", *ROBOT 2023 - Sixth Iberian Robotics Conference*, 2023.

L. Borges, A. Rocha, **E. Ferreira**, and J. Ribeiro, "Planetary Gear Train for Dragster Robot", *TEEM'23 - Eleventh International Conference on Technological Ecosystems for Enhancing Multiculturality*, 2023.

A. Barbosa, **E. Ferreira**, V. Grilo, L. Mattos, and J. Lima, "Impact of EMG Signal Filters on Machine Learning Model Training: A Comparison with Clustering on Raw Signal", *OL2A 2023 - International Conference on Optimization, Learning Algorithms and Applications*, 2023.

V. Grilo, **E. Ferreira**, A. Barbosa, F. Chaves, and J. Lima, "Development of a Controller for the FANUC S-420FD Industrial Robot: a description of the graphical user interface", *ROBOT 2023 - Sixth Iberian Robotics Conference*, 2023.

Bibliography

[1] N. Slack, S. Chambers, and R. Johnston, *Operations Management*, 5th ed. Harlow, England: Financial Times Prentice Hall, Nov. 2006.

[2] M. P. Groover, *Fundamentals of modern manufacturing*, en, 4th ed. Chichester, England: John Wiley & Sons, Jan. 2010.

[3] L. Sabattini, V. Digani, C. Secchi, *et al.*, "Technological roadmap to boost the introduction of agvs in industrial applications," in *2013 IEEE 9th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2013, pp. 203–208. DOI: 10.1109/ICCP.2013.6646109.

[4] S. K. Das and M. K. Pasan, "Design and Methodology of Automated Guided Vehicle - A Review," *IOSR Journal of Mechanical and Civil Engineering*, vol. 3, no. 3, pp. 29–35, 2016.

[5] A. J. Moshayedi, L. Jinsong, and L. Liao, "AGV (automated guided vehicle) robot: Mission and obstacles in design and performance," *Journal of Simulation and Analysis of Novel Technologies in Mechanical Engineering*, vol. 12, no. 4, pp. 5–18, 2019.

[6] A. Haleem, M. Javaid, R. P. Singh, *et al.*, "Exploring the potential of 3d scanning in industry 4.0: An overview," *International Journal of Cognitive Computing in Engineering*, vol. 3, pp. 161–171, 2022, ISSN: 2666-3074. DOI: https://doi.org/10.1016/j.ijcce.2022.08.003.

[7] Z. Wang, Y. Liu, Q. Liao, H. Ye, M. Liu, and L. Wang, "Characterization of a RS-LiDAR for 3D Perception," in *2018 IEEE 8th Annual International Conference*

on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), 2018, pp. 564–569. DOI: 10.1109/CYBER.2018.8688235.

[8] U. Wandinger, "Introduction to lidar," in *Lidar: range-resolved optical remote sensing of the atmosphere*, Springer, 2005, pp. 1–18.

[9] T. Raj, F. Hanim Hashim, A. Baseri Huddin, M. F. Ibrahim, and A. Hussain, "A survey on LiDAR scanning mechanisms," *Electronics*, vol. 9, no. 5, p. 741, 2020.

[10] Y. Li and J. Ibanez-Guzman, "Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 50–61, 2020. DOI: 10.1109/MSP.2020.2973615.

[11] D. Hutabarat, M. Rivai, D. Purwanto, and H. Hutomo, "Lidar-based obstacle avoidance for the autonomous mobile robot," English, in *Proceedings of 2019 International Conference on Information and Communication Technology and Systems, ICTS 2019*, ser. Proceedings of 2019 International Conference on Information and Communication Technology and Systems, ICTS 2019, United States: Institute of Electrical and Electronics Engineers Inc., Jul. 2019, pp. 197–202. DOI: 10.1109/ICTS.2019.8850952.

[12] C. Dehong, Z. Liangqi, S. Pengpeng, T. Zaiyang, M. Yuhao, and W. Yong, "Design and implementation of lidar navigation system based on triangulation measurement," in *2017 29th Chinese Control And Decision Conference (CCDC)*, 2017, pp. 6060–6063. DOI: 10.1109/CCDC.2017.7978258.

[13] P. Dong and Q. Chen, *LiDAR Remote Sensing and Applications* (Remote Sensing Applications Series), en. London, England: CRC Press, Dec. 2017, ISBN: 978-1-138-74724-1.

[14] J. Liu, Q. Sun, Z. Fan, and Y. Jia, "TOF Lidar Development in Autonomous Vehicle," Sep. 2018, pp. 185–190. DOI: 10.1109/OGC.2018.8529992.

[15] Amad-ud-Din, I. A. Halin, and S. B. Shafie, "A review on solid state time of flight tof range image sensors," in *2009 IEEE Student Conference on Research and Development (SCOReD)*, 2009, pp. 246–249. DOI: 10.1109/SCORED.2009.5443066.

[16] J. G. Webster and H. Eren, Eds., *Measurement, Instrumentation, and Sensors Handbook* (Electrical Engineering Handbook), en, 2nd ed. Boca Raton, FL: CRC Press, Dec. 2017, ISBN: 978-1-4398-4889-0.

[17] N. Kolev, Ed., *Sonar Systems*. InTech, Sep. 2011.

[18] H. Weber, *LiDAR sensor functionality and variants*, Whitepaper, SICK AG, Jul. 2018.

[19] W. Y. Amaglo, "Volume calculation based on lidar data," M.S. thesis, KTH, Real Estate and Construction Management, 2021, p. 43.

[20] J. Park, Q.-Y. Zhou, and V. Koltun, "Colored point cloud registration revisited," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 143–152. DOI: 10.1109/ICCV.2017.25.

[21] A. E. Johnson and S. B. Kang, "Registration and integration of textured 3d data," *Image and vision computing*, vol. 17, no. 2, pp. 135–147, 1999.

[22] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. "Building rome in a day." (2009), [Online]. Available: https://grail.cs.washington.edu/rome/ (visited on 10/12/2023).

[23] F. Remondino, A. Guarnieri, and A. Vettore, "3d modeling of close-range objects: Photogrammetry or laser scanning?" In *Videometrics VIII*, SPIE, vol. 5665, 2005, pp. 216–225.

[24] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*, IEEE, 2012, pp. 3354–3361. DOI: 10.1109/CVPR.2012.6248074.

[25] I. Alonso, L. Riazuelo, L. Montesano, and A. C. Murillo, "3d-mininet: Learning a 2d representation from point clouds for fast and efficient 3d lidar semantic segmentation," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5432–5439, 2020.

[26] J. Behley, M. Garbade, A. Milioto, *et al.*, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019.

[27] J. Behley, M. Garbade, A. Milioto, *et al.*, "Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI Dataset," *The International Journal on Robotics Research*, vol. 40, no. 8-9, pp. 959–967, 2021. DOI: 10.1177/02783649211006735.

[28] J. Behley, M. Garbade, A. Milioto, *et al.*, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019.

[29] J. Behley, A. Milioto, and C. Stachniss, "A Benchmark for LiDAR-based Panoptic Segmentation based on KITTI," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 13 596–13 603. DOI: 10.1109/ICRA48506.2021.9561476.

[30] A. Duong, A. Almin, L. Lemarié, and B. R. Kiran, *Evaluating the effect of data augmentation and BALD heuristics on distillation of Semantic-KITTI dataset*, 2023. arXiv: 2302.10679 [cs.CV].

[31] Z. Wang, Y. Liu, Q. Liao, H. Ye, M. Liu, and L. Wang, "Characterization of a rs-lidar for 3d perception," in *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, 2018, pp. 564–569. DOI: 10.1109/CYBER.2018.8688235.

[32] S. Bi, C. Yuan, C. Liu, J. Cheng, W. Wang, and Y. Cai, "A survey of low-cost 3D laser scanning technology," *Applied Sciences*, vol. 11, no. 9, p. 3938, 2021.

[33] J. R. Rosell, J. Llorens, R. Sanz, *et al.*, "Obtaining the three-dimensional structure of tree orchards from remote 2D terrestrial LIDAR scanning," *Agricultural and Forest Meteorology*, vol. 149, no. 9, pp. 1505–1515, 2009.

[34] P. Moghadam, W. S. Wijesoma, and D. J. Feng, "Improving path planning and mapping based on stereo vision and lidar," in *2008 10th International Conference on Control, Automation, Robotics and Vision*, 2008, pp. 384–389. DOI: 10.1109/ICARCV.2008.4795550.

[35] C. Debeunne and D. Vivet, "A review of visual-lidar fusion based simultaneous localization and mapping," *Sensors*, vol. 20, no. 7, 2020, ISSN: 1424-8220. DOI: 10.3390/s20072068.

[36] X. Zhang, J. Lai, D. Xu, H. Li, and M. Fu, "2D Lidar-Based SLAM and Path Planning for Indoor Rescue Using Mobile Robots," *Journal of Advanced Transportation*, vol. 2020, Z. Cao, Ed., pp. 1–14, Nov. 2020. DOI: 10.1155/2020/8867937.

[37] N. Bolourian and A. Hammad, "LiDAR-equipped UAV path planning considering potential locations of defects for bridge inspection," *Automation in Construction*, vol. 117, p. 103 250, 2020, ISSN: 0926-5805. DOI: 10.1016/j.autcon.2020.103250.

[38] J. Bierende, J. Braun, P. Costa, J. Lima, and A. Pereira, "Volume Estimation of an Indoor Space with LiDAR Scanner," in *Optimization, Learning Algorithms and Applications*, Springer International Publishing, 2022, pp. 78–92, ISBN: 978-3-031-23236-7. DOI: 10.1007/978-3-031-23236-7_6.

[39] Autodesk, *Autodesk Inventor 3D CAD Software*. [Online]. Available: https://www.autodesk.com/products/inventor/overview (visited on 04/12/2023).

[40] W. Kowalski and M. Roman, "Autodesk Inventor Professional as common platform CAD for designer from mechanic and electromechanic profession," in *Proceedings of the International Conference Modern Problems of Radio Engineering, Telecommunications and Computer Science, 2004.*, 2004, pp. 303–305.

[41] Autodesk, *EAGLE | PCB Design And Electrical Schematic Software*. [Online]. Available: https://www.autodesk.com/products/eagle/overview (visited on 04/12/2023).

[42] M. Simon, *Make your own PCBs with EAGLE: From schematic designs to finished boards*. New York, NY: TAB Books, Jun. 2014.

[43] M. Scarpino, *Designing circuit boards with EAGLE - Make High-Quality PCBs at Low Cost*. Philadelphia, PA: Prentice Hall, Mar. 2014.

[44] V. S. Code, *Code Editing. Redefined*. [Online]. Available: https://code.visualstudio.com/ (visited on 03/21/2023).

[45] PlatformIO, *A professional collaborative platform for embedded development*. [Online]. Available: https://platformio.org/ (visited on 03/21/2023).

[46] E. Systems, *ESP32 Wi-Fi & Bluetooth MCU*. [Online]. Available: https://www.espressif.com/en/products/socs/esp32 (visited on 03/21/2023).

[47] A. Maier, A. Sharp, and Y. Vagapov, "Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things," in *2017 Internet Technologies and Applications (ITA)*, 2017, pp. 143–148. DOI: 10.1109/ITECHA.2017.8101926.

[48] E. Systems, *ESP32 Series Datasheet*, v4.3, 2010. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (visited on 03/22/2023).

[49] R. Agrawal, *Hacking the Neato XV-11 Wiki*. [Online]. Available: http://xv11hacking.rohbotics.com/ (visited on 03/18/2023).

[50] C. Taylor, *Neato Robotics XV-11 tear-down*, Nov. 2010. [Online]. Available: https://www.sparkfun.com/news/490 (visited on 03/21/2023).

[51] V. Athani, *Stepper motors: fundamentals, applications and design*. New Age International, 1997.

[52] P. Lawrenson, "Stepping motors: A guide to modern theory and practice," *Electronics and Power*, vol. 29, no. 9, p. 659, 1983.

[53] B. Aranjo, P. K. Soori, and P. Talukder, "Stepper motor drives for robotic applications," in *2012 IEEE International Power Engineering and Optimization Conference Melaka, Malaysia*, 2012, pp. 361–366. DOI: 10.1109/PEOCO.2012.6230890.

[54] W. M. C. Ltd, *China Hybrid Stepper Motor, Hybrid Stepper Motor Driver, Closed Loop Stepper Motor Suppliers, Manufacturers, Factory - WANTAI*. [Online]. Available: https://www.wantmotor.com/hybrid-stepper-motor/2-phases-hybrid-stepper-motor/42byghw-stepper-motor.html (visited on 03/22/2023).

[55] T. Instruments, *DRV8825 Stepper Motor Controller IC Datasheet*, Rev. F, 2010. [Online]. Available: https://www.ti.com/lit/ds/symlink/drv8825.pdf (visited on 03/22/2023).

[56] M. H. Rashid, *Power Electronics: Devices, Circuits and Applications*, 3rd ed. Pensacola, FL: Butterworth-Heinemann, 2011.

[57] M. Instruments, *MP1482 2A, 18V Synchronous Rectified Step-Down Converter Datasheet*, Rev. 1.31, 2012. [Online]. Available: https://pt.mouser.com/datasheet/2/277/MP1482-1383982.pdf (visited on 03/22/2023).

[58] S. Liu, J. Zheng, X. Wang, Z. Zhang, and R. Sun, "Target detection from 3d point-cloud using gaussian function and cnn," in *2019 34rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2019, pp. 562–567. DOI: 10.1109/YAC.2019.8787705.

[59] K. Sreedhar, "Enhancement of images using morphological transformations," *International Journal of Computer Science and Information Technology*, vol. 4, no. 1, pp. 33–50, Feb. 2012. DOI: 10.5121/ijcsit.2012.4103.

[60] P. Soille *et al.*, *Morphological image analysis: principles and applications*. Springer, 1999, vol. 2.

[61] R. Gonzalez and R. Woods, *Digital Image Processing*, 3rd ed. Prentice Hall, 2002, ISBN: 9780201180756.

[62] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968. DOI: 10.1109/TSSC.1968.300136.

[63] M. Goldenberg, "The heuristic search research framework," *Knowledge-Based Systems*, vol. 129, pp. 1–3, 2017.

[64] A. Nash and S. Koenig, "Any-angle path planning," *AI Magazine*, vol. 34, no. 4, pp. 85–107, 2013.

[65] F. Duchoň, A. Babinec, M. Kajan, *et al.*, "Path planning with modified a star algorithm for a mobile robot," *Procedia Engineering*, vol. 96, pp. 59–69, 2014, Modelling of Mechanical and Mechatronic Systems, ISSN: 1877-7058. DOI: 10.1016/j.proeng.2014.12.098.

[66] J. Yao, C. Lin, X. Xie, A. J. Wang, and C.-C. Hung, "Path planning for virtual human motion using improved a* star algorithm," in *2010 Seventh International Conference on Information Technology: New Generations*, 2010, pp. 1154–1158. DOI: 10.1109/ITNG.2010.53.

[67] G. Tang, C. Tang, C. Claramunt, X. Hu, and P. Zhou, "Geometric a-star algorithm: An improved a-star algorithm for agv path planning in a port environment," *IEEE Access*, vol. 9, pp. 59 196–59 210, 2021. DOI: 10.1109/ACCESS.2021.3070054.

[68] J. Stewart, *Calculus*, 7th ed. Florence, AL: Cengage Learning, Jan. 2011.

[69] OpenCV, *OpenCV Library Documentation*. [Online]. Available: https://docs.opencv.org/4.x/ (visited on 06/28/2023).

[70] E. Ferreira, V. Grilo, J. Braun, *et al.*, "Development of a Low-cost 3D Mapping Technology with 2D LIDAR for Path Planning Based on the A* Algorithm," *ROBOT 2023 - Sixth Iberian Robotics Conference*, 2023.

[71] V. Grilo, E. Ferreira, A. Barbosa, A. A. Chellal, and J. Lima, "Design and Development of a Differential Drive Platform for Dragster Competition," *ROBOT 2023 - Sixth Iberian Robotics Conference*, 2023.

[72] L. Borges, A. Rocha, E. Ferreira, and J. Ribeiro, "Planetary Gear Train for Dragster Robot," *TEEM'23 - Eleventh International Conference on Technological Ecosystems for Enhancing Multiculturality*, 2023.

[73] A. Barbosa, E. Ferreira, V. Grilo, L. Mattos, and J. Lima, "Impact of EMG Signal Filters on Machine Learning Model Training: A Comparison with Clustering on Raw Signal," *OL2A 2023 - International Conference on Optimization, Learning Algorithms and Applications*, 2023.

[74] V. Grilo, E. Ferreira, A. Barbosa, F. Chaves, and J. Lima, "Development of a Controller for the FANUC S-420FD Industrial Robot: a description of the graphical user interface," *ROBOT 2023 - Sixth Iberian Robotics Conference*, 2023.

Appendix A

Original Project Proposal

INSTITUTO POLITÉCNICO DE BRAGANÇA
Escola Superior de Tecnologia e Gestão

Mestrado em Engenharia Industrial – Eletrotécnica
Ano letivo de 2022/2023

Three-dimensional mapping of an unknown environment based on 2D laser measurement

Aluno: Edilson Santos Ferreira
Orientador: José Luís Sousa de Magalhães Lima
Coorientador: Murillo Ferreira dos Santos / João Afonso Braun Neto

1 Objetivo

O objetivo deste projeto é desenvolver um sistema autônomo de baixo custo para realizar escaneamento 3D de ambientes internos, visando o planejamento de trajetórias. Será utilizada a tecnologia LiDAR 2D combinada à um motor de passo, permitindo o mapeamento e navegação autônoma em espaços internos.

2 Detalhes

Este sistema se concentra na tarefa específica de escaneamento 3D e planejamento de trajetória. Ele utiliza um LiDAR 2D para capturar dados do ambiente e um motor de passo para controlar a orientação do LiDAR. Esses dados são utilizados para mapear e criar um modelo 3D do espaço interno, um aspecto fundamental para a navegação autônoma.

Por meio de uma interface gráfica intuitiva, os usuários podem visualizar o modelo 3D do ambiente e definir pontos para o planejamento de trajetórias.

3 Metodologia de trabalho

O desenvolvimento deste trabalho seguirá as seguintes etapas:

1. Estudo do hardware: ESP32, motor de passo e LiDAR 2D, seguida pela sua integração.
2. Estudo das linguagens de programação C++ e Python, e das bibliotecas necessárias, para desenvolver o algoritmo de controle e processamento de dados.
3. Criação da estrutura física do protótipo, para acomodar e interligar os componentes de hardware.
4. Testes do circuito eletrônico em protoboard e desenvolvimento da placa de circuito.
5. Desenvolvimento da interface gráfica do usuário.
6. Realização de testes e validação do sistema.
7. Escrita da dissertação e preparação para a defesa final do trabalho.

Dimensão da equipa: 4 pessoas

Recursos necessários: Recursos de hardware disponibilizados nos laboratórios do Centro de Investigação em Digitalização e Robótica Inteligente (CeDRI) e recursos de software.

Appendix B

Printed Circuit Board

This appendix provides extra information about the PCB, including the full-size PCB with dimensions (Figure B.1) and the complete schematic circuit (Figure B.2).

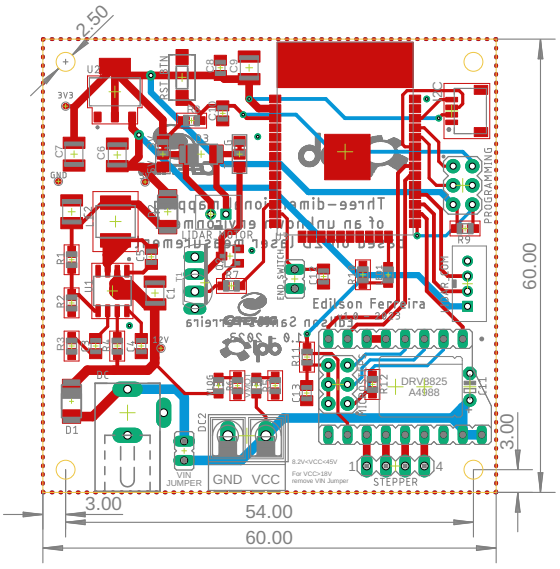


Figure B.1: Full-size PCB with dimensions.

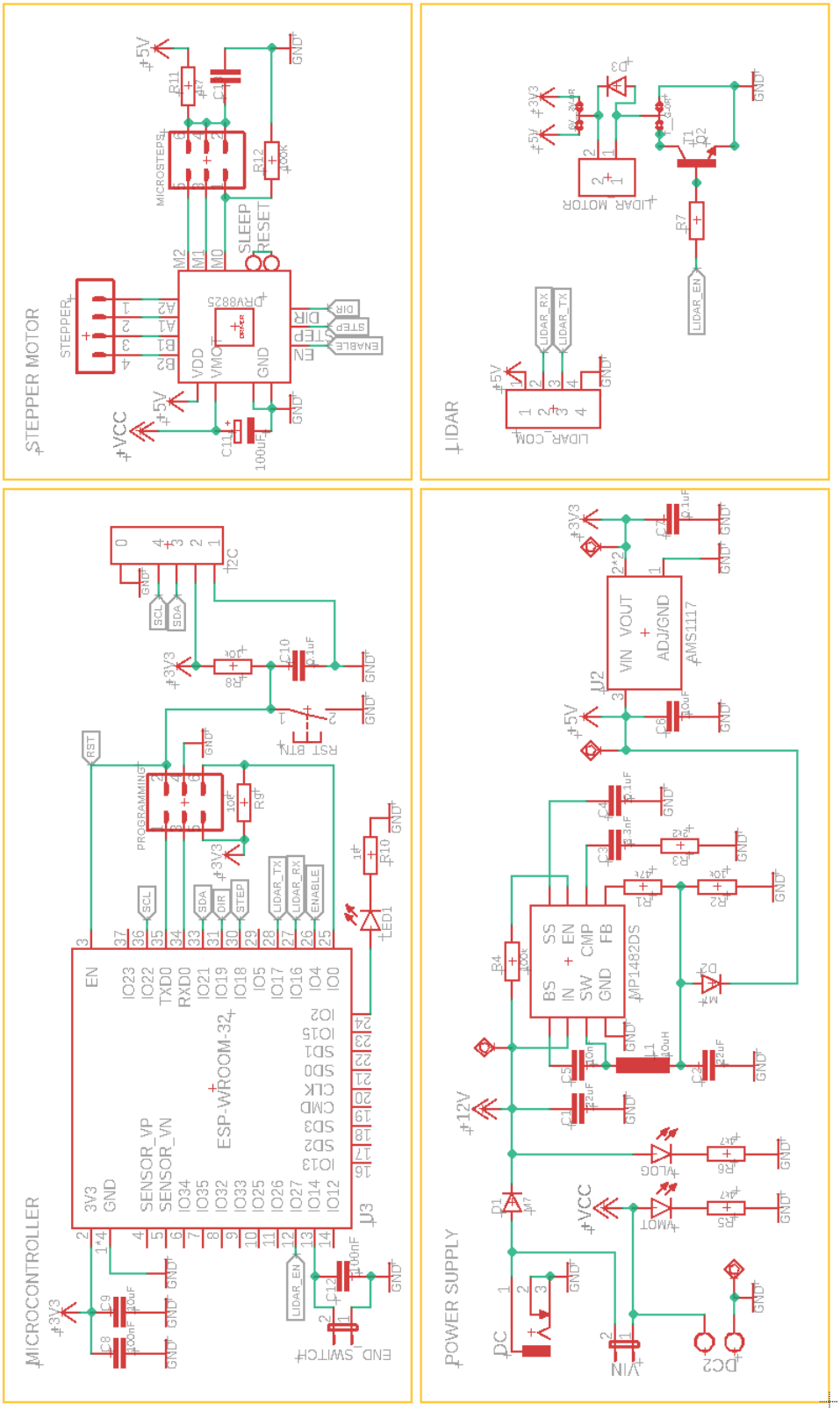


Figure B.2: Complete Circuit Schematic.

Appendix C

Project Repositories

This appendix provides links to the GitHub repositories that contain the source code, 3D drawings and PCB files related to this research work. The following resources are essential for understanding and reproducing the results discussed in this thesis.

C.1 Microcontroller

The source code for the microcontroller used in this project is available in the following GitHub repository:

`https://github.com/edilsonsfc/Tese-Lidar-Code`

This repository includes all the programs, libraries and schematics related to the microcontroller, allowing detailed analysis and reproduction of the experiments.

C.2 Graphical User Interface

The GUI developed for this project can be found in the following GitHub repository:

`https://github.com/edilsonsfc/Tese-Lidar-GUI`

In this repository, you will find the GUI source code, along with all the information

you need to configure and run the user interface.

C.3 3D Drawings

The 3D drawings of the structure used in this project are available in the following GitHub repository:

`https://github.com/edilsonsfc/Tese-Lidar-Structure`

This repository includes CAD files, 3D models and documentation related to the physical structure, making it easier to visualize and reproduce the structure used in the experiment.

C.4 Printed Circuit Board

The circuit board files developed for this project can be found in the following GitHub repository:

`https://github.com/edilsonsfc/Tese-Lidar-PCB`

In this repository, you will find the schematics, PCB layouts and any documentation associated with the circuit board, allowing detailed analysis and replication of the board used in the experiment.