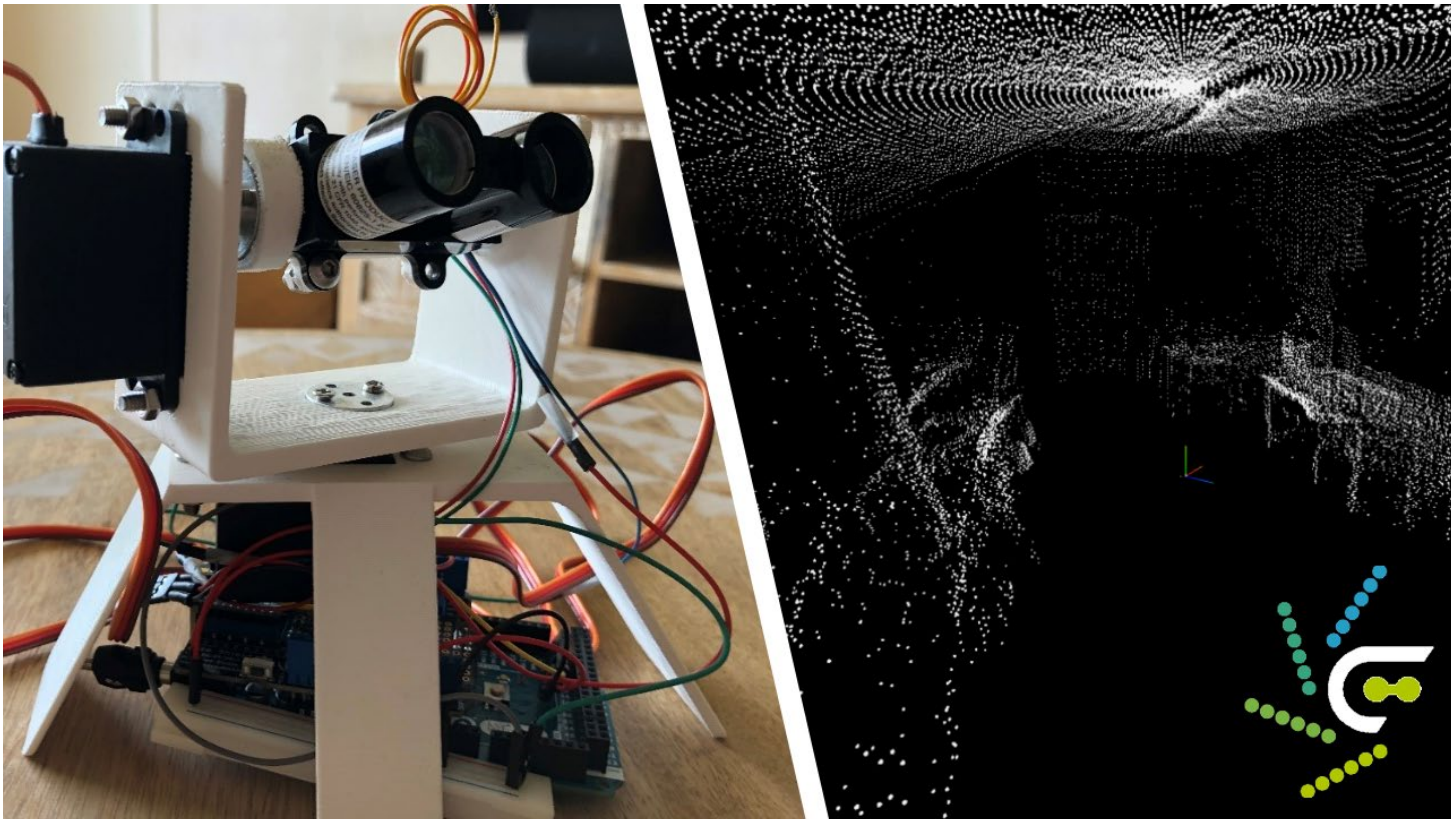


Engineering Sciences & Physics and Mathematics Specialties

LiDAR 3D Cartography

Project file

Project led by Théo Vidal and Kylian Capitanio

2020-2021

Table of Contents

1. Project description

A. The identified need

The goal is to propose a technical solution for **mapping our real environment** in three dimensions, as in the digital era, acquiring environmental data is a major challenge for numerous fields such as photography, cinema, architecture, and history.

Figure 1 : cameras and LiDAR sensor on the back of the iPhone 12 Pro
© Apple Inc.

In 2020, the technology company Apple introduced its latest generation of mobile phones, the iPhone 12 Pro. Among its new features was an onboard **LiDAR sensor** , in addition to the three cameras. Until then, the device could only capture a two-dimensional image of its surroundings, relying on artificial intelligence to distinguish reliefs, which did not always work well… This new LiDAR, capable of **capturing in three dimensions** , allows for a better understanding of the environment by overlaying the 3D model onto the 2D image for a more accurate rendering. It thus enables more precise spatial measurements or photographic portraits, and third-party developers have already leveraged this technology through augmented reality applications available on the App Store.

Digitization allows for both long-term data preservation and easier exploitation using specialized computer tools. We can identify several situations where 3D mapping is important or even essential:

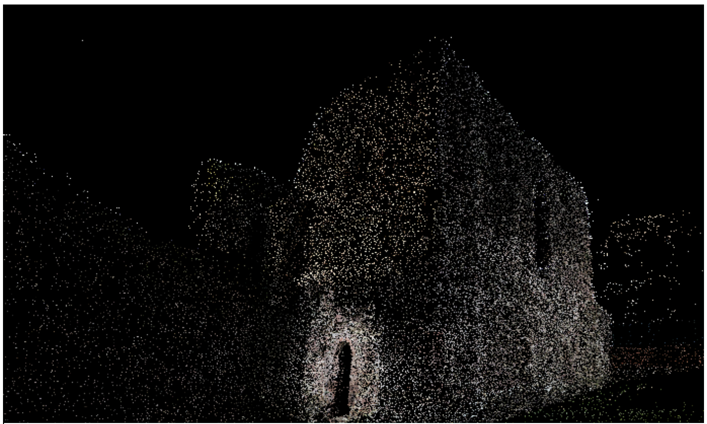


Figure 2 : Example of a point cloud modelling a ruined castle
© John Doe / Wikimedia Commons / CC-BY-SA-3.0

- During deep underwater reconnaissance, where vision is useless as sunlight is absorbed by the upper layers of the ocean.
- Recognition of hard-to-access or dangerous locations: confined spaces (*and no, I'm not talking about COVID…*), subject to extreme constraints

- Preservation of historical monuments or significant locations, threatened by destruction due to human activity or natural disasters
- Calculations in the real world enable the autonomous operation of cars, reconnaissance drones, and can provide assistance to visually impaired individuals

B. Project goal

We will design a technical system capable of autonomously performing a **3D scan** of its surrounding environment and transmitting the data to a computer for processing and storage. The output can be **used in various ways** (exploration, printing, calculations…) and remains flexible for use across a **wide range of fields** , ensuring it is not restricted to specific tools. The digitized data can also be integrated with other technologies, such as artificial intelligence and machine learning algorithms, for object recognition in space. In the future, we can envision extending the system to a mobile platform to achieve a comprehensive and continuous mapping of a location .

Figure 3: Our determination when we embarked on the project. Otherwise, for the actual caption: SpaceX Crew-2 launch in April 2021.

2. A multidisciplinary project

A. The project from a Mathematical perspective

Overall, we use **spatial geometry** with an orthonormal reference frame, allowing us to perform a wide variety of calculations based on the points acquired by the system.

Our mathematical knowledge enables us to perform **trigonometric calculations** in space using the distance returned by the LiDAR and the two angles around the axes formed by the servomotors. These three values represent the polar coordinates of the point.

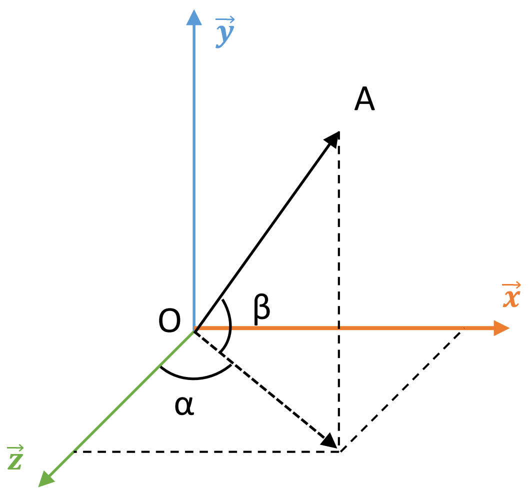


Figure 4 : Diagram of the situation in the direct orthonormal reference frame $(O; \vec{x}; \vec{y}; \vec{z})$

Point A corresponds to the LiDAR acquisition, with $OA = \|\overrightarrow{OA}\|$ representing the distance returned by the sensor, α the angle formed by the azimuth scan, and β by the elevation scan (see structural description of the system, section 6.A).

The conversion from polar to Cartesian coordinates, for simplified processing, is performed using the formulas:

$$x_A = OA \,.\cos\beta \,\,.\sin\alpha$$
$$y_A = OA \,.\sin\beta$$
$$z_A = OA \,.\cos\beta \,\,.\cos\alpha$$

Mathematics will also enable us to perform calculations on the digitized data, including **distances, surfaces, and volumes in space** , as well as approximations using planes and geometric shapes. The creation of multi-parameter functions for **integration** also requires knowledge in this field.

We have implemented several solutions to carry out initial calculations. To achieve this, we use the point cloud obtained from mapping to create a **convex geometry** —a volume in which all interior angles are less than 180°, meaning it has no recesses. Indeed, the raw point cloud is largely unusable, except for an initial exploration to obtain an overview. We use the Quickhull algorithm, which constructs a single surface by assembling non-flattened triangles, connecting as many vertices as possible while maintaining a shape with minimal noise.

i. Surface of the geometry

We first calculate the length of the triangle's sides based on the position of its vertices:

$$AB = \|\overrightarrow{AB}\| = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2}$$

The calculation of the triangle's surface from its three sides is then performed using Heron's Formula:

$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

with a, b and c being the lengths of the three sides (the order does not matter) , and $p = \frac{a+b+c}{2}$

To obtain the total surface area of the convex geometry, it is sufficient to sum the surface areas of each triangle.

Source (in French): https://www.alloprof.qc.ca/fr/eleves/bv/mathematiques/l-aire-des-triangles-al-aide-de-la-formule-de-h-m1295

ii. Volume of geometry

We decompose the geometric shape into tetrahedra , with one of their vertices at the origin of the reference frame and their base being any triangle with vertices v_1, v_2 and v_3 defined by vectors originating from the reference point.

Figure 5: Diagram of the decomposition of a geometry into tetrahedra to determine its volume

The signed volume of such a tetrahedron, positive or negative depending on the order of the vertices, is defined by:

$$V = \frac{1}{6}(v_1 \wedge v_2) \cdot v_3$$

The cross product $\vec{u} \wedge \vec{v}$, allowing the calculation of the orthogonal vector with a norm of $\|\vec{u}\|.\|\vec{v}\|.\left|\sin \widehat{\vec{u},\vec{v}}\right|$, is defined in 3D by:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \wedge \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} yz' - y'z \\ xz' - x'z \\ xy' - x'y \end{pmatrix}$$

The dot product $\vec{u} \cdot \vec{v}$, which associates a unique scalar with two vectors, is defined in space by:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \cdot \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = xx' + yy' + zz'$$

To obtain the total volume of the convex geometry, it is sufficient to sum the signed volumes of each tetrahedron. The overlapping regions of the tetrahedra will cancel out during summation due to the sign of the volumes, ensuring that each part is counted only once.

Source: https://nervous.com/blog/?p=4415

B. The project from a Physical Sciences perspective

Our knowledge of Physical Sciences will enable us to understand the physical phenomena involved, particularly **light**, and specifically the flux of laser photons in the infrared spectrum, which is central to the operation of the LiDAR. This is described in greater detail in section 3.B of the report.

We also apply **Newton's Second Law for Rotation** to achieve balance in the system's structure (see section 6.A). Indeed, the goal is to reduce the effort exerted on the servomotor to prevent its premature degradation.

C. The project from an Engineering Sciences perspective

Our knowledge in Engineering Sciences enables us to create and manage a **scientific project**, following all the necessary steps such as defining specifications, conducting simulations, and performing tests. This field also provides the required **Arduino programming** skills for the remote system (see structural description of the system, section 6.A), as well as combinational logic using Boolean algebra and solid mechanics for the system's movements via servomotors. Finally, the concept of **moments** is considered when seeking balance to optimize scanning efficiency and ensure the long-term preservation of components.

3. Description of the available technical support

A. Analyzes is of the available technical solutions

To develop our technical system, we first need to acquire environmental data. For this, several options are available:

Figure 6 : HC-SR04 Ultrasonic sensor

- Ultrasonic sensor: returns distance but lacks precision, as mechanical waves are more susceptible to interference.
- 3D LiDAR scanner: requires minimal effort as the mapping is almost entirely automated, but it is quite expensive (over €200), and the project's purpose would be lost.

Figure 7 : YDLIDAR X2 3D LiDAR scanner

- LiDAR sensor: the best compromised, as it is precise, reliable, and affordable (around €40) .

To implement the system's control strategy for scanning, we will reuse servomotors from a previous project, assembling them differently with the help of 3D-printed parts. The precision of rotation and real-time knowledge of the angles formed around the axes are essential for reliable data acquisition.

Figure 8 : GARMIN LiDAR-Lite v3 LiDAR scanner

B. Physical phenomena involved

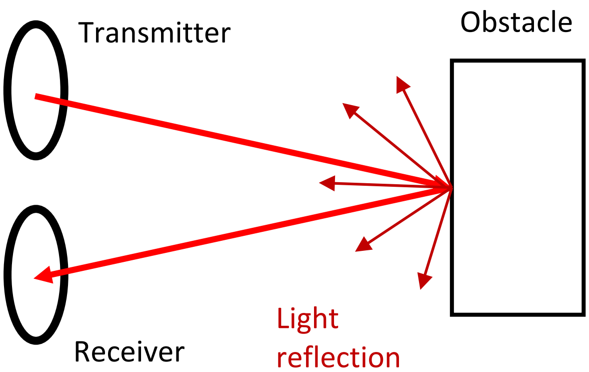


Figure 9 : Diagram of LiDAR operation with light emission and reception

Light and the photon flux it involves is the most crucial phenomenon in this project, as it is used by the LiDAR sensor to measure the distance to the nearest obstacle. The emission of infrared light, its reflection on obstacles, and its reception by the sensor enable precise distance calculation. LiDAR operates in several internships:

- Correction of polarization and variations in external light
- Initial measurement of the "zero" distance, adjusted over multiple readings
- Transmission of multiple acquisitions via laser signal recorded on the device
- Acquisition of the most intense light peak in the captured spectrum

- Integration until the peak is reached or the maximum number of acquisitions is reached
- Based on this peak and the ambient light signal threshold, it calculates the signal intensity
- If the threshold is exceeded, the distance is calculated; otherwise, the sensor returns 1 cm
- The recording is then erased, and the sequence can restart.

The distance is calculated based on the signal intensity using the following equation: $I = \frac{1}{d^2}$.

We will neglect the inertia of the servomotors and the offsets it may cause, as our step size is very small (1°), meaning the error will not be noticeable when processing the data.

4. Definition of a Specifications document

A. The initial need

Defined in section 1.A

B. Purpose and mission of the system

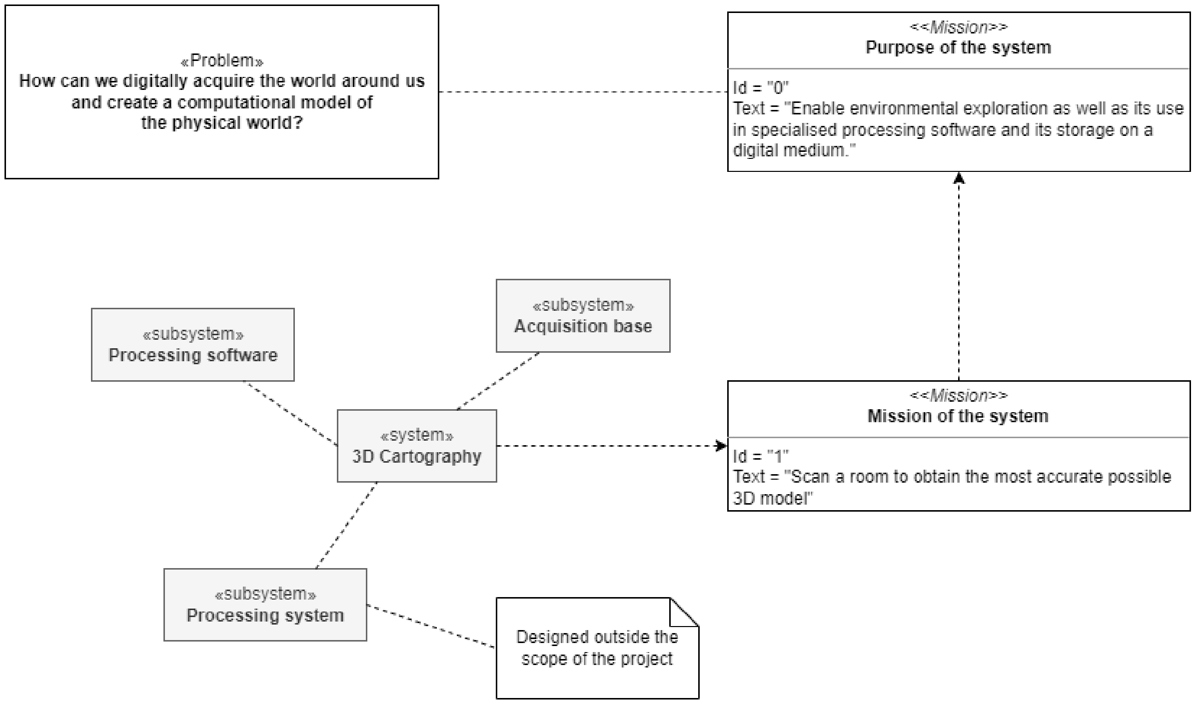


Figure 10: Diagram illustrating the purpose of the system

Defined in greater detail in section 1.B

C. Taking into account the context of use

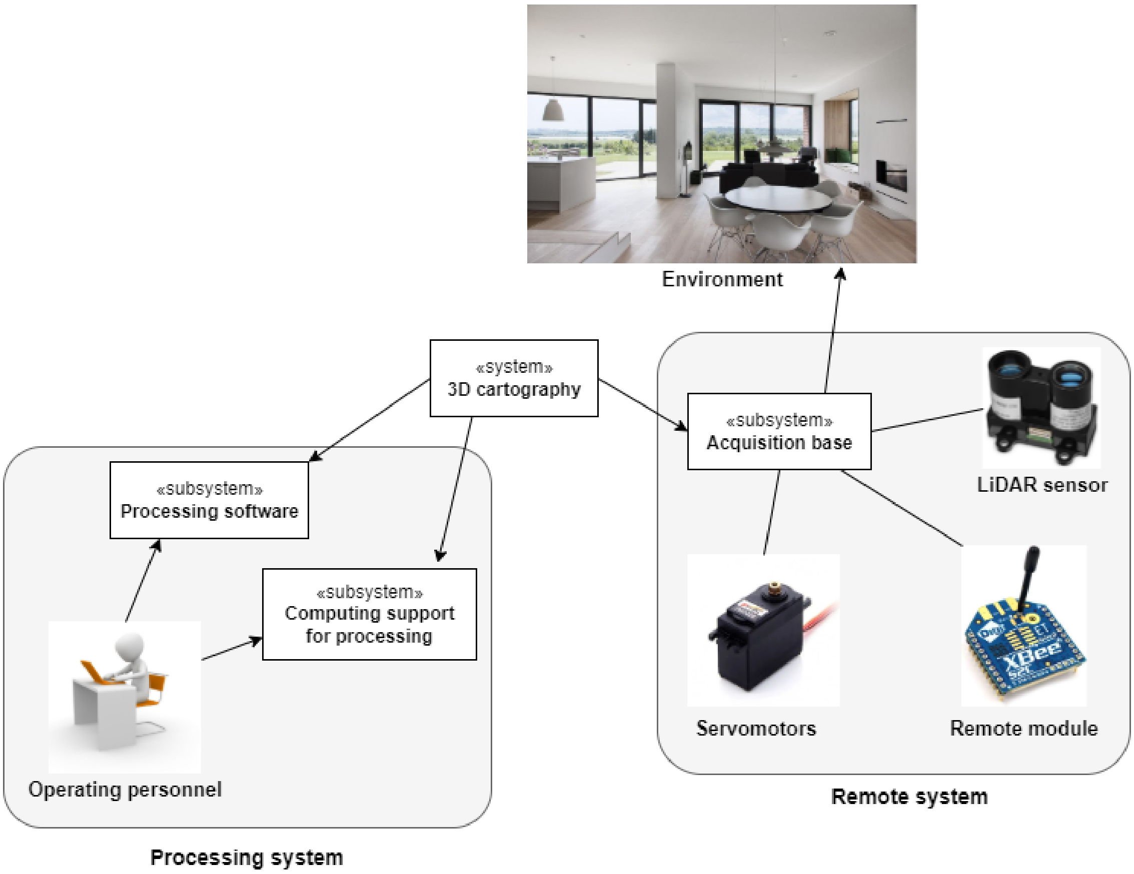


Figure 11: System context diagram

D. System Use Cases

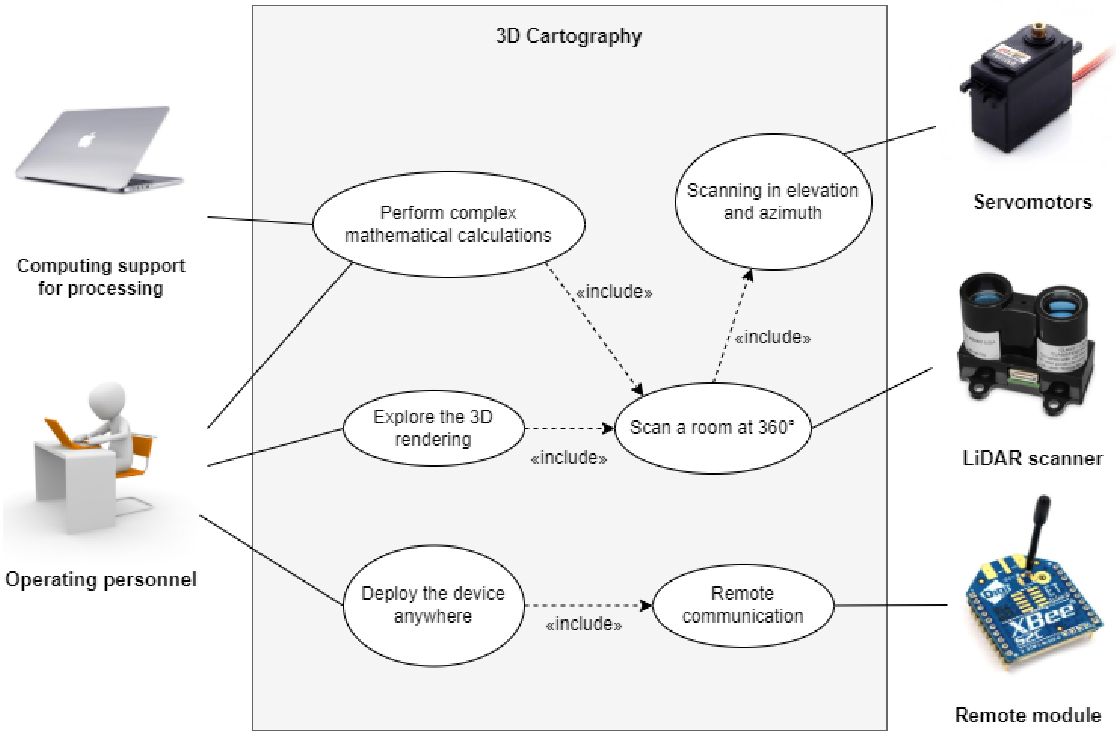


Figure 12: System use case diagram

E. Requirement diagram

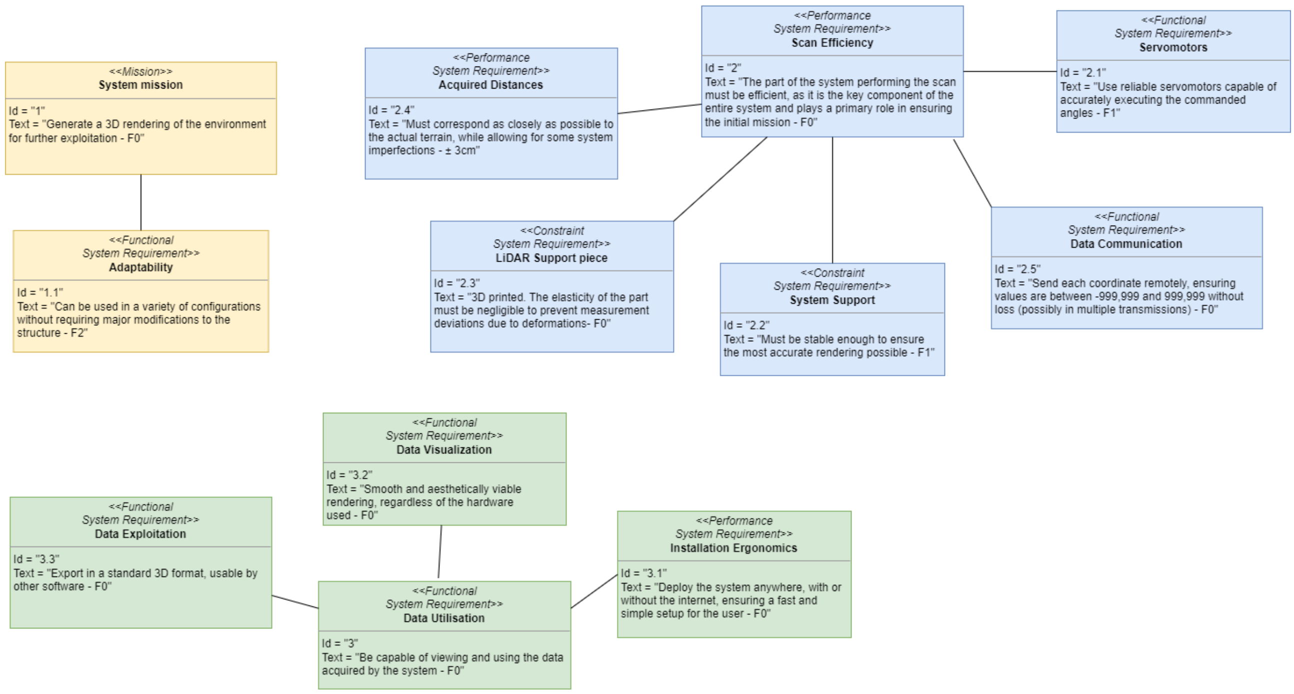


Figure 13: System Requirements Diagram

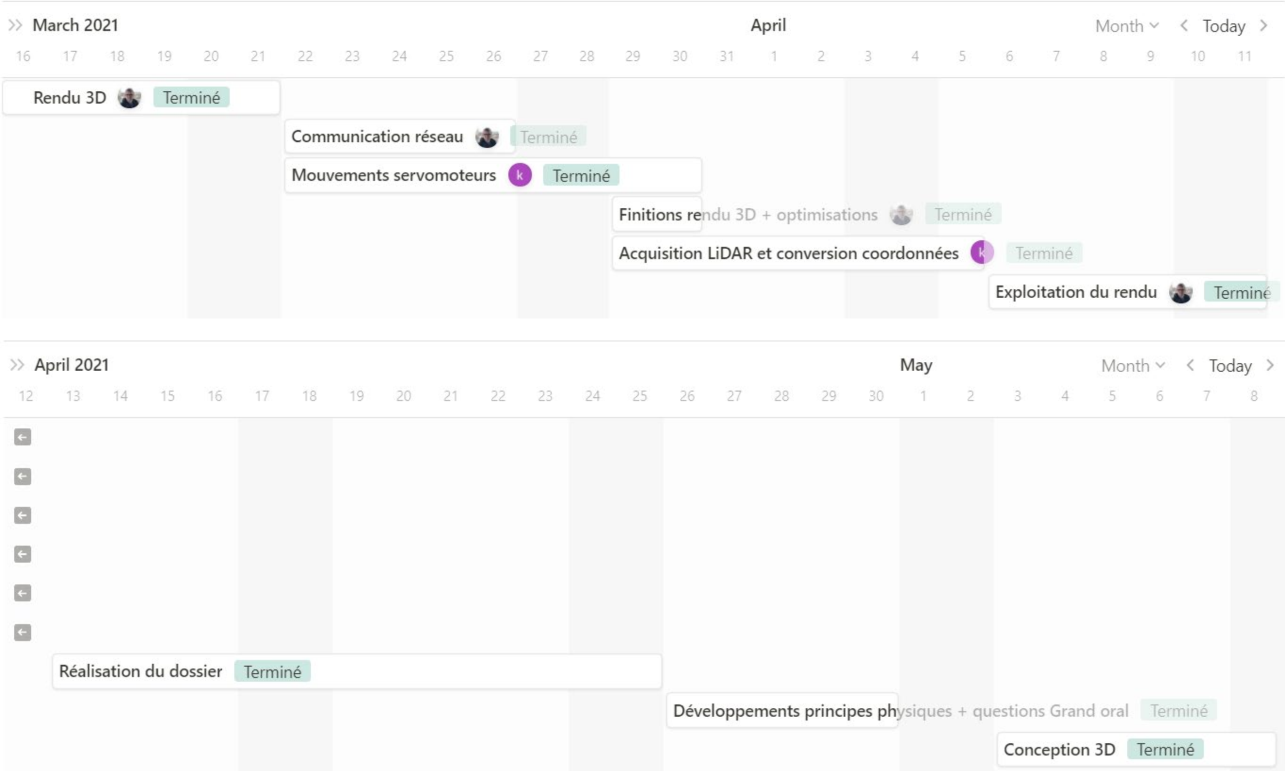
F. Summary of requirements

N°	Flexibilit...	Nom	Description
1	F0	Mission du système	Faire un rendu 3D de l'environnement proche pour exploitation
1.1	F2	Adaptabilité	Peut être utilisé dans une variété de configurations sans devoir apporter d'importantes modifications à la structure
2	F0	Efficacité du scan	La partie du système opérant le scan doit être efficace, car constituant la pièce maitresse de tout l'ouvrage et étant au front pour assurer la mission initiale
2.1	F1	Servomoteurs	Utiliser des servomoteurs fiables, capables d'appliquer à la lettre les angles commandés
2.2	F1	Support du système	Doit être sufisamment stable pour avoir un rendu le plus net possible
2.3	F0	Pièce de support LiDAR	Imprimée en 3D. L'élasticité de la pièce doit être nulle pour éviter les écarts de mesure par des déformations
2.4	± 3cm	Distances acquises	Correspondre au maximum à la réalité du terrain tout en admettant que le système peut avoir des imperfections
2.5	F0	Communication des données	Envoi à distance de chaque coordonnée de valeur comprise entre -999,999 et 999,999 sans pertes (éventuellement en plusieurs fois)
3	F0	Données acquises	Être capable de visualiser et d'exploiter les données acquises par le système
3.1	F0	Ergonomie de l'installation	Déploiement du système en tout lieu, avec ou sans Internet, rapide et simple pour l'exploitant
3.2	F0	Visualisation des données	Rendu fluide et estéthiquement viable peu importe le matériel
3.3	F0	Exploitation des données	Exportation dans un format 3D standard, exploitable par d'autres logiciels

Figure 14: Project requirements summary table

5. Project approach and planning

A. Initial approach and planning



Nom	Assigner	Date	Status
Rendu 3D	Théo Vidal	@Mar 16, 2021 → Mar 21, 2021	Terminé
Communication réseau	Théo Vidal	@Mar 22, 2021 → Mar 26, 2021	Terminé
Mouvements servomoteurs	kylian capitanio	@Mar 22, 2021 → Mar 30, 2021	Terminé
Finitions rendu 3D + optimisations	Théo Vidal	@Mar 29, 2021 → Mar 30, 2021	Terminé
Acquisition LiDAR et conversion coordonnées	kylian capitanio	@Mar 29, 2021 → Apr 5, 2021	Terminé
Exploitation du rendu	Théo Vidal	@Apr 6, 2021 → Apr 11, 2021	Terminé
Réalisation du dossier		@Apr 13, 2021 → Apr 25, 2021	Terminé
Développements principes physiques + questions Grand oral		@Apr 26, 2021 → Apr 30, 2021	Terminé
Conception 3D		@May 3, 2021 → May 8, 2021	Terminé
Montage, tests et ajustements		@May 10, 2021 → May 28, 2021	

The complete planning is freely available on the project's Notion page .

B. Actual progress: logbook

Name	Assigner	Date
Brainstorming en classe		@Mar 16, 2021
Étude communication XBee	Théo Vidal	@Mar 22, 2021
Rendu 3D : première application	Théo Vidal	@Mar 21, 2021
Étude structure servo-moteurs	kylian capitanio	@Mar 22, 2021
Essais Xbee	Théo Vidal	@Mar 23, 2021
Protocole de communication	Théo Vidal	@Mar 26, 2021
Avancement servo-moteurs	kylian capitanio	@Mar 26, 2021
Commande des deux servos	kylian capitanio	@Mar 29, 2021
Étude + branchements du LiDAR	kylian capitanio	@Mar 30, 2021
Optimisations rendu	Théo Vidal	@Mar 30, 2021
Dataset d'exemples pour le rendu	Théo Vidal	@Apr 2, 2021
Galères de câbles avec le LiDAR	kylian capitanio	@Apr 2, 2021
Premiers mouvements du lidar	Théo Vidal	@Apr 4, 2021
Intégration calcul deuxième angle	Théo Vidal	@Apr 5, 2021
Débuts de calculs avec la 3D	Théo Vidal	@Apr 7, 2021
Import et export + d'autres rendus	Théo Vidal	@Apr 22, 2021
Calcul de volume	Théo Vidal	@Apr 23, 2021
Modélisation assemblage et pièce support LiDAR		@May 7, 2021
Dessin de la structure + modélisation LiDAR et servo		@May 3, 2021
Impression 3D pièces et montage		@May 10, 2021
Essai en conditions réelles	Théo Vidal	@May 11, 2021
Tests infructueux	Théo Vidal	@May 15, 2021
Essais en classe	Théo Vidal	@May 21, 2021
Essais en classe		@May 18, 2021
Essais dehors		@May 25, 2021

The complete logbook is freely available on the project's Notion page .

The items in the Gantt chart and calendar can contain additional elements (explanatory texts, images, etc.) which can be viewed by clicking on them.

6. Description of the proposed response to the need

A. Structural description of the solution

Our solution uses the following components:

- 2 servomotors: one for the “Body” which scans in azimuth, the other for the LiDAR which scans in elevation.
- An Arduino Mega board with ATMega 2560 processor.
- XBee communication modules: one on the Arduino using a hat, and the other on the host computer with a USB adapter.
- GARMIN® LiDAR-Lite v3 sensor – 40-meter range, accuracy to plus or minus 2.5 cm for distances greater than one meter.
- Software to receive data and display a 3D rendering (described in section 6.C).
- Assembly parts modeled by CAD and prototyped by 3D printing (described in section 6.B).

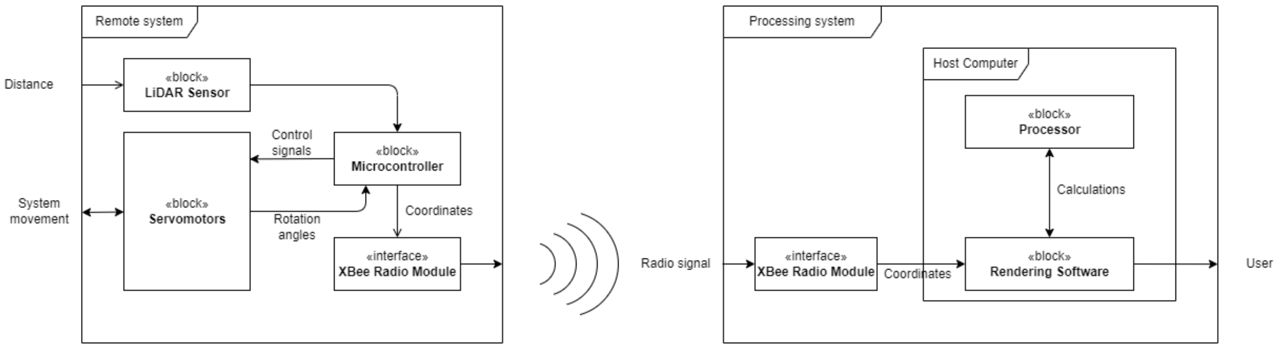


Figure 15: System block diagram

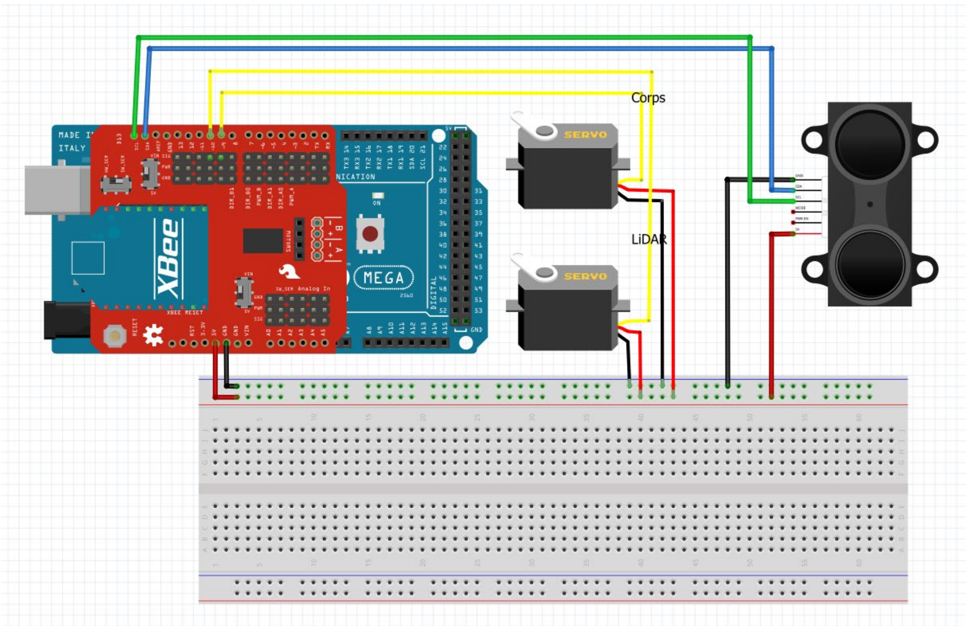


Figure 16 : Electronic wiring diagram of the components for the remote system

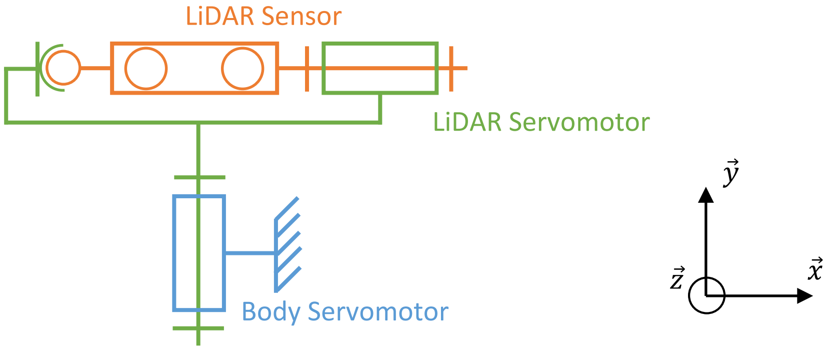


Figure 17: Kinematic diagram of the remote system

B. Modeling and simulation results

Being confined with little equipment, we had to improvise in order to carry out tests of the remote system. We replaced the 3D printed part with an ingenious design made of cardboard and tape worthy of MacGyver, which we regret not having as an official sponsor. A preview video is also available on [YouTube](#).

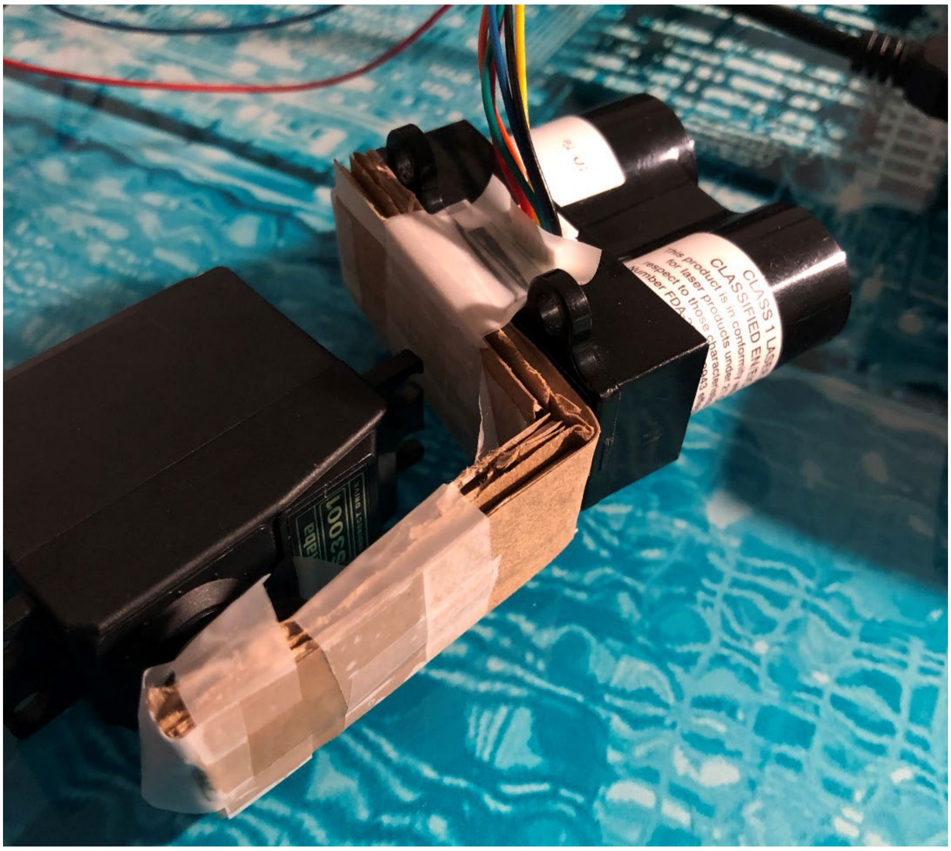


Figure 18LiDAR sensor attached to the servomotor by a piece of cardboard
© Théo's iPhone

This first test, although not very successful in terms of rendering precision, allowed us to adjust the rendering software to real conditions as well as to review the design in order to make the sensor less off-axis.

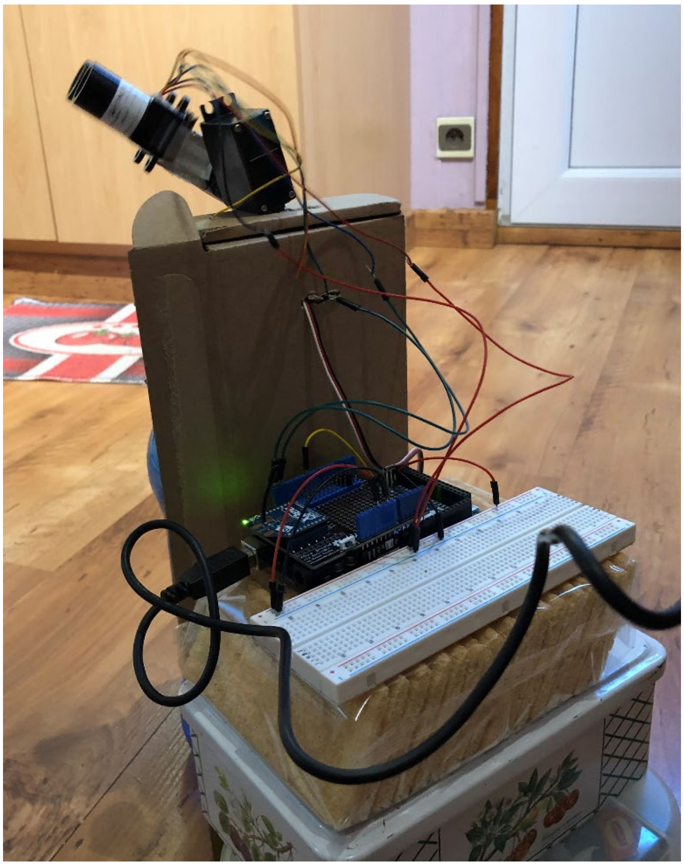


Figure 19: Autonomous positioning test of the remote system, using on-board resources for stabilization

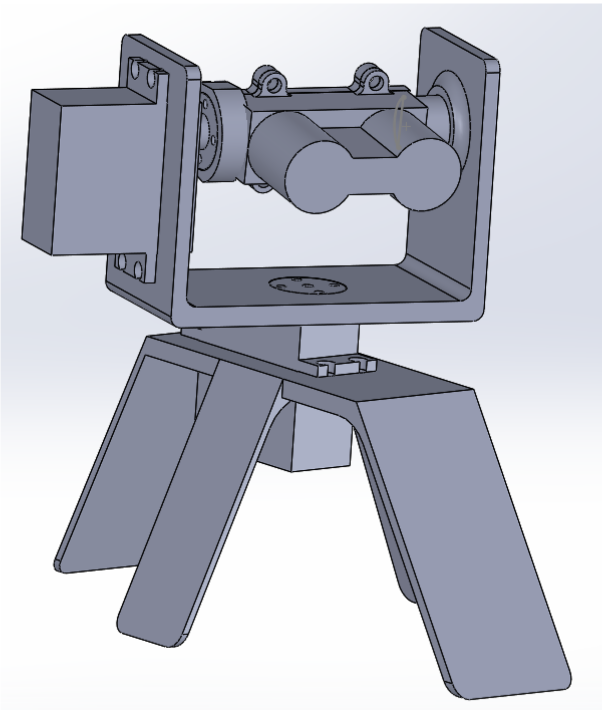


Figure 20 : Modeling the acquisition base in SOLIDWORKS

After the lockdown, we modeled the acquisition base in SOLIDWORKS by building the missing parts around the servomotors and the LiDAR sensor, also designed in this software. The final structure ensures the centering of the LiDAR on the two axes of rotation, for optimal scanning accuracy: this is how the servomotor driving the sensor is off-center and not centered as in the first iteration. After 3D printing, we assembled and wired it, then we started scanning tests of closed parts at our disposal. We therefore obtained much more realistic supports thanks to the complete scanning as well as the stability of this base. These tests are described in section 6.D of the file.

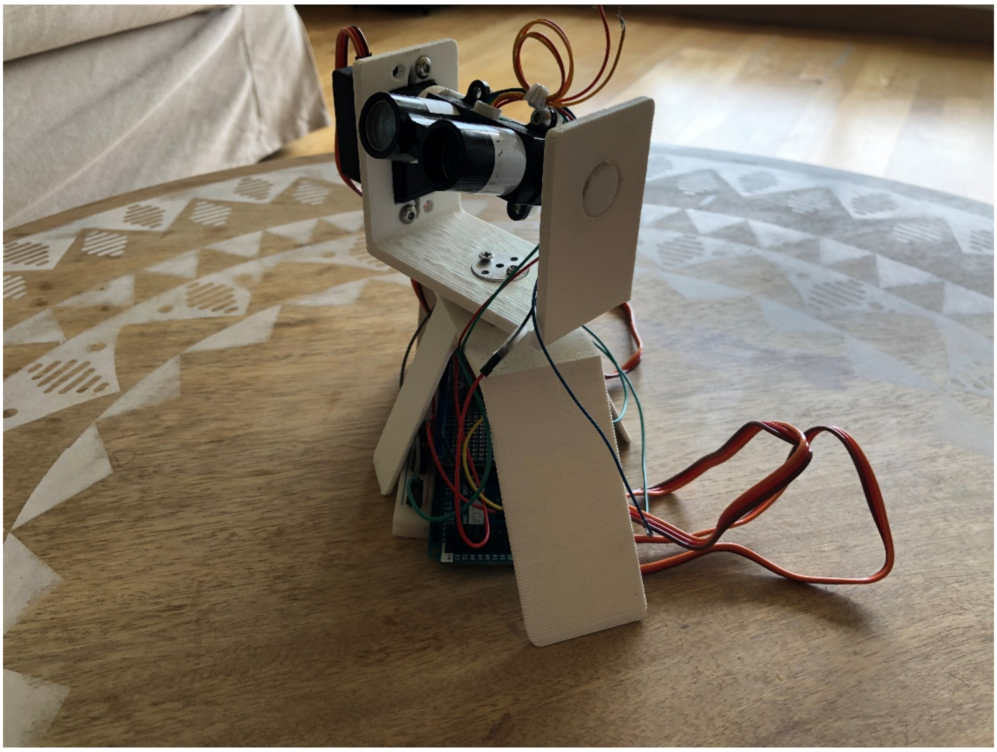


Figure 21: View of the acquisition base, named Charlie for the occasion

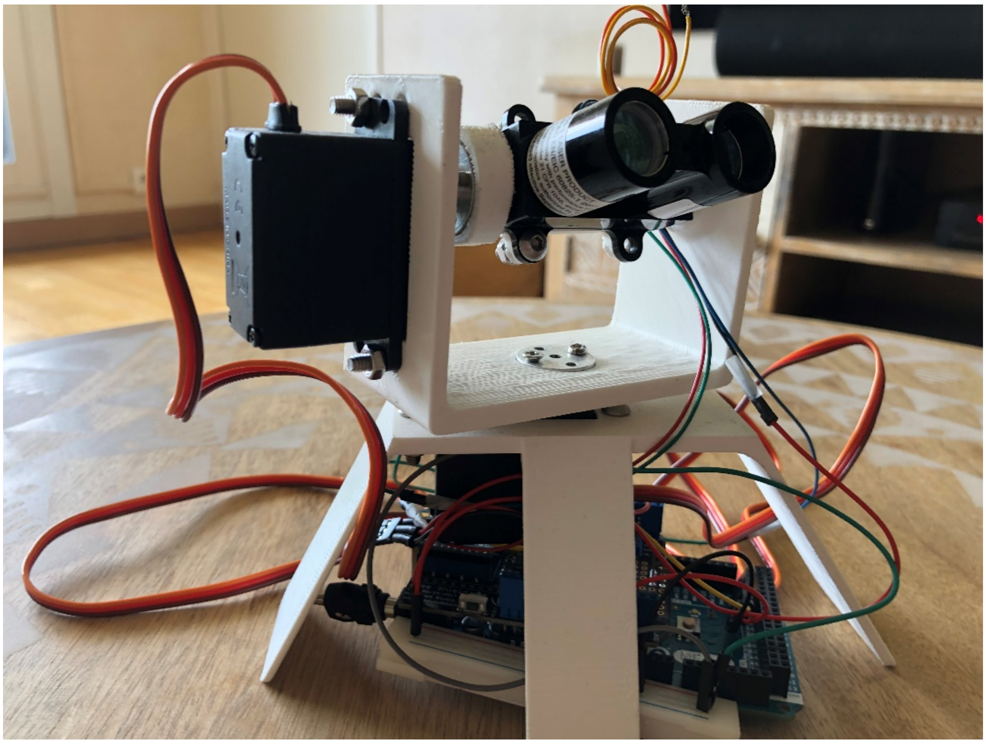


Figure 22: A splendid side shot of Charlie, ~~his gaze~~ his LiDAR turned towards the future

C. Programming/piloting the solution

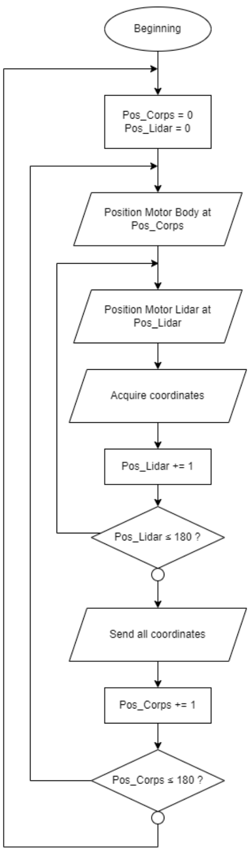


Figure 23 : Flowchart for controlling the remote system

All the code written as part of this project, for the remote and processing system, as well as the SOLIDWORKS assemblies and all the tests performed, are available in an open-source repository under the GNU GPL v3 license on GitHub.

The remote system uses an Arduino board , the script compiled and uploaded to it is therefore written in the Arduino language which is a derivative of C and C++.

The steering strategy consists of rotating the sensor 180° using the "LiDAR " servomotor, acquiring a point every degree, then shifting by one degree on the $\vec{y}$ axis with the "Body" servomotor and starting again like this. The process is described in the algorithm opposite.

To transmit the data to the computer responsible for storing and rendering it, we use the XBee radio module. The protocol for communicating the coordinates of each point is defined as follows:

- Each coordinate acquired in the form of a decimal number is multiplied by 1000 and rounded to the nearest unit. We obtain a number in the form XXX YYY with on the one hand the integer part of the coordinate, and on the other hand the decimal part rounded to 10^{-3}.
- These coordinates are then coded on four bytes (2 for the integer part and 2 for the decimal part), the whole is preceded by a start byte of value 1 and followed by a stop byte of value 0 then sent.

In total, 14 bytes are received in one or two times depending on the situation, a sharing which is decided autonomously by the XBee radio protocol.

The processing and rendering software is developed in JavaScript with the node.js environment and compiled as a native Windows application using the Electron library. The program is an instance of Chrome, a famous web browser: this makes application development much faster and more productive because web technologies are more accessible than machine code. 3D rendering is done by the Three.js library, using the standardized WebGL

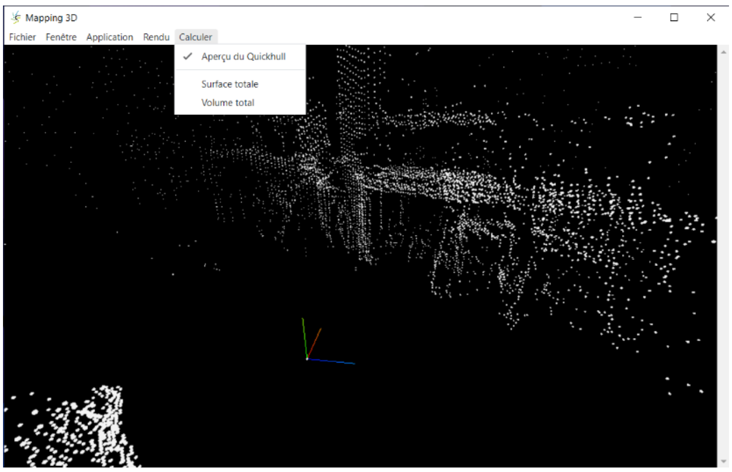


Figure 24 : Screenshot of the processing software under Windows 10

technology. Many open-source libraries are used, particularly for communication with the radio module.

D. Actual performance measurements and gap analysis

i. Real-world testing

Figure 27 : Modeling a house part. The circular arcs in the middle of the rendering are caused by a disconnection of the servomotors during the acquisition phase.

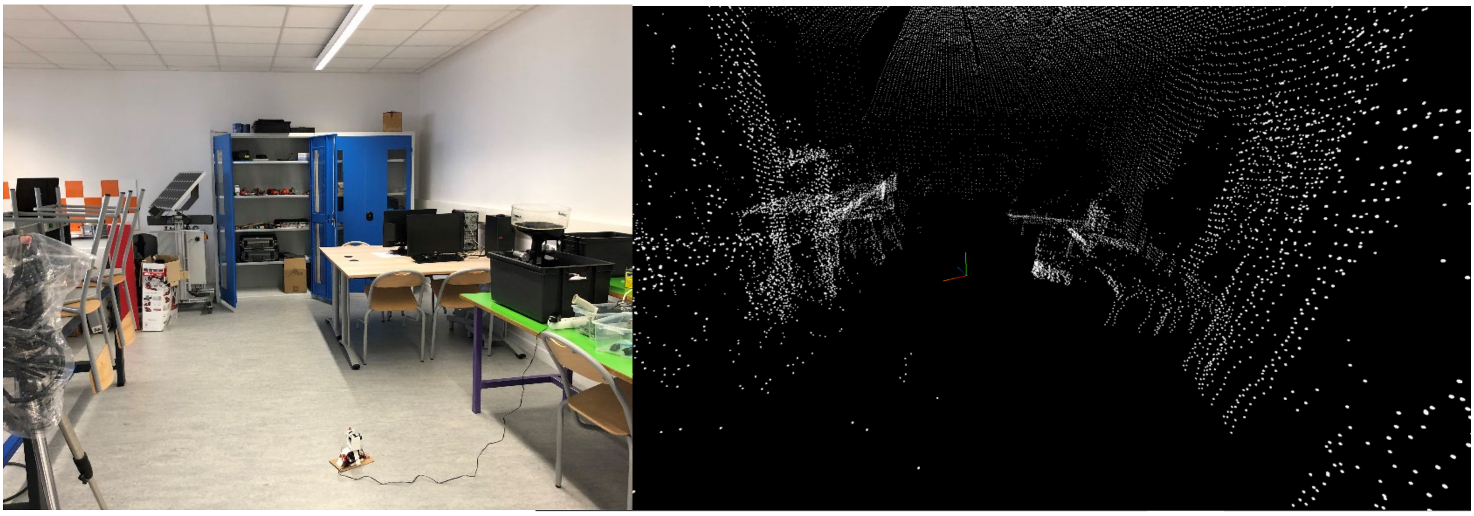


Figure 28 : Modeling a corner of the Engineering Sciences laboratory.

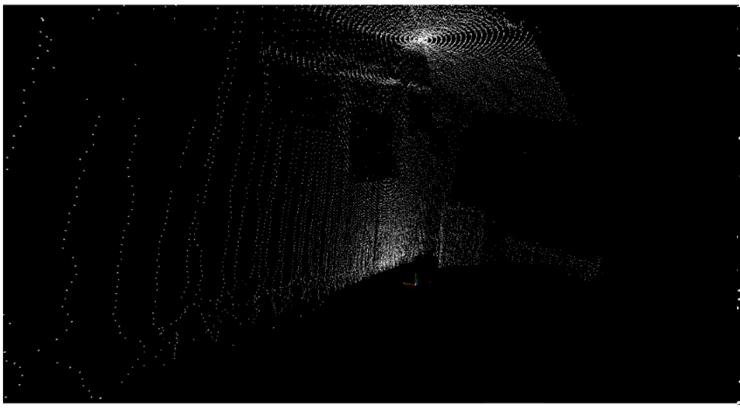


Figure 26 : Exterior in front of the laboratory, with the notable element being the solar radiator from another Engineering Sciences

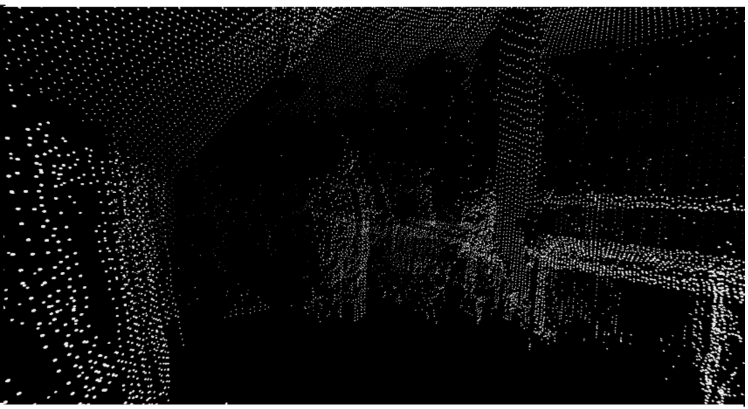


Figure 25 : Acquisition of our Engineering Sciences professor in the laboratory.

ii. Deviations and conclusions

- Distance deviations of 5 cm were observed between the LiDAR measurements and the actual distances, but this is easily corrected in the remote system program.

- Flat surfaces are clipped and curved, which is probably due to a minor misalignment but has a big impact when rotating step by step. In addition, distant objects are distorted: the distance scales are not the same everywhere. With more time in this project, we could have integrated point cloud smoothing and surface reconstruction algorithms.

- The servomotor driving the LiDAR does not rotate beyond 180°, so the ground and objects below it are not in its scan field.

- Some positioning problems were observed with the servomotors, which tended to jerk. This is probably due to the injected power being lower than the demand and their quality. The addition of capacitors is particularly recommended to guarantee the stability of the current injected into the motors.

- The system is generally slow (a full scan takes about half an hour), especially because the control strategy is synchronous: only one operation is executed at a time. A possible solution would be to use a processor instead of a microcontroller that would open up multithreading, for example by using a Raspberry Pi card instead of the Arduino card . This would also allow us to create a real "on-board computer" with on-site data storage and a rendering web server so as not to be dependent on a single external machine. However, this new structure requires major modifications that we will not make in this project.

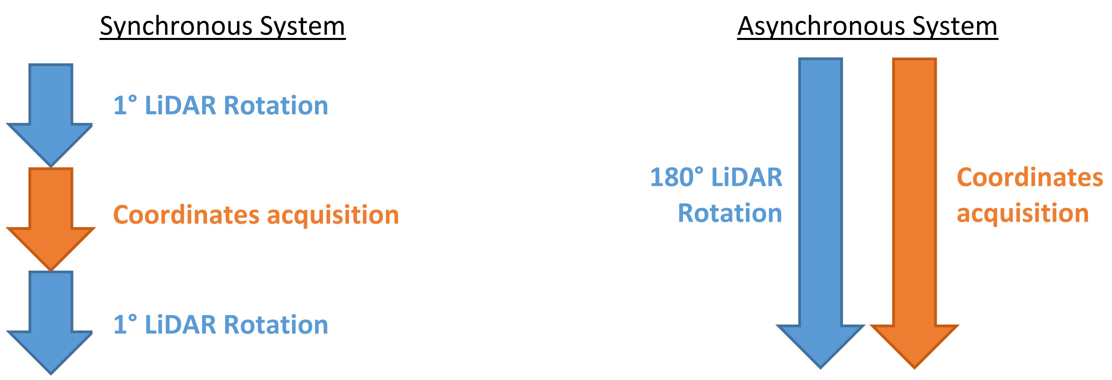


Figure 29: Diagram of the order of execution of processes in synchronous and asynchronous systems