# Portable 3D Scanner Feasibility Study

The University of Central Florida

Department of Electrical Engineering and Computer Science

Senior Design I

Group 17

Members and Sponsor:

1. Jean Cestin (Computer Engineering)

2. Sergio Arciniegas (Computer Engineering)

3. Rayan Hamada (Computer Engineering)

4. John Paszynski (Electrical Engineering)

**Abstract — The goal of this project is to study the feasibility of a portable 3D scanner. The scanner is intended to take an image of an object in its field of view (FOV), and proceed to create an image file which a user can access on their PC. The components that will be needed are a microcontroller to establish communication with sensors and a PC. Additionally we are using ultrasonic sensors, as they provide an affordable robust way of collecting data in a FOV.**

## I. INTRODUCTION

From the creation of the daguerreotype camera to the advanced camera's implemented into devices we keep in our pockets; cameras have constantly evolved allowing us to capture moments in time to forever remember. With the introduction of 3D camera's ability to capture objects in three dimensions from multiple points of view, technology will eventually advance and perfect it to the point 3D video's will be commonplace. Most 3D cameras, however, cost extremely large amounts of money, which is not ideal for commercial use. 3D cameras have many uses today. From fun photos of the family to 3D modeling, 3D cameras have many practical uses. From creating models for customers to view online, to models or references for video games, whether it be for work or education, 3D cameras will continue to be useful for a great number of applications. If 3D cameras become cheaper and cheaper, more people will purchase them, therefore increasing the demand to advance technology and improve the capabilities 3D cameras are lacking in.

Currently the market for scanners that can create a 3D model can cost easily up to $3000. Our group is motivated to find a way to build a reliable and efficient scanner that can also model 3D, but at a way lower cost. We feel that this is important as it can allow a lower barrier of entry for people that are interested in this area of study, which would be positive for the overall advancement of modeling. This would allow for advancements in space exploration, automated vehicles and even in certain medical practices with nanotechnology. So, we are trying to use a camera that can 3D scan an object using ultrasonic sensors for depth, upload images to a website/interface, and have an interface that a user can save and edit images, along with exporting them. These will give us access to create a low-cost 3D camera scanner with sensors that are portable and accurate enough to be similar to those that are already available on the market.

Additionally, 3d scanners are playing a much more important role in society and will be an essential part of the future. With artificial intelligence and the path towards automation, and bridging reality with technology, the role of having data to be recorded from the environment surrounding us would require sensors to capture and render images of it. There are many companies attempting to achieve autonomy in some form, whether it be Tesla with autonomous driving or a company focusing on worker safety, productivity, costs, precision and quality. One example of these applications has been by the company caterpillar which has implemented autonomous equipment for their mining operations ranging from drilling equipment to haul trucks. Given the shortage of workers during the pandemic, having these automated systems/equipment allowed for the company to relatively keep up to demand with limited workforce which shows another positive example that autonomous technology can provide, which is that even though a pandemic occurred, and many people could not show up to work, the job could still get done. We view the advancement of autonomous systems as imperative, which is why we chose to undergo studying the feasibility of a portable 3d scanner as it provides a means to further improve the space of autonomous technology.

## II.SYSTEM COMPONENTS

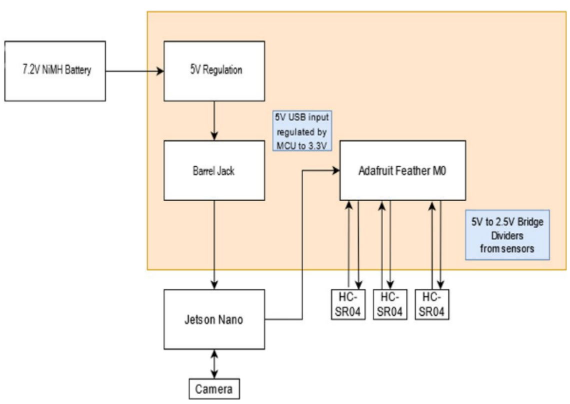



Fig 1. General Hardware Flowchart

### A. Adafruit Feather M0 and the STM32F1 microcontroller

The Adafruit is a lightweight board that is a suitable candidate for implementing it into the portable scanner. The controller boasts 32KB of on-board RAM and a connector specifically for 3.7V Lithium batteries. Additionally the board has the ability to be powered straight through USB 5V as this board regulates the USB voltage to 3.3V, along with the BAT and USB pins serving as more power options available on the controller.

While this microcontroller was a good choice for testing while in the development phase, we have to design our own PCB with its own dedicated microcontroller. This is where the STM32F103C8T6 comes into play. This microcontroller also uses a voltage of 3.3V, so when using the 12V battery, we had to make sure that after the 12V is regulated to 5V, we use an additional buck converter to convert the 5V down to 3.3 V, which then can safely be used for the microcontroller.

### B. Jetson Nano

For our development board, we needed something very powerful in order to process the images captured by the sensor data and also operate multiple accessories such as MIPI-CSI-2, although the FPGA is the ideal solution, it does come at a high cost along with a longer development going against the goals of our project which is why we settled with the NVIDIA jetson nano. This board suited us nicely to go with our camera sensors as it has a 2-lane CSI interface that are used by our camera. Also, we needed to find a development board that would be able to connect wirelessly to a PC, and the NVIDIA Jetson Nano has the ability to add a 802.11ac wireless adapter for uploading image captures. The board is also powerful enough to process our images as it has a dedicated GPU. We also needed to create a power management system on our PCB when mounting the battery with the development board and MCU.

### C. Ultrasonic subsystem

Our scanner is dependent on a set of 3 ultrasonic sensors that are collecting data from the FOV. The sensors collaborate with the microcontroller. The sensors which are utilized are the HC-SR04 sensors, which run on %V logic while the our microcontroller runs on 3.3V. The ultrasonic sensors will operate sequentially, which is done in order to prevent crosstalk from occurring.

| Feature | HY-SRF05 | HC-SR04 | URM37 V5.0 |
|---|---|---|---|
| Operating Voltage | 4.5V ~ 5.5V | 5V | 3.3V ~ 5.5V |
| Resolution | 0.3cm | 0.3cm | 1cm |
| Shape | Sqaure | Rectangle | |
| Current Draw | 10 to 40mA | 15mA | 20mA |
| Pins | 5 | 4 | 9 |
| Operational Range | 2cm - 450cm | 2cm - 400cm | 2cm - 800cm |
| Precision | ~ 3mm | ~ 2mm | |
| Dimensions | 45 x 15 x 27 mm | 45 x 20 x 15 mm | 51 x 22 x 13 mm |
| Price | $2.49 | $3.95 | $13.90 |

Fig 2. Sensors specs

How we solve the problem

At the core of our development is attempting to not only create an affordable scanner, but one that is affordable relative to the ones already existing in the market. One of the ways we approached affordability was by researching ways to scan an object, and then determining which option would be able to be optimized to lower the cost. We ultimately settled on using ultrasonic sensors in a way that can measure the depth of the object, and then be able to use our data to construct an image file.

We came up with some specifications for our design. Firstly, our device should operate as a portable device that can transmit data wirelessly to a PC over Wi-Fi. The device itself should not be longer than 10 inches, this is to make it user friendly and not something uncomfortable to be using. The device should also not weigh more than 5 pounds, we achieved this by focusing on components that were relatively light weight.

Figure showing our Power Management System on PCB

Lastly, for the image creation, our system consists of using 3 ultrasonic sensors that operate sequentially, preventing crosstalk, that collect an array of data in a form of triangles. This data is then processed as we must use trigonometry in order to get a true measurement of distance as the scanner scans at each point, otherwise not doing this can cause a flat surface to seem curved when the data is written to the image file.

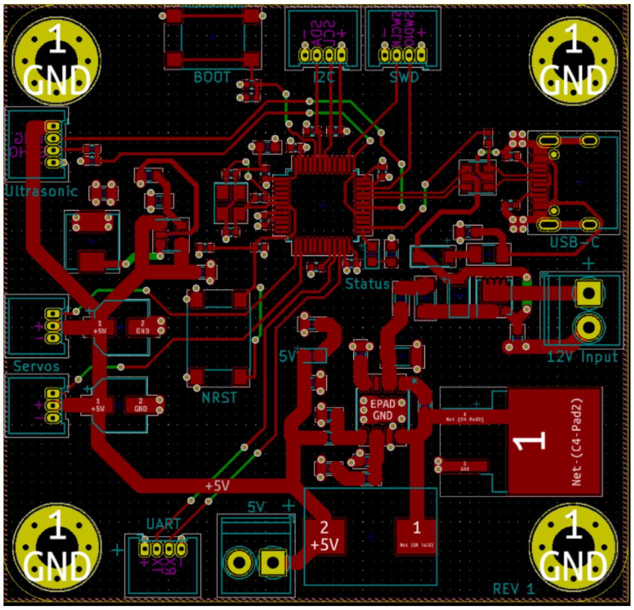
Design Implementation Efficiency

For the camera design we use one (1) camera to be able to provide image for the scan. The way that the system structure is similar to an MRI scanner. Having the camera focuses on an object that exists in a box. The camera is controlled by a shutter remote which connects to an MCU, the camera is mounted to a structure that can hold it at an elevated and stable place aiming directly towards the box. The front end of the box is illuminated by the object using high powered LED's and the object is rotated by utilizing a motor and an MCU. The collection of the images then be reconstructed to create a 3d scan of the object.

The limitation of this design is that although the design is accomplishing our goal of 3d scanning. The 3d scan would be pertained to the size of the box we made; therefore, we are limited to scanning a small object and not of a large open area which we find more useful for practical purposes. Additionally, the box that we need to situate the object, and basically have the camera attached to bring the portability of the scanner down a lot, as it is not really feasible to bring a large structure with a box around to scan. However, the novelty of using a design that mirrors that of how an MRI scan works did seem interesting and is something we keep in mind for our actual design when it comes to taking a creative approach to our system.

One of the core goals of our project was to study the feasibility of a portable scanner. With that said, we focused on designing our scanner to be lightweight, so that a user would be able to easily move and use the scanner. Components were selected for their smaller size, such as the ultrasonic sensors which allowed for us to not use bulky cameras that would make the scanner not so portable. Ultimately our prototype was created in such a way

This is our PCB design below.




Our software implementation is fairly simple, we have it split into three different phases. Our first phase is called the data collection phase, in which our Jetson Nano communicates with the microcontroller, which then communicates with the ultrasonic sensor and servos to scan points in a 2D array. These points on the 2D array represent the Z axis of our 3D environment scan, the X and Y being either pre-determined or calculated later.

The raw data is then sent onto the next phase of the software design, the Data Processing phase. The reason why we have to process the data is because when the data is scanned, it's scanned at an angle (an angle we know). Essentially using simple trigonometry we make it so that the angled line we scanned (the hypotenuse) is then transformed into a line parallel to the direction the ultrasonic sensor is facing. We also then account for the fact that someone might scan the object from a meter away, we find the shortest point and subtract all points by the shortest point in order to "zero in" on the image.

We then move on to the final phase, the image creation phase. In a .stl file, you can see that a 3D image is made up of facades (triangles basically). Each triangle is made up of three points on the xyz plane. X and Y are predetermined currently, and the Z is the depth we find while using the ultrasonic sensor. With this, we can traverse through the 2D array of processed data 4 points at a time to write two triangles at a time. After the program is done traversing through the array the .stl file is found on the Jetson Nano itself, which the user can then download through wifi or ethernet.

Here is an example of what one of our .stl files look like

```
solid model
facet normal 0.0 1.0 0.0
outer loop
vertex 0.0 8.023909236273086 0.0
vertex 6.3235837518209355 8.060209829871884 0.0
vertex 0.0 8.435103310331042 6.3235837518209355
endloop
endfacet
facet normal 1.0 0.0 0.0
outer loop
vertex 6.3235837518209355 8.473264167865171 6.3235837518209355
vertex 6.3235837518209355 8.060209829871884 0.0
vertex 0.0 8.435103310331042 6.3235837518209355
endloop
endfacet
facet normal 0.0 1.0 0.0
outer loop
vertex 6.3235837518209355 8.060209829871884 0.0
vertex 12.234328718899402 8.094055205475497 0.0
vertex 6.3235837518209355 8.473264167865171 6.3235837518209355
endloop
endfacet
facet normal 1.0 0.0 0.0
outer loop
vertex 12.234328718899402 8.225215854149702 6.3235837518209355
vertex 12.234328718899402 8.094055205475497 0.0
vertex 6.3235837518209355 8.473264167865171 6.3235837518209355
endloop
endfacet
facet normal 0.0 1.0 0.0
outer loop
vertex 12.234328718899402 8.094055205475497 0.0
vertex 17.730434431059756 8.030953250507503 0.0
vertex 12.234328718899402 8.225215854149702 6.3235837518209355
endloop
endfacet
```

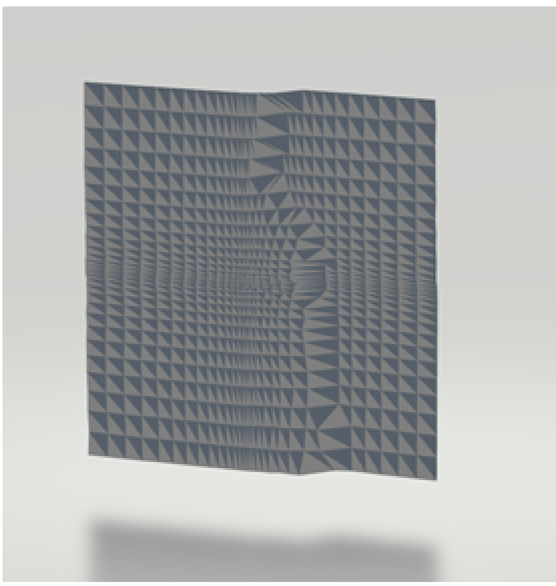
In the end our software design is straight forward, the only hang-ups are mainly component related with inaccuracies with the ultrasonic sensor or servo's (more on that later). One way to combat that is to have the ultrasonic sensor scan the same environment multiple times. We would repeat the data collection phase and data processing phase however many times needed. We would also have to add a data "averaging" phase where we take the average of the data collected, then continue with the image creation phase.

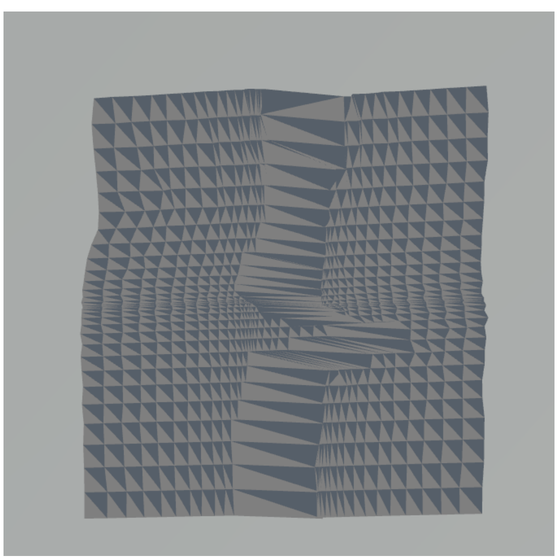
Functional Performance of prototype

Our prototype right now takes less than 15 minutes to scan the environment it's looking at. This is while the scanner had an FOV of 31 by 31, which brings the total number of points to 961. In the future after more testing and fine tuning is complete, we will be increasing the FOV to 45 by 45 so that our scanner can view a larger area. However, this would increase the number of points to 2025, which is more than double the points originally scanned. This would lead to the scanner taking more than

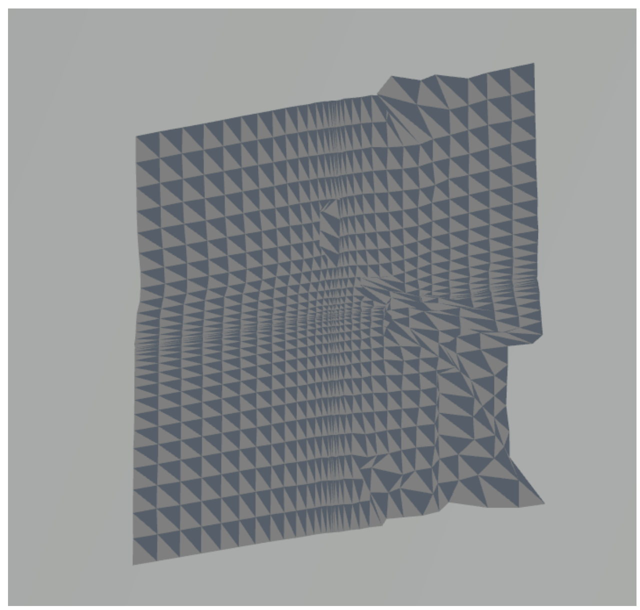
30 minutes to complete one scan. We are currently testing whether or not making the scanner run faster will negatively affect the scan. If the scanner running faster does not hurt the scan we will increase the speed, because if we want to be able to run multiple scans at the same time to average out the scan, we need to be able to run it faster so we don't end up spending 5 hours on one scan (with multiple runs), 10 if we increase the FOV.

Previously, when trying to scan a flat surface, we were unable to scan flat edges. The ultrasonic sensor would detect bumps or spikes when scanning just a flat edge.

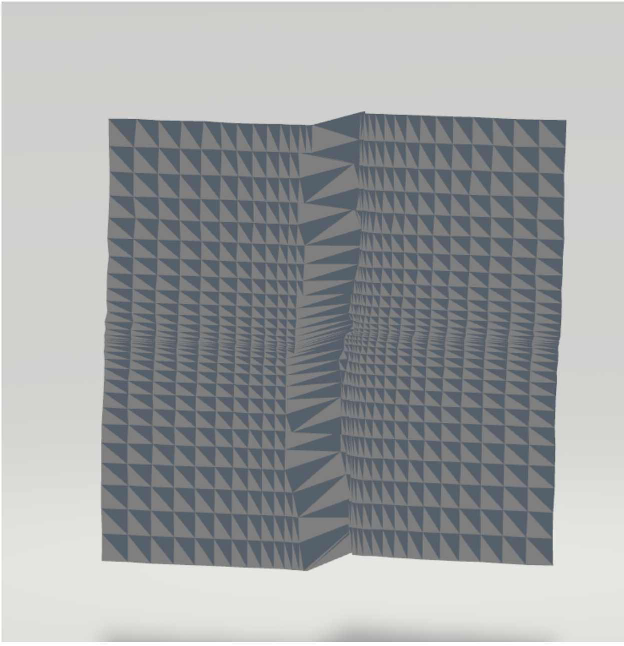




As you can see, the flat edge between the two surfaces is not straight. There were some issues that could have caused this problem. One is that the ultrasonic sensor we use is not accurate enough. This is simply a matter of changing the ultrasonic sensors to a different one that boasts more accuracy, or by changing the way that we collect our data, either infrared or lidar. The other reason might be that the servo's jittering causes inaccuracies, this is because the servo goes in a set path collecting data at every point, sometimes the servo might shake too hard when going in its set path that it begins to slowly deviate the actual point at which it moves to. We were given recommendations as to how to solve this problem as well. One, if the scan was not consistent in the deviations on the flat edge, we could have the scanner scan multiple times and average the data out. The other solution was if the scan deviations are consistent, then we should calibrate the machine. At first trying to average out the data was our plan, but unfortunately the final product would appear almost flat, decreasing the quality of the scan.

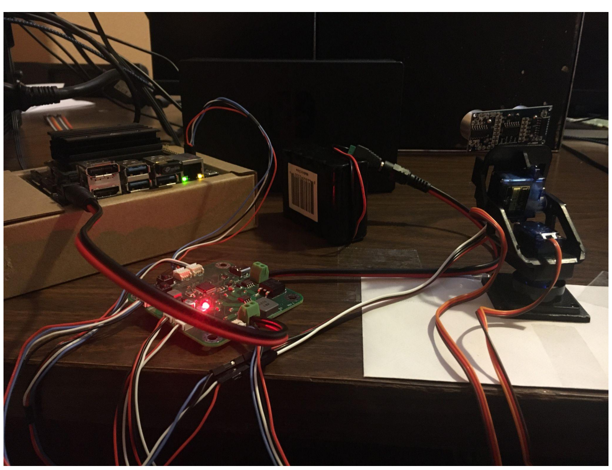


As this method seemed unviable, we determined that calibrating the scanner would be our best option. Upon further testing we discovered that the lower our scanner scanned, the more distorted the image would be. We raised the FOV by a few angles and the scanner was able to detect a flat edge.

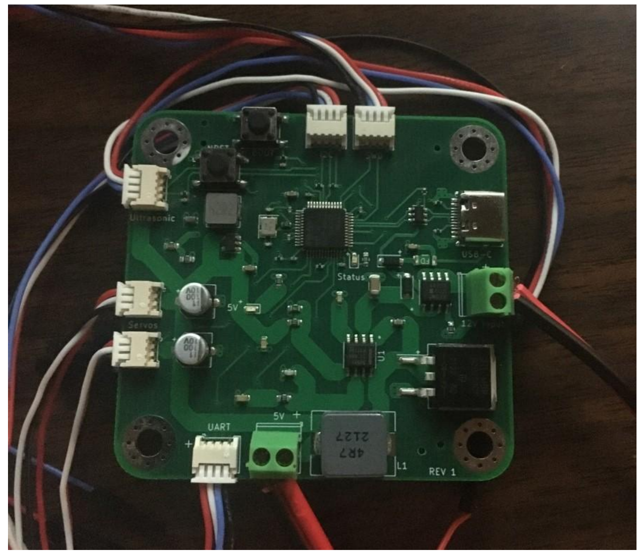


This shows that the scanner was able to scan flat surfaces with flat edges. This led to further testing with different edges and surfaces. When scanning a curved or round surface, the ultrasonic sensor will either determine the object is flat or nonexistent. This is most likely due to the nature of an ultrasonic sensor, using the bounce of sound waves to determine distance. The only way to fix this would be to change the type of sensor used. The other test we did was testing a flat surface with a round edge, this led to results where we could see the curve of the edge.

Prototype appearance and completeness



The Portable 3D scanner is complete in terms of components. We have the Jetson Nano with a WiFi adapter for communication between the computer and the Nano. The Nano is connected to the PCB through I2C, providing communication between the Nano and the Microcontroller.




The PCB also supplies power to the Nano. The PCB, powered by a 12V battery, is also connected to the Ultrasonic sensor and the two servos that move the sensor. We have no casing for the scanners components, so unfortunately the scanner looks a bit loose and messy.

Budget

The total cost of the project is imperative to our study as our intent was to create not only a portable scanner, but one that is affordable that gives our scanner an edge compared to the traditional scanners that already exist in the market. We wanted to ensure that the cost of the scanner was kept as low as possible, but also not at the cost of accuracy or ability. We wanted to find the most optimal way of ensuring quality engineering and design, but also affordability. Many of our component selections took into account their features, but also their price. Our PCB was designed to be small and lightweight, which led to a more compact (but more layered) design. The cost for our PCB was only increased due to the fact we ordered multiple, in fear of one malfunctioning. On the other hand we had more flexibility in addressing more affordable options on the sensors as SR04 sensors which we chose were extremely powerful for the price compared to the premium URM37 sensors which were nearly 4x more expensive. The flexibility in picking sensors, allowed for more budget allocation towards other components that had to be constrained by our projects core goals such as size and portability of the scanner. The scanner prototype did fit our vision of creating a scanner that is smaller in size, and relatively lightweight. Although it is not made in a way that can be handheld, there is definitely a way to make the scanner in a handheld shape,  our main goal for the prototype was to assemble the scanner to just scan and realize our goal of scanning for an object in its FOV.

When ordering the PCB, we ordered 5 boards with 2 of them assembled. The cost of all this was around $113, so the cost of one assembled PCB is about $50. So if added to the chart below, the final total becomes around $325. The total cost for one scanner as we have it would be $185 dollars. This is a much better cost compared to the $10,000 to $20,000 scanners on the market.

| Item | Price |
|---|---|
| Adafruit Feather M0 | $23.35 |
| 3pcs Ultrasonic Sensors | $9.99 |
| 7.2V NiMH Battery | $19.95 |
| Stereo Binocular Camera | $48.95 |
| Jetson Nano 4GB | $99.99 |
| Pan/Tilt Servo kit | $9.99 |
| Total ----------------------------------------------------- | $212.22 |

Table 1. Budget Summary

Work distribution within the group

We have two teams for the hardware and the software. The Hardware team composed of John Paszynski and Jean Cestin. The Software team consists of Rayan Hamada and Sergio Arciniegas.

For the software team of Rayan and Sergio, work is split between what phases in the software there are. Sergio is in charge of the data processing phase, while Rayan is in charge of the image creation phase. The data collection phase is split into two parts, what the microcontroller does and what the Jetson Nano does. Rayan is in charge of the microcontroller and its interactions with the servos and ultrasonic sensor, while Sergio is in charge of the Jetson Nano, communicating with the microcontroller to obtain and store the data.

The hardware team is less defined than the software team, as in work isn't selected for one person or the other. Both members focus on working together for design and testing. However, if one were to "split" the work between the two, it would be clear that John was mainly focused on design (of the PCB) and Jean was more focused on testing parts and components.

Conclusion

The portable 3D scanner prototype which we assembled shows that with enough time and effort it is definitely feasible to create a portable 3D scanner. The only thing we would change is the type of sensor we would use. Different sensors we could use would be Lidar, infrared, or Time of Flight. The reasons for why those weren't options during our time prototyping is because of multiple reasons, the most impactful is that we didn't have enough time. So overall, in terms of feasibility, it is very feasible to create a good 3D environment scanner for a much lower cost than what is on the market. Testing has shown that some scans are really good for something that is worth less than 2% of 3D scanners on the market today. With more time, more improvements could be made, but with what results we have gained we can safely say that is is definitely feasible to create a low cost 3D Portable Scanner.

The Engineers

**Jean Cestin** is a computer engineering student at the University of Central Florida.

**Sergio Arciniegas** is a computer engineering student at the University of Central Florida.

**Rayan Hamada** is a computer engineering student at the University of Central Florida.

**John Paszynski** is an electrical engineering student at the University of Central Florida.