Mobile 3D Printing Robot
Project Document

Team 14

Konstantin Markov, Ethan Masiel, Oluwaseun (Sam) Popoola, Xinrui Zhou

10/18/2022

1. Overview

1.1. Executive Summary

This mobile 3D printer is designed to print on surfaces at all angles using computer vision and a unique configuration. The basic printer architecture is now successfully operating after a year of work. This year the mobile 3D printer project is a multidisciplinary project involving 3 teams. The ECE team, the ME team, and the Graduate student team.

As the ECE team, our ultimate goal is to pass the coordinates of the surface's position relative to the robot at that point to the computer when the robot is moved to a position suitable for printing on the specified surface. In this process, we need to collect information about the environment, the robot, plan the path, move the robot, compare and select the surface, and calibration. Some features that involve getting the robot to the print surface and accurately print required us to collaborate with the ME team. They added a stable, liftable balancing structure to the robot, and the physical movement system allowing the robot to move with the position information collected. At this stage, we have completed and tested the whole system.

The ME team has made a lot of progress and their initial prototype is almost complete. All the components needed for the ECE team module are in place, although more may be added to facilitate/debug. The decision was made to include bluetooth capabilities in order to interface with a machine capable of running slicing software.

1.2. Team Contacts and Protocols

TABLE I
GROUP MEMBER

Members	Email	Roles
Konstantion Markov	markovk@oregonstate.edu	Lidar, Planner
Ethan Masiel	masiele@oregonstate.edu	Wheels, Touch sensor, Bluetooth
Oluwaseun (Sam) Popoola	popoolol@oregonstate.edu	Microcontroller, Software
Xinrui Zhou	zhouxin@oregonstate.edu	Camera, Planner

TABLE II
TEAM PROTOCOLS AND STANDARD

Topic	Protocol	Standard

On-time Deliverables and Team Collaboration	All individual work should be fully completed by the set deadline so that the team can review and make final changes together.	Work includes all necessary content and formatting requirements for a complete assignment as listed in Canvas and is as free of errors as possible.
Internal Communication Management	Team will use Discord to communicate, and Teams to post any links related to the project.	Team communicates in Discord, with questions and answers, and when links are shared on Teams, a message will be sent out in Discord. No confusion together.
Assignment and Note Management	Team will use Google drive for class assignment, and Onenote to record team agenda, weekly meeting, and weekly update.	Team creates assignments in Google Drive, and writes necessary content in Onenote. No confusion together.
Individual Work Management	Team will use Onenote to record the individual weekly update following the fixed format.	Team writes self update on Onenote followed the format, date, name, update.
Related to Project Partner	Team will need to meet with Dr.Campbell each week, and use Teams for all team communication.	Team meets with Dr. Campbell every Friday at 1pm in person, and talks about any achievements and problems in this week. Teams use Teams for communication with Dr. Campbell and other teams.
Cross Team Communication	Team will meet with the MechE team every Friday at 9:00 AM.	Meeting with MechEs ensures that each group clearly understands what the other is doing and any changes that may impact the other group will be communicated and confirmed by the other group. Meeting notes will be uploaded to Team Drive.

1.3. Gap Analysis

This project is to make the 3D printer more suitable for different environments as well as expand printer capabilities to the level of adding material to partly manufactured parts. The project will also work as a proof of concept for printing on irregular surfaces, as well as mobile 3D printing in general.

The printer can automatically gather environment information, determine its location, move itself to the surface and align the printing path. Some assumptions we are making involve the requirements of the project partner. These include a boundary of a 10x10 square when navigating to the part, as well as knowing the dimensions of the part to be worked on.

Our group has mainly gathered information from our project partner, as well as the graduate student team who have been working on the project and the previous team who were assigned the project in the previous year.

The end user of this project will likely be those who are studying the technology we used in our design to create a more capable version that will be both easier to use and more independent. The derivations of this project will likely be put to use by NASA, various manufacturing companies, and wider society as the technology becomes cheaper and more refined.

1.4. Timeline/Proposed Timeline

Please see 1.5.2 "timeline" link.

1.5. References and File Links

1.5.1. References (IEEE)

[1] "IEEE referencing style: Images, figures and tables," *Guides*. [Online]. Available: https://guides.library.uwa.edu.au/IEEE/images_tables_figures#:~:text=Place%20the%20Figure%20title%2Fcaption,captions%20should%20be%20centre%2Djustified. [Accessed: 13-Oct-2022].

[2] "How to cite references: IEEE documentation style - IEEE dataport." [Online]. Available: https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf. [Accessed: 14-Oct-2022].

1.5.2. File Links

[1] "Timeline," Google Sheets. [Online]. Available: https://docs.google.com/spreadsheets/d/1VcrObmRsxS33vXSDnUR2YAtS8Ef2u0aOw4VM8GPqeLg/edit?usp=share_link. [Accessed: 18-Nov-2022].

1.6. Revision Table

TABLE III
REVISE TABLE

Date	Author	Revise content

10/21/22	Sam	Executive Summary revised: to include project statement
10/28/22	Konstantin	Grammar errors corrected
11/04/22	Xinrui	Combined contacts and roles and update members' role
11/11/22	Xinrui	Revised executive summary: update the current state, project purpose
11/18/22	Xinrui	Revised team protocols: update the new protocols and standards
11/18/22	All members	Update the timeline
11/18/22	Xinrui	Deleted reference file links
11/18/22	Ethan	Updated formating
11/18/22	Ethan	Revised Gap analysis
11/28/22	Ethan	Revised executive summary: update the current state
05/14/2023	Xinrui	Updated summary and timeline

2. Impacts and Risks

2.1. Design Impact Statement

2.1.1. Introduction

This paper contains an assessment of the potential impacts for the Mobile 3D Printing Robot, with the aim of providing the reader with an idea of the possible consequences of the project or the possible continuation of the consequences. The assessment will remain realistic and unbiased, and we will make great effort to minimize bias, but it cannot be denied that there will be subjective bias in any assessment.

This assessment will first discuss the public health, safety, and welfare impacts that need to be considered. Then, the cultural and social impacts of the project will be explored.

This will be followed by environmental impacts, and finally, economic impacts will be discussed. The potential benefits this project can provide are massive, however there are some negative effects that will need to be addressed. The negative impacts raised in this paper may not be resolved within the scope of the project, but they need to be recognized and acknowledged so that they are more likely to be improved in the future.

For background on the project, we are upgrading an existing 3D printing robot architecture to allow it to automatically scan the environment, move, find print surfaces, calibrate and print. In this process, we need to use the vision system, sensing system, movement system, etc.

2.1.2. Public Health, Safety, and Welfare impacts

The benefits this robot will have for safety are potentially enormous. The functions performed by the Mobile 3D Printer Robot would be identical to those performed by humans in certain repair and manufacturing jobs. By supplanting the jobs traditionally performed by humans the robot could lower the risk of injury or death to near zero by replacing valuable manpower with expendable robots. 3D printing in construction is rapidly becoming more prevalent[1], with construction automation reducing the amount of physical labor construction workers must perform. For example, when 10 onsite construction robots were put into use across a variety of construction projects in several different countries, they were found to reduce repetitive site work between 25 and 90% and reduce time spent on hazardous tasks by 72% on average[2]. While the robot will have less of an impact on Public Health, overall there should be a small decrease in injuries caused by manufacturing, as well as an increase in the amount of repairs that can be performed at any given time - potentially preventing accidents from occurring through the use of preventative maintenance. The potential effects on welfare are hard to predict, besides the aforementioned safety and public health effects mentioned earlier. However, with automation becoming more prevalent, injury prevention could be massively reduced.

Impacts to public welfare could be more detrimental. With the reduced risk of health hazards to the workforce, companies could use this technology to replace human labor and reduce the expenditure of industries harmful to the environment and public health. This could replace workforces in developing nations, limiting income streams many people depend on and forcing them to perform less well paying jobs. Solutions to these measures would be reliant on government legislation that either taxes companies that produce profit with robotic workers, or even a specific tax levied upon computer automated jobs.

2.1.3. Cultural and Social Impact

When talking about the social impacts of our robot, we can talk about how 3D printing is changing how we interact with our cultural heritage. According to article[3], the recent rise of 3D-printed art in museums throughout the world shows how there is a market for it. It states how one of the most notable digital recreations - Tutankhamun's tomb - allows tourists to experience the inside of the tomb without “harming the original burial site”. There is also the topic of the accuracy that digital recreation can get. The same article talks about how the digital recreation of previous pieces allows for efficient ways to replicate historic pieces. With the growing availability of 3D printing robots like this, digital versions of pieces can allow one to download scans and print out replicas for personal and educational use. Another effect that the 3D printer can have is the streamlining of emerging products. It can help[4] the development of emerging products. That is to say that the existence of the printer could aid the rapid development and prototyping of ideas. That means if a team is trying to express ideas during the research and development phase of the project, they can use bots like this to visualize previously unattainable or prohibitively expensive designs to members of the team to improve communication between departments.

While this project will allow the creation of many copies of actual historical objects, the ability to have an item that is functionally identical to the actual artifact may cheapen the cultural impact of seeing the actual item. With the availability of 3D printing, this could allow a museum to create any display it wants without having to acquire any cultural artifacts to make the experience more momentous. It may even devalue the actual artifacts that are used to create such culturally enriching experiences. The way to solve this issue would be to use such technology to enhance actual exhibits, and allow actual artifacts to be enhanced by being displayed with recreations and other displays.

2.1.4. Environmental Impacts

In terms of environment, 3D printing has the characteristics of material saving and high precision compared with traditional manufacturing technology. Traditional manufacturing technology is a serious waste of resources, for larger and complex fine section often need to take cutting assembly processing, most of the raw materials will be cut off as waste and discarded, causing serious harm to the environment. The 3D printer uses the principle of one-time molding, regardless of the complexity of the object, can be formed once, the waste of resources is zero. At the same time, compared to the large amount of waste gas and noise pollution generated by traditional manufacturing, 3D printing has the characteristics of high material utilization and no noise during the printing process. Also, as a mobile printer, it reduces harms to the environment caused by the traditional manufacturing industry due to a fixed address and a long time production.

However, it also may affect our surrounding environment. For the 3D printer itself, its product production cycle is not zero pollution or zero waste. During the use of the printer, toner dust will leak, and dust containing a large amount of iron and silicon powder will be emitted[5]. The printing ink also contains heavy metals such as lead, cadmium, and tribute. For 3D printers, the fumes formed during the printing process are called UFPs (Ultrafine Particles), and in experiments with animals and humans, ultrafine particles can affect changes in lung function, allergic reactions, thrombosis, or change the heart rate. Also,3D printers mainly come from electricity supply[6]. Anyone who has witnessed the whole process of 3D printer printing will know that a three-dimensional disc model with an irregular shape about 30 cm in size requires a 3D printer to work for 28 and a half hours. In addition to the high energy consumption throughout the manufacturing process, 3D printing production relies heavily on plastics. Unlike injection molding (the traditional method of making plastic objects), which can leave a very small amount of unused plastic raw material behind, industrial-grade 3D printers for plastics print using powdered or molten polymers but leave a large amount of unused raw material behind. Although materials can sometimes be reused during the printing process, most of them cannot be recycled due to damage to material properties and have to be discarded directly. Although the ultrafine particles formed by general 3D printers will not have an excessive impact on the human body and are within the acceptable range of the human body, it is certain that the printer must be placed in a ventilated place and a location far away from the human body is more beneficial. Also, mobile 3D printing robots may damage ground vegetation during the movement if they are outdoors. To address these effects, improve the vision system to identify vegetated areas and detour. Also, find alternative materials to reduce pollution or ways to recycle existing materials and improve the efficiency of 3D printing and reduce energy consumption.

2.1.5. Economic Factors

Implementing 3D printing robots to help with repairs, for example, will help lower the amount of money large corporations have to spend on repair fees. The addition of the robots will create more time and money spending efficiency in warehouses. The robot has potential to do various things such as 3D print, weld, and spray making it all the more beneficial for large companies. The use of the robot reduces dealing with sickness, fatigue, stress, and scheduling, taking away possible human error that can occur[7]. Along with the robots more positions for technicians and engineers will open up in order to maintain the robots. Using one robot for repetitive repairs will be cheaper than having to either hire someone every time or keep someone on a payroll. But the robot will unfortunately supplant a large amount of jobs that are currently filled by humans, potentially decreasing demand for those types of jobs. A possibility is that the robots will fill the role as those jobs increase in demand, and current workers will not be affected. A study done by MIT showed that in a span of 17 years, increasing the amount of robots reduced the average employment-to-population ratio in a zone by 0.39 percentage points, and average wages by 0.77%[8]. A potential mitigation for the unemployment the introduction of this robot could cause would be to retrain current workers to operate the Mobile 3D Printer Robot.

Although the price of a few robots and an employee may be large, the robots are a one time purchase. So, economically through the course of time the benefits of having mobile 3D printing robots out ways keeping hundreds of employees on payroll.

2.1.6. Conclusion

Overall, it is clear that our project could have a variety of different positive and negative effects. By recognizing these, to the extent that we can address them, we can improve our design, and for the impacts outside our scope, we provide more information and possible solutions for the future development and application. For sure, our contribution to the project will contribute only a small part of the total effort being directed towards the normalization of robots in unstructured environments. The most significant impact of our project will be the data it generates on the best way to create automated 3D printing robots, and the impact it will have on future projects.

2.2. Risks

TABLE IV
RISK ASSESSMENT AND ACTION PLANS

Risk ID	Risk Description	Risk Category	Risk Probability	Risk Impact	Performance Indicator	Action Plan
R1	Using LiDAR system	Cost	L	H	Loss of significant amount of budget and time	Acquire part ASAP
R2	Camera will not function within scope	Technical	L	H	Computer vision doesn't work	Troubleshoot, and look into replacement
R3	Component outside of ECE team responsibility falls	Technical	L	M	Components from ECE and ME team don't work well together	Collaborate with other team and attempt to finish without reaching
R4	Camera is not stable	Technical	M	M	Images captured come back blurred	Work with ME team to find a mechanical solution to stabilize camera
R5	Team member drops class	Organization	L	H	Reorganize group roles	Review team member areas of responsibility and distribute the mission to the rest of the members

R6	Touch Sensor overheats	Technical	L	M	Melted or just dysfunctional touch sensor	Buy a new touch sensor (same one) and reapply the code. Adjust position of the sensor
R7	Communication between Camera/LiDAR and microcontroller doesn't work	Technical	M	H	Robot doesn't drive to the correct location and/or stabilize to print surface	Going through the code to fix the issue (debugging).
R8	Sipeed Maixduino cannot establish communication to Duet	Technical	L	H	Loss of time in implementation and delay to finished prototype.	Consider raspberry pi. Which is known to communicate with the Duet and move over platform for the projects implementation

2.3. References and File Links

2.3.1. References (IEEE)

[1] Brosque, Cynthia, and Martin Fischer. “Safety, Quality, Schedule, and Cost Impacts of Ten Construction Robots - Construction Robotics.” *SpringerLink*, Springer International Publishing, 27 June 2022, https://link.springer.com/article/10.1007/s41693-022-00072-5. https://link.springer.com/article/10.1007/s41693-022-00072-5

[2] Hossain MA, Zhumabekova A, Paul SC, Kim JR. A Review of 3D Printing in Construction and its Impact on the Labor Market. *Sustainability*. 2020; 12(20):8492. https://doi.org/10.3390/su12208492

[3] Samaroudi PhD Candidate, M., & Rodriguez Echavarria Principal Lecturer, K. (2022, October 20). “*How 3D printing is transforming our relationship with Cultural Heritage”*. The Conversation. November 4, 2022, from

https://theconversation.com/how-3d-printing-is-transforming-our-relationship-with-cultural-heritage-112642

[4] Yarlagadda, Ravi Teja, Future of Robots, AI and Automation in the United States (February 8, 2015). IEJRD - International Multidisciplinary Journal, vol. 1, no. 5, p. 6, Feb. 2015, Available at SSRN: https://ssrn.com/abstract=3803010

[5] M. Shuaib, A. Haleem, S. Kumar, and M. Javaid, "Impact of 3D printing on the environment: A literature-based study," *Sustainable Operations and Computers*, vol. 2, pp. 57–63, 2021.

[6] "The environmental impact of 3D printing," *Tangible Creative*, 18-Jul-2021. [Online]. Available: https://tangiblecreative.com/the-environmental-impact-of-3d-printing/. [Accessed: 04-Nov-2022].

[7] Siraphet, Madison. "What Are the Benefits of Automated Warehouse Robots." *SRSI*, 20 July 2022, https://www.gosrsi.com/the-benefits-of-automated-warehouse-robots

[8] Brown, Sara. "A New Study Measures the Actual Impact of Robots on Jobs. It's Significant."
MIT Sloan, 29 July 2020, https://mitsloan.mit.edu/ideas-made-to-matter/

2.3.2. File Links

[1] "Design assessment document," *Google Docs*. [Online]. Available: https://docs.google.com/document/d/1kcCi1JGLbXWbsUNgIUETYfZCfY4cVWMEoLcLaqXSTKA/edit?usp=sharing. [Accessed: 04-Nov-2022].

2.4. Revision Table

TABLE V
REVISE TABLE

Date	Author	Revise content
11/2/2022	Konstantin	Risk assessment table filled and annotated
11/4/2022	Xinrui	Update risk assessment and add impact file link
11/18/2022	Konstantin & Sam	Added risks 6 through 8

04/20/2023	All members	Revised the contents

3. Top-Level Architecture

3.1. Block Diagram

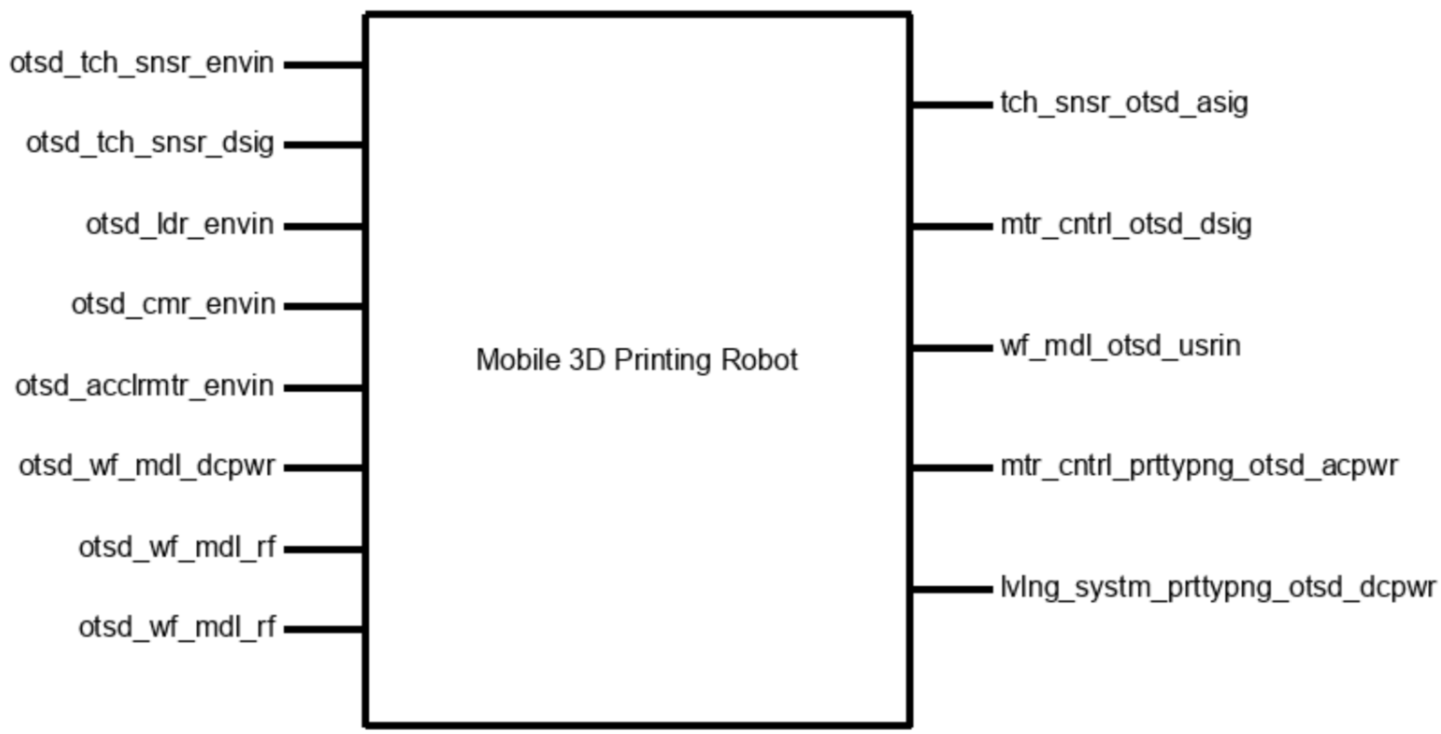


Fig. 3.1.1. Black Box Diagram

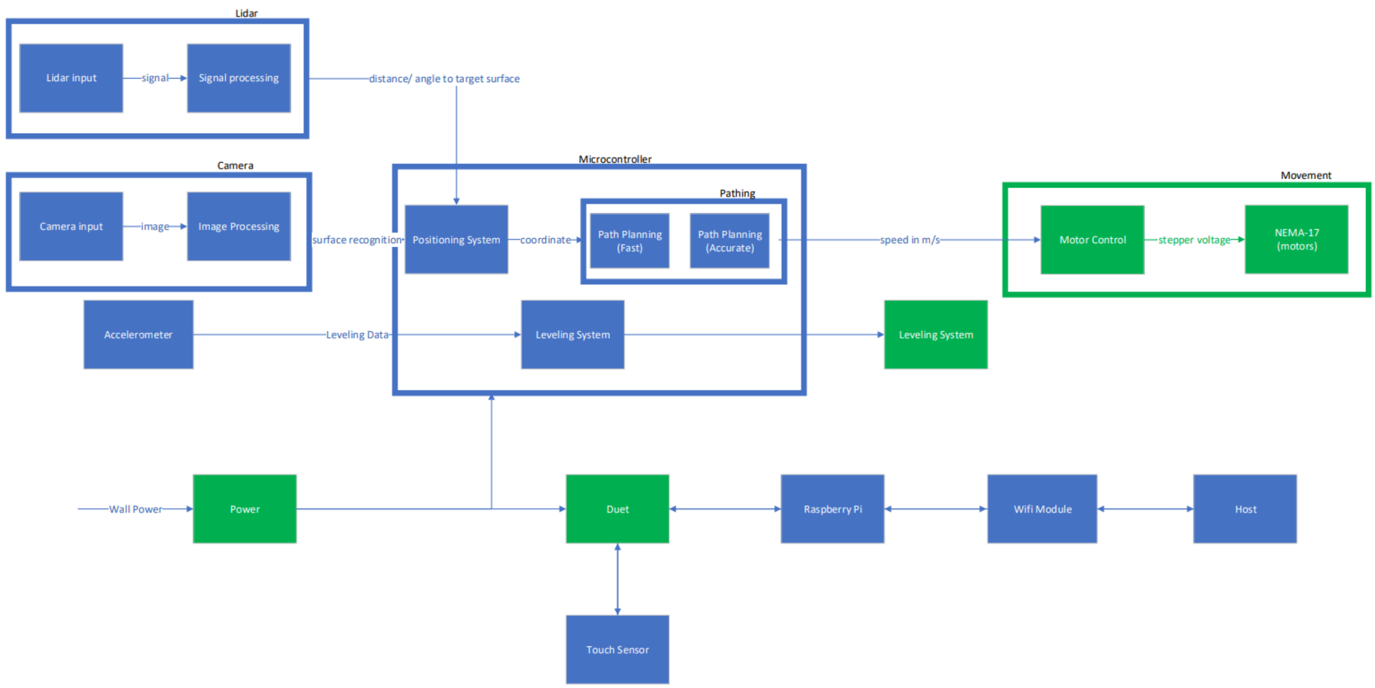


Fig. 3.1.2. Block Diagram

Konstantin: Lidar, Accelerometer

Ethan: Touch sensor, Wifi module

Sam: Leveling system prototyping, Motor control prototyping

Xinrui: Camera, Positioning system

3.2. Block Descriptions

3.1.1 Accelerometer

The accelerometer will send leveling data to the Arduino microcontroller for the purpose of providing proper leveling data. This helps the robot auto level when it comes up to the print surface.

3.1.2 LiDAR

The lidar provides object distance data to the microcontroller for the purpose of navigation and object identification. The lidar will update the microcontroller every second with the current coordinates of the robot relative to the target object. The lidar block will allow the robot to make precise movements in relation to the print surface.

3.1.3 Touch Sensor

The touch sensor will allow the 3D printer module to measure and record the precise coordinates of the print surface. The touch sensor has an extendable probe that will send an electrical signal when a small amount of pressure is applied to the probe. This will allow the Mobile 3D Printing Robot to automatically calibrate itself and send an accurate mapping of the print surface to the host running the slicing software by providing the printer with accurate environmental data.

3.1.4 WiFi Module

Allows the Duet to connect to a remote host using wifi. Allows the host to upload gcode files and directly controlling 3D printer's various stepper motors. The wifi module is enabled through a raspberry pi setup controlling the Duet 3 mainboard of the 3D printer.

3.1.5 Leveling System

This block is in charge of making sure that the robot is normal to the force of gravity. It makes sure that our robot is not meeting the print surface at an angle. Note: the destination of this block would be the level controlling motor control system.

3.1.6 LS Prototyping (COMBINE WITH LEVELING SYSTEM)

This block includes the work done to make sure the level of the axes of the print robot is normal to the axis of the print surface. It uses 3 stabilization motors, two located at the front and one on the back. It uses NEMA 17 motors. It is driven by the TMC2209 motor driver circuit.

3.1.7 Microcontroller

This block includes the code that is the brain of our system as well as the I/O connections that are made to connect the computer vision blocks to the motor control blocks that manage the movement of our system. It will have the image processing of the camera and Lidar inputs, the path planning and syntax to move from a point to normal to the surface from a top-down view and the equivalent instructions sent to the motor controller to drive to the required location.

3.1.8 Motor Control

This means that from power up, to ready to print should be within 5 minutes. How the motor control block comes into play is when our computer vision system has identified the destination surface of the robot. It converts this imaging data to data that the motor control block is designed for in code, then it transmits the right digital signals to the drive circuit to drive the motors in the right order and allow the robot to orient itself towards the surface. Afterwards it. It will let out feet(screwed in NEMA 17) motors that lift the robot off its wheel motors and stabilize for the print. During this stabilization, it will take in accelerometer values to actively update the feet motors to make sure we are normal to the surface we want to print on.

This block is a code block that consists of the code to ensure that allows the movement/mobility of the 3D printing robot. The controller we will be using for this project is Sipeed Maixduino which is built on the Arduino uno form factor. The block will interpret the output data from the microcontroller block and convert it into direct stepper instructions which will be transmitted outside of code to the stepper driver. The inputs to the block will be the direction to turn in, and the angle of attack for the turn(for now not sure what the angles will be). The angles will be from a choice given by the block that the microcontroller takes into account when giving the angle as an input. The output of the block is stepper motor controls. We will be using SN754410NE H-bridge to drive 7 NEMA 17 stepper motors. We will have for wheel motors and 3 feet motors.

3.1.9 Motor Control Prototyping

This block is used to represent the voltage and current values sent to the motors in our system and how they react and behave based on those values. There are seven motors in total in our system. Four of them are mounted to the four corners of the 3D printing robot and are used to drive the "Hulk" in the direction of the print surface. They are attached to "mecanum" wheels which allow rotation on the center point axis of the system. The other three are threaded through and are connected to the feet on both sides. With a triangle arrangement, two are on the back side and 1 at the front. They will serve as the feet and stability system of the Hulk for finer movements before printing. The name of the motor being used in our system is the 2-phase bipolar NEMA 17 stepper motor.

The type of stepper motor we use in our system has divided a rotation up to 1.8 degrees per “step”. This means that it takes 200 "steps" of the motor to complete a rotation. This motor was chosen because our system requires finer movements. The ability to make up to a millimeter of adjustments when factoring in the friction of the ground was required. When our vision system coordinates have been converted into drive signals, the motors will be driven with precision autonomously through the room. It satisfies the "distance x" and "distance y" requirements of the project by aiding that movement. It also satisfies the "time limitation" requirement because it can be driven with a variable speed. Since the last checkoff, the motors have been increased in size.

3.1.10 Pathing

This block includes the code that will take in the camera and Lidar data and produce motor directs/speeds to get us to our desired location. It will take the distance to the surface from the lidar, and the surface marker recognition data from the camera, and it will calculate an offset from the surface, accounting for the size of the robot by a radius. Then it will navigate to a rough estimate of a location using the path planning(fast) code. Once we are below a certain distance to the surface using proportional speed, we then transition to path planning(fine/accurate) code, where we make smaller and more useful movements. We will also sample data from the camera and lidar more often to get more accurate results.

3.1.11 Power Supply

This block (WHAT EXACTLY IS OUR PWR SUPPLY) (NOT OUR BLOCK)

3.1.12 Camera

The camera block is a critical component in the functioning of the Mobile 3D Printing Robot. Its main function is to assist in navigation and surface recognition by continuously capturing images and sending the processed data to the microcontroller. This allows the robot to have a real-time understanding of its surroundings, and make necessary adjustments to its movements to ensure precision and accuracy for subsequent printing. Currently, the camera will be a microcontroller-equipped camera with a 24P connection and will be placed on the front of the robot as its "eyes".

The camera block contains imaging sensors and processing units that work together to identify important features in the captured image. It also has the ability to recognize colored circles on the surface, which is critical for adjusting the robot's position. A green circle is placed in the center of a given surface and a red circle is placed equally spaced on each side of the green circle. By determining whether the green circle is in the center of the image, the system will get the approximate orientation of the robot relative to the specified surface and thus guide the robot's motion. If the three circles are in a straight line and the green circle is centered in the image, the robot is considered to be in the center of the surface.

3.1.13 Positioning System

The location system will collect data from the lidar and camera blocks, and send the coordinates and angle of the surface relative to the robot to the path planning block. This information will enable the path planning block to guide the robot's movement.

3.3. Interface Definitions

Name	**Properties**
otsd_tch_snsr_envin	• **Electromagnetic:** The sensor will function in an environment up to 90 degrees Fahrenheit • **Humidity:** The sensor shall function in an environment with a humidity between 0% and 60% • **Other:** The sensor shall function while facing up to 90 degrees from straight down
otsd_tch_snsr_dsig	• **Other:** Probe will retract when servo is set to position 80 • **Other:** Probe will retract and extend with a delay of at most 1 second • **Other:** Probe will extend when servo is set to position 10

otsd_ldr_envin	• **Light:** Reflected light from the surrounding environment which was initially sent from the LiDAR sensor. It will have a pulse frequency of 20kHz. • **Light:** Reading the distance that the light traveled from and back to the LiDAR at up to three different rates. • **Light:** Reading the values off of any surface in the surrounding environment within a range of 10 meters from the target object.
otsd_cmr_envin	• **Other:** Object to detect include colors: red and green • **Other:** Objects between 0.5ft and 2 ft • **Other:** Must have at least a 90 degree field of view
otsd_acclrmtr_envin	• **Other:** It can read values tilting right. • **Other:** It can read values tilting forward. • **Other:** It can read values tilting left. • **Other:** It can read values tilting back.
otsd_wf_mdl_dcpwr	• **Inominal:** 500mA • **Vmax:** 5.5V • **Vmin:** 4.5V
otsd_wf_mdl_rf	• **Messages:** Recieves temperature data

	• **Messages: Recieves** information about stepper motor position • **Other:** Gives remote access to duet3.local when on same wifi
otsd_wf_mdl_rf	• **Messages:** Recieves temperature data • **Messages:** Recieves information about stepper motor position • **Other:** Gives remote access to duet3.local when on same wifi
tch_snsr_otsd_asig	• **Max Frequency:** Greater than 0.5 Hz • **Other:** Goes into state of error when signal lasts longer than 1 second • **Vrange:** 3-6 V
ldr_mcrcntrllr_data	• **Datarate:** X, y and theta coordinates should be provided within the 10 meter radius of the target object. Ex: (5.16, 7.25 and 45 degrees) • **Messages:** Every three seconds, second, or half second the data is updated (x, y, and theta coordinates) providing the microcontroller with the robot's current location. • **Protocol:** The LiDAR sensor will be able to read values of the surface as the robot is up against the target object. Approximately up to one foot away from the surface.

cmr_mcrcntrllr_data	• **Other:** Data update at least once 5 seconds. • **Other:** Pixel coordinates of up to three circles' centers • **Other:** Movement directions. Eg. Left, Right, etc...
mcrcntrllr_pstnng_systm_data	• **Other:** Movement directions. Eg. Left, Right, etc... • **Other:** The distance from Lidar measured. • **Other:** The pixel coordinates of the circles' center.
pstnng_systm_pthng_data	• **Other:** Movement directions. Eg. Left, Right, etc... • **Other:** The angle of the surface based on the robot. • **Other:** The coordinate of the surafce based on the robot (x, y).
pthng_mtr_cntrl_data	• **Datarate:** 115200 bps • **Messages:** calls and passes parameters to movement functions to be executed by the motor controller
lvlng_systm_mtr_cntrl_dsig	• **Logic-Level:** 3.5 V min for high input voltage • **Logic-Level:** 1.5 V max for input low voltage • **Other:** gives values for stability (stepper) motors that keeps the robot parallel to the ground when printing

acclrmtr_lvlng_systm_dsig	• **Other:** Code values show y-axis rotation between -1000 and 1000. Positive for upwards rotation, negative for downward rotation. • **Other:** Code values show x-axis rotation between -1000 and 1000. Positive for upwards rotation, negative for downward rotation. • **Other:** Runs at 115200 baud rate
wf_mdl_otsd_usrin	• **Other:** Allows user to modify extrusion • **Other:** Allows user to change stepper motor position • **Usability:** GUI needs to be understood by at least 4 out of 5 users
mtr_cntrl_prttypng_otsd_acpwr	• **Inominal:** 0.6A • **Other:** Vlogic: 5V • **Other:** Per Motor Drive: Imin: 0.9A, Imax: 2.4A • **Vmax:** 24V • **Vnominal:** 24 V
pwr_spply_tch_snsr_dcpwr	• **Inominal:** 12 mA when extended • **Ipeak:** 250mA • **Vmax:** 6V • **Vmin:** 3.3V

pwr_spply_acclrmtr_dcpwr	• **Inominal:** 3.8mA • **Vmax:** 3.46V • **Vmin:** 2.375V
pwr_spply_mtr_cntrl_prttypng_dcpwr	• **Inominal:** Drive: 1.5A per motor • **Vmax:** 24 V • **Vmin:** 24V
pwr_spply_lvlng_systm_prttypng_dcpwr	• **Inominal:** 0.6 A • **Ipeak:** I ascending: 2.1A • **Other:** I descending: 1.9A • **Vmax:** 24 V • **Vmin:** 24V • **Vnominal:** 24V
lvlng_systm_prttypng_otsd_dcpwr	• **Inominal:** Holding per motor phase: 0.63A • **Ipeak:** I ascending: 1.5A • **Other:** I descending: 1.5A per motor phase • **Vmax:** Per Motor Phase: 1.8V • **Vmin:** Per Motor Phase: 1.8V • **Vnominal:** Per Motor Phase: 1.8V

3.4. References and File Links

3.4.1. References (IEEE)

[1] “IEEE referencing style: Images, figures and tables,” *Guides*. [Online]. Available: https://guides.library.uwa.edu.au/IEEE/images_tables_figures#:~:text=Place%20the%20Figure%20title%2Fcaption,captions%20should%20be%20centre%2Djustified. [Accessed: 13-Oct-2022].

3.4.2. File Links

3.5. Revision Table

Date	Author	Revise Content
3/10/2023	Xinrui	Initial document created
4/28/2023	Konstantin	Filled in Section 3.2
4/28/2023	Xinrui	Added Section 3.3
5/14/2023	Xinrui	Added the block owners

4. Block Validations (one subsection per block)

4.1. Accelerometer

4.1.1. Description

This block is used in the stabilization portion of the Mobile 3D Printing Robot. The robot will be able to start anywhere from up to 10 meters away from a target object and autonomously find its way to the print surface and begin a highly accurate printing job. In order to make sure the printing is as accurate as possible the robot needs to be level or perpendicular relative to the print surfaces normal. To make sure we can achieve this we are using an accelerometer as it is a highly used device for leveling systems.

The purpose of the [1] accelerometer block is to provide accurate data about the current level to the robot via the microcontroller. These values are measured when the robot reaches its final destination before the print calibration begins. The live data is given to the stabilizing stilts that are on the outside of the frame. This way the robot levels using the stilts to lower into the ground until the accelerometers data shows that the robot has reached a level state. This ensures that regardless of how the flooring may be uneven, the robot will be able to print as if it is on a flat floor.

4.1.2. Design

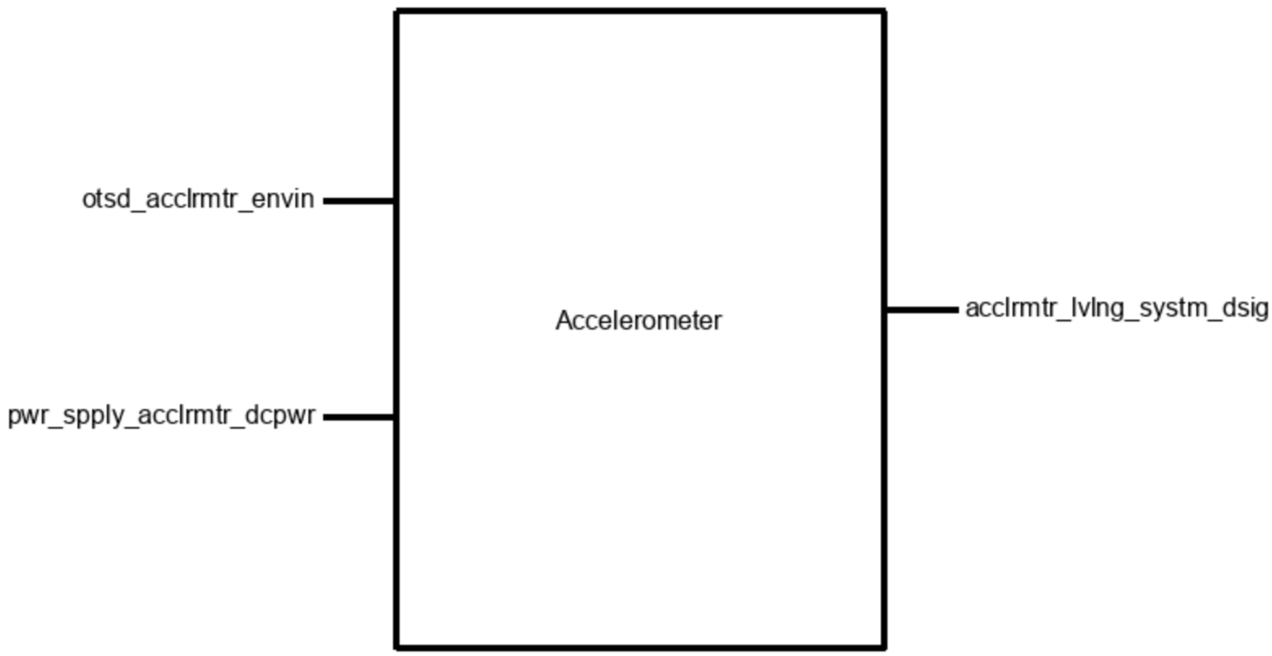


Fig. 4.1.2.1 Block of Accelerometer

As shown in the figure above, the accelerometer takes in environmental input and input from a power supply. The accelerometer is able to read values when the environment forces the robot to be tilted in any direction and recognize that the robots leveling should be adjusted. Using that information the accelerometer is able to tell the leveling system how to level the robot if necessary. The accelerometer also takes in power to work.

4.1.3. General Validation

The simplicity of the diagram above makes it much easier to communicate with the rest of the robot. Accelerometers are very simple and straight forward devices that are commonly used. They have a few base functions, and we are using the motion sensor function. As seen above the accelerometer simply takes environmental input and power input. The environmental input is really how much the robot is tilted along the x and y axis due to uneven flooring. The power input is just the power connection to the microcontroller. The microcontroller provides all of the power necessary for the accelerometer to function. The only output is the code values of the rotation about the x and y axis. These code values are sent to the microcontroller to communicate to the motors that control the stability stilts on the outside of the robot.

4.1.4. Interface Validation

Interface Property	Why is this interface this value?	Why do you know that your design details <u>for this block</u> above meet or exceed each property?

otsd_acclrmtr_envin : Input

Other: It can read values tilting left.	This is one of the premade functions that the accelerometer can do. If the flooring is uneven in a way so that our robot tilts more to the left the accelerometer as able to tell.	The accelerometer correctly outputs the x and y values caused by this tilt.
Other: It can read values tilting back.	This is one of the premade functions that the accelerometer can do. If the flooring is uneven in a way so that our robot tilts backwards the accelerometer as able to tell.	The accelerometer correctly outputs the x and y values caused by this tilt.
Other: It can read values tilting right.	This is one of the premade functions that the accelerometer can do. If the flooring is uneven in a way so that our robot tilts more to the right the accelerometer as able to tell.	The accelerometer correctly outputs the x and y values caused by this tilt.
Other: It can read values tilting forward.	This is one of the premade functions that the accelerometer can do. If the flooring is uneven in a way so that our robot tilts forward the accelerometer as able to tell.	The accelerometer correctly outputs the x and y values caused by this tilt.

acclrmtr_lvlng_systm_dsig : Output

Other: Code values show x-axis rotation between -1000 and 1000. Positive for upwards rotation, negative for downward rotation.	The robot needs to be able to read the accelerometers data. So, the more negative the x value the more the robot is tilted down on the x-axis. The more the robot is tiled upward the more positive the value will become. The values make it easy for the microcontroller to communicate to the motors.	As the robot is manually being tilted you can see the correct values appear from the microcontroller's perspective.
Other: Runs at 115200 baud rate	This is the baud rate that the rest of the leveling system runs at by default. Therefore, in order for the	The microcontroller is able to communicate with the accelerometer without issue. Meaning the baud rate is correct and works.

	accelerometer to communicate properly it is at the same baud rate.	
Other: Code values show y-axis rotation between -1000 and 1000. Positive for upwards rotation, negative for downward rotation.	The robot needs to be able to read the accelerometers data. So, the more negative the y value the more the robot is tilted down on the y-axis. The more the robot is tiled upward the more positive the value will become. The values make it easy for the microcontroller to communicate to the motors.	As the robot is manually being tilted you can see the correct values appear from the microcontroller's perspective.

pwr_spply_acclrmtr_dcpwr : Input

Inominal: 3.8mA	This is the value from the accelerometers manufacture.	When connected to the microcontroller the accelerometer functions properly. The values read when testing matching the needed values for the accelerometer.
Vmax: 3.46V	This is the value from the accelerometers manufacture.	When connected to the microcontroller the accelerometer functions properly. The values read when testing matching the needed values for the accelerometer.
Vmin: 2.375V	This is the value from the accelerometers manufacture.	When connected to the microcontroller the accelerometer functions properly. The values read when testing matching the needed values for the accelerometer.

4.1.5. Verification Process

1. First test if the accelerometer works using the power supply which is the microcontroller.
2. Manually tilt the robot forward and see if the accelerometers values show the correct values indicating that the robot is tilting forward.
3. Manually tilt the robot backward and see if the accelerometers values show the correct values indicating that the robot is tilting backward.
4. Manually tilt the robot to the left and see if the accelerometers values show the correct values indicating that the robot is tilting left.
5. Manually tilt the robot to the right and see if the accelerometers values show the correct values indicating that the robot is tilting to the right.
6. Check to see if the microcontroller is showing the same values the accelerometer is reading as each manual tilt check is being made.
7. After manually tilting the robot in whatever direction, start the auto leveling system.

4.1.6. References and File Links

[1] *Hiletgo GY-521 MPU-6050 MPU6050 3 Axis Accelerometer Gyroscope Module 6* www.amazon.com/HiLetgo-MPU-6050-Accelerometer-Gyroscope-Converter/dp/B078SS8NQV Accessed 15 May 2023.

4.1.7. Revision Table

Revisor	Date	Revision
Konstantin Markov	3/15/2023	Add sections 4.1.1., 4.1.2., and 4.1.5.
Konstantin Markov	5/12/2023	Updated all sections of 4.1

4.2. Lidar

4.2.1. Description

The project this block is for is the Mobile 3D Printing Robot. The robot will be able to start anywhere from up to 10 meters away from a target object and autonomously find its way to the print surface and assist the calibration of the 3D printing nozzle. In order for the robot to make its way to the target object it needs to know its location throughout its movement. For this to happen we are using a LiDAR sensor because the accuracy obtained from using laser detection is higher than most other methods of calculating distance.

The purpose of this block is to get distance values to help provide location awareness for the robot and surface identification when the robot is used in nonideal circumstances such as anything besides a flat wall. This block will also serve as an update from the current calibration method which is manual calibration with a touch sensor to part of the new automated calibration system. This will improve the time it takes to calibrate from about fifteen minutes to about three minutes max, ideally. After searching for a LiDAR sensor that would meet the engineering and customer requirements I purchased the LW20/C LiDAR sensor from [1]

The LiDAR sensor provides object distance data to the microcontroller for the purpose of navigation and helps with object identification. It does this by transmitting light as a laser and once the laser is reflected from a surface the sensor can tell how far the light traveled which in turn shows how far the sensor is from the surface. This function allows the LiDAR sensor to be able to map out surfaces which will help the robot know the type of surface it is going up to and how to use the touch sensor to calibrate the 3D printer. The LiDAR sensor will update the microcontroller every either three, one, or half a second(s) with the current coordinates of the robot relative to the target object. The LiDAR block will allow the robot to make precise movements in relation to the print surface in order to place the robot in the best position for print surface calibration to be done easily.

4.2.2. Design

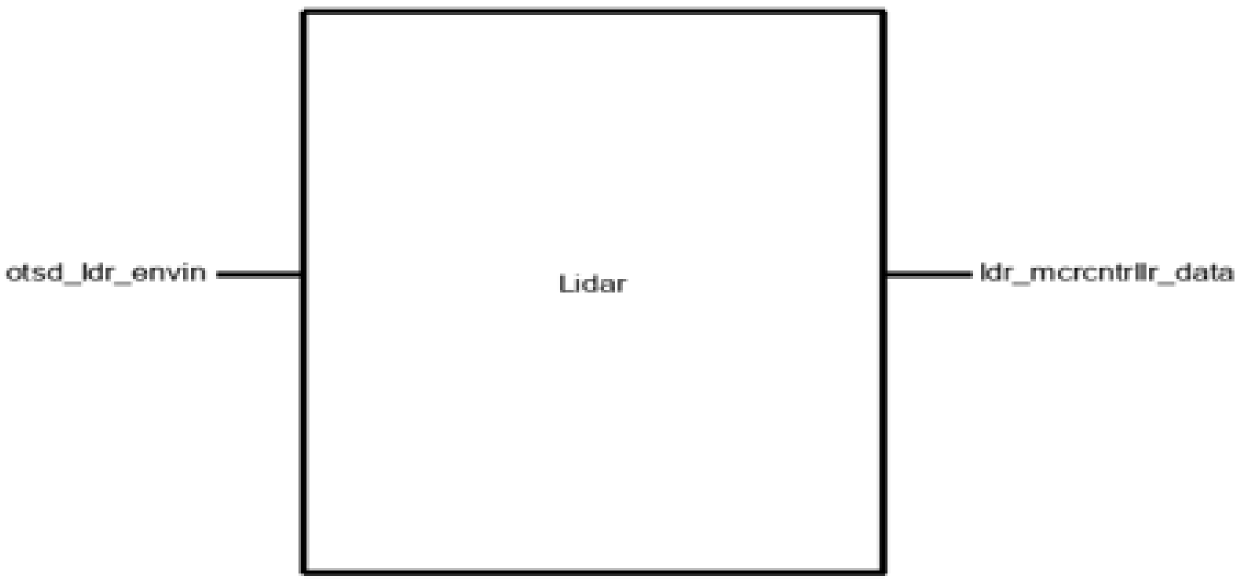


Fig. 4.2.2.1. Block of Lidar

As shown in *Figure 1* this block, LiDAR sensor, takes in values from the environment and outputs the organized data to the microcontroller. The environmental data is read by light via a laser sent from the LiDAR and reflected back to the sensor which measures the distance that light traveled. There is no specific type of environment required for the LiDAR to work as long as it is within the range for the sensor to read. Once all of the necessary values have been captured the coding algorithm will do the calculations of the global coordinates of the robot that the microcontroller will use for the pathing system.

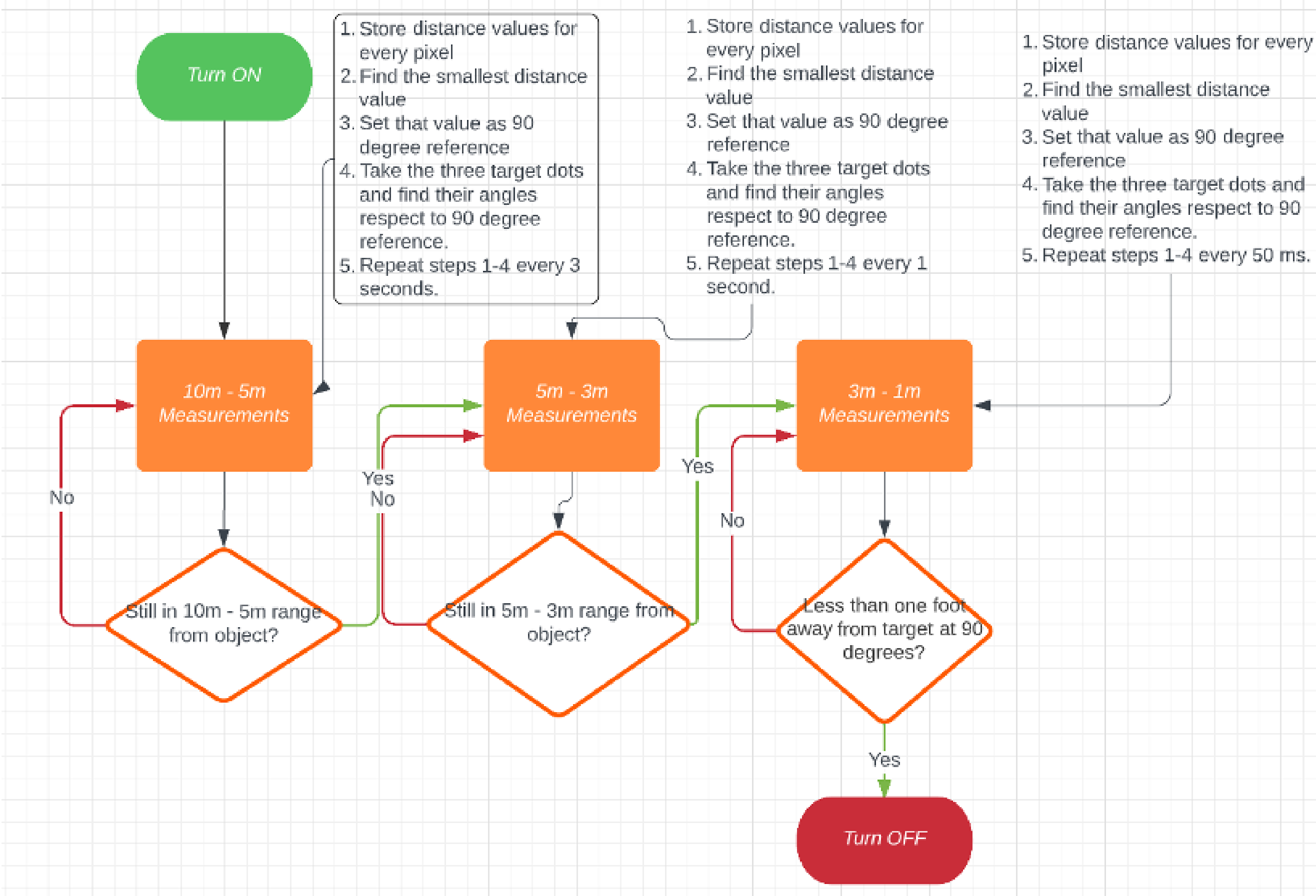


Fig. 4.2.2.2. Flow chart of Lidar

In *Figure 2* the process that this block goes through is shown. Once the entire system, including the LiDAR, the sensor will start measuring. The robot will be placed at most 10 meters away from the target object. From further distance the robot doesn't need updated information as frequent because there is more room for error. But as the robot closes in on the target object the data, from the LiDAR, update rate needs to be faster to get more accurate locational awareness for the robot to make calibration easier. This process is very repetitive with small changes as the robot moves and at the end the LiDAR will be turned off.

4.2.3. General Validation

The black box diagram shows a general simplicity of having one input and one output for the LiDAR sensor block. As described above, the one function this LiDAR sensor has is measuring distances by transmitting and receiving light in laser form. Using a laser device is much more accurate and efficient than any other method. After the LiDAR collects the distance values of the surface around the target object, using an algorithm, select values will be used to tell the robot where it is relative to the target object. The output of the LiDAR sensor goes to the microcontroller that is being used in the overall system. The output will be provided in coordinate value form, meaning it will provide the required x and y coordinates as well as the angle theta of the robot relative to the target object. The flowchart design above is simple as the main idea of the LiDAR camera is simple.

Even though there could be just three blocks (ON, find distance, and OFF) that encompasses all of the measurements and changes that control the upload rate, depending on its distance, I split it into three. This design choice was a way to make it easier to debug any problems that may occur. Additionally, having the option to change the information update rate increases efficiency and location accuracy when and/or if needed. Simplicity for the function helps when working with an advance technology/something I have never worked with before. The idea is that it loops through a measurement cycle while actively updating the microcontroller to help the robot reach the object at best or near best case scenario. The cycle entails decision points that to help control the information update rate by checking what the current distance of the robot is from the target object. The best-case scenario being, less than a foot away so the LiDAR sensor can still function and perpendicular to the object.

4.2.4. Interface Validation

otsd_ldr_envin : Input

Interface Property	**Why is this interface this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Light: Reading the values off of any surface in the surrounding environment within a range of 10 meters from the target object.	This is a customer requirement that must be followed which accounts for multiple potential situations the 3D printing robot may be used in.	The LiDAR sensor being used for this block has a range of 100 meters and does not require specific surfaces in order to work properly.
Light: Reflected light from the surrounding environment which was initially sent from the LiDAR sensor. It will have a pulse frequency of 20kHz.	This is the only function of the LiDAR sensor: sending light as a laser then reading the reflected lights value to find the distance the light traveled.	Again, this is the only function of the sensor, so this property comes with the purchase of the part. As long as the sensor turns on, which according to its specifications and it is matching power specifications of the robot, it will.
Light: Reading the distance that the light traveled from and back to the LiDAR at up to three different rates.	This accounts for keeping the accuracy of the robot's location as it gets closer to the target object. Larger movements require less precision.	As seen in the flowchart above, it includes decision points to know when to switch the update rate if needed.

Table 1: Interface one properties that account for engineering and customer requirements

ldr_mcrcntrllr_data : Output

Datarate: X, y and theta coordinates should be provided within the 10 meter radius of the target object. Ex: (5.16, 7.25 and 45 degrees)	The customer requires the robot to be able to start within a 10 meter radius from the target object. Having a global coordinate system makes telling the robot its location much simpler. The angle, theta, helps provide locational awareness for the robot by letting it know if and by how much it needs to turn.	Depending on the current distance the LiDAR camera will actively provide coordinates at a set rate starting from at most 10 meters away. This portion is controlled through the code for the LiDAR.
Messages: Every three seconds, second, or half second the data is updated (x, y, and theta coordinates) providing the microcontroller with the robot's current location.	The customer requires that the current location is updated at least every three seconds. Increasing update rate as the robot gets closer increases efficiency and accuracy to help the pathing system block.	The block diagram includes decision points to know when to switch the update rate if needed. Depending on the current distance the LiDAR camera will actively provide coordinates at a set rate. This is controlled by the coding that will be used for the LiDAR and pathing system.
Protocol: The LiDAR sensor will be able to read values of the surface as the robot is up against the target object. Approximately up to one foot away from the surface.	This choice makes sure that the LiDAR will be place on the robot in a way for it to be able to read values up close to the surface. Otherwise, most LiDAR sensors stop working at about eight inches from a surface.	The placement of the LiDAR sensor on the finalized robot will be further back to avoid glitches in the sensor.

Table 2: Interface two properties that account for engineering and customer requirements

4.2.5. Verification Process

1. The first step is to test that the LiDAR sensor works and works long enough for how long it will be used for navigation. To do this connect to the power source that is being used for the robot. Does the LiDAR sensor turn on and stay on for at least 15 minutes?
2. This step is used to check the range of the sensor. The LiDAR sensor needs to be able to read values up close and far away. Does the LiDAR sensor read values between 1 foot and 10 meters?
3. In order to get multiple points along the surface to calculate distances and angles the LiDAR sensor should be able to see a wide view. This can come from either the sensor itself or using a motor to rotate the sensor. Does the LiDAR sensor see objects within a 60 degree field of view?
4. The sensor should be able to map out different types of surfaces to help the touch sensor calibrate for the printing process. Connect to computer (for testing otherwise

microcontroller) and test sensor against different types of surfaces. Is the sensor able to provide enough data points so that the robot will know the type of surface it is about to print on accurately?

5. Select two or three target points (around the target object) on the wall with the sensor, Using the distances of those points find the angle the robot is looking at the target object.
6. Is the information able to be updated every three seconds? One second? Half a second? [This is more of the coding aspect]

These steps were chosen in order to verify that the LiDAR block meet the engineering requirements. The steps are also set so the block can be verified that it meets the customer requirements. To find the specifications of the LiDAR sensor I used the datasheet for it [2].

4.2.6. References and File Links

[1] "LW20/C (100 m)." LightWare Lidar LLC, https://lightwarelidar.com/products/lw20-c-100-m?variant=8806944145507¤cy=USD&utm_medium=product_sync&utm_source=google&utm_content=sag_organic&utm_campaign=sag_organic&googlead=yes&gclid=CjwKCAiAoL6eBhA3EiwAXDom5pozrUbO05c6CZvLQ86VxEcFcYJNg_NlqnkMY97ZtkuUOvmqtYP5rxoC_-cQAvD_BwE.

[2] LW20 / SF20 Lidar Sensor Product Manual - LightWare Lidar LLC. https://www.documents.lightware.co.za/LW20%20SF20%20-%20LiDAR%20Manual%20-%20Rev%201.pdf.

4.2.7. Revision Table

1/17/2023	I completed the description section.
1/20/2023	I completed the remaining sections of the Block Validation Draft. Still need to find some exact values for the LiDAR camera.
2/7/2023	I worked on fixing the description and general validation sections of the document.
2/11/2023	I completed the updated version of the validation paper. Specified below in section eight.

4.2.8. Revision Statement

During the revision process I made many changes that were recommended from the peer reviews and general changes I have made to the block itself. Overall, I added more depth to each section to make it more clear for the reader what the block is. In the description section the addition was the connection between this block and the overall project. Mentioning the purpose of the block and the customer/engineering requirements. In the design section I added descriptions of what each figure represents. As well as the choices that made up the design of the block diagram and flowchart. The general validation section was just specified even more by talking more in depth about the design of the block.

As for larger changes, I replaced one of the properties in the output interface. Initially one of the properties was about the FOV of the LiDAR sensor but we decided that the sensor can also just be rotated by a motor to disregard any FOV issues. I removed on of the steps in my verification plan but talked more in depth for the remaining steps to make it more clear for how someone would go about verifying the block.

4.3. Touch Sensor

4.3.1. Description

The touch sensor will allow the 3D printer module to measure and record the precise coordinates of the print surface. The touch sensor has an extendable probe that will send an electrical signal when a small amount of pressure is applied to the probe. This will allow the Mobile 3D Printing Robot to automatically calibrate itself and send an accurate mapping of the print surface to the host running the slicing software by providing the printer with accurate environmental data.

4.3.2. Design

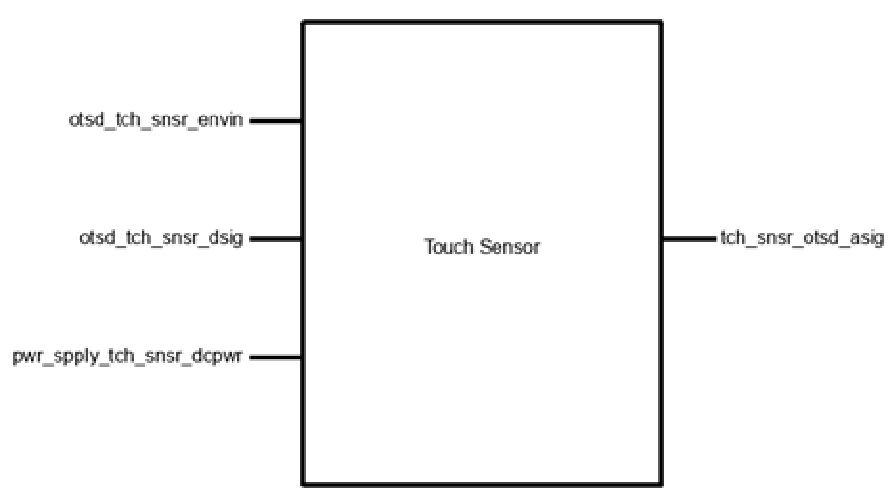


Fig. 4.3.2.1. Block of Touch sensor

The design of the touch sensor is simple, being a commercial auto bed leveling sensor being adapted to the purposes of this project. Due to the printer mainboard being identical to those used in traditional 3D printers made for flat surfaces, with the only difference being the instruction set uploaded to the board, means that the touch sensor has documentation as well as a terminal that accepts the sensor input gives the team an easy time adapting this block into the project.

The interfaces for this block are extremely simple. Being a simple touch sensor, the device contains an extendable probe that can retract when the sensor is not in use. When the probe is extended the sensor sends a digital signal to the main board when pressure is applied, forcing the probe to retract, and creating an electrical short. This will send a voltage of 3.3 volts back to the printer board and in tandem with the software identifying the position of the printer head will allow the printer hot end to calibrate itself before printing on the surface. The expected environmental data will be a curved print surface likely made from PLA plastic or tinfoil. The 5v_power input will be provided through the printer main board and has been verified by various companies that produce and sell 3D printers.

4.3.3. General Validation

This block was designed to be both cheap and replaceable, being mounted near a 3D printer hot end that reaches approximately 215 degrees Fahrenheit when printing PLA, it was a design decision to have a hot end that could be damaged without too much cost being incurred to the project. The touch sensor will allow the printer to determine the distance between the hotend and the print surface. As the project calls for the autonomous printing on a curved object, manual levelling is not possible. The touch sensor will allow the printer to determine the distance between the object and the printer hotend and adjust the gcode which controls the printer during the print. As the 3D printer mainboard we are using does not have any error detection functionality, it is essential that we are able to get an accurate calibration of the printer before the printing process starts.

The 3D Touch brand sensor was chosen due to its cheapness and reliability. Another option that may be purchased before the project is complete is the BL touch auto bed leveling kit, as both are functionally identical, except the BL touch is manufactured with higher quality materials.

Using a touch sensor is much more accurate than any calibrations made using computer vision and measurements can be made extremely quickly. The touch sensor is being adapted to work with our printer that can print on curved surfaces rather than being implemented on a traditional flat print surface. Due to this, changes in the algorithm used to calculate and generate gcode will be necessary. The block will affect both the graduate student and mechanical engineering teams and close collaboration will be required for both.

4.3.4. Interface Validation

Interface	Properties
otsd_tch_snsr_envin	· Humidity: The sensor shall function in an environment with a humidity between 0% and 60% · Other: The sensor shall function while facing up to 90 degrees from straight down
tch_snsr_otsd_asig	· Max Frequency: Greater than 0.5 Hz · Other: Goes into state of error when signal lasts longer than 1 second · Vrange: 3-6 V
pwr_spply_tch_snsr_dcpwr	· Inominal: 12 mA when extended · Ipeak: 250mA · Vmax: 6V · Vmin: 3.3V
otsd_tch_snsr_dsig	· Other: Probe will extend when servo is set to position 10 · Other: Probe will retract when servo is set to position 80

4.3.5. Verification Process

The testing for this block is mainly to provide a verification for the functionality of the touch sensor, as well as some tests that prove the block can communicate with the printer main board. The tests will have to be somewhat rudimentary, as the printer mainboard cannot easily be transported, and functionality will be simulated with a microcontroller.

A. Power Supply Test

This test will ensure that the touch sensor is functioning under the correct power range.

1. Connect the touch sensor power and grounds to a controllable voltage source.
2. Set the voltage to 6 volts.
3. Verify that the probe functions using a microcontroller to send signals that simulate printer functionality.
4. Verify that the sensor sends signals receivable by the microcontroller.
5. Verify that the touch sensor does not send a signal when the touch sensor has no pressure applied to it.
6. Repeat steps iii-v with the voltage set to 3.3v

B. Analog Signal Test

1. Connect the touch sensor output pin to a microcontroller with serial plotter capability or an oscilloscope and input correct voltage sources.
2. Verify that the probe triggers properly and a signal is sent.
3. Verify that the probe can be triggered at least 10 times in 20 seconds.

C. Environment Test

1. Verify that the touch sensor functions in environments with a range of 0% to 60% humidity.
2. Verify that the sensor functions when tilted at most 90 degrees from straight down.
3. Input Signal Test
4. Connect the touch sensor to a microcontroller and set the servo to 10.
5. Verify that the probe extends.
6. Test current and verify that it has a I-nominal of ~12mA.
7. Set the servo to 80.
8. Test current and verify that it never exceeds 250mA.
9. Verify that the probe retracts.

If the block passes all tests, then the block is ready to be a part of the finished system.

4.3.6. References and File Links

https://www.amazon.com/Printer-Leveling-Sensor-Touch-Heatbed/dp/B09M9V8Y4Y/ref=asc_df_B09M9V8Y4Y/?tag=hyprod-20&linkCode=df0&hvadid=563646990927&hvpos=&hvnetw=g&hvrand=13948976404109837661&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9032981&hvtargid=pla-1834500606472&th=1

4.3.7. Revision Table

Revisor	Date	Revision
Ethan Masiel	1/20/2023	Document Created
Ethan Masiel	2/12/2023	Interface changes, Validation plan updated
Ethan Masiel	3/12/2023	Updated for Project Document

4.4. Wifi Module

4.4.1. Description

Allows the Duet to connect to a remote host using wifi. Allows the host to upload gcode files and directly control the 3D printer's various stepper motors. The wifi module is enabled through a raspberry pi setup controlling the Duet 3 mainboard of the 3D printer.

4.4.2. Design

The Duet 3 mainboard of the 3D printer has been modified with a Raspberry Pi setup that enables a wifi module for remote connectivity. This functionality allows the user to upload gcode files and control the 3D printer's various stepper motors directly from a remote host. It also allows the user to monitor the printer’s temperature, current instructions, change configuration files, and adds access to various utilities that would otherwise be unavailable.

The modification was achieved through the use of a modified OS image that includes the necessary drivers and software for the raspberry pi to work with the Duet 3 mainboard. The modified OS image includes a custom-built driver that interfaces with the wifi module and communicates with the Duet 3 mainboard. The driver translates commands from the remote host and sends them to the mainboard, allowing for user control over the 3D printer's stepper motors. The modified OS image includes a user interface that simplifies the process of uploading gcode files and controlling the 3D printer. The user interface is intuitive and user-friendly, allowing users of all levels of technical expertise to use the wifi module with ease. The Raspberry Pi allows a host with a connection to the same wifi network to access the user interface remotely, enabling remote access to the interface.

4.4.3. General Validation

The Wifi Module needs to provide consistent and reliable access to the Duet interface

4.4.4. Interface Validation

4.4.5. Verification Process

4.4.6. References and File Links

4.4.7. Revision Table

4.5. Leveling System Prototyping

4.5.1. Description

This block includes the work done to make sure the level of the axes of the print robot is normal to the axis of the print surface. It uses 3 stabilization motors, two located at the front and one on the back. It uses NEMA 17 motors. It is driven by the TMC2209 motor driver circuit. The other three are threaded through and are connected to the feet on both sides. With a triangle arrangement, two are on the back side and 1 at the front. They will serve as the feet and stability system of the Hulk.

The motors used in our system is the 2-phase bipolar NEMA 17 stepper motor. The type of stepper motor we use in our system has divided a rotation up to 1.8 degrees per “step”. This means that it takes 200 "steps" of the motor to complete a rotation. This motor was chosen because our system requires finer movements. The ability to make up to a millimeter of adjustments when factoring in the friction of the ground was required. When our vision system coordinates have been converted into drive signals, the motors will be driven with precision autonomously through the room. It satisfies the "distance x" and "distance y" requirements of the project by aiding that movement. It also satisfies the "time limitation" requirement because it can be driven with a variable speed.

4.5.2. Design

Below is the design of the leveling control system. There are three motors in general. All of them are used for the stability of the robot. They have a free spinning screw going through them that allows the lateral mounting of flat feet-like materials on both sides. The name of the motor used in our design is the version of the NEMA 17 that is rated for 24V and 1.7A per phase. All the motors in our system are two-phase devices. This means each motor takes about 3.4 A per motor. But because of the nature of the requirements for the design, all seven motors in the entire project are never in use at the same time. The four drive motors are used to get to the print surface and the other three motors are used to stabilize the system. So our theoretical max current at any given time would be 3.4 X 4 = 13.6 A. All three motors are controlled by our MAIXduino which will not be covered in this block, but they are driven via the TMC2209 driver. The TMC2209 allows the use of variable speed and current in our motors. It also controls the PWM sequence that will be required to run these AC motors from dc power. This can be done both through hardware with the potentiometer on the board and software with the libraries that accompany the part. **Figure** 4.5.2.1. has a 3D image of the TMC2209 part. **Figure** 4.5.2.2. is a black-box diagram of the leveling system prototyping block.

Because our motor needs to be run with 24V which is much higher than the max voltage from the Arduino which is 5V, we run our motors directly from the power supply. **Figure x** has the wiring diagram to the protoboard currently being used for testing. Once testing is concluded a PCB will be built to replace the protoboard.

testing. Once testing is concluded a PCB will be built to replace the protoboard.

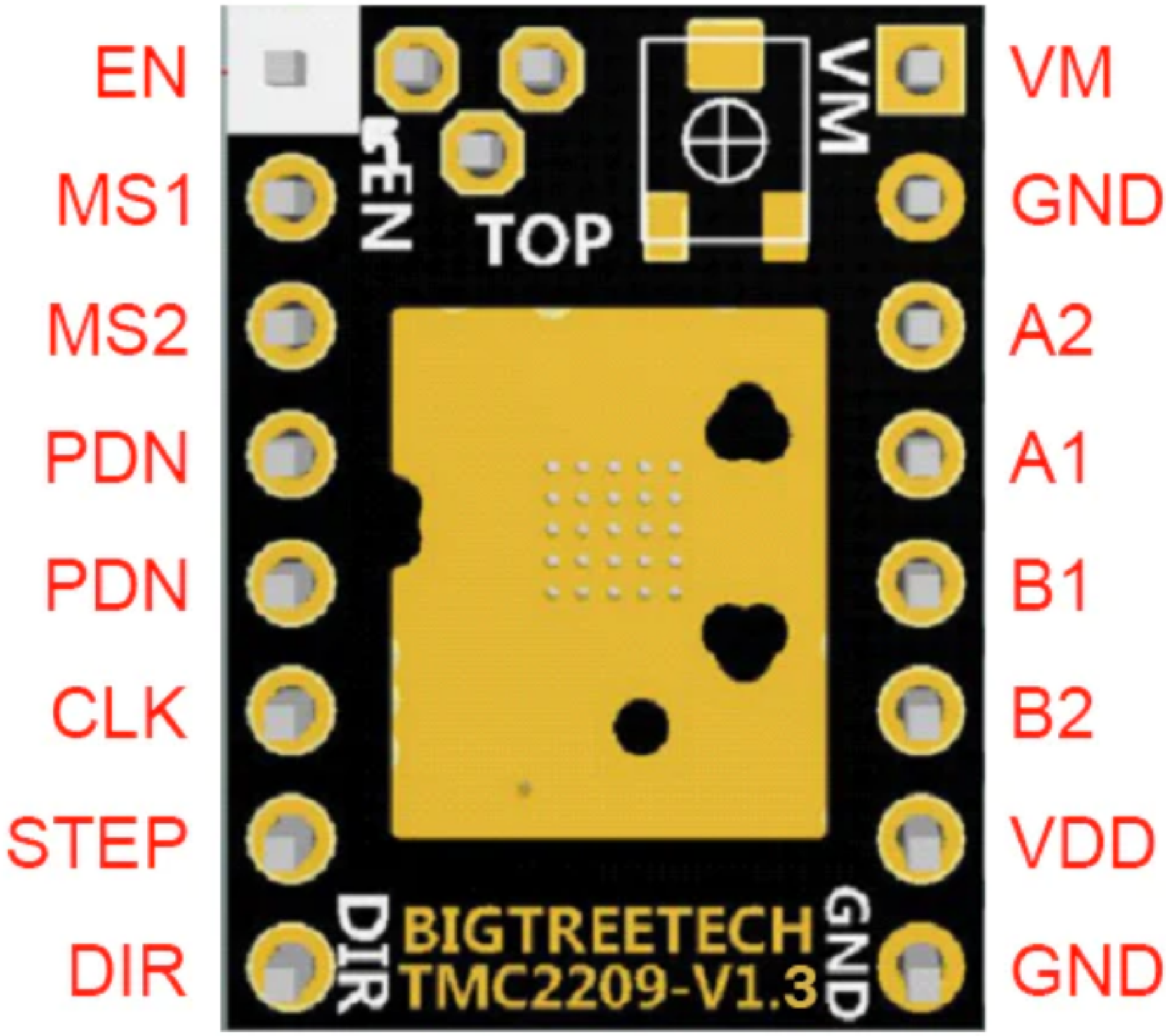


Fig. 4.5.2.1. TMC2209

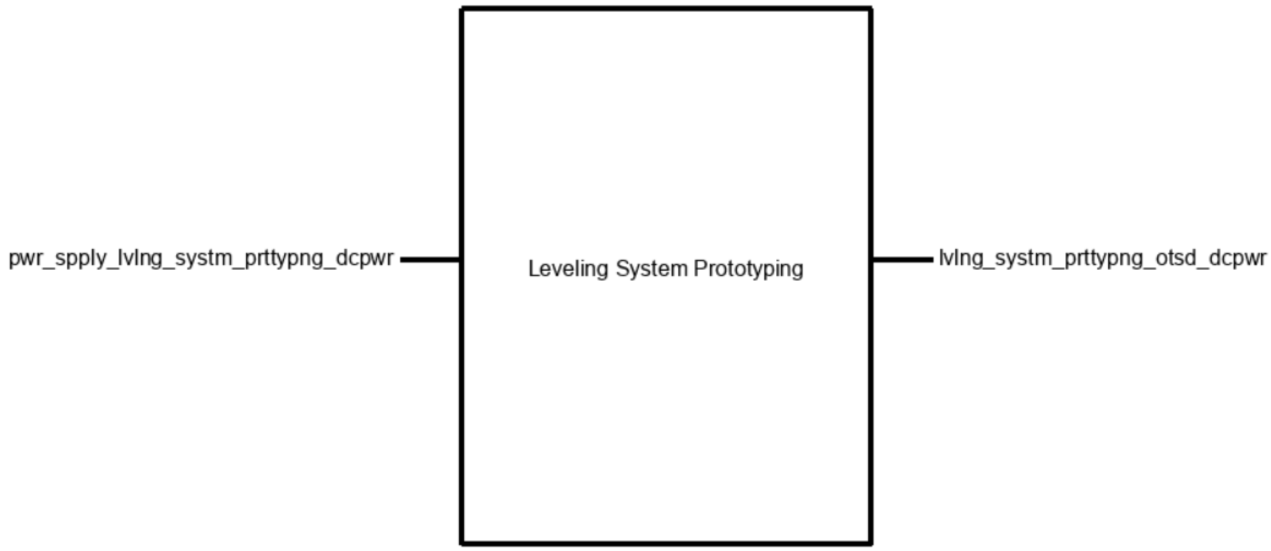


Fig. 4.5.2.2. Block of Leveling system prototyping

4.5.3. General Validation

This block is required to be able to support the weight of our system and do so accurately. That means we desire smooth movement and fine movement.

At first, we tried to use regular dc motors to drive the free wheel motors of the system, but we found that those did not have the fine movements required for keeping our system level.. This led us to search and find the NEMA 17 stepper motors, which are rated for 200 steps per revolution meaning that we gained more than enough resolution to the accuracy of the movement.

The ME team created 2 support beams on the side of the motor free spinning and a slab on both ends to create a surface with the force evenly spread across the 3 motors. Attached below is an image of the motor and said motor housing.

Fig. 4.5.3.1. Image of prototyping

4.5.4. Interface Validation

Interface Property	Why is this interface this value?	Why do you know that your design details <u>for this block</u> above meet or exceed each property?

pwr_spply_lvlng_systm_prttypng_dcpwr : Input

Inominal: 0.6	This is the expected current draw to supply 4 of the motors with current when driving. It is also	While it could be significantly less or more because we have a 24 V power supply and the ability for

	about the same amount of current used to drive the holding motors.	variable speed and torque, our “nominal” value is relative. If we are moving slowly but want to move the weight of the bot which is about 60 lbs, we would need a high torque but a relatively low voltage at the output. So this is just an example.
Ipeak: I ascending: 2.1A	This is the max draw possible by the NEMA 17 stepper motors when driving maximum torque	This here is a situation in our system where are at the max speed which depends on the voltage on the NEMA 17 motors and the max torque which depends on the current through those motors.
Other: I descending: 1.9A	This is the max draw possible by the NEMA 17 stepper motors when driving maximum torque	there is a situation in our system where are at the max speed which depends on the voltage on the NEMA 17 motors and the max torque which depends on the current through those motors.
Vnominal: 24V	This is the voltage range required to power the driver logic which controls the current phase distribution on the motors	This is the power to the logic on board the TMC2209 drivers we use to send PWM signals to the NEMA 17 motors.
Vmax: 24 V	This is the temporary max voltage our system can be before failure.	This is the max current our TMC2209 part can take before it will fail. This is not the only limiting factor cause the motor is also rated at a max value of 28 V.
Vmin: 24 V	This is the cutoff minimum voltage our system can be before we cannot drive enough speed to meet our time limitation requirement	This is the minimum amount of voltage that can be put across the motors that it will still rotate. Any lower and it will not.

lvlng_systm_prttypng_otsd_dcpwr: Output

Inominal: Holding per motor phase: 0.63A	This is the current per phase for each drive motor under steady-state operation	This is the current required to drive the movement motors when within “nominal” movement. Here it is described per phase. Each motor has 2 phases.
Ipeak: I ascending: 1.5A	This is the holding torque current for the feet motors	These are the versions of the motor that have a free spinning screw down the center. This means that this is the required current to hold the torque of the current position of the system
Vmax:Per Motor Phase: 1.8V	This is the max available power to us in our system from the power supply	Highest DC voltage available from our power supply.
Vmin: Per Motor Phase: 1.8V	This is the minimum voltage that can be used to drive the motors in our system	This is the minimum amount of voltage where if it were to be passed the driver and the motors would function near identical to or max value.
Vnominal: Per Motor Phase: 1.8V	This is the voltage at any point per motor phase in our system	It meets the specification
Other: I descending:1.5A Per Motor Phase.	This represents the value of the current per motor phase while the level of the system is being reduced	It meets the spec as it does not draw more than required of the power supply.

4.5.5. Verification Process

The following steps will be used to verify the functionality of this block.
First:
Connect the “Hulk” to wall power.
Listen for the beep from the buzzer system.
Use the joysticks which are simulating movement from our vision system.
Move the left stick up and observe all stabilization motors ascend until they meet the limit switch and stop.

Move the left stick down and observe all stabilization motors descend until they meet the limit switch and stop.

At all times during this test. There should be no slip between the motor and the stabilization. That visibly and with the help of a level gauge. No changes when no motion is initiated with the joystick.

4.5.6. References and File Links

“Pololu - AMIS-30543 Stepper Motor Driver Carrier.” *Www.pololu.com*, www.pololu.com/product/2970. Accessed 15 Mar. 2023.

4.5.7. Revision Table

3/15/2023	First iteration of this block. Filled out all sections of the block

4.6. Motor control prototyping

4.6.1. Description

This block is used to represent the voltage and current values sent to the motors in our system and how they react and behave based on those values. There are four motors in total in this block. Four of them are mounted to the four corners of the 3D printing robot and are used to drive the "Hulk" in the direction of the print surface. They are attached to "mechanum" wheels which allow rotation on the center point axis of the system. The name of the motor being used in our system is the 2-phase bipolar NEMA 17 stepper motor.

The type of stepper motor we use in our system has divided a rotation up to 1.8 degrees per “step”. This means that it takes 200 "steps" of the motor to complete a rotation. This motor was chosen because our system requires finer movements. The ability to make up to a millimeter of adjustments when factoring in the friction of the ground was required. When our vision system coordinates have been converted into drive signals, the motors will be driven with precision autonomously through the room. It satisfies the "distance x" and "distance y" requirements of the project by aiding that movement. It also satisfies the "time limitation" requirement because it can be driven with a variable speed.

4.6.2. Design

Below is the design of the motor control system. There are four motors in general. Four of them are used for the movement of the robot on wheels. The other three have a free spinning screw going through them that allows the lateral mounting of flat feet-like materials on both sides. The name of the motor used in our design is the version of the NEMA 17 that is rated for 24V and 1.7A per phase. All the motors in our system are two-phase devices. This means each motor takes about 3.4 A per motor. But because of the nature of the requirements for the design, all seven motors are never in use at the same time. The four drive motors are used to get to the print surface and the other three motors are used to stabilize the system. So our theoretical max current at any given time would be 3.4 X 4 = 13.6 A. All seven motors are controlled by our MAIXduino which will not be covered in this block, but they are driven via the AMIS-30543 driver. The AMIS-30543 allows the use of variable speed and current in our motors. It also controls the PWM sequence that will be required to run these AC motors from dc power. This can be done both through hardware with the potentiometer on the board and software with the libraries that accompany the part. **Figure** 4.6.2.1 has an image of the AMIS-30543 part. **Figure** 4.6.2.2 is a black-box diagram of the motor control prototyping block.

Because our motor needs to be run with 24V which is much higher than the max voltage from the Arduino which is 5V, we run our motors directly from the power supply. **Figure x** has the wiring diagram to the protoboard currently being used for testing. Once testing is concluded a PCB will be built to replace the protoboard.

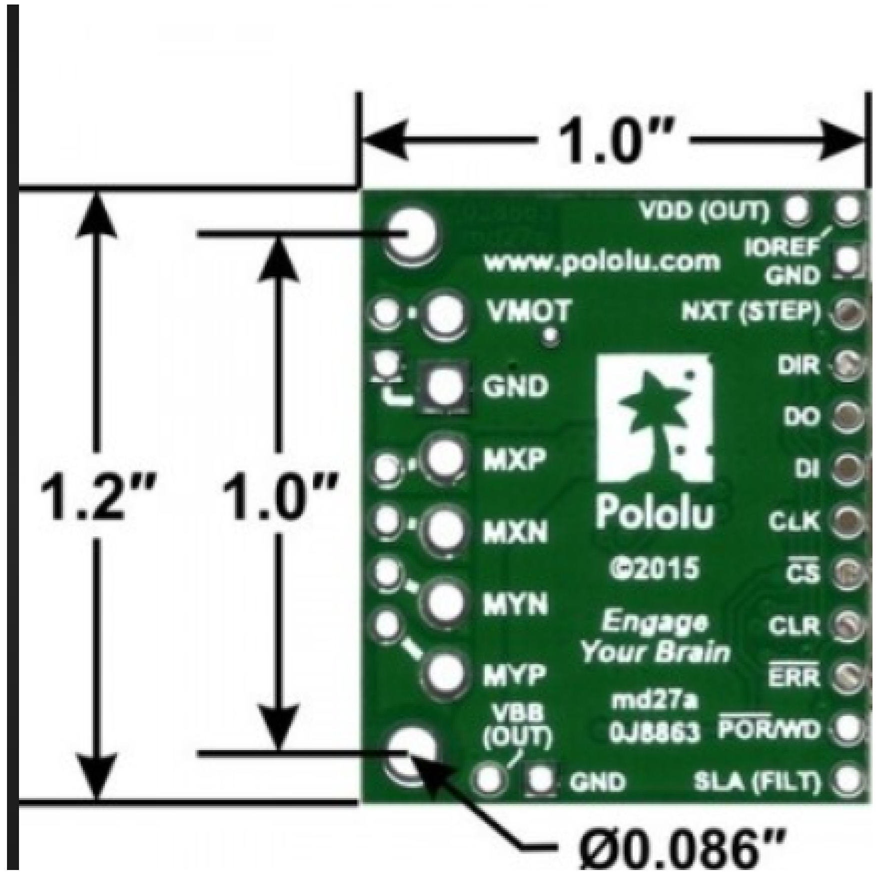


Fig. 4.6.2.1. AMIS-30543

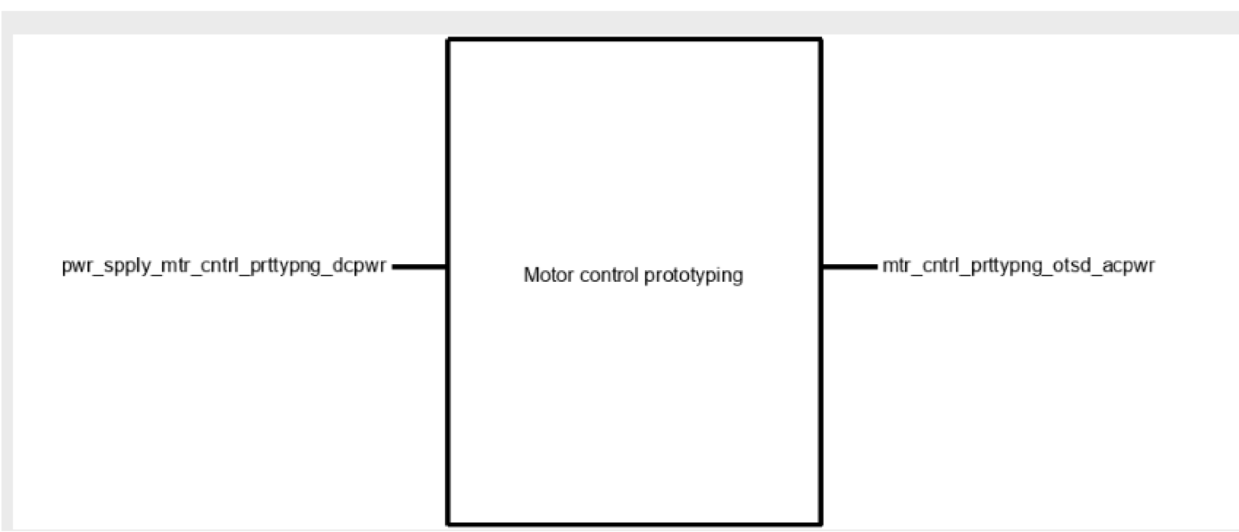


Fig. 4.6.2.2. Block of Motor control prototyping

4.6.3. General Validation

This block is required to be able to transport our system and do so accurately. That means we desire smooth movement and fine movement.

At first, we tried to use regular dc motors to drive the wheels of the system, but we found that those were much too weak to drive the “Hulk” (3D printing robot) because it was over 30 pounds before we added our drive system and will weight in close to 50 pounds after it is included. This led us to search and find the NEMA 17 stepper motors, which are rated for about 10-12lbs each meaning we can hold about 40-46 pounds of weight and drive it around within our 10 ft requirement.

The ME team also decided on Mechanum wheels as they were the most cost-effective wheels for the drive system. They also 3d printed flats slabs that went on either side of the free-spinning motors for the robot to stay on. This was also the most cost-effective method as drivers with more features were more expensive and the dc motors while being cheap[er could not achieve the level of precision needed for movement in the system.

4.6.4. Interface Validation

Interface Property	**Why is this interface this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**

mtr_cntrl_prttypng_otsd_acpwr : Output

Inominal: 8A	This is the expected current draw to supply 4 of the motors with current when driving. It is also about the same amount of current used to drive the holding motors.	While it could be significantly less or more because we have a 24 V power supply and the ability for variable speed and torque, our “nominal” value is relative. If we are moving slowly but want to move the weight of the bot which is about 60 lbs, we would need a high torque but a relatively low voltage at the output. So this is just an example.

Ipeak: 13.6A	This is the max draw possible by the NEMA 17 stepper motors when driving maximum torque	This here is a situation in our system where are at the max speed which depends on the voltage on the NEMA 17 motors and the max torque which depends on the current through those motors.
Iripple:	N/A	N/A
Other Vlogic: 3-5V	This is the voltage range required to power the driver logic which controls the current phase distribution on the motors	This is the power to the logic on board the TMC2209 drivers we use to send PWM signals to the NEMA 17 motors.
Vmax: 28 V	This is the temporary max voltage our system can be before failure.	This is the max current our TMC2209 part can take before it will fail. This is not the only limiting factor cause the motor is also rated at a max value of 28 V.
Vmin: 4.75 V	This is the cutoff minimum voltage our system can be before we cannot drive enough speed to meet our time limitation requirement	This is the minimum amount of voltage that can be put across the motors that it will still rotate. Any lower and it will not.

pwr_spply_mtr_cntrl_prttypng_dcpwr : Input

Inominal: Drive: 1.5A	This is the current per phase for each drive motor under steady-state operation	This is the current required to drive the movement motors when within “nominal” movement. Here it is described per phase. Each motor has 2 phases.
Inominal: Stability: 1A	This is the holding torque current for the feet motors	These are the versions of the motor that have a free spinning screw down the center. This means that this is the required current to hold the torque of the current position of the system
Vmax: 24 V	This is the max available power	Highest DC voltage available from

	to us in our system from the power supply	our power supply.
Vmin: 21 V	This is the minimum voltage that can be used to drive the motors in our system	This is the minimum amount of voltage where if it were to be passed the driver and the motors would function near identical to or max value.

4.6.5. Verification Process

1. Open the sample movement code on the Arduino IDE
2. Make sure the analog joysticks “simulating our computer vision system” is connected
3. Put a Digital Multimeter in line with the 24 V DC power supply and set it to read current.
4. Run the code on the Arduino.
5. Log the current value on the multimeter. If it is < 0.5 A it is undesired behavior
6. Reconnect DMM from 24V DC power supply to the ground and set it to measure DC voltage
7. VDC must be 24 V.
8. When you pull the left analog stick forward, you should observe forward movement.
9. You should also observe an increase in speed based on how far you push or pull the left stick
10. The right stick should allow lateral movement, you should be able to move side to side
11. After pushing in the left stick, you should be able to turn left or right on the center point of the robot.
12. After pushing in the right stick, the feet should deploy and should stabilize the robot.

4.6.6. References and File Links

“BIGTREETECH TMC2209 v1.3 stepper motor driver,” *Biqu Equipment*. [Online]. Available:

https://biqu.equipment/products/bigtreetech-tmc2209-stepper-motor-driver-for-3d-printer-board-vs-tmc2208. [Accessed: 11-Feb-2023].

G. Styger, B. M., Greg, Moe, and J. Finlayson, “Nema 17 Stepper Motor,” *OpenBuilds Part Store*. [Online]. Available: https://openbuildspartstore.com/nema-17-stepper-motor/. [Accessed: 11-Feb-2023].

“Stepper Motors - Everything You Need To Know about Stepper Motors,” *Oriental Motor U.S.A. Corp.* [Online]. Available: https://www.orientalmotor.com/stepper-motors/technology/everything-about-stepper-motors.html. [Accessed: 11-Feb-2023].

4.6.7. Revision Table

1/20/2023	Oluwaseun Samuel Popoola: Started Motor controller block validation paper. Added beginnings of description
1/23/2023	OSP: added design and general validation sections, started interface validation sections,
2/11/2023	OSP: rewrote the description and design sections, added the interface validation, verification plan, and references, and file links sections.
3/15/2023	Made changes to description and design. Included updated images of new driver

4.7. Camera

4.7.1. Description

The camera block is a critical component in the functioning of the Mobile 3D Printing Robot. Its main function is to assist in navigation and surface recognition by continuously capturing images and sending the processed data to the microcontroller. This allows the robot to have a real-time understanding of its surroundings, and make necessary adjustments to its movements to ensure precision and accuracy for subsequent printing. Currently, the camera will be a microcontroller-equipped camera with a 24P connection and will be placed on the front of the robot as its "eyes".

The camera block contains imaging sensors and processing units that work together to identify important features in the captured image. It also has the ability to recognize colored circles on the surface, which is critical for adjusting the robot's position. A green circle is placed in the center of a given surface and a red circle is placed equally spaced on each side of the green circle. By determining whether the green circle is in the center of the image, the system will get the approximate orientation of the robot relative to the specified surface and thus guide the robot's motion. If the three circles are in a straight line and the green circle is centered in the image, the robot is considered to be in the center of the surface.

4.7.2. Design

The Mobile 3D Printing Robot should autonomously perform tasks and print vertically on surfaces. Its camera block gives it "eyes" to guide movement and determine the surface's normal position. To accomplish this, the camera block must be able to capture the image and perform circle detection, color recognition, and computations. Inside this block, the camera will be operated by the Maixduino microcontroller using code built on Maixpy IDE and written in Micropython (python for microcontrollers).

As we can see in Fig 1, this block has one input interface, otsd_cmr_envin, and one output interface, cmr_mcrcntrllr_data. The input is an image captured from the surrounding environment, and through processing, the coordinates and the motion instruction will send to the microcontroller.

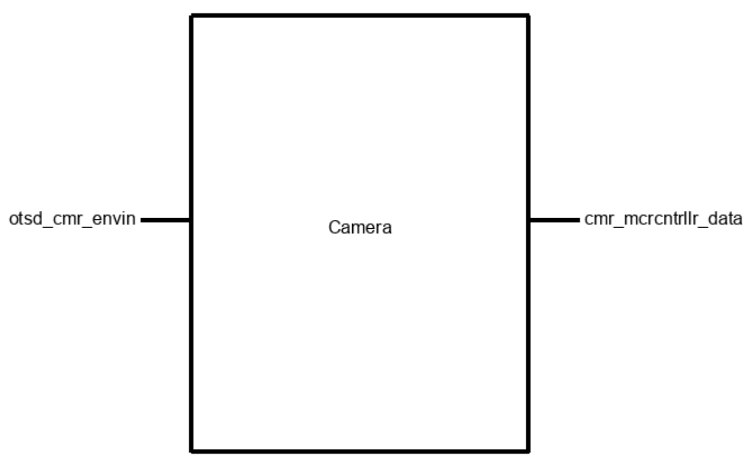


Fig. 4.7.2.1. Black Box Diagram of Camera Block

The first step in the code flowchart (Fig 2) is preparation. This includes initializing the camera, setting the pixel format and frame size for optimal image quality, skipping frames to stabilize the image, adjusting the image gain for improved identification, and disabling auto gain and auto white balance to prevent color distortion. Next, the system captures an image and detects circles within it. If no circles are detected, the system will sleep for one second and send a "No target circles detected" notification to the microcontroller. If circles are detected, the system will count the pixel color of the detected circular area, then compare it to the set color range. Circles in green or red will be circled in black, while others will be framed in white. If a green circle is detected, the system will compare its x-coordinate center to the image center to determine the robot's orientation and guide its movement. If two red circles and a centered green circle in a straight line are detected, the robot is assumed to face the print surface vertically. Otherwise, the camera will wait for one second and send a notification to the microcontroller.

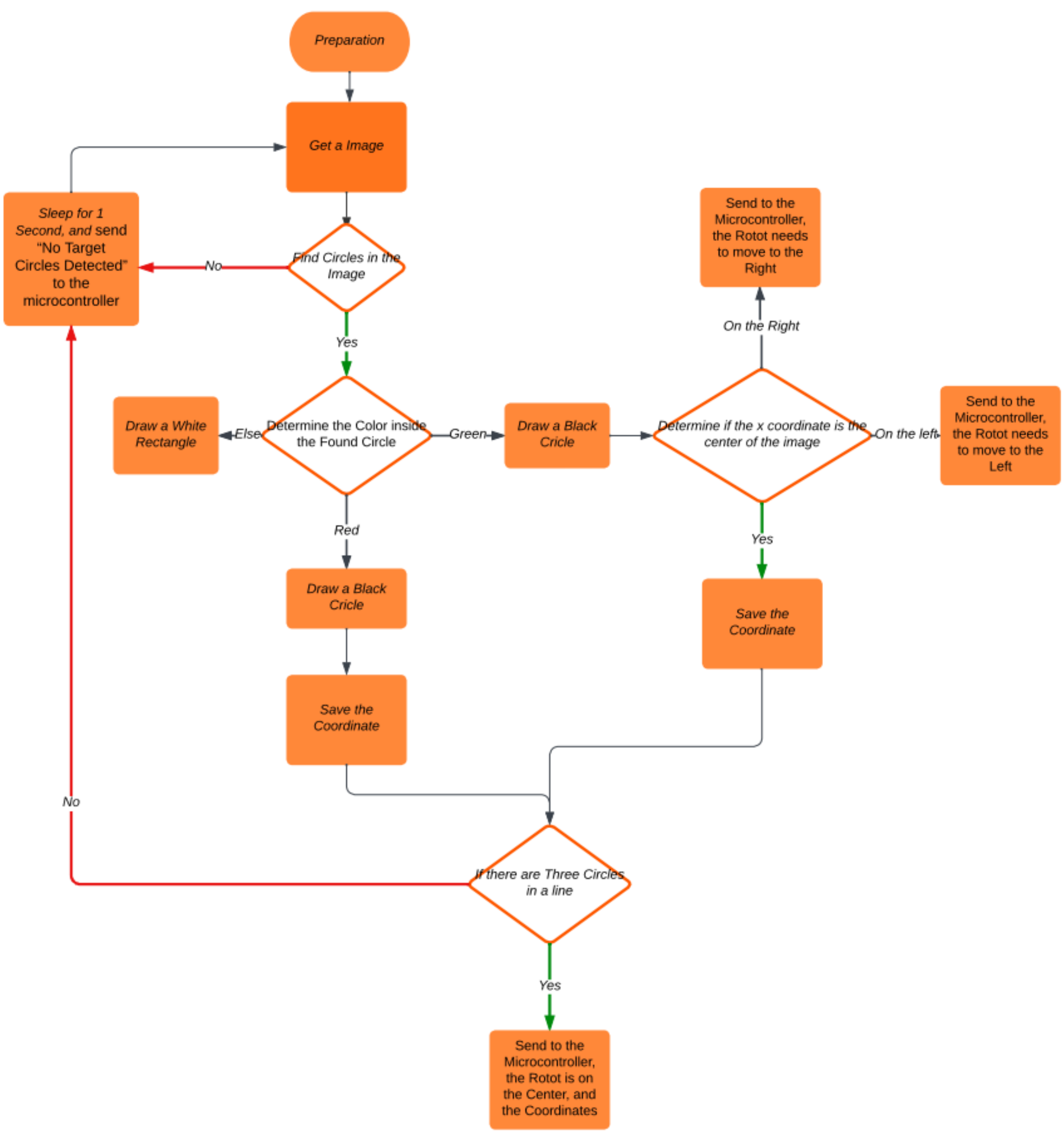


Fig. 4.7.2.2. Code Flowchart

4.7.3. General Validation

The camera block exists because it is necessary for the system to achieve its requirements of completing tasks without human intervention and ensuring vertical alignment of the robot with the print surface. The camera block is controlled by the microcontroller. With the built code, the camera can automatically take images once per second, providing the robot with “eyes” and automatically processing the images to detect circles and recognize color, a part of the complete automation requirements. Also, by checking the positioning of the green circle in the image, the system can estimate the orientation of the robot and guide its movement accordingly. If the three circles are aligned in a straight line, with the green circle in the center of the image, it can be concluded that the robot is positioned at the center of the surface, otherwise, it can give the robot movement instruction to ensure the robot can print vertically.

We selected Maixduino instead of the commonly used Arduino as it offers both Arduino functions and the ability to perform computer vision algorithms, making it ideal for our needs of collecting environmental data and processing picture information. It also includes an ESP32 module for WiFi and Bluetooth[1], meeting our additional requirements for other modules. At 32 dollars for Maixduino plus a camera, compared to just a 5-dollar difference for an Arduino Uno, Maixduino offers excellent value for money.

The camera is an OV2640 with a 24P interface[2], which can be easily connected to the development board for seamless data transfer. It has a wide angle of 140 degrees compared to the camera equipped with the purchase of Maixduino, allowing for a larger viewing range[2], and is not too expensive at only 9 dollars. Its capability to perform vision algorithms using sensors and image libraries allows us to access real-time images directly from the camera through code. This enables the system to analyze colors and perform calculations to determine the robot's position and give movement commands in real time.

An alternate solution to the camera block for ensuring the robot is vertical to the surface is a depth sensor. A depth sensor uses triangulation to calculate the distance of objects in the environment and detect the print surface. The sensor sends out beams of light, which reflect off the surface and is received by the sensor. By measuring the time it takes for the beams of light to travel to the surface and back, the sensor can determine the distance to the surface.

In addition to measuring distance, depth sensors can also detect obstacles in the environment and adjust the robot's position to avoid the collision. This allows for greater safety in the printing process, as the robot can automatically avoid obstacles and prevent damage to itself or the surrounding environment.

The use of a depth sensor also eliminates the need for a camera, reducing the cost and complexity of the system. The depth sensor is also less prone to errors, as it does not require the processing of images and is less affected by lighting conditions.

Moreover, a depth sensor can be combined with a tilt sensor to ensure the robot is in a vertical position. The tilt sensor will detect the angle at which the robot is positioned, and the depth sensor will determine the distance of the surface. By combining these two pieces of information, the robot will be able to calculate its position relative to the surface and adjust itself accordingly to ensure it is in a vertical position.

In addition, the depth sensor can be used in conjunction with an accelerometer, which will detect the orientation of the robot and help to ensure it is vertical. The accelerometer will detect changes in the robot's orientation and send this information to the control system, which will adjust the robot's position to maintain its vertical alignment with the surface.

4.7.4. Interface Validation

Interface Property	**Why is this interfac this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**

otsd_cmr_envin : Input

Other: Objects between 0.5ft and 2 ft	The system needs to identify objects between 0.5fft and 2ft	The object on 0.5ft and 2ft can be circles in the image.
Other: Must have at least a 90 degree field of view	The system with 90 degree field of view could enlarge the field of robot.	The whole 8.5x11 inches paper could be seen in the camera when the distance is 5.5 inches.
Other: Object to detect include colors: red and green	The systems needs to identify the colors of objects.	Red and green circles could be circled in black in the image.

cmr_mcrcntrllr_data : Output

Other: Pixel coordinates of up to three circles' centers	The system will perform the secondary verification of the robot's position in the next block, the positioning system.	The coordinates will be stored in the list which is convenient to use again and they will be printed on the screen.
Other: Movement directions. Eg. Left, Right, etc...	The robot needs to find the correct surface, and the print position needs to be precise, so it needs to send instructions to guide the robot to move.	The code will store movement directions and print them on the screen.
Other: Data update at least once 5 seconds.	Since we all think we need to save energy and getting images and sending updated data every 5 seconds are enough, during this time, the system does some calculations and moves.	There is a function call used in the "sleep for 1 second" in the flowchart.

4.7.5. Verification Process

The camera block code is built on Maixpy IDE using Micropython language. The verification will be done on Maixpy.

1. Connect the camera to the Arduino board, and the board and the computer by type-C connector.
2. Run the code.
3. Observe the right side of the Maixpy to see if the current image captured by the camera is displayed.
4. Print the image frame size.
5. Keeping no circles in the field of view, watch Maixpy's terminal to see if a prompt pops up.
6. Place green and red rectangles and circles in the field of view, and check whether they will be marked in the image.
7. Place green, red, and blue circles in the field of view, and check if they will be marked in the image.
8. Place two red and green circles in the field of view, and check if their coordinates are saved.
9. Randomly change these three circles' positions, and check the notifications.

Pass condition:

1. The image frame size is 320 x 240.

2. Rectangles don't be marked in the image.
3. Circles of colors other than red and green will be framed in white.
4. Red and green circles will be circled in black.
5. The notification "the robot is in the center" shows on the screen only when the green is in the center of the picture and in between the two reds and their circle centers are in a straight line, and their center coordinates will be printed on the screen.
6. The notification " the robot is on the left of the surface, move to the right" shows on the screen when the green circle is at the right of the image center.
7. The notification " the robot is on the right of the surface, move to the left" shows on the screen when the green circle is at the left of the image center.
8. After the image is captured, it will pause for a second before the next image is captured.
9. The notification/instruction will be printed on the screen once per second.

4.7.6. References and File Links

[1] "Sipeed Hardware - Sipeed Wiki," Sipeed.com, 2022. https://wiki.sipeed.com/hardware/en (accessed Feb. 11, 2023).

[2] "Amazon.com: JESSINIE OV2640 Camera Module 200W Pixel Large Wide Angle 140 Degree Monitoring Identification Cam Module 2MP CMOS Mini Camera 24P Connector Support JPEG Output : Electronics," *www.amazon.com*. https://www.amazon.com/JESSINIE-Monitoring-Identification-Connector-Support/dp/B0B5XGMTSZ/ref=sr_1_2?crid=2K8A5QFJ14511&keywords=24p+camera&qid=1676179240&sprefix=24p+camer%2Caps%2C142&sr=8-2 (accessed Feb. 11, 2023).

4.7.7. Revision Table

3/12/2023	Updated the interface validation
2/11/2023	Modified sections based on Rachel's suggestions: add the content about what requirements produce the block, why the block meets the requirements, add camera reference, and revise the IEEE format. Modified sections to enrich content: add alternate solutions and so on.
1/20/2023	Updated the whole content
1/19/2023	Initial document created

4.8. Positioning System

4.8.1. Description

The location system will collect data from the lidar and camera blocks, and send the coordinates and angle of the surface relative to the robot to the path planning block. This information will enable the path planning block to guide the robot's movement.

4.8.2. Design

As we can see in Fig 1, this block has one input interface, mcrcntrllr_pstnng_systm_data, and one output interface, pstnng_systm_pthng_data. The input is the values from the Lidar and Camera blocks, and through calculations, the final coordinates and angle will be passed to the path planning block.

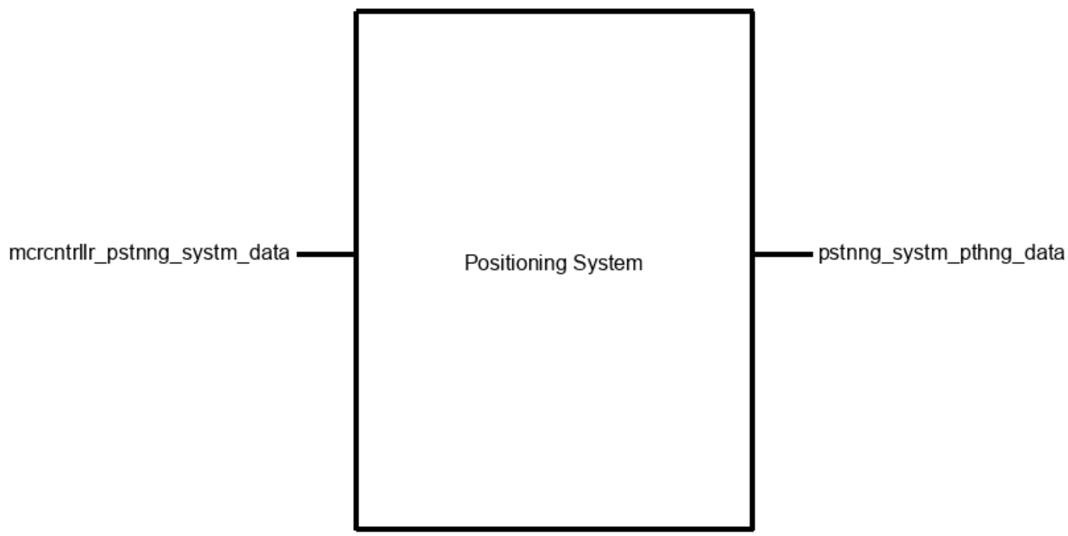


Fig. 4.8.2.1. Black Box Diagram of Positioning System Block

4.8.3. General Validation

The positioning systen block is of an intermediate level and is concerned with extracting information from the data that is stored on the SD card by both the camera and the lidar. The lidar data undergoes calculations in order to obtain both angle and coordinate data, while the camera data undergoes operations on the pixel coordinates to determine the direction of movement, which is then verified and updated. The resulting coordinates, angles, and movement directions are then saved as a backup on the SD card and transmitted to the path planning block.

4.8.4. Interface Validation

Interface Property	Why is this interface this value?	Why do you know that your design details <u>for this block</u> above meet or exceed each property?

mcrcntrllr_pstnng_systm_data : Input

Other: The distance from Lidar measured.	Come from the Lidar block	The system can print the distance on the screen to check the value.
Other: The pixel coordinates of the circles' center.	Come from the Camera block	The coordinate can be printed on the screen and calculated again to check the movement.
Other: Movement directions. Eg. Left, Right, etc...	Come from the Camera block	The movement directions can be printed and checked again based on the result of coordinate calculation.

pstnng_systm_pthng_data : Output

Other: The angle of the surface based on the robot.	The system needs the angle to help align the printing path.	The code will store angles and print them on the screen.
Other: The coordinate of the surafce based on the robot (x, y).	The system needs the coordinate to help movement and record the location.	The coordinates will be stored in the list which is convenient to use again and they will be printed on the screen.
Other: Movement directions. Eg. Left, Right, etc...	The system needs the movement direction to guide the robot.	The code will store movement directions and print them on the screen.

4.8.5. Verification Process

The positioning system block code is built on Maixpy IDE using Micropython language. The verification will be done on Maixpy.

1. Connect the board to the computer using type-C cable.
2. Run the code
3. Check the distance, pixel coordinates and movement directions are read from the file in the SD card.
4. Observe the coordinates and angle the system computed, and the updated movement directions.

5. Manually calculate the coordinates and angle.

Pass conditions:

1. The coordinates and angles are same values from manually calculation and code.
2. Manually calculate the pixel coordinates, and the result is fit the update movement directions.

4.8.6. References and File Links

[1] “Sipeed Hardware - Sipeed Wiki,” Sipeed.com, 2022. https://wiki.sipeed.com/hardware/en (accessed Feb. 11, 2023).

4.8.7. Revision Table

Date	Author	Content
3/10/2023	Xinrui	Added content of section 4.8

5. System Verification Evidence

5.1. Universal Constraints

5.1.1. The system may not include a breadboard.

This image is from the inside of our robot.

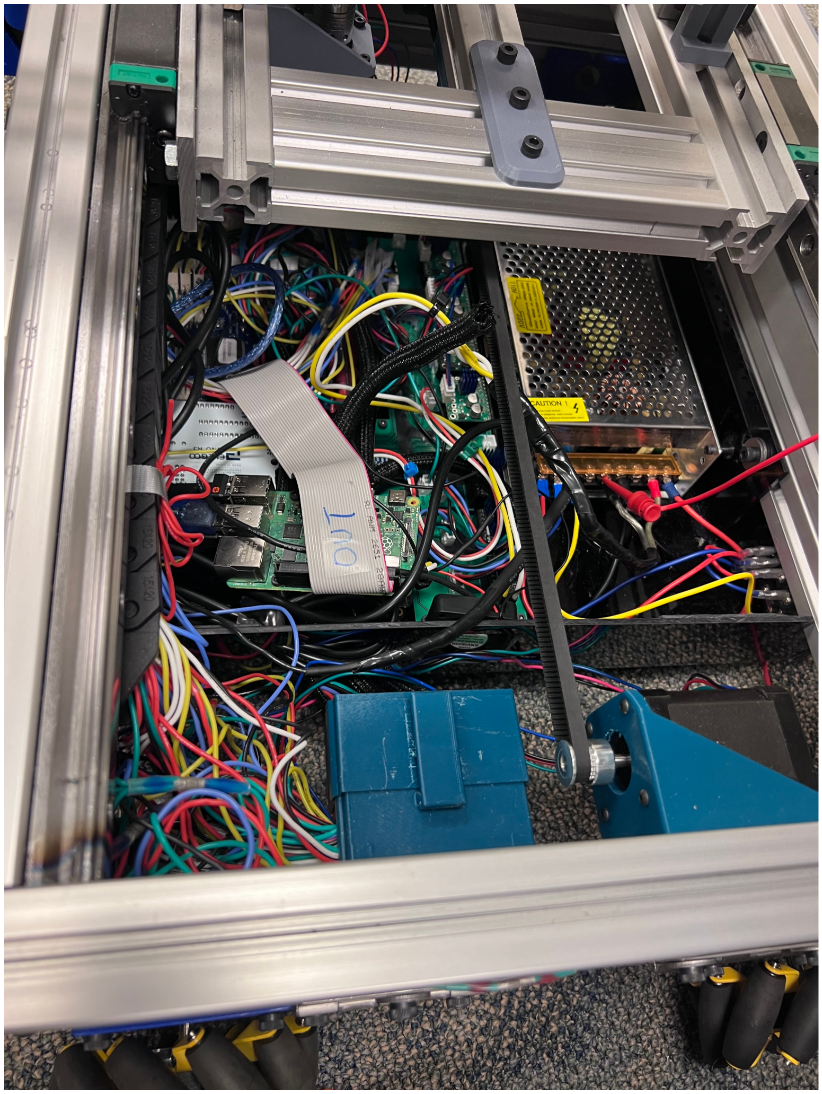


Fig. 5.1.1.1. Inside of robot

5.1.2. The final system must contain a student designed PCB.

The below figure shows our designed PCB for motor driving. It was designed in KICAD. Below attached as well is an image of the schematic and the order information:

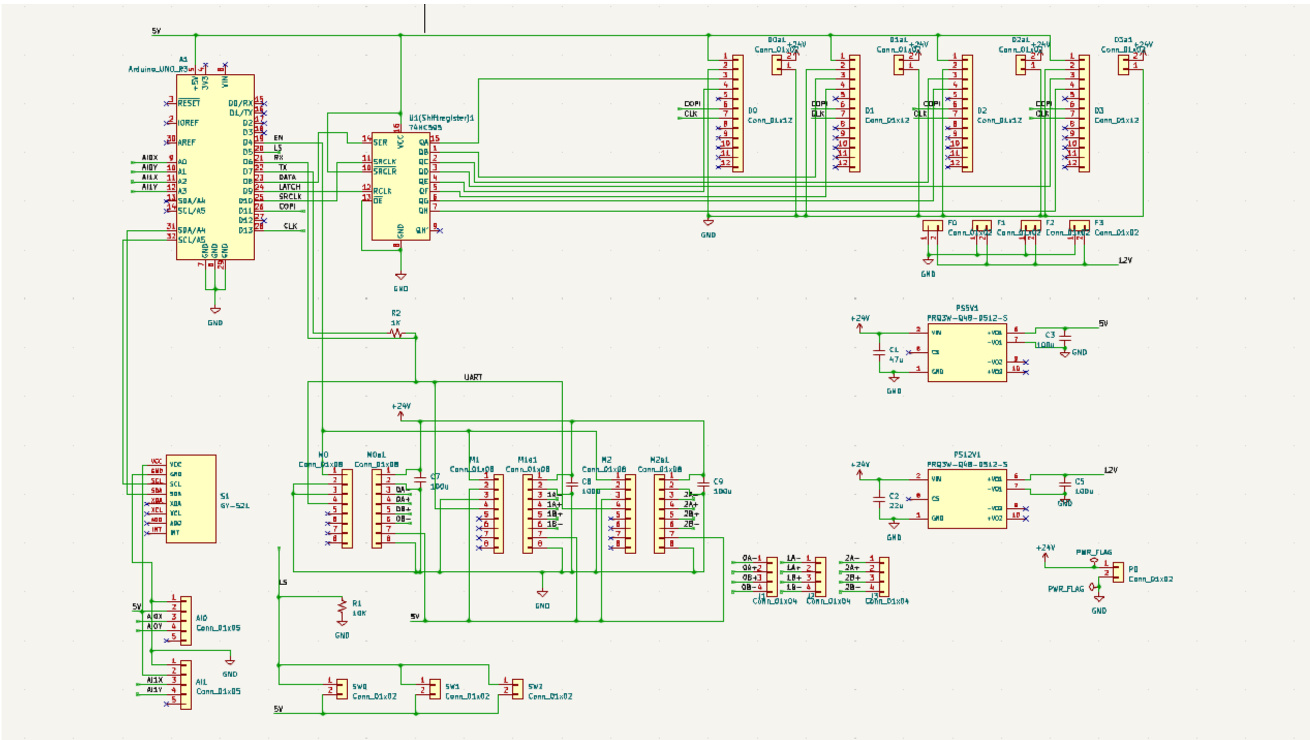


Fig. 5.1.2.1. Student designed PCB schematic

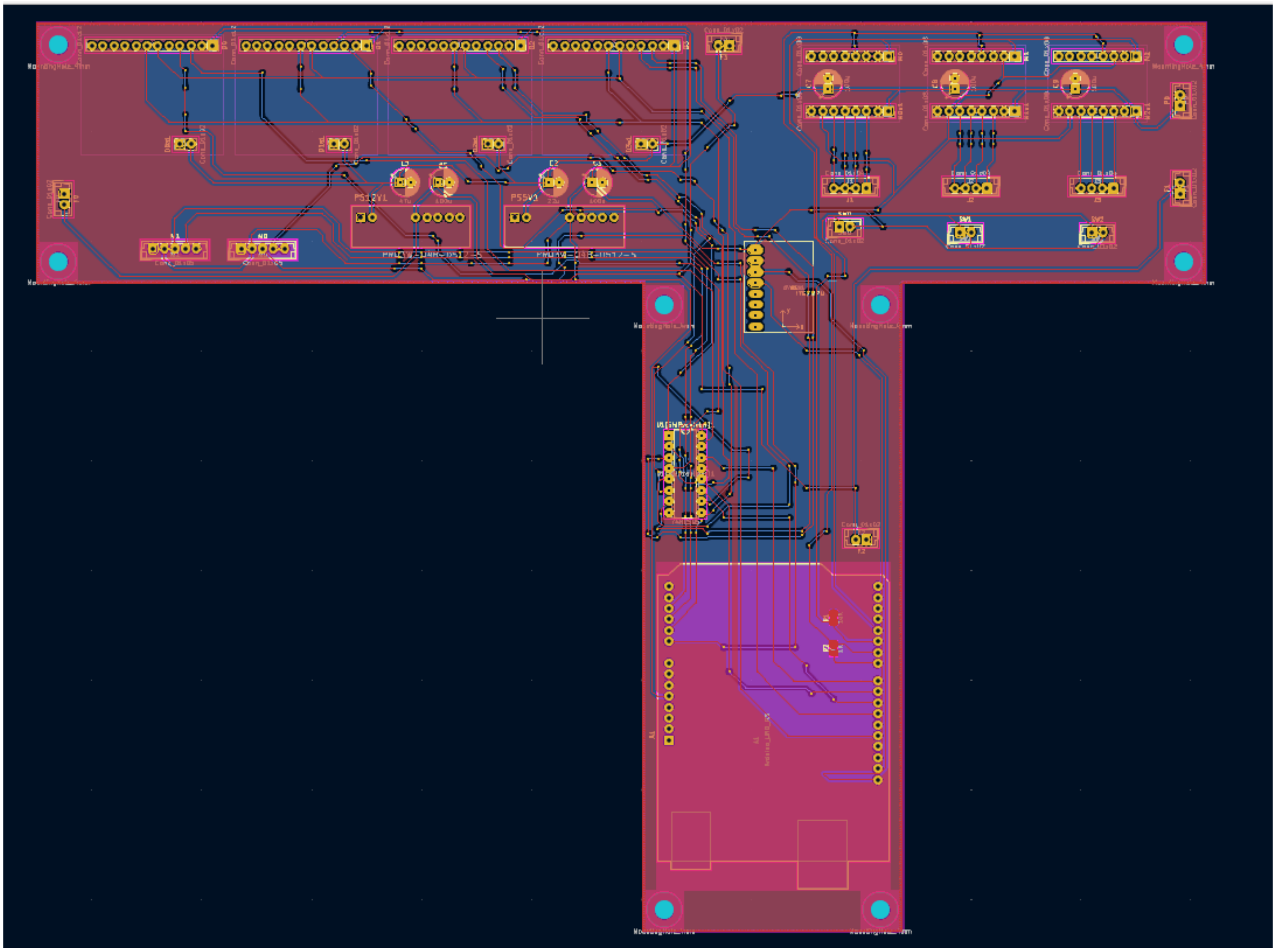


Fig. 5.1.2.2. Student Designed PCB

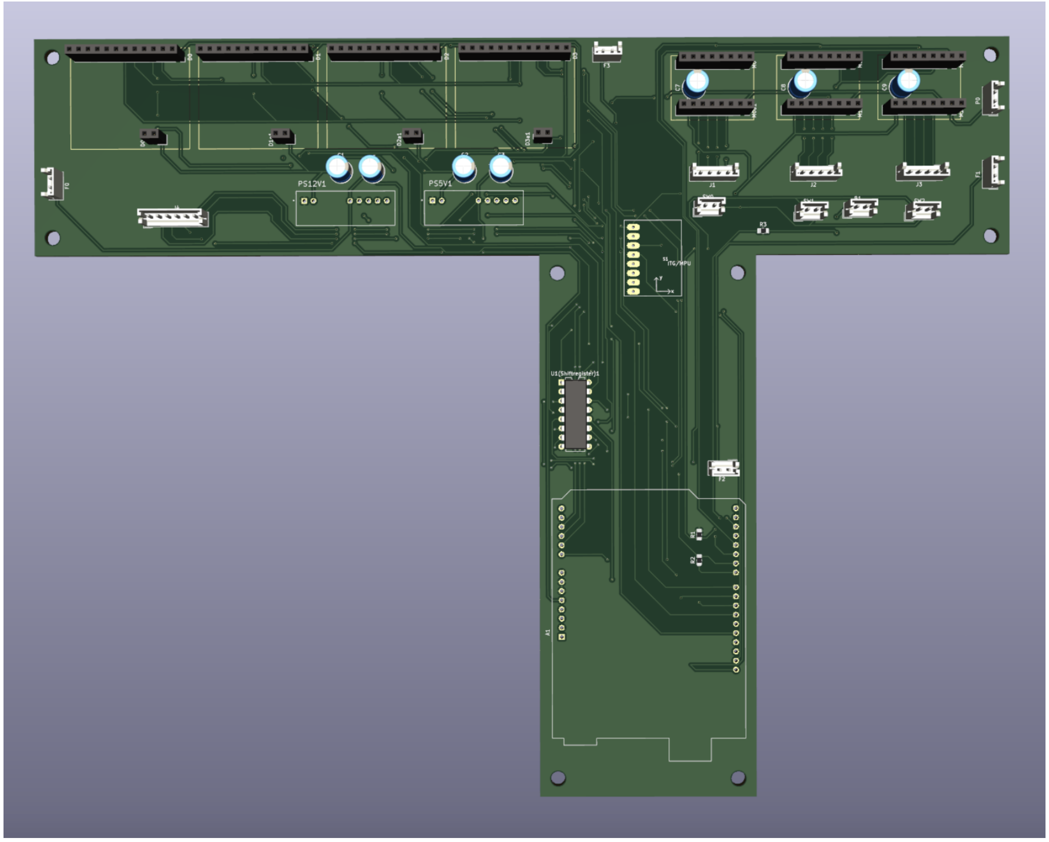


Fig. 5.1.2.3. 3D viewer version of PCB

5.1.3. All connections to PCBs must use connectors.

Our connections use through holes instead. This allows for future customization of the PCB and the PCB itself did not need to be any more advance. This is something our project partner wanted.

5.1.4. All power supplies in the system must be at least 65% efficient.

RealSense: 2.25W

https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/Intel-RealSense-D400-Series-Datasheet.pdf

Raspberry Pi: 6W

https://www.ecoenergygeek.com/raspberry-pi-power-consumption/

https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-4-Product-Brief.pdf

Lidar: 0.33W

https://www.documents.lightware.co.za/LW20%20-%20LiDAR%20Manual%20-%20Rev%2012.pdf

Arduino: 3.5W

From ME team

Power Supply: 5V * 3A = 15W

https://www.amazon.com/ALITOVE-Transformer-Universal-Regulated-Switching/dp/B078RYWZMH/?th=1

Calculation:

80%

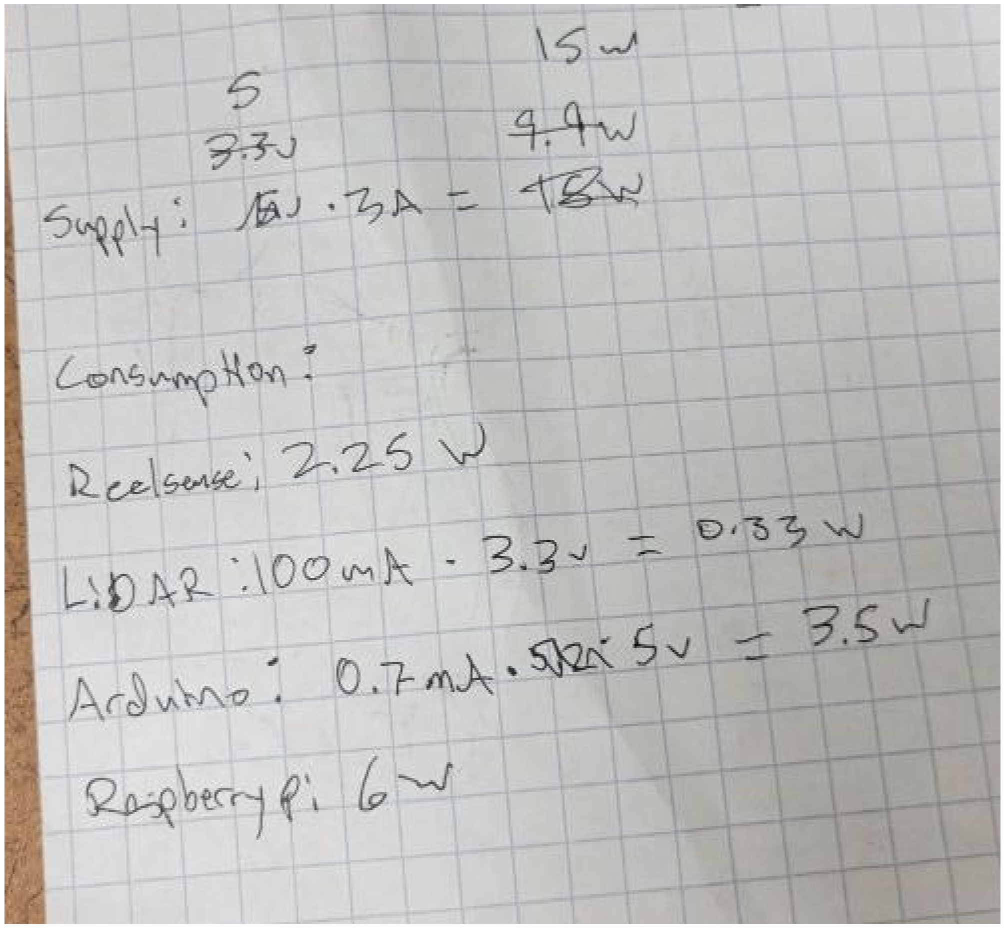


Fig. 5.1.4.1. Calculation of power efficiency

5.1.5. The system may be no more than 50% built from purchased 'modules.'

Our system has a total of eight modules, and seven are built modules. Each of these built modules have been customized through coding that we have implemented.

Built Modules	**Purchased Modules**
LiDAR sensor	RaspberryPi
RealSense camera	
Accelerometer	
Positioning system	

Touch sensor	
WiFi Module	
Motor control	

5.2. Requirements

5.2.1. Angle z

5.2.1.1. Project Partner Requirement

Develop a movement system to approach a desired print surface.

5.2.1.2. Engineering Requirement

The system will orient the printer normal to the printer surface (rotation about z) with error less than 1.5 degrees because the 3D printer's probe is normal to the printing surface in order to successfully run.

5.2.1.3. Testing Method

Inspection. Start the location process of the robot and see if the printer normal matches the printer surface.

5.2.1.4. Verification Process

Move the robot, and when the robot automatically arrives at the target surface, measure the angle between the probe and surface.

Pass Condition: When the robot reaches the target surface, the robot’s probe is normal to the surface with error less than 1.5 degree.

5.2.1.5. Testing Evidence

We have evidence of this requirement being met on video. In the video you will see that after movement, the probe is normal to the surface where the robot stops.

Link to video: https://youtu.be/KomwuV73EnY

5.2.2. Autonomous

5.2.2.1. Project Partner Requirement

Robot must operate without human interference.

5.2.2.2. Engineering Requirement

The system will complete all tasks without interaction with a human operator and will be able to return to the original position after completing a job.

5.2.2.3. Testing Method

Demonstration

5.2.2.4. Verification Process

Run the robot, and observe whether the robot will automatically finish all tasks and return to the starting location.

Pass Condition: The robot completes all tasks automatically and return the starting location after all tasks have been completed.

5.2.2.5. Testing Evidence

We have evidence of this requirement being met on video. In the video you will see that the robot autonomously moves to the print surface and returns to the starting position. Everything done in the video is the robot working autonomously (finding the surface and driving up to it).

Link to video: https://youtu.be/KomwuV73EnY

5.2.3. Calibration

5.2.3.1. Project Partner Requirement

Robot will be able to calibrate itself automatically faster than doing it manually.

5.2.3.2. Engineering Requirement

The system will complete auto calibration in under 10 mins.

5.2.3.3. Testing Method

Inspection

5.2.3.4. Verification Process

When the robot starts to calibrate, observe whether the robot will complete it without human interface and the time is under 10 mins.

Pass Condition: The auto calibration is finished within 10 mins.

5.2.3.5. Testing Evidence

We have evidence of this requirement being met on video. In the first video you will see the robot manually being calibrated to achieve the precision required for the robot to be able to print. You will see that it takes roughly two minutes for a user to calibrate the robot. In the second video you will see the robot starting ten feet away from the target object at a randomly placed offset angle and find its way to the object autonomously. The robot will calibrate itself so that it's parallel to the target object. You can notice by the time in the video it takes about 40 seconds for the to autonomously calibrate using the computer vision system.

Link to video one: https://youtu.be/9OZ6-oPxjfs

Link to video two: https://youtu.be/KomwuV73EnY

5.2.4. Distance x

5.2.4.1. Project Partner Requirement

Develop a movement system to approach a desired print surface.

5.2.4.2. Engineering Requirement

The system will move the printer to target distance from the print surface (translation in x) with error less than 30mm.

5.2.4.3. Testing Method

Inspection

5.2.4.4. Verification Process

Move the robot, and when the robot automatically stops, measure the distance in x axis between the target location and actual stopping location.

Pass Condition: When the robot stops, the distance between the target location and the actual location in x axis is less than 30mm.

5.2.4.5. Testing Evidence

We have evidence of this requirement being met on video. In the video you will see that after movement, the distance between the target location and the actual stopping location has an error less than 30mm. The perfect stop location is marked by duct tape on the floor for the wheels to line up to. Specifically looking at the X-direction.

Link to video: https://youtu.be/KomwuV73EnY

5.2.5. Distance y

5.2.5.1. Project Partner Requirement

Develop a movement system to approach a desired print surface.

5.2.5.2. Engineering Requirement

The system will align the printer parallel to the print surface (translation in y) with error less than 30mm.

5.2.5.3. Testing Method

Inspection

5.2.5.4. Verification Process

Move the robot, and when the robot automatically stops, measure the distance in y axis between the target location and actual stopping location.

Pass Condition: When the robot stops, the distance between the target location and the actual location in y axis is less than 30mm.

5.2.5.5. Testing Evidence

We have evidence of this requirement being met on video. In the video you will see that after movement, the distance between the target location and the actual stopping location has an error less than 30mm. The perfect stop location is marked by duct tape on the floor for the wheels to line up to. Specifically looking at the Y-direction.

Link to video: https://youtu.be/KomwuV73EnY

5.2.6. Operating Area

5.2.6.1. Project Partner Requirement

The Robot must be able to function wirelessly within a certain ranged.

5.2.6.2. Engineering Requirement

The system will function wirelessly at least 10 ft away from the operating host.

5.2.6.3. Testing Method

Inspection by seeing if the system will function properly with the user 10 feet away.

5.2.6.4. Verification Process

We will use a measuring tape to make sure the operator is 10 feet away from the robot and observe whether the robot is able to use the touch sensor and move the printer head when the PC is outside 10ft.

Pass Condition: All system requirements must function within a 10ft by 10ft square of space.

5.2.6.5. Testing Evidence

We have evidence of this requirement being met on video. Video one shows an operator who is 10 feet away from the robot controlling all six types of movements. The movements shown on the video are: forward, back, left, right, clockwise rotation, and counterclockwise rotation. Video two shows the entire run through of the system that proves all of the functions working together properly.

Link to video 1: https://youtu.be/F_q8e9yDIgo

Link to video 2: https://youtu.be/KomwuV73EnY

5.2.7. Signal feedback

5.2.7.1. Project Partner Requirement

Printers should alert a user when switching between printing steps.

5.2.7.2. Engineering Requirement

The system will indicate when it is changing between printing steps such that 9 out of 10 viewers can clearly state the current printing step.

5.2.7.3. Testing Method

Demonstration

5.2.7.4. Verification Process

User interface will inform the user as to the current printing step, as well as the current printer function. We should be able to see each printing step on the PC that is connected to the printer.

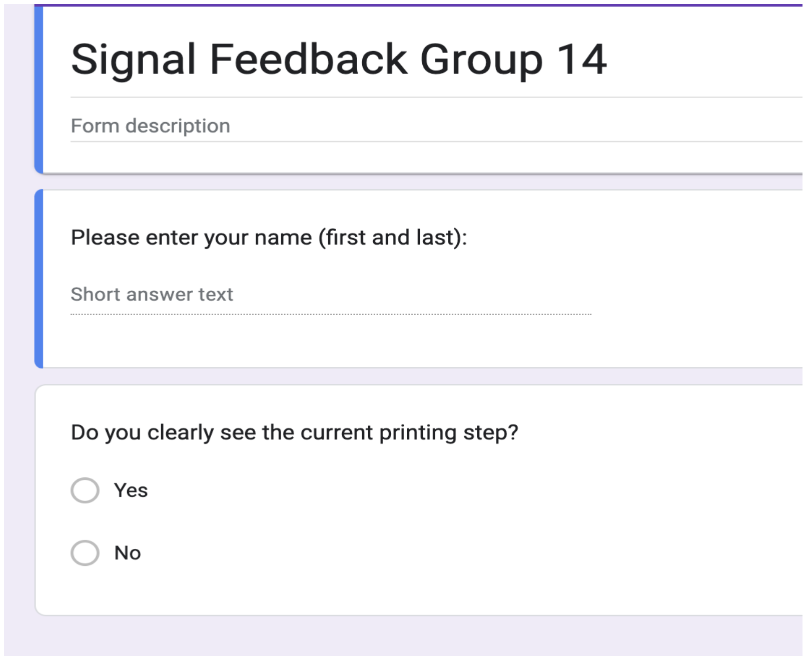
Pass Condition: The 9 out 10 viewers can know the current steps when changing between the printing steps.

5.2.7.5. Testing Evidence

Ten people have seen this requirement being met and signed that the system works. In the video you will see what each person saw, the printer changing steps as the probe moves.

Link of the video: https://youtu.be/qM_7IxinIgQ

The following two figures are about the question we used for signal feedback and the responses from 12 students.



Signal Feedback Group 14

Form description

Please enter your name (first and last):

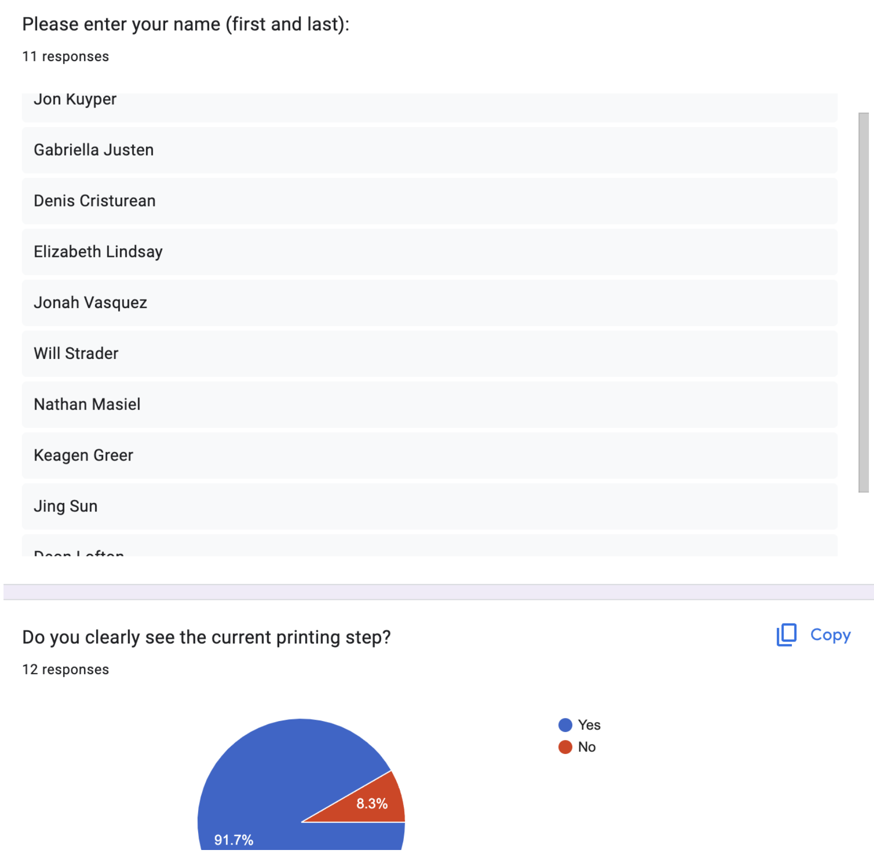
Short answer text

Do you clearly see the current printing step?

Yes

No

Fig. 5.2.7.1. Screenshot of google form question



Please enter your name (first and last):

11 responses

Jon Kuyper

Gabriella Justen

Denis Cristurean

Elizabeth Lindsay

Jonah Vasquez

Will Strader

Nathan Masiel

Keagen Greer

Jing Sun

Do you clearly see the current printing step?

Copy

12 responses

Figure. 5.2.7.2 Screenshot of responses

5.2.8. Time limitation

5.2.8.1. Project Partner Requirement

Beginning a print on a surface should be faster using the vehicle, compared to manually moving and orienting the printer.

5.2.8.2. Engineering Requirement

The system will take less than 5 mins to finish movement, and orientation prior to beginning printing.

5.2.8.3. Testing Method:

Inspection. We will use a timer to see if the robot can complete its movements to the print surface within the required time.

5.2.8.4. Verification Process

Start the time recording when the robot starts to move, and stop recording when the robot has reached the print surface at the correct orientation (facing the print surface straight on) and starts to calibrate. We will check to see if those movements were completed within five minutes.

Pass Condition: The time the robot uses is under 5mins.

5.2.8.5. Testing Evidence

Based on the length of video, you will observe that movement and orientation are finished in 5 mins. Everything our system is required to do is done within the time limit.

Link to video: https://youtu.be/KomwuV73EnY

5.3. References and File Links

5.3.1. References (IEEE)

[1] "IEEE referencing style: Images, figures and tables," *Guides*. [Online]. Available: https://guides.library.uwa.edu.au/IEEE/images_tables_figures#:~:text=Place%20the%20Figure%20title%2Fcaption,captions%20should%20be%20centre%2Djustified. [Accessed: 13-Oct-2022].

5.3.2. File Links

[1] "X,Y,Z,Autonomous, Calibration,Time,OA," *www.youtube.com*. https://youtu.be/KomwuV73EnY (accessed May 14, 2023).

[2] “Operating Area,” *www.youtube.com*. https://youtu.be/F_q8e9yDIgo (accessed May 14, 2023).
[3] “Signal Feedback,” *www.youtube.com*. https://youtu.be/qM_7IxinlgQ (accessed May 14, 2023).

5.4. Revision Table

Date	Author	Revise Content
3/14/2023	Xinrui Zhou	Downloaded the content fro student portal
3/15/2023	All members	Filled in the rest of section 5
5/10/2023	Xinrui Zhou	Modified the test process, and corrected portions of section 5
5/10/2023	Konstantin Markov	Added video links for each requirements

6. Project Closing

6.1.Future Recommendations

6.1.1. Technical recommendations

To improve the project's functionality, it is recommended to integrate the various modules more seamlessly. This can be achieved by ensuring that the communication between the Arduino, Raspberry Pi, and Duet 3 mainboard is reliable and efficient. It is also recommended to use a unified software platform that can control all the modules, making it easier to manage and troubleshoot any issues that may arise. A recommendation that would make the project much more efficient is a custom mainboard that performs the functions of all the modules contained within one circuit board.

Another recommendation would be to simplify the structure of the robot with a purpose manufactured design, rather than using t slot aluminum for the structure. This can both increase the strength of the robot’s chassis and reduce weight of the system. An additional benefit is making the robot more compact - aiding in storage and transportation. The printer parts, which includes the actual 3d printer can also be upgraded; using high end extruders, nozzles, and filaments which will result in prints of better quality. Another upgrade discussed during the design phase is the use of non-plastic materials such as metal and glass. The necessary upgrades to enable this capability will require technology not included in the scope of this project, however it would represent a significant leap in the project's capabilities.

Also, improving motors can have a significant impact. By incorporating better motors, we can achieve higher efficiency, increased power output, and improved overall reliability. Consider using motors with higher torque and reduced weight to optimize the performance of your system. Additionally, selecting motors with advanced control mechanisms, such as brushless DC motors or stepper motors, can provide better precision and smoother operation.

Another significant difficulty encountered was the strain on the system itself due to its own weight causing significant strain on the stepper motor drive system. This often caused the robot to slip or become unable to move. This also caused the computer vision system to have inaccuracies due to the pathfinding instructions not being followed accurately due to aforementioned slippages. Some recommendations to fix this would be to use a sturdier frame, lighten the robot, or use stepper motors with higher torque.

Last one enhanced the camera system to improve functionality and enhanced user experience. Consider incorporating additional cameras to capture a wider field of view or enable 3D depth sensing. Moreover, explore the integration of high-resolution cameras with improved low-light performance and image stabilization to ensure high-quality imaging in various conditions. Implementing advanced image processing algorithms can further enhance the camera system's performance by enabling features like image denoising, real-time object tracking, or intelligent scene analysis.

A significant risk the project faced was damage to fragile electrical components. Improving the durability and stability of the robot will allow the risk of damage occurring to critical components of the projects to be minimized. The stability can be improved by adding shock absorbers and stabilizers to reduce the impact of vibrations. This will also improve the quality and speed of 3D prints made by the robot.

Current simplified block diagram link:

[1] “Blank diagram | Lucidchart,” *lucid.app*. https://lucid.app/lucidchart/95d703d2-c2e9-45d4-9b18-27db1bbc9f53/edit?viewport_loc=-652%2C-64%2C2219%2C1065%2C0_0&invitationId=inv_d0def9ab-d587-4a7b-a186-2e51bbe59db0 (accessed Apr. 29, 2023).

Current Intel Realsense camera link:

[2] “Depth Camera D435,” *Intel® RealSense™ Depth and Tracking Cameras*. https://www.intelrealsense.com/depth-camera-d435/

Current protocol link:

[3] “Main Page - Firmata,” *Firmata.org*, 2010. http://firmata.org/wiki/Main_Page (accessed Dec. 09, 2019).

6.1.2. Global impact recommendations

The mobile 3D printer robot project has the potential to make a positive global impact, provided that it is appropriately designed and implemented. In order to achieve this, several recommendations should be taken into consideration.

Firstly, sustainable materials should be used in the project. The printer should be designed to use recycled plastics as feedstock for printing. This will help reduce the amount of plastic waste in the environment and mitigate the negative impact of plastic waste. In addition, the use of biodegradable materials is recommended to decrease the project's environmental impact. Additionally, renewable energy sources should be utilized to power the printer. The mobile printer robot should be designed to run on solar or wind energy, which are renewable energy sources. This will help reduce the carbon footprint of the project and promote sustainable development.

Collaborating with local communities could also help improve the global impact of the project. Local communities should be involved in the design and execution of the project to ensure that it meets their needs. Collaboration with local communities will also promote the project and increase its acceptance in the community. The project should train local communities on how to use the printer and provide technical support when needed. This will ensure that the printer is used correctly and effectively, and will also promote the project's acceptance in the community.

As we all know, 3D printing is time-consuming, so it is very necessary to save energy as much as possible. Therefore, another recommendation is energy efficiency. Implement power-saving features on our 3D printer, such as automatic sleep mode or power-off after a period of inactivity. Also, energy-efficient components, such as selecting 3D printers equipped with energy-efficient components, including power supplies and motors, to minimize power consumption during operation.

Lastly, investing in research and development is necessary to improve the project's efficiency and effectiveness. The project should prioritize research and development to identify ways to enhance the printer's performance, reduce its energy consumption, and improve its printing quality. This will ensure that the project continues to have a positive global impact and remains relevant in the future.

Current design impact document link:

[1] “Design assessment document,” *Google Docs*. [Online]. Available: https://docs.google.com/document/d/1kcCi1JGLbXWbsUNgIUETYfZCfY4cVWMEoLcLaqXSTKA/edit?usp=sharing. (accessed: 20-Apr-2023).

6.1.3. Teamwork recommendations

The mobile 3D printing robot project is a multi-team, multi-disciplinary project, therefore, good and sufficient communication is significant. Before continuing the project, it’s necessary to look over documents and ask the corresponding team for clarification.

During the project, encourage open communication. It’s important to let all members share their ideas, thoughts and concerns openly and honestly. It can be achieved by setting up regular meetings to discuss their progress and any issues they are facing. During the meeting, make sure everyone has the chance to speak and be heard. Also, determine tools for communication and file store in advance.

Also, setting clear expectations is another important point. Make sure everyone understands their roles and responsibilities, as well as the project goals and deadlines. It can be achieved by developing a project plan that outlines each member's tasks and expected deadlines, and regularly reviewing progress against this plan. Furthermore, cosider setting up a regular check-in time during the meeting.

Current Teams communication channel link:

[1] "Join conversation," *Microsoft Teams*. https://teams.microsoft.com/l/channel/19%3a-OcnmhVW_w7g1OEkctqIEjoJY1MvQ5frDWhLZRZvtjo1%40thread.tacv2/General?groupId=824ca9de-39b4-427f-aff1-524842df52ee&tenantId=ce6d05e1-3c5e-4d62-87a8-4c4a2713c113 (accessed Apr. 29, 2023).

6.2. Project Artifact Summary with Links

Github link:

[1] popoolol, "HulkVehicleControlSystem-," *GitHub*, Feb. 28, 2023. https://github.com/popoolol/HulkVehicleControlSystem- (accessed May 14, 2023).

PCB:

[1] "Ebox-Design.kicad_pcb," *Google Docs*. https://drive.google.com/file/d/1BATOPt3J1iTAl3qP7_5pXAYK5Mjy9WD8/view?usp=share_link (accessed May 15, 2023).

6.3. Presentation Materials

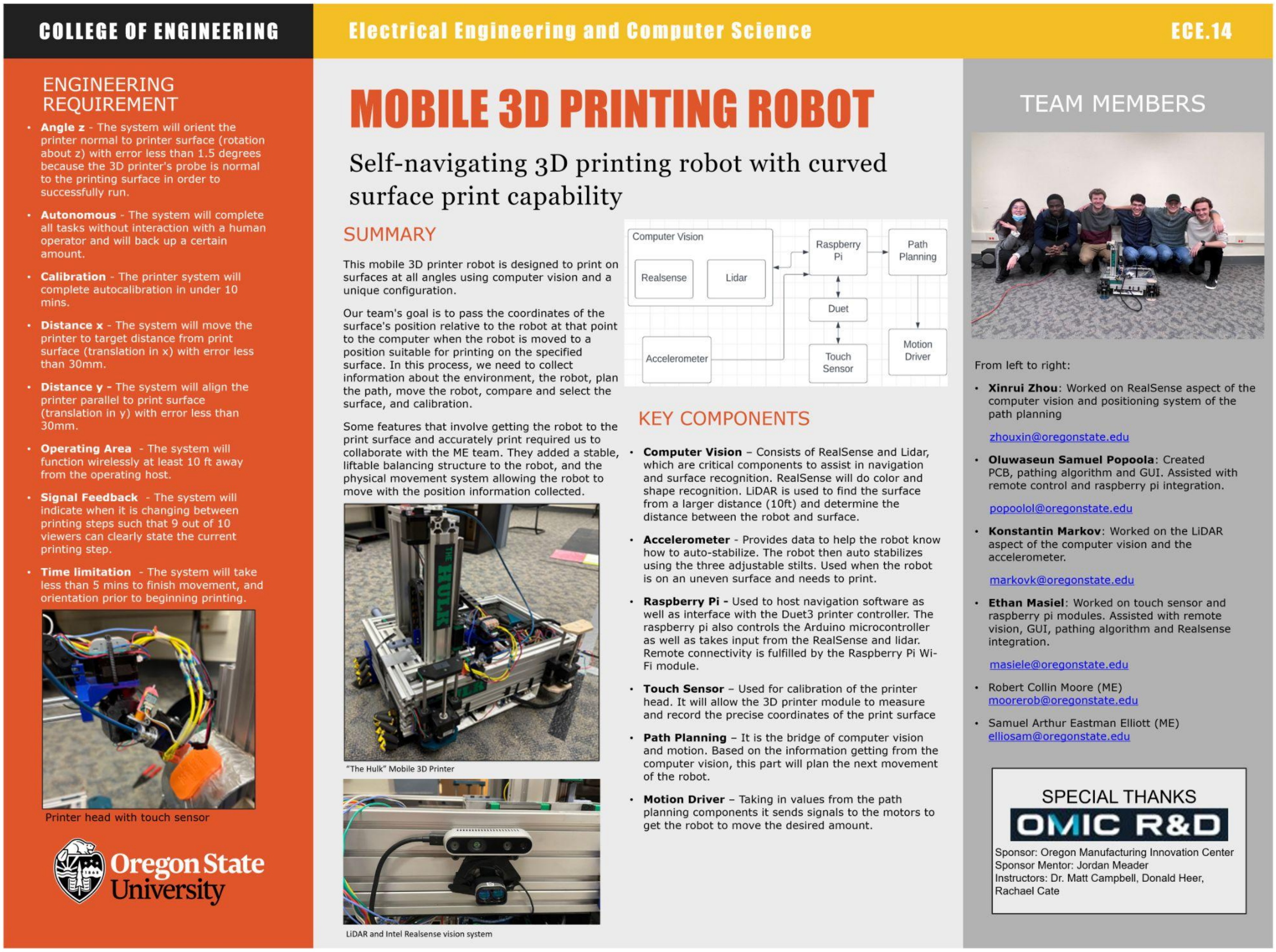


Fig. 6.3.1. Project Poster

Project Showcase link:

[1] "Showcase Project | OSU," *eecs.engineering.oregonstate.edu*. https://eecs.engineering.oregonstate.edu/project-showcase/projects/?id=xrF5IasytAx1QJbq (accessed May 14, 2023).