Critical Design Review (CDR) report

Pseudo GPS for Romi
Critical Design Review

June 7, 2024

For Charlie Refvem

Team
Emmanuel Baez – embaez@calpoly.edu
Gabriel Coria – gacoria@calpoly.edu
Owen Guinane – oguinane@calpoly.edu
Conor Schott – cjschott@calpoly.edu

College of Engineering
California Polytechnic State University, San Luis Obispo

2024

Table of Contents

Statement of Disclaimer

Since this project is a result of a class assignment, it has been graded and accepted as a fulfillment of the course requirements. Acceptance does not imply technical accuracy or reliability. Any use of information in this report is at risk to the user. These risks may include catastrophic failure of the device or infringement of patent or copyright laws. California Polytechnic State University at San Luis Obispo and its staff cannot be held liable for any use or misuse of the project.

Four Panel Chart

Table 1. Four-Panel Chart

<table>
<tr><th>Project Overview</th><th>Concept Description</th></tr>
<tr><td>

- Project Objective: Currently, the Romi robots used in the Mechatronics lab have no way of obtaining absolute orientation or location from onboard sensors. The Cal Poly ME 405 instructor, Charlie Refvem, and Mechatronics students need a way to deliver precise and real-time location data for multiple Romi robots for the development of better algorithms. Our aim is not just to improve tracking but also to enhance communication standards among multiple Romis by ensuring geolocation accuracy, precision, range, latency, and update rates.
- Documentation and Impact: The project involves comprehensive documentation, including detailed electrical schematics and code explanations.

</td><td>

The idea with our structural prototype is to prove that our system will be able to detect the Romi bots as well as that it will properly mount to a ceiling tile. In the picture above is our Raspberry Pi and Camera Module wired together ready to be used.

</td></tr>
<tr><th>Concept Justification</th><th>What's Next</th></tr>
<tr><td>

The decision to integrate camera technology into the project is based on its accuracy, providing precise locational data within ±5mm. The Raspberry Pi camera's real-time data capabilities and fast update rate enable monitoring of multiple Romi robots. Additionally, the camera's cost-effectiveness and plentiful documentation online further supports its selection for enhancing tracking capabilities.

</td><td>

This quarter we will have to keep working on our software to get the detection to work, as well as implementing a grid system to help with the position of the Romi bots. We will perform multiple pass/fail tests on the detection and position and safety tests with our system mounted. The safety test for mounting the system will consist of FEA on our ceiling tile, and a shake test to see if the system will fall.

</td></tr>
</table>

Overview

We have finalized the concept selection and are currently expanding further iterations on our concept prototype. This report aims to describe our chosen concept, our reasoning behind it, and a table of the alternate solutions considered. In addition, this report also includes a preliminary analysis, a weighted design matrix, an updated Gantt chart, a design hazard matrix, and our references. We hope to get feedback on our reasoning for the sensor choice and on the housing system.

For our system, we decided to use a Raspberry Pi with a camera to obtain absolute orientation and positional data. The basic layout is depicted in Figure 1 below.

Concept Description and Justification

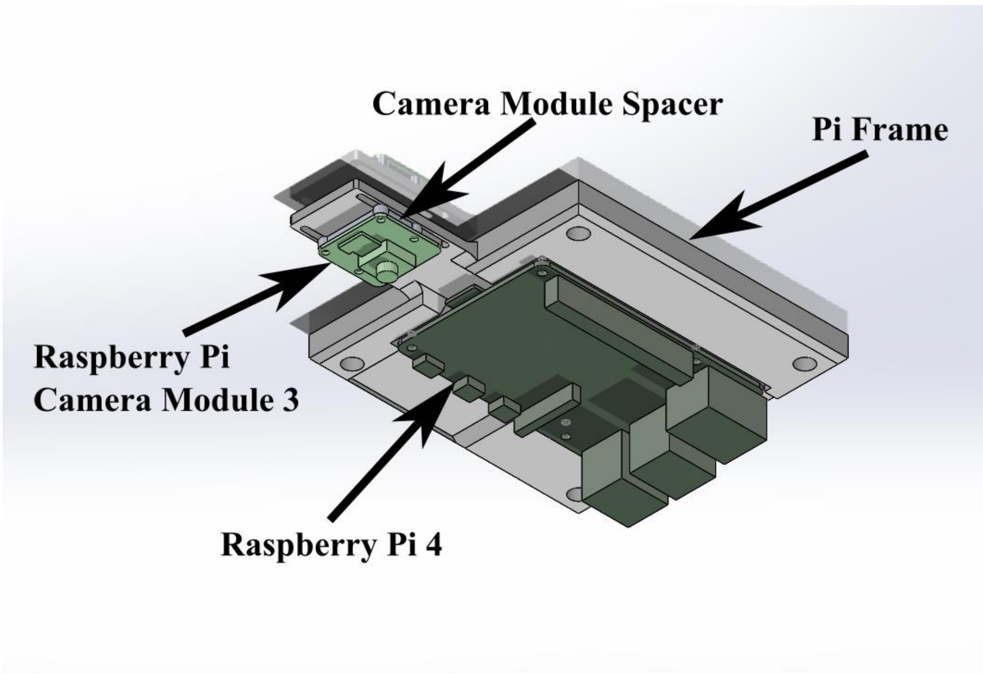


Figure 1. General layout of system

Mounted overhead, the camera captures images of the lab table, which are processed to generate a grid overlay in 2-dimensional space, facilitating precise localization and differentiation of objects. The device creates an origin point on the table, which it uses to measure Romi bots and obstacles relative to it. It constantly monitors Romi bots and obstacles within its field of view. Data such as grid representation and object positioning are handled by our microcontroller for processing.

To assist the microcontroller with processing, we will create custom libraries that will reference our specific objects, such as the Romi bots, table, and obstacles. These libraries will be created and trained through an online AI software called Teachable Machine developed by Google. Orientation of Romi bots could also be achieved if they had unique identifiers on their circular base plate, such as N, S, E, W signs or a 3D printed arrow.

We are confident that our hardware will be able to meet the project specifications. By using trigonometry and the manufacturer-given data on the camera modules, we can estimate an accuracy of 0.55 mm per pixel for the Raspberry Pi Camera Module 3 and an accuracy of 1mm per pixel with the Raspberry Pi Camera Module 3, assuming a mounting height of 6.75ft above the table. This falls well-within our target accuracy of +- 5mm for Romi tracking.

Model Training and Integration

We will use Teachable Machine to train our models by uploading images of Romi bots, obstacles, and the table in various conditions and orientations. The platform leverages transfer learning, which uses pre-trained models and fine-tunes them with our specific dataset. This approach ensures quick and efficient model training, enabling real-time predictions on new data. Once trained, models can be downloaded in Keras (.h5) format. These models will be loaded into our Python environment using the Keras library and integrated with OpenCV for various computer vision tasks. By combining Keras and OpenCV, we can utilize the trained model for real-time image or video analysis and processing. This data package will be used as our library for object detection.

Grid Representation and Object Coordinates

In parallel, we will use OpenCV for color detection, grid creation, and coordinate calculation. OpenCV provides robust tools for image processing and computer vision tasks. To create the grid representation, we will use OpenCV functions to draw a grid overlay on the table. The grid cells will help in precisely localizing and differentiating objects. For object coordinates, we will use OpenCV's contour detection to find the objects and calculate their centroids, translating these into coordinates relative to the grid's origin.

Software and Algorithms

The software integrates custom-trained models and OpenCV techniques to detect Romi bots, obstacles, and the table. Initially, we will import essential components such as the COCO dataset's class names and configure our SSD MobileNet V3 model. However, we will adapt this pre-trained model by retraining it with our specific dataset of Romi bots and obstacles to enhance accuracy. For optimal performance, we will set specific parameters, such as a confidence threshold of 0.5 for object detection to filter out low-confidence predictions, and non-max suppression (NMS) with an IoU threshold of 0.4 to eliminate redundant overlapping boxes. During operation, the software will continuously capture frames from the overhead camera feed and perform object detection using our retrained SSD MobileNet V3 model. Detected objects will be identified based on their confidence scores and class labels, with bounding boxes drawn around them for visualization.

Grid and Coordinate System

Simultaneously, the software will establish a grid overlay on the detected table and define an origin point, facilitating precise localization and measurement of objects within the environment. This will involve using OpenCV to draw the grid and calculate object centroids. Coordinates of Romi bots and obstacles relative to the table's origin will then be transmitted to the microcontroller for further processing and monitoring. The microcontroller will use these coordinates to calculate movement instructions for the Romis, avoiding obstacles and navigating the environment efficiently. Through the integration of object detection, grid generation, and coordinate calculation functionalities, the software will ensure accurate detection and monitoring of Romis and obstacles, enabling real-time data processing and reliable obstacle avoidance.

Testing and Validation

To test our system, we will conduct a series of trials under varying conditions to ensure robustness and accuracy. We will start with controlled environments where Romis and obstacles are placed at known locations to verify the precision of our object detection and coordinate calculation algorithms. Subsequently, we will introduce dynamic scenarios with moving Romis and varying obstacle placements to evaluate the real-time tracking and obstacle avoidance capabilities. Data from these tests will be collected and analyzed to fine-tune our models and algorithms for optimal performance, meaning the system will consistently deliver high accuracy and robust obstacle avoidance within a precision tolerance of ±5mm. We will also perform stress tests to evaluate the system's performance under high processing loads and ensure it meets our precision requirements. For software implementation, we will use Python with libraries such as OpenCV for image processing, TensorFlow/Keras for loading and running the trained models, and ESP-IDF for programming the ESP32 microcontroller. This combination will allow us to achieve seamless integration of computer vision and control systems, ensuring robust and reliable performance of our Romi tracking and obstacle avoidance solution.

Safety

To ensure the safe operation of our system, several main concerns need to be addressed. The largest safety concern with our system comes with mounting. Because the system is roof-mounted, we need to design the system to be mounted securely and minimize the chance of falling out, due either to structural failure of mounting parts or by not fitting properly within a drop-down roof tile frame. To ensure this does not occur, we plan on performing a test involving shaking the ceiling tile with the hardware attached and monitor the behavior of the mounting hardware and tile. Minor safety considerations include adding fillets and chamfers to 3-D printed parts to eliminate sharp corners and edges which can cut the user when handling.

Justification Table

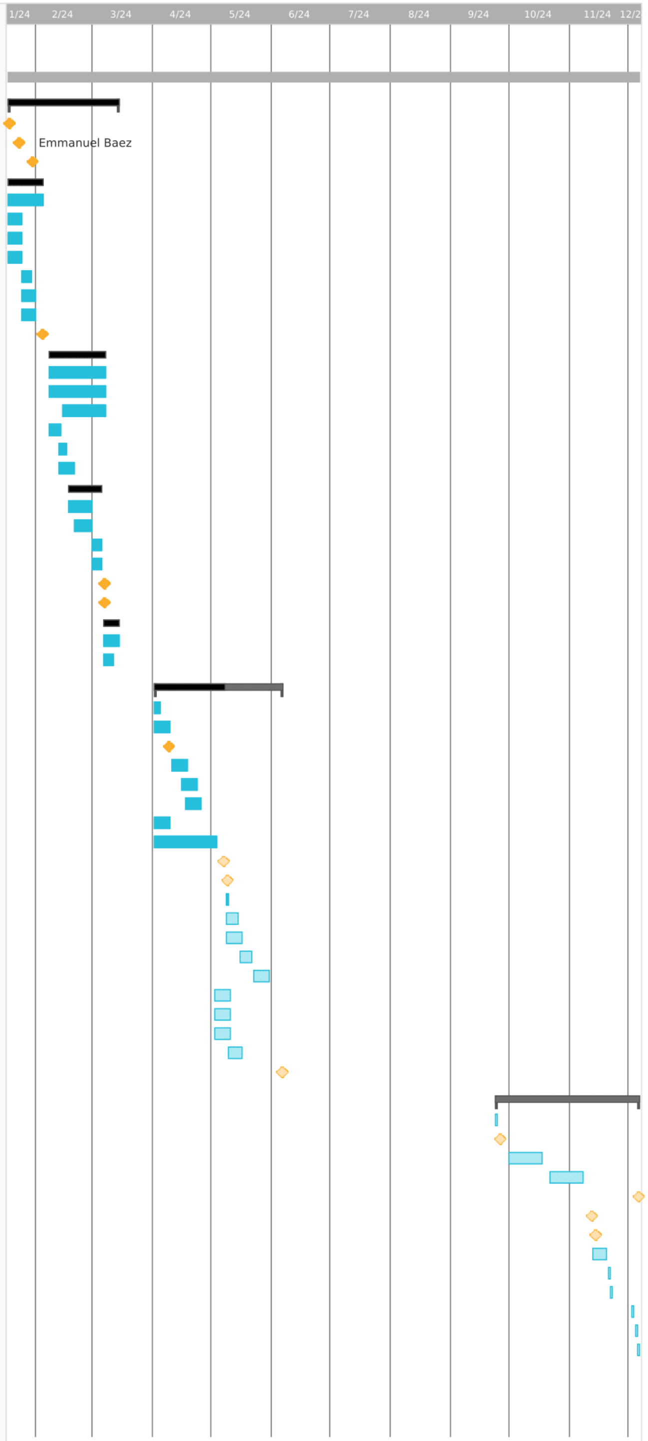
Table 2. Justification Table

Spec #	Specification Title	Specification Description	Target (units)	Justification
1	Total System Length	The system would prefer to be able to fit onto a Romi.	Romi Size	Currently, we plan to attach an ESP32 module to the Romi bots for communication. The size of the module is 18 x 31.4 x 3.3mm. Our sponsor has approved.
2	Total System Weight	The Romi will also have other sensors attached to it, so we would not want to encumber it further.	2 lbs.	Currently, we have plans to attach an ESP32 module to the Romi bots for Communication, and our sponsor said it's okay.
3	Field of View	The system should track a Romi placed on the classroom table.	96x48 in.	The Raspberry Pi Camera 3 modules have a wide enough field of view to capture the table in its entirety, depending on the mounting distance.
5	Data Transfer	The system should be able to transfer data at a high rate.	6 Hz	The Raspberry Pi 4 that we purchased has the specs to transfer data at 6 Hz.
6	Number of Romi Bots Being Detected	The system should be able to track multiple Romi bots.	6 bots	Our tracking camera algorithm combined with the camera positioning will be able to track 6 Romi bots.
7	Tracking Resolution	The system should be able to track the position of Romi.	+/- 5 mm	The Raspberry Pi Camera 3 modules will have enough resolution to track Romi position, it has a resolution of 0.55 mm/pixels (Table 7).
8	Production Cost	The budget is $500 (could be higher).	$500	We have an Excel (Table 6) that we put in every purchase to make sure we are within budget and notify our sponsor when we need to make a purchase.

Appendix

Updated Gantt Chart

GPS for Romi	**0h**	**71%**
Winter Quarter 2024	**0h**	**100%**
Create team contract	0	100%
Send introductory sponsor email	0	100%
Meet with sponsor	0	100%
Scope of Work (SOW)	**0h**	**100%**
Write Scope of Work (SOW) Report	0	100%
Research current solutions	0	100%
Conduct patent research	0	100%
Conduct technical issue research	0	100%
Create Gantt Chart	0	100%
Create QFD House of Quality	0	100%
Create a Functional Decomposition	0	100%
Submit final SOW report	0	100%
Preliminary Design Review (PDR)	**0h**	**100%**
Write Preliminary Design Review (...	0	100%
Create PDR presentation	0	100%
Address Scope of Work Issues	0	100%
Brainstorm ideas	0	100%
Build Ideation Models	0	100%
Create a Pugh matrix	0	100%
Concept Prototype	**0h**	**100%**
Create Prototype CAD Models	0	100%
Build Concept Prototype	0	100%
Research parts functionality	0	100%
Rank researched parts	0	100%
Present Preliminary Design Review	0	100%
Submit Preliminary Design Review	0	100%
Safety, FMEA, and Design Analys...	**0h**	**100%**
Perform Design Analysis	0	100%
Develop a Failure Mode Effects Ana...	0	100%
Spring Quarter 2024	**0h**	**55%**
Perform Design Analysis	0	100%
Select Parts for prototype	0	100%
Write Interim Design Review	0	100%
Create Design Verification Plan / Dra...	0	100%
Begin Building Structural Prototype	0	100%
Complete Purchasing Plan	0	100%
Review Drawing & Manufacturing Pla...	0	100%
Write Critical Design Review (CDR)	0	100%
Present CDR in Lab	0	0%
Submit CDR & Present to Sponsor	0	0%
Order materials for prototype	0	100%
Perform Risk Assessment	0	0%
Perform Safety Review	0	0%
Experimental Design	0	0%
Discuss Ethics & Test Procedures	0	0%
Start building Structural Prototype	0	0%
Finalize Structural Prototype	0	0%
Finalize Part Selection	0	0%
Revise Critical Design Review	0	0%
Prepare Manufacturing & Test Update	0	0%
Fall Quarter 2024	**0h**	**0%**
Hold Ethics Debates	0	0%
Take Senior Exam / Survey	0	0%
Finalize Verification Prototype	0	0%
Perform Verification Prototype Testing...	0	0%
Verification Prototype Sign off	0	0%
Design Verification Project Report Si...	0	0%
Peer Review Expo Posters	0	0%
Finish FDR Report/Print Poster	0	0%
Edit FDR/Prep for Expo	0	0%
Present at Senior Project EXPO	0	0%
Clean and clear workspaces	0	0%
Complete Checklist	0	0%
Submit FDR to Sponsor (5pm)	0	0%



Design Hazard Checklist

Table 4. Design Hazard Checklist Table

Y	N	
	N	1. Will any part of the design create hazardous revolving, reciprocating, running, shearing, punching, pressing, squeezing, drawing, cutting, rolling, mixing or similar action, including pinch points and sheer points?
	N	2. Can any part of the design undergo high acceleration/deceleration?
	N	3. Will the system have any large moving masses or large forces?
	N	4. Will the system produce a projectile?
Y		5. Would it be possible for the system to fall under gravity creating injury?
Y		6. Will a user be exposed to overhanging weights as part of the design?
	N	7. Will the system have any sharp edges?
	N	8. Will any part of the electrical systems not be grounded?
	N	9. Will there be any large batteries or electrical voltage in the system above 40 V?
	N	10. Will there be any stored energy in the system such as batteries, flywheels, hanging weights or pressurized fluids?
	N	11. Will there be any explosive or flammable liquids, gases, or dust fuel as part of the system?
	N	12. Will the user of the design be required to exert any abnormal effort or physical posture during the use of the design?
	N	13. Will there be any materials known to be hazardous to humans involved in either the design or the manufacturing of the design?
	N	14. Can the system generate high levels of noise?
	N	15. Will the device/system be exposed to extreme environmental conditions such as fog, humidity, cold, high temperatures, etc?
	N	16. Is it possible for the system to be used in an unsafe manner?
	N	17. Will there be any other potential hazards not listed above? If yes, please explain on reverse.

Manufacturing Plan

Table 5. Manufacturing Plan table

Sub system	Component	Purchase (P) Modify (M) Build (B)	Raw Materials Needed to make/modify the part (only M & B)	Where/how procured?	Equipment and Operations anticipate using to make the component	Key limitations of this operation places on any parts made from it
Ceiling Mount	Ceiling Tile	M	Ceiling Tile	Provided by Wood Workshop at Cal Poly for use in Lab 192-118	11/16" Drill Bit, Cordless Drill to make holes that the threaded inserts will screw into.	n/a
	1/4"-20 Zinc Plated Insert Nut	P	n/a	Purchased at Home Depot	n/a	n/a
Housing	Main Housing	B	PLA Plastic	3D printed	3D Printer to create the housing, M2.5 x .45, 1/4"-20 tap drill bit and a tap s to create internal threads on the large outer holes and small inner holes. Deburring may be necessary after tapping.	Small size may prevent M2.5 holes from printing properly. Additional drilling may be required
	1/4"-20 1" Machine Screws	P	n/a	Purchased at Home Depot	n/a	n/a
	M2.5 10mm Screws	P	n/a	Purchased from Amazon	n/a	n.a
	M2 10mm screws	P	n/a	Purchased from Amazon	n/a	n/a
	M2.5 Threaded Inserts	P	n/a	Purchased from Amazon	n/a	n/a
	1/8" Mushroom-Head Toggle Bolt	P	n/a	Part Provided by Sponsor	n/a	n/a
Microcontroller	Raspberry Pi 4 Model B	P	n/a	part already provided by sponsor (Adafruit)	n/a	n/a
	Micro HDMI Cable	P	n/a	part already provided by sponsor (Adafruit)	n/a	n/a
	SD card	P	n/a	part already provided by sponsor (Adafruit)	n/a	n/a
	Heat Sink	P	n/a	part already provided by sponsor (Adafruit)	n/a	n/a
	Power Supply	P	n/a	part already provided by sponsor (Adafruit)	n/a	n/a
Camera	Raspberry Pi Camera Module 3	P	n/a	part already provided by sponsor	n/a	n/a
	Camera Spacer	B	PLA Plastic	3D printed	3D Printer to create the part	Small Size may prevent M2 holes from printing to size. Screws may have to be threaded through.

Manufacturing Steps

Ceiling Tile Assembly

<u>Ceiling Tile</u> (110)

11/16" Drill Bit with Cordless Drill

1. Drill four holes that are 11/16" in diameter on the top (floor-facing) side of the ceiling tile to allow the threaded inserts to screw in.
2. Screw 1/4"-20 Zinc Plated inserts into drilled holes. These will connect the main housing to the ceiling tile. ***[Zinc inserts could not support force. Replaced with Mushroom Toggle Bolts per recommendation from tech support (safety review)]***

Housing Assembly

<u>Main Housing</u> (210)

3D Printer

1. Using the provided file "MainHousing.STL", use a 3D printer to print the housing. Refer to printer documentation for instructions on how to print an STL file.

1/4"-20 Tap drill bit and Tap

2. On the larger outer 4 holes and center hole, tap these holes all the way through. These holes will secure the housing to the ceiling tile. ***[Removed due to over-constraint.]***

M2.5 x 45 Tap drill bit and Tap

3. On the smaller inner 4 holes, tap these holes all the way through. These will be used to secure the Raspberry Pi 4 to the housing. ***[Removed in Favor of using heat-set M2.5 Threaded inserts]***

Soldering Iron

4. Press fit the M2.5 Threaded inserts (280) into the inner 3.5mm Holes until they are snug. Heat the soldering iron to 300 degrees Celsius. Use the soldering iron to gently push the threaded inserts in until they are flush with the plastic surface. ***[Added instead of internal threading to increase lifespan per recommendation from manufacturing TAs]***

Camera Assembly

<u>Camera Spacer</u> (2320)

3D Printer

1. Using the provided file "CamSpacer.STL", use a 3D printer to print the spacer. Refer to printer documentation for instructions on how to print an STL file.

Design Verification Plan

Table 3. Design Verification Plan Table

Planned Test	What it will test (user need or specific spec, whichever is most applicable for your project and sponsor)	Required resources
Detection Test	Our system needs to be able to detect what a Romi bot is. We need to test whether the system can detect the Romi bots throughout the duration of the lab (at least 3 hours). This test should be pushed further by adding more Romi bots and making sure all are detected and known individually.	Raspberry Pi, camera, computer
Ceiling Mount Test	Our sponsor suggests we attach our system to the ceiling. So, we need to test to make sure the system doesn't fall. This should include a shake test for our new tile with system as well as monitoring it for the lab period to see if it would fall over time.	Ceiling tile, Raspberry Pi, Housing, Camera
Rate of Transfer test	Our sponsor wants to be able to use the data right away from our system. So, if our system detects an object in front of the Romi bot, it should transfer that data relatively quickly. So, a data speed test will be needed.	Raspberry Pi, Computer, Romi bot, ESP32
Fitment and Weight test	Our sponsor wants the attachments to not encumber the Romi bots, not intervene with possible sensor other mechatronic students will use. A test for how much all the components weight will be conducted as well as several tests runs on The Romi bots to see if performance is hindered.	ESP32 Module, Romi Robots
Positional Resolution test	Our sponsor would like the system to locate the Romi within +-5mm accuracy. A test will be conducted where the system will determine the position of the Romi center from the specified origin point for six different locations. The X and Y positions will then be evaluated to determine if it falls within specification.	Romi Robot, Calipers, digital tape measure, computer, mechatronics lab
Orientation test	Our sponsor would like the system to determine the orientation of the Romi. A test will be conducted where the system will observe and determine the heading of the Romi as it is rotated 360 degrees in 5-degree increments. The goal of the heading test is to ensure a Romi can be tracked within 1 degree of true heading.	Romi Robot, Protractor, mechatronics lab.

Budget

Table 6. Budget Chart

Romi Budget Plan	
Part	Cost ($)
¼"-20 Threaded Inserts	3.87
1/4"-20 1" Machine Screws	1.38
M2.5 10mm Machine Screws	10.86
M2 10mm Machine Screws	9.99
M2 Nuts	[-]
M2.5 Threaded Inserts	8.99
1/8" Mushroom-Head Toggle Bolt Anchors	8.48
Raspberry Pi 4 Model B - 8GB Ram	150.00
SD/MircoSD - 16GB class 10	19.95
Raspberry Pi Camera Module 3 Standard - 12MP Autofocus	25.00
Micro HDMI to HDMI Cable	8.95
Power Supply 5.1V 3A w/ USB C	7.95
Parts Total	255.42
Initial Budget	500
Current Budget	255.42

Indented Bill of Materials

Psuedo GPS
Indented Bill of Material (iBOM)

Assy Level	Part Number	Descriptive Part Name	Qty	Part Cost	Source	URL	More Info
		Lvl0 Lvl1 Lvl2 Lvl3 Lvl4					
0	1	Final Assy					
1	10	Ceiling Tile Assy					
2	110	Ceiling Tile	1		Cal Poly Wood Shop		Ask for tiles for lab 192-118
2	120	Threaded Inserts	4	$3.87	Home Depot	https://www.homede	
2	20	Housing Assy					
3	210	Main Housing	1		3D Printed		
3	220	Microcontroller Assy					
4	2210	Raspberry Pi 4 Model B - 8GB Ram	1	$150.00	Adafruit	https://www.adafruit	
4	2220	SD/MircoSD - 16GB class 10	1	$19.95	Adafruit	https://www.adafruit	
3	230	Camera Assy					
4	2310	Raspberry Pi Camera Module 3 Sta	1	$25.00	Adafruit	https://www.adafruit	
4	2320	Camera Spacer	1		3D Printed		
3	240	1/4"-20 1" Machine Screws	4	$1.38	Home Depot	https://www.homede	
3	250	M2.5 10mm Machine Screws	4	$10.86	Amazon	https://www.amazon	Sold in 840 Pc Kit
3	260	M2 10mm Machine Screws	4	$9.99	Amazon	https://www.amazon	Sold in 660 Pc Kit
3	270	M2 Nuts	4	[-]	Amazon		In M2 Screw kit
3	280	*M2.5 Threaded Inserts*	4				
3	*290*	*1/8" Mushroom-Head Toggle Bolt Ancho*	*4*	*$8.48*	*Cal Poly Techn*	*https://www.homede*	*Obtained through tech support*
1	3	Cables					
2	30	Micro HDMI to HDMI Cable	1	$8.95	Adafruit	https://www.adafruit	
2	31	Power Supply 5.1V 3A w/ USB C	1	$7.95	Adafruit	https://www.adafruit	
	Total Parts		**36**	**$246.43**			`

Figure 2. Indented Bill of Materials (iBOM) Updated 6/6/2024

Supporting Evidence

We can confirm that our field of view and tracking resolution will meet our specifications by performing an analysis of the camera's field of view. For a camera, we have two types of viewing distances that define our viewing area, horizontal distance Dhor and vertical distance Dvert. In our desired configuration where the camera system is mounted to the roof and facing down, we can calculate the viewing area using trigonometry.

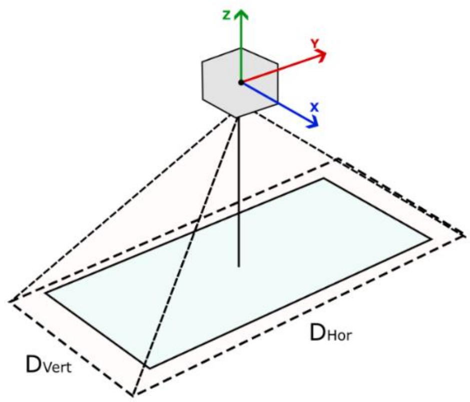


Figure 3. 3D Sketch of camera setup.

Camera manufacturers specify the horizontal and vertical angles for the field of view for any given lens which we'll define as HFOV for the horizontal field of view and VFOV for the vertical field of view. Let's take the case of the horizontal field of view. Assuming a perfectly centered camera, we can model it as a symmetrical triangle of angle HFOV, height Hmount, and distance Dhor. Hmount will be the height at which the system is mounted above the table.

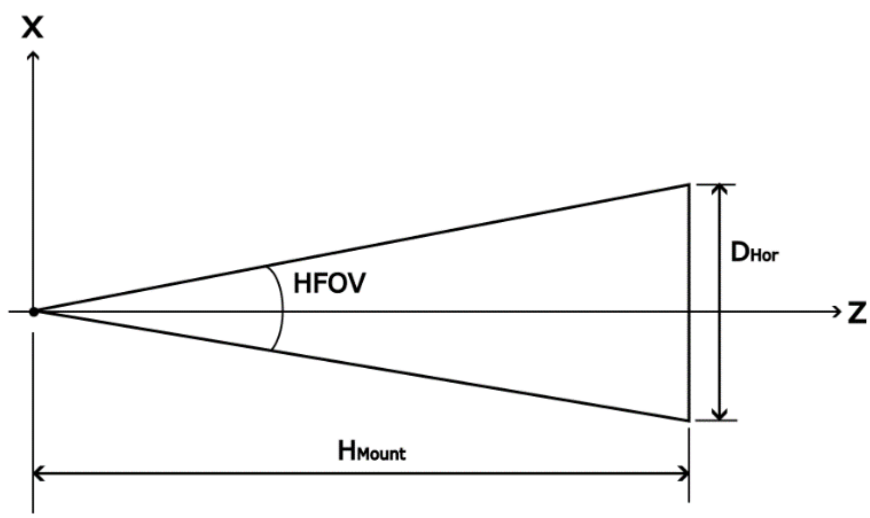


Figure 4. Viewing triangle of camera for horizontal distance

Since the triangle is isosceles, we can split it into two identical right triangles with half the angle and horizontal distance.

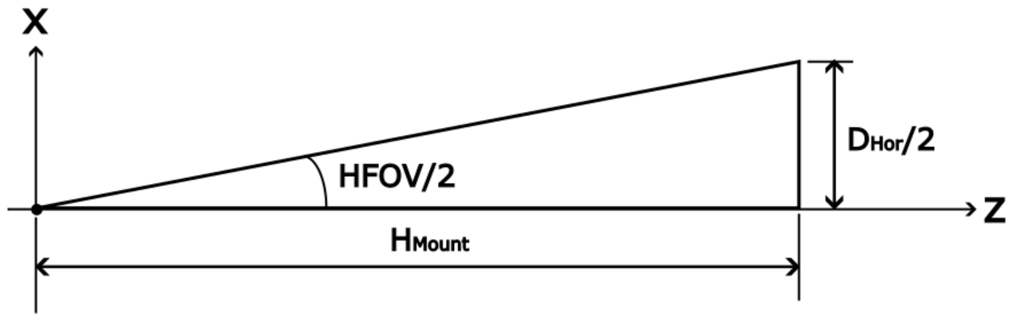


Figure 5. Half of viewing triangle. Trigonometry can now be used to solve for distance.

Using trigonometry, we obtain

$$\tan\left(\frac{HFOV}{2}\right) = \frac{(D_{Hor}/2)}{(H_{mount})}$$

Rearranging the equation,

$$D_{Hor} = 2H_{mount}\tan\left(\frac{HFOV}{2}\right)$$

Solving for the vertical distance D_{Vert} follows a similar process, so similar equation can be used.

$$D_{Vert} = 2H_{mount}\tan\left(\frac{VFOV}{2}\right)$$

To track the whole table, the camera must have a vertical viewing distance of at least 4 feet and a horizontal viewing distance of 8 feet. Using these two distances, we can calculate the viewing area of the camera.

$$A_{View} = D_{Hor} * D_{Vert}$$

Finally, by dividing the viewing area by the pixel count, we can determine the area of each pixel in the image, which will be our tracking resolution area.

$$A_{pix} = \frac{A_{View}}{\#\, of Pixels}$$

A pixel is a square, so using the area of the pixel, we can determine the pixel width, which will be our tracking resolution.

$$W_{pix} = \sqrt{A_{pix}}$$

Now that we have our basic equation, we can substitute our values to determine if our cameras will meet specifications. Measurements taken in the lab give us a maximum mounting height of 7 ft above the table. Accounting for the fact that our camera will not sit flush with the drop-down ceiling, we can assume a reasonable mounting distance of 6.25 ft above the table. To streamline the process of calculating viewing areas, we used an Excel worksheet to run the calculations

Table 7. Camera Calculation

Camera Lens	Resolution	HFOV	VFOV	Dhor	Dvert	Aview		Apix	Wpix
	[pixels]	[deg]	[deg]	[ft]	[ft]	[ft^2]	[mm^2]	[mm^2/pixel]	[mm/pixel
Raspberry Pi Camera Module 3	1.19E+07	66	42	8.12	4.80	39.0	3.62E+06	0.30	0.55
Raspberry Pi Camera Module 3 Wide	1.19E+07	102	67	15.44	8.27	127.7	1.19E+07	1.00	1.00

From our analysis, it appears that both the Raspberry Pi camera module 3 and Raspberry Pi Camera Module 3 wide will both meet our design specifications.

The ESP32 effectively communicates with our Pi4's Wi-Fi module, ensuring seamless connectivity. Its integration with the Romi Nucleo board, equipped with GPIO pins, ADC, and PWM, facilitates data transmission to and from the Pi4. With low power consumption, especially when powered by the Nucleo, it enables smooth continuous tracking and monitoring tasks. Costing between $10 to $15, the ESP32 offers a cost-effective solution. Its compact dimensions (55mm x 27.9mm x 25mm) and lightweight construction (less than 1 ounce) ensure minimal impact on the Romi's weight distribution.