

MOTORIZED TRACK LIGHTING

By

Pratik Ainapure

Adithya Bellary

Ethan Hickman

Final Report for ECE 445, Senior Design, Spring 2019

1 May 2019

Project No. 80

Abstract

This report describes a motorized, modular track lighting system with multiple independent light sources that can illuminate any area below and move back and forth along the track. The track and the individual light carts, which control the direction of the light sources, can be controlled via Bluetooth using a phone application. The reason for creating a track lighting system is the fact that existing systems are static and require manual adjustment, which creates possible safety hazards. This solution allows users to configure custom lighting for their specific use cases from the convenience of a mobile application. The way the system is designed also allows users to add additional light carts if their lighting needs change in the future.

Table of Contents

1. Introduction.....	3
1.1 Objective.....	3
1.2 High-Level Requirements.....	3
2. Design.....	4
2.1 Introduction.....	4
2.2 Traversal System.....	5
2.2.1 Track Physical Design.....	5
2.2.2 Track Electrical Design.....	7
2.3 Power System	9
2.4 Application Interface.....	11
3. Verification.....	14
3.1 Traversal System Verification.....	14
3.2 Pan-Tilt System Verification.....	15
3.3 Power System Verification.....	16
3.4 Application Interface Verification.....	17
4. Cost Analysis.....	18
4.1 Materials Cost.....	18
4.2 Labor Cost:.....	20
5. Conclusion.....	21
5.1 Accomplishments.....	21
5.2 Uncertainties.....	21
5.3 Ethical Considerations.....	21
5.3.1 Ethics.....	21
5.3.2 Safety.....	21
5.4 Future Work.....	22
References.....	23
Appendix A Requirement and Verification Tables.....	25
Appendix B: Figures and Reference Tables.....	36

1. Introduction

1.1 Objective

Many current lighting solutions for home, museum, and studio use are static systems. While static systems are useful in fixed environments, if the environment they are meant for use in is rearranged regularly, manually adjusting the lighting in the space is an annoying and potentially dangerous process. Light fixtures are often hung from high ceilings and may require hand tools to adjust. When modifying current track lighting configurations manually, users risk burns, electric shock, and high falls.

Track lighting is a popular power distribution method to avoid running wires across ceilings. They consist of concealed wires at mains voltage running through a stiff insulating case called “track”. Track lighting “heads” are designed to clip into place in a channel on the bottom of the track. The heads consist of a housing, socket, and contacts to house and power a lightbulb. Though there are thousands of stationary track light heads available for purchase [11], few, if any, are designed to move along the track. There are some lighting systems that are motorized to move along linear track, but they are designed for slow periodic motion for greenhouse applications [13].

The objective of our project is to create a remote controlled ceiling-mountable track lighting system. Each head (which we call “carts,” because they roll along the track in our system) is able to move left and right along shared linear track. Additionally, each cart is able to manipulate the direction that the light source points in by changing the pan and tilt angles, analogous to longitude and latitude respectively, of the cart’s light socket.

The carts in our system are controlled through a mobile application developed for Android in React Native. Our system makes adjusting the lighting configuration in a space easier by giving the user the ability to change the focus of their light sources.

1.2 High-Level Requirements

In order to sufficiently accomplish our design objective, the following three requirements were chosen to ensure that the alternative product being developed would solve the given problem.

- The carts must be able to move independently from one another along the track and accelerate to a speed of 3 ± 0.5 inches per second.
- The directional light source must be able to pan 360 degrees around a vertical axis and tilt from 90 to 180 degrees from vertical upright. This range constitutes a hemispherical surface below the track, which the light source will point normal to.
- The control module must be able to turn on and off individual light bulbs when there are multiple carts on a track.

2. Design

2.1 Introduction

Our final design is comprised of three major subsystems: application interface, traversal system, and power system. Together, these systems provide the functionality of our proposed design.

Referencing the block diagram below, a visualization of all the subsystems as well as their internal components can be seen.

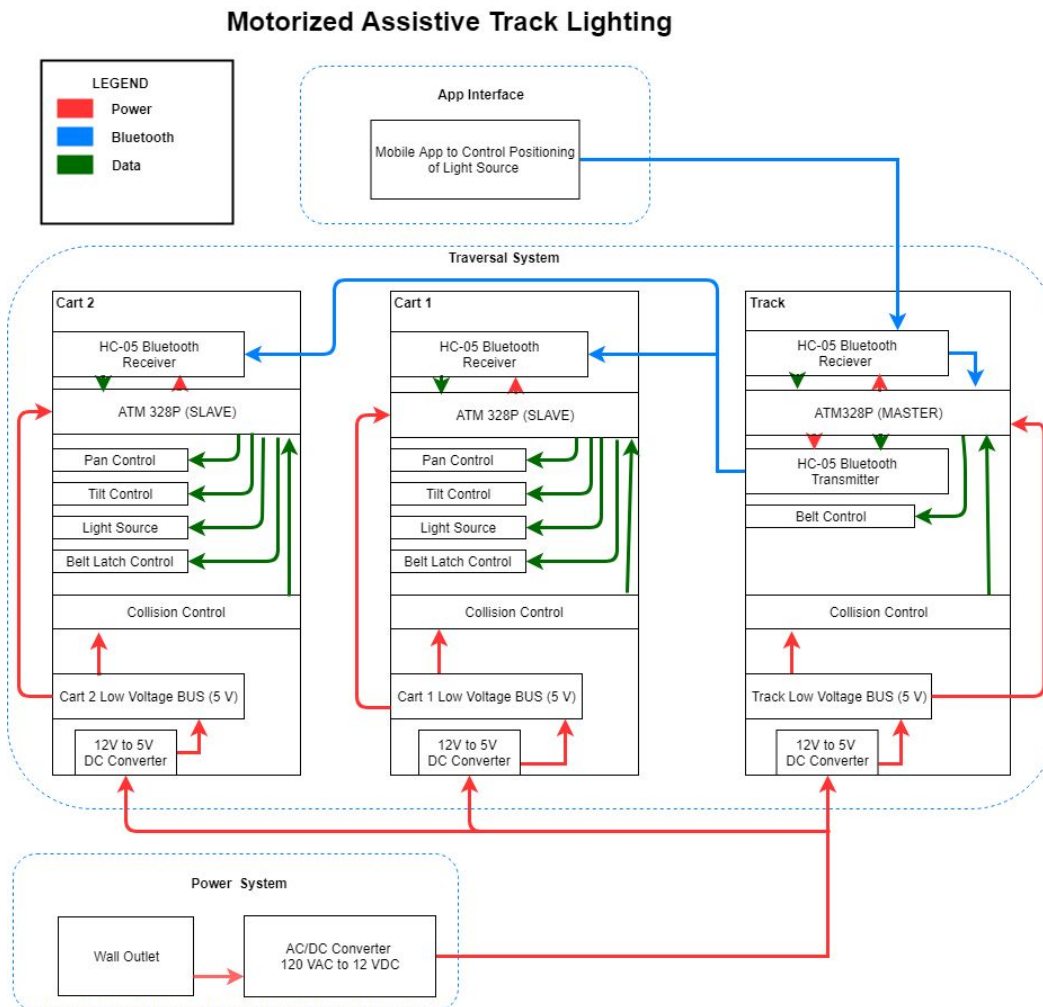


Figure 1. Overall system block diagram

The first step in our system execution cycle is the application interface. This module represents a bluetooth enabled mobile application that communicates directly with our track. Data is transferred from the connected device to our track via a Bluetooth serial line and then decoded by our processor on the traversal system.

The traversal subsystem is composed of the two slave carts, as well as the master node which receives the instructions from the mobile application and then decodes and transmits the relevant instructions to each cart. On the master node side, an ATmega328p processor handles all of the processing of the user sent instructions as well as controls the motor belt that allows the user to control the lateral position of each cart. In addition, collision controllers are mounted on each end of the track to allow for the master node to prevent a cart from colliding with the edge of a track as a result of bad input from the user.

The power system encapsulates a 120 V ac supply that is stepped down to 12 V dc that is driven every submodule of the traversal system. Using local 12 V dc to 5 V dc step down converters, low voltage devices are able to receive sufficient power to operate as desired. Each module is powered in a serial chain to ensure consistent power is being delivered to all components.

2.2 Traversal System

2.2.1 Track Physical Design

To begin our design process, we determined the core features that make our product a viable replacement for standard track lighting systems. We narrowed the key functionality down to controlling power to the light source, moving the carts laterally and independently across the track, and adjusting the pan and tilt angles of the light source.

With specification for these functionalities laid out in our high level requirements, we began investigating viable options to deliver these functionalities in our track system.

We considered several options to deliver lateral motion. We determined rack and pinion systems would deliver the precision characteristics we desired, but would require a gearbox to operate, and become expensive as track length increases. Ballscrew systems would also deliver precise motion, but would require extra rails, would drive too slowly, and make independent motion difficult or impossible to design for assuming a shared screw. We briefly considered electrical solutions like linear actuators, but were dissuaded by high cost and limited flexibility. We decided that a belt drive system with a single stepper motor would be the best solution to provide the functionality required for our prototype. It was relatively simple to implement, and the cost of a belt drive system scales to whatever the user's specific needs are. One drawback to the belt drive system that we implemented is that the carts don't have a brake other than latching to the belt. If a cart is using the latch mechanism as a brake, other carts can't use it to drive at the same time without moving both carts.

The carts access the belt drive deflecting the belt with a hobby servo horn that the belt runs through. Because the carts interact with the drive system by deflecting the belt, the single drive system also allows the carts to move independently of one another. Referencing figure B.2 in appendix B, the belt deflection mechanism is . A latch servo is placed on each cart that is engaged when the user wishes to move that specific cart. The latching and unlatching functionality is performed using a servo horn on each latching servo that physically deflects the belt at a 45 degree angle. Once engaged, the belt drive motor guides the cart to the desired position and then unlatches the servo.

Two standard size Hitec hobby servos are used per cart to provide the range of motion for panning and tilting the light source. One servo controls each degree of freedom. To obtain the 360 degree range for panning, we investigated and experimented with several approaches. Most hobby servos have feedback control and significant torque over only a 180 degree range, but they are modifiable to rotate continuously at the cost of losing the internal position feedback. There are some highly specialized servos for model truck winches and model sailboat masts that can make multiple rotations, but trade single degree precision for the increased rotation range.

After finding the SPT-200 direct drive pan & tilt kit [4], we tried to implement the pan functionality with a 4.3 kg.cm servo modified for continuous rotation. What we found when experimenting with the continuous rotation modified servo was that without the precision of feedback control, the servo's holding position was extremely unreliable. Outside of a small window around the modified servo's dead zone, the servo would drive forwards or backwards near its maximum rate of about 50 rpm, an unacceptable angular velocity for our precision requirements. We quickly determined we had to use unmodified servos and find a workaround for the standard 180 degree limit. I discuss these limitations and workarounds in 3.2 Pan-Tilt System Verification.

Light power control for this system is operated by a power circuit that steps down the 12 V dc operating voltage of the system to 5 V dc and applies an NMOS FET switch to control power to the light source.

With these design features in mind, we made 3D models of the entire system using Autodesk Inventor. The physical components in the system were either sourced from their respective online manufacturers website or machined by ECE Machine Shop Services. A full render as well as a sub-assembly of the cart and individual assembly drawings of components can be seen in figure B.3 in Appendix B.

After generating the 3D model, the design was then fabricated and assembled utilizing the ECE Machine Shop Services. The final machined design can be seen below.

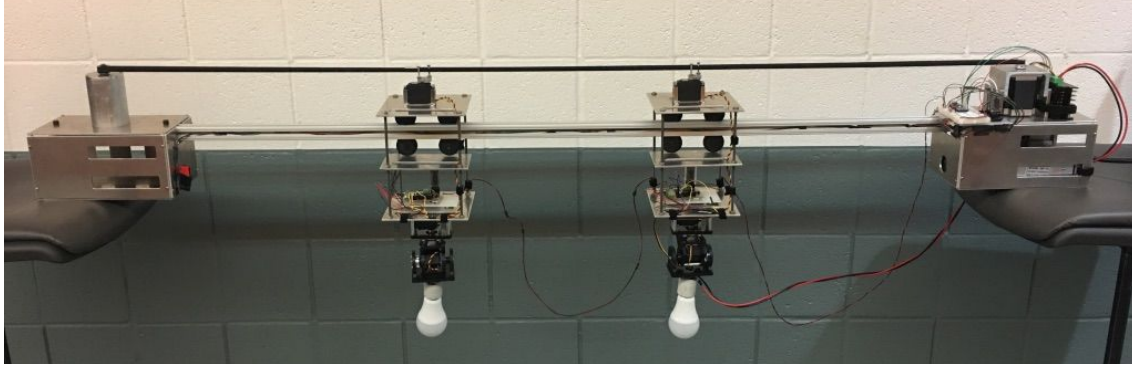


Figure 2. The final design of the track system

2.2.2 Track Electrical Design

The track electrical system utilizes the ATmega328P processor to handle all of the processing done for this system. One processor is placed on each of the carts as well as the master track node. To allow for bluetooth communication, the HC-05 bluetooth module is utilized to allow for bluetooth serial communication. The master node contains three of these HC-05 modules while the carts only have one.

The master node requires three bluetooth modules to allow for communication between the mobile application and track, as well as communication to each cart respectively.

The circuit schematic for both the master and slave control module can be seen below. A modular design approach was taken to design this system. Due to the overlap between hardware components, the decision to use a modular design was made to reduce manufacturing time as well as reduce the complexity of the verification process.

The key differences between the functionality of the cart and master node PCBs are that the master node allows for stepper motor control using a 4 wire breakout that controls the systems TH6600 motor driver. The master node also has a separate breakout for the three bluetooth modules. The cart PCBs utilize the breakouts for full RC servo motor control as well as has a separate breakout for the bluetooth communication module.

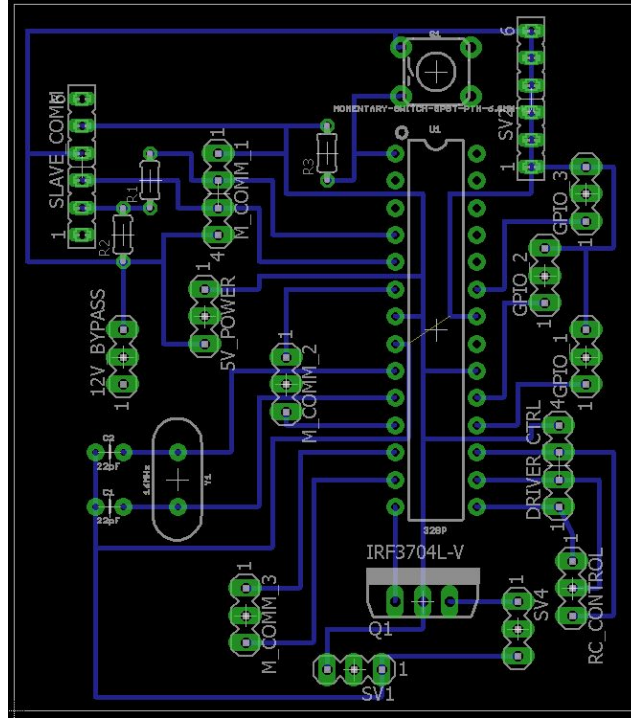


Figure 3. The control module PCB layout

The detailed circuit schematic for this PCB can be found in figure B.2 in Appendix B. This particular PCB module is powered by a single 12 V dc to 5 V dc converter to ensure that current overdraw will not damage the components.

The ATmega processors are all flashed with the Arduino bootloader to allow us to program with the arduino sketch environment. By utilizing this environment we are able to take advantage of the multiple built in I/O libraries provided by the Arduino communication.

To handle all the data communication, We utilized the HC-05 chip [3], a Bluetooth Serial Port Protocol (SPP) module capable of wireless serial communication.

The instructions encoded command the carts to perform these functions:

- Toggle power to light source
- Pan the light source
- Tilt the light source
- Latch on to the traversal belt
- Move the cart forwards and backwards

Utilizing a modular string parsing process, ascii characters are decoded and then the appropriate functions are called to complete. The exact process used is discussed in greater detail in section 2.4, App Interface.

2.3 Power System

The power system for the motorized track utilizes 3 different operational voltages: 120 V ac, 12 V dc, and 5 V dc. The power system step down conversion begins with our 120 V ac to 12 V dc converter that regulates our output voltage and allows for a max current draw of 30 A. This particular supply was chosen in the situation that the user adds more than 10 light carts on a single track.

The overall current draw from all the 5 V dc peripherals like RC servos could lead to stalling behavior as the system will not be able to provide the necessary power. Once the 12 V dc conversion is done, this voltage is carried across the masternode and to the carts using serial wiring and pass throughs. This decision was made to ensure that power was being delivered to each component and the possibility of wires slipping out of the device would be prevented by using screw terminal blocks as the entry and exit points of the pass through system.

Once the 12 V dc is brought to each respective module, the 12 V dc is regulated down to 5 V dc using the LM317T linear regulator with a current limit configuration. By using the linear regulator the appropriate operating voltage can be provided to control PCB as well as the RC servos being used on each cart.

Due to the output current limitation of the LM317T being 1.25 A, each RC servo was given its own regulator to prevent unstable behavior in the case that a motor ends up stalling. To ensure that the lightbulb receives enough power, the 12V passthrough on each power module is used to directly power the LED bulb with the appropriate amount of power.

An abstract breakdown of the flow of power can be seen in figure 3 below.

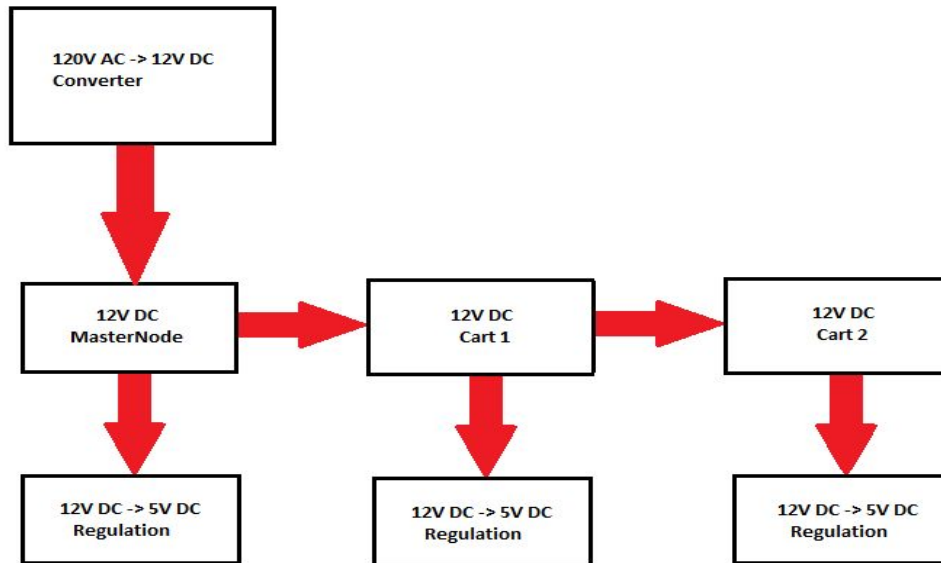


Figure 4. Power system diagram for various operating voltages

The system is overall rather simple as it takes advantage of the passthroughs and the redundancy between the multiple LM317T modules. Unfortunately due to computer issues, the PCB footprint was lost for this board, but the schematic was retained. The circuit schematic for the power regulation PCB can be seen below in figure 5.

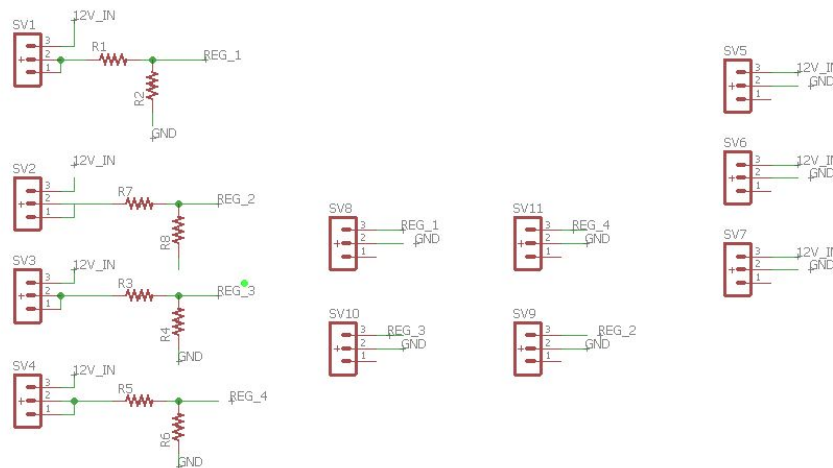


Figure 5: power regulation board PCB

2.4 Application Interface

The application interface portion of the block diagram in figure 5 is how users interact with the track. It is a mobile application written in React Native. We chose the React Native framework because it provides the capability to easily deploy the same application on both Android and iOS devices. In order to give our app Bluetooth capability in order to connect to the track, we used an open source library called React Native Bluetooth Serial Next [15]. This library gave our app basic Bluetooth functionality: connect to, read from, and write to Bluetooth devices. A simulated screenshot of the app on the Google Pixel 2 is shown in Figure 10. During the later part of the app design process, we came across the fact that iOS devices could not connect to the HC-05 Bluetooth module. The reason for this is because Apple uses the MFI Licensing Program, and the HC-05 is not a licensed device under that program [16]. Therefore, although if the app is deployable to both Android and iOS devices, only Android phones can connect to the track with this iteration of the design. In the future, we will search for alternative Bluetooth modules that allow both Android and iOS connectivity.

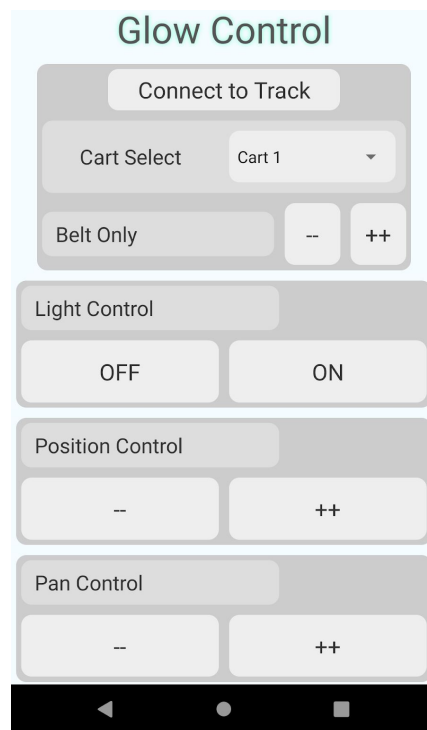


Figure 6. App UI Layout for Functionality Testing

There are eight commands that can be sent by the user to a specified cart. The application will allow the user to change the cart under control by selecting the cart from the “Cart Select” dropdown near the top of the app. The cart closest to the stepper motor is referred to as “Cart 1”

and the cart second closest to the stepper motor is “Cart 2.” If more carts were added to the system, they would be numbered in increasing order away from the stepper motor. The light for that specific cart can also be set to the on and off state by using the “ON” and “OFF” buttons in the “Light Control” box. The lateral movement, pan, and tilt operations each represent a single degree of freedom in the system that can be increased or decreased, so each have two buttons. When a button is pressed, the app sends the appropriate instruction to the master node, where it is parsed and the instruction content is relayed to the target cart. These instructions are sent in the form of short strings. The first set of characters defines the target cart number (if applicable), and the following character(s) are the instruction code to be carried out.

To give the user greater control over their lighting system, we incorporated two types of instructions: step instructions and continuous motion instructions.

The step instruction is triggered by a short press on any of the position, pan, or tilt buttons. The effect of a step instruction is a short, predefined movement. For position control, we set this to 100 steps, or roughly 0.8 inches. For pan and tilt control, the servo changes its angle by one degree. Step instructions are meant for making fine adjustments to the position and angle. Figure 6 details the message passing protocol between the app, master node, and target cart to execute a step instruction.

Continuous motion instructions are triggered on a long press of any of the motion control buttons (as well as the belt control buttons). This instruction type commands movement until the user lifts his or her finger off of the button. These instructions are meant to allow the user to rapidly approximate his or her desired configuration. Figure 7 details the message passing system between the app, master node, and target cart to execute a continuous motion instruction.

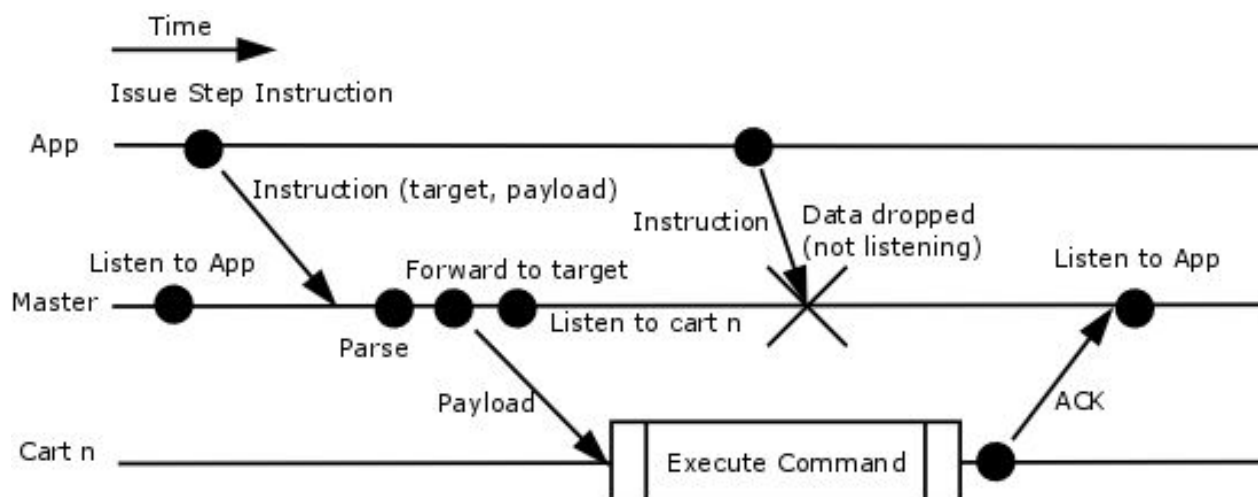


Figure 6. Step instruction message passing protocol

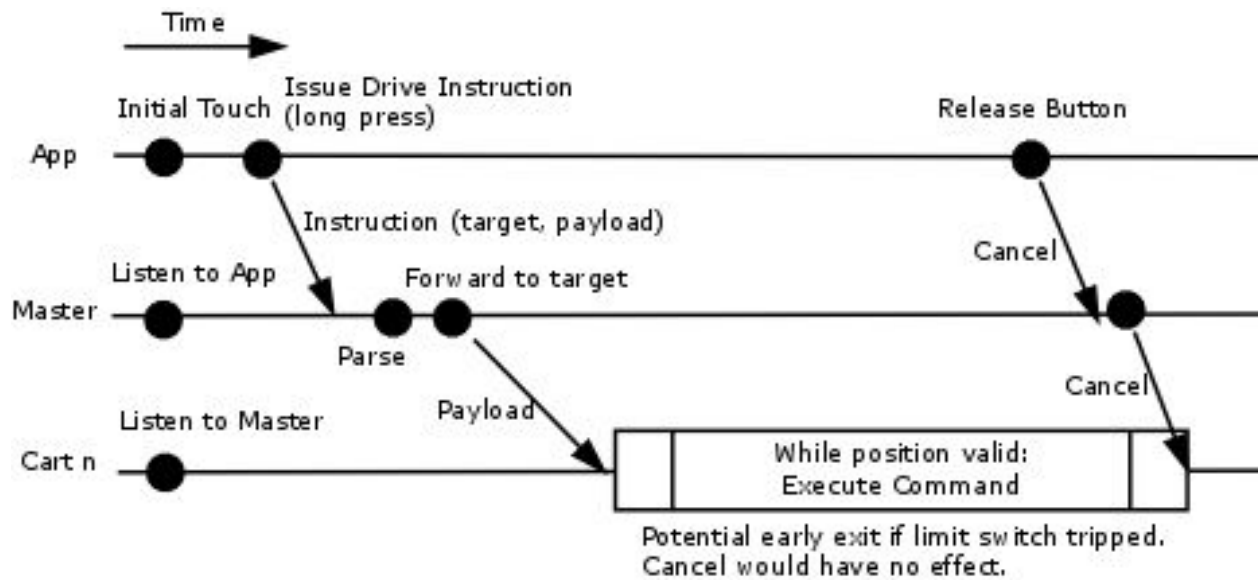


Figure 7. Continuous motion instruction message passing protocol

3. Verification

In order to ensure that the design proposed met the requirements we set out at the start of this semester, rigorous verification procedures were made to determine if the system was capable of delivering the desired core functionality. This section is broken down into the three subsystems described in the block diagram seen in figure 1. Subsystems with requirement and verification (R&V) procedures to determine expected functionality which are referenced in the text and can be found in appendix A.

3.1 Traversal System Verification

One of the major points of emphasis in our design of this system is scalability to more than two carts and to longer runs of track. When scaling track length, we don't want the user to have to wait uncomfortably long periods of time to move the carts. We have characterized through both simulation and experimentation is the speed and acceleration of a cart as it moves along the track.

Because our design incorporates a stepper motor and 2mm GT2 timing belt, we can efficiently quantify the distance that the cart travels per step of the motor.

Our NEMA-23 stepper motor makes one revolution per 200 steps. The neoprene GT2 belt has a pitch of 2mm, where pitch is the separation between teeth on the inside surface of the belt. Then we need to select a drive pulley for the stepper motor with a number of teeth that translates the number of steps per rotation into a movement speed that upholds the utility of our project even when scaled beyond our implementation this semester.

Our motor, and many standard stepper motors divide a revolution into 200 steps, or 1.8 degrees per step ($\pm 5\%$ in the case of our motor).

Then we know that revolutions per second is defined by equation 1:

$$\frac{\text{revolutions}}{\text{second}} = \frac{\text{rotation}}{200 \text{ steps}} \cdot \frac{\text{steps}}{\text{second}} \quad (1)$$

We predict the motor can be accelerated to upwards of 2000 steps/second.

From equation 1, we get that the speed of our motor caps out at 10 rotations/second.

Next, we solve for the distance the belt travels per rotation of the stepper motor shaft. The *pitch* of the timing belt is the distance between adjacent teeth. For our GT2 timing belts, the pitch is 2 mm (assume negligible tolerance amount for neoprene over the length of the belt, actual tolerance not listed). GT2 pulleys have some number of teeth. We found a company that sells timing belts and pulleys for 3D printers. Listed pulleys had 16, 18, 20, or 36 teeth (spaced for the 2mm pitch of the GT2 belts).

To solve for the distance the belt is moved per rotation, we multiply the number of teeth on one pulley by the pitch of the teeth, yielding equation 2:

$$\frac{\text{distance}}{\text{rotation}} = (\text{number of teeth on pulley}) \cdot (\text{pitch between teeth}) \quad (2)$$

It follows that the speed of a fixed point on the belt is governed by equation 3:

$$\text{belt speed} = \frac{\text{distance}}{\text{rotation}} \cdot \frac{\text{rotations}}{\text{second}} \quad (3)$$

Graphing the belt speed against the motor's steps per second allows us to estimate the number of steps per second at which our belt drive motor should be driven. We see in Figure 12 that our anticipated operating speeds of 3 ± 0.5 in/s are well within the operating range of our stepper motor.

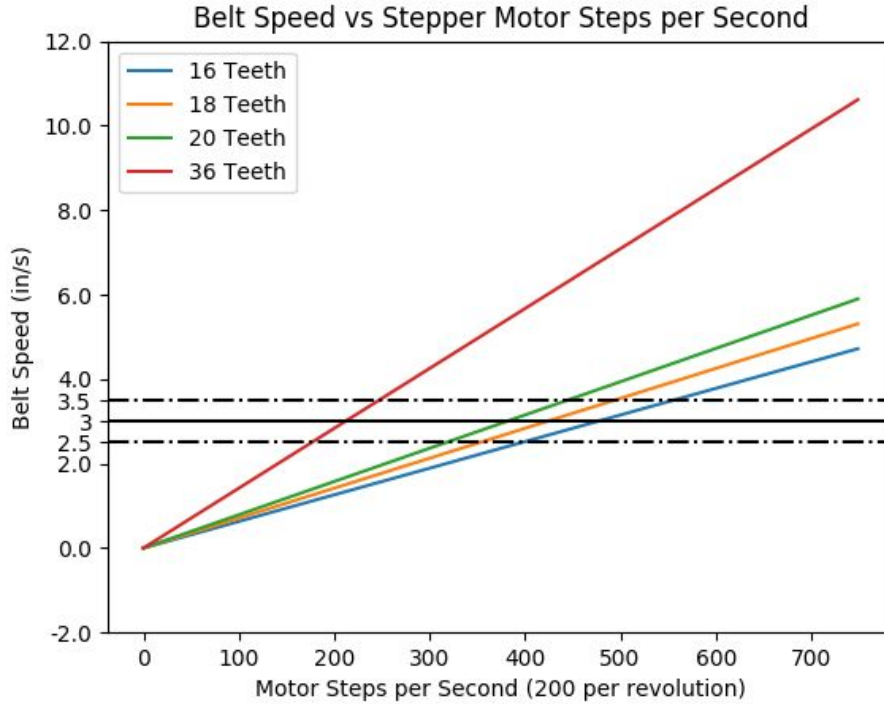


Figure 12. Belt Speed vs Stepper Motor Stepping Rate

3.2 Pan-Tilt System Verification

Upon transitioning from the continuous modified rotation pan servos to using standard 180 degree hobby servos for both pan and tilt, we had to design some software workarounds to maximize the range our light source can access. To do this, we recognized that though we only wanted the tilt to operate in the range of horizontal to vertical downwards at any given time, the SPT-200 pan tilt kit allows the servo to tilt from 0 degrees (defined pointing horizontal to ground), past 90 degrees up to 150 degrees. For safety reasons we software limit the range to only tilt to 145 degrees, to avoid the servo trying to push the plates against each other and stalling.

The range when pan is limited to 180 degrees and tilt is limited to 90 degrees is shown in B.4. By tilting past 90 degrees, the light source can point into the other half of the hemisphere, as shown by the orange region in B.6. Unfortunately, the SPT-200 doesn't allow for the full 180 degree tilt servo range to be taken advantage of, so the orange side of B.5, B.6, B.7 is limited to 55 degrees.

Thankfully, we project that most lighting applications will not require lighting incident at an angle from the floor less than 25 degrees (56-90 degrees of tilt away from vertical down). We believe that is a reasonable expectation given that track lighting is ceiling mounted, and the final implementation for this course does not miss the high level design requirement for the tilt range by a significant amount.

3.3 Power System Verification

Out of all the subsystems that were verified, the power system had the most issues with respect to working in the expected R&V tolerances.

Although each LM317T module delivered the required output voltage within the 10% tolerance range. The poor selection of gain resistors lead to the necessary output current being incorrect and therefore making the LM317T ineffective at distributing the necessary power to the sub system. The likely issue for this problem was a poor selection of the R1 resistor in the voltage divider subcircuit. That R1 resistor sets the current limit given the output voltage. The poor selection of that led to not enough amperage being delivered to the PCB devices. For that reason, our LM317T's failed their respective R&V procedure seen in table A.6.

The poor performance of the power regulation board resulted in our group utilizing the already stepped down 5 V power provided from the lab kit for some of the hardware functionality.

With respect to the wall outlet power supply, it was able to appropriately convert and step down the 120 V ac to 12 V dc with a measured output voltage of 12.13 V dc. Looking at table A.7, the requirement that the component needed to deliver at least $12\text{ V} \pm 10\%$ of the desired value. Since the 12 V bypass was not needed due to the power system board failure, the power supply was only driving a load of 0.268 A to the TH6600 stepper motor driver which was controlling the NEMA-23 motor. This value fell under our 10 A maximum supply draw requirement.

Overall, however a consideration that was not initially made was the total power draw of this system. One of the major issues that genuinely would have failed many of our R&V requirements is the inefficient power consumption. In the case of the 12 V dc to 5 V dc conversion using the LM317T:

Power Step conversion of 12 V dc to 5 V dc:

$$12\text{ V dc} - 5\text{ V dc} = 7\text{ V dc loss} \quad (4)$$

If an RC servo is at full stall on the light cart, it pulls 1.2 A, meaning the power loss for 1 RC servo is,

$$P = I * V = 1.2 * 7 = 8.4 \text{ W Consumed} \quad (5)$$

This power loss multiplied by the max number of regulators per power board is,

$$8.4 \text{ W} * 4 \text{ Regulators per board} = 33.6 \text{ W consumed per board} \quad (6)$$

33.6 watts is consumed at worst case by each light cart. If the system has multiple light carts, this is a massive power loss that is inefficient for a product that will likely be on at hours at a time.

3.4 Application Interface Verification

We unit tested the mobile application to verify it functions according to the interface we established in the code. The instruction set that would communicate with the track was entirely up to us, so we could make it as complicated or simple as we wanted. First, we focused on getting simple instructions sent. We need to establish how information sent from the app is being received on the ATmega side.

It is possible that some data types could be converted to a different format, so the first step is discovering how this information is sent. To start out, we sent simple characters such as 'A' and 'B' to the ATmega and then printed out exactly what we read in onto the Serial monitor of the ATmega. That way, we can know exactly how the information sent is being interpreted. We found that the baud rate at which the Serial port of the ATmega is expecting data affects how the data is read in. When we set the baud rate to 9600 Bd, the data was read in as garbage. The processor could not make any sense of the data. However, when we changed it to 38400 Bd, the data read in matched the data written from the app exactly. It was also necessary that we sent strings data types from the app, not character data types.

Once we had established exactly how data is received at the master node, we began constructing conditional statements. For example, we would blink a test LED using the ATmega if the data read in was a 'A.' Then we verified that the data we sent from the app could trigger this action on the processor side. Now, we could construct our instruction set. Depending on what action the user wanted to take, different buttons on the app would send different Strings to the processor. We set the processor to perform different blink functions based on different buttons pressed on the app. The reason why we used the blink LED to test the instructions being sent from the app is because the other commands the processor had to perform to pan/tilt as well as activate the stepper motor were also relatively simply one-line commands. This way, the simple blink lines could be easily swapped out for the actual commands that we would need to write when testing on the actual product. The instruction set is straightforward to expand from this point. The full instruction set can be found in Table B.4. Each of these instructions were mapped to different button on the app UI, shown in Figure 6. Each of these instructions were verified in a similar fashion to what was mentioned above.

4. Cost Analysis

4.1 Materials Cost

Table 1. Physical Design Costs

Part	Manufacturer	Retail Cost	Quantity	Total Cost
72" 1"x1" T-slot track	80/20	\$11.73	1	\$11.73
1" Roller Bracket Assembly	80/20	\$38.85	4	\$155.40
12 V dc Lightbulb	GE	\$9.99	2	\$19.98
SPT-200 Pan Tilt Kit	Servo City	\$39.09	2	\$78.18
HS-645MG	Hitec	\$25.49	4	\$101.96
HS-422	Hitec	\$10.45	2	\$20.90
Servo Hub Horn	Servo City	\$4.24	2	\$8.48
Vertical Aluminum Servo Mounts	Bilda	\$4.89	2	\$9.78
NEMA-23 Stepper Motor	Zyltech	\$16.99	1	\$ 16.99
TH6600 Stepper Motor Driver	Toshiba	\$12.99	1	\$12.99
4m G2 Timing Belt	Inventables	\$20.00	1	\$20.00
.25" Pulley Wheel	Inventables	\$6.99	1	\$6.99
4' x 4' x 1/8" Aluminum Sheet	Metals Depot	\$34.60	1	\$34.60
			Total	\$497.98

Table 2: Electrical hardware parts

Part	Manufacturer	Retail Cost	Quantity	Total Cost (\$)
LM317T Regulator	Texas Instruments	\$1.64	9	\$14.76
ATMega328p processor	Atmel	\$8.62	3	\$25.86
120 V ac to 12 V dc Power Supply	Supernight	\$18.95	1	\$18.95
2 Pin 5mm PCB screw terminal block	Kangwei	\$.15	12	\$1.80
16 MHz crystal oscillator	Uxcell	\$.95	3	\$1.92
HC-05 Bluetooth module	DSD Tech	\$7.62	5	\$38.10
IRF951 PMOS FET	Vishay	\$1.32	2	\$2.64
6mm Tactile Button	Uxcell	\$.35	3	\$1.05
Limit Switch	Cylewet	\$1.05	2	\$2.10
Total	\$107.18			

Total Parts Cost: $497.98 + \$107.18 = \605.16

4.2 Labor Cost:

For the purposes of estimating labor cost, the national average for an Electrical Engineer was slightly more than \$71,000/year [6].

Hourly salary can be computed from annual salary by:

$$\text{hourly salary} = (\text{annual salary}) \cdot \frac{\text{year}}{52 \text{ weeks}} \cdot \frac{\text{week}}{40 \text{ hours}} \quad (7)$$

Taking \$71,000 as the annual salary, equation 4 gives:

$$\frac{\$71,000}{\text{year}} \cdot \frac{\text{year}}{52 \text{ weeks}} \cdot \frac{\text{week}}{40 \text{ hours}} = \frac{\$34.13}{\text{hour}} \quad (8)$$

Table 3: Breakdown of total labor cost

Name	Rate (dollars/hr)	Hours/Week	Project Total (dollars)
Pratik Ainapure	\$34.13/hr	12	\$5,329
Adithya Bellary	\$34.13/hr	12	\$5,329
Ethan Hickman	\$34.13/hr	12	\$5,329
Total Labor Cost			\$15,987
Adjusted Labor Cost (Total Labor Cost x 2.5)			\$39,967

After taking into consideration of material and labor cost, the total estimated cost to implement to develop this motorized track lighting system is:

$$\text{Total Cost} = \text{Parts Design Labor} \quad (10)$$

$$605.16 + 39,967 = \$40,572.16$$

5. Conclusion

5.1 Accomplishments

As of the end of the course, the system meets two of the three high level requirements we set out to fulfill. The carts can be driven anywhere along the track, and belt operation is quick and accurate. Though the system can't direct the light anywhere in the hemisphere range below the track, the loss of the top thirty degrees of the range on one side of the hemisphere is acceptable. Though we weren't able to integrate light power switching and dimming functionality onto the carts, we were able to demonstrate the switching functionality separate from the rest of the system. Additionally, we have a viable application built on an extensible platform.

5.2 Uncertainties

During the design process, we came across some challenges. One of them was regarding the way we had structured the software application and the Bluetooth network. In retrospect, our use of delays could have been optimized. The reason we added delays into the code was to allow for time for the instruction to be carried out. However, during testing times, we noticed that these delays could put our track system in states that were difficult to get out of. The delays on the ATmega side essentially froze the entire processor, preventing it from reading any instruction sent to it. We never want to allow the track to get into a state that we cannot get it out of. This could create some safety hazards, such as if a cart on the track could not stop. It could either crash into the end of the track or another cart, possibly causing falling debris. If given more time, we would reduce the number of delays and more efficiently allow for the system to carry out the instructions passed to it.

5.3 Ethical Considerations

5.3.1 Ethics

Throughout the development of the system we drew from IEEE Code of Ethics [1] to ensure that the final product that we create aligns with the Code established by IEEE. We understand that there are safety hazards regarding this product and therefore will abide by the statement #1 "hold paramount the safety...of the public". Appropriate safety circuitry and mounting practices will be implemented. Individual components will be verified to reduce the chances of malfunction and injury. The safety considerations that we made are expanded in greater detail in the following safety section.

5.3.2 Safety

We worked to mitigate the inherent danger of designing with powerful electronics as we prototyped this product. One potential hazard could have been the manner in which we powering the system as whole. We are designing the product to draw power from a standard wall outlet, which is 120 V ac. The

individual components on our track take in DC voltage, so we will need to use a AC-DC transformer. This could be dangerous. Light bulbs could also pose a fire hazard in the case that they overheat. The Led bulbs will be regulated to ensure that they operate within respective power ratings.

Additional considerations we have made are operation in the face of current overdraw or power loss. To minimize the impact of power fluctuations, this system is designed with the intentional selection of parts with higher power ratings to minimize the likelihood of damage from overdrawing current. In the event of power failure, the system will likely need to be reconfigured to determine it's zero positions. Currently, the only portion of the system that does get saved is the addressing between the master and slave processors on the track system and the software that will be flashed onto each processor.

Another safety consideration we made is security. The user connects to the track system through Bluetooth communication, theoretically anyone in the vicinity could connect to it. In order to prevent any unauthorized users from controlling the track system, we designed the system so that the user needs to know the hardware address of the bluetooth module the application communicates with.

5.4 Future Work

There are several improvements that can be made to this system to improve the user experience, as well as the overall build quality of the track lighting system.

To improve the physical design we could optimize the size and weight of the carts and pan-tilt kit. A reimagined housing enclosure would significantly improve the aesthetic of the system as well. Due to the strict timeline for designing the track, we put less emphasis on the appearance of the track relative to ensure high level functionality.

Improvements that could be made in software would be the inclusion of a scene recall feature as well as configurable presets. The scene recall feature would be beneficial to the user in the event of a power outage or the need to reset the lighting to its previous configuration. This functionality would sequence the carts into their last known positions and adjust their pan and tilt positions as well. Allowing the user to have presets would also be a nice software feature that would allow the user to store multiple lighting configurations on their device for a scene recall when they desire.

References

- [1] Ieee.org. (2019). *IEEE Code of Ethics*. [online] Available at: <https://www.ieee.org/about/corporate/governance/p7-8.html> [Accessed 18 Feb. 2019].
- [2] “IP 66, IP 65 and IP 67 Ratings Explained,” *BUD Industries*, 18-Aug-2017. [Online]. Available: <http://www.budind.com/blog/2014/02/the-mysteries-of-ip-rated-enclosures-explained/>. [Accessed: 18-Feb-2019].
- [3] “HC-05 Manual,” *GM Electric*. [Online]. Available: <https://www.gme.cz/data/attachments/dsh.772-148.1.pdf>. [Accessed: 19-Feb-2019].
- [4] “SPT200 Pan & Tilt Kit,” *ServoCity.com*. [Online]. Available: <https://www.servocity.com/spt200>. [Accessed: 19-Feb-2019].
- [5] Actobotics Horizontal Standard Gripper Kit A,” *Robotshop*. [Online]. Available: https://www.robotshop.com/en/actobotics-horizontal-standard-gripper-kit-a.html?gclid=Cj0KCQiAtbnjBRDBARIsAO3zDI91dzsh7SzsdolUbW8WfRmgo2os51iVRd1JCxcWPiqzOib31i1PQuEaApc2EALw_wcB. [Accessed: 21-Feb-2019].
- [6] “Top 3 Industries for Highest Electrical Engineer Salary,” *Interesting Engineering*, 12-Mar-2018. [Online]. Available: <https://interestingengineering.com/top-industries-highest-electrical-engineer-salary>. [Accessed: 20-Feb-2019].
- [7] “Arduino Software Serial Example,” *Arduino - Introduction*. [Online]. Available: <https://www.arduino.cc/en/tutorial/SoftwareSerialExample>. [Accessed: 21-Feb-2019].
- [8] *Allegromicro.com*, 2019. [Online]. Available: <https://www.allegromicro.com/~media/Files/Datasheets/A3901-Datasheet.ashx>. [Accessed: 21-Feb- 2019].
- [9] ATMEL, “8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash DATASHEET,” *microchip*. [Online]. Available:

http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATMega328P_Datasheet.pdf. [Accessed: 21-Feb-2019].

[10] “TB6600HG,” *Microcontrollers in everyday life | TOSHIBA Semiconductor & Storage Products | Asia-Pacific*. [Online]. Available: <https://toshiba.semicon-storage.com/us/product/linear/motordriver/detail.TB6600HG.html>. [Accessed: 22-Feb-2019].

[11] “Track Lighting Heads | Lamps Plus,” *Ideas & Advice | Lamps Plus*. [Online]. Available: https://www.lampsplus.com/products/track-lighting/usage_track-heads/. [Accessed: 21-Feb-2019].

[12] “Installing Track Lighting,” *The Home Depot*. [Online]. Available: <https://www.homedepot.com/c/ah/installing-track-lighting/9ba683603be9fa5395fab90814b79eb>. [Accessed: 21-Feb-2019].

[13] “Indoor Grow Light Systems & Equipment,” *LightRail® Light Movers*. [Online]. Available: <https://www.lightrail3.com/>. [Accessed: 21-Feb-2019].

[14] “MotoLux Series,” *Formal Lighting*. [Online]. Available: https://l.messenger.com/l.php?u=https://www.formalighting.us/products/motolux-series&h=AT3ZOMk4icfx7dd7qlPPgYFFMQulUzd4WvTOkJiwY1Ru90VMP-Qj8yiimsH-YPBnumS0Mv_L4SSQtNNPqUfUljwecwc9IYXV2u1qfNczpRvlGpiFuNOm4iIdDSg24A0-GizQ. [Accessed: 21-Feb-2019].

[15] “react-native-bluetooth-serial-next,” *npm*. [Online]. Available: <https://www.npmjs.com/package/react-native-bluetooth-serial-next>. [Accessed: 01-May-2019].

[16] AlokAlok 2, hcheunghcheung 1, and GrooverFromHollandGrooverFromHolland 5721713, “Connecting hc-05 with iPhone SE iOS(v11.0),” *Stack Overflow*. [Online]. Available: <https://stackoverflow.com/questions/50443456/connecting-hc-05-with-iphone-se-iosv11-0>. [Accessed: 01-May-2019].

Appendix A Requirement and Verification Tables

Table A.1 AC-DC Converter Requirement and Verification

Requirement	Verification
1. Convert 120 V ac power from the wall outlet to 12 V dc ($\pm 5\%$) needed to power the motorized track system	1. <ol style="list-style-type: none">Plug the AC-DC converter into a standard 60 Hz 120 V dc grounded outletUsing a multimeter, verify that the voltage at the converter output is 12 V dc ($\pm 5\%$)

Table A.2 Traversal System Requirement and Verification

Requirement	Verification
<ol style="list-style-type: none"> 1. The track stepper motor driver must be able to receive forwards and reverse signals from the master ATmega328P 2. The track motor must be able to move forwards and backwards on command from signals sent from the master ATmega328P 	<ol style="list-style-type: none"> 1. With an oscilloscope: <ol style="list-style-type: none"> a. Validate the signal produced by the ATmega328P being sent to the motor drivers against datasheet for the driver [10] b. Validate the signals produced by the motor drivers to accelerate the stepper motor to cart cruising speed of 2 ± 0.5 in/s 2. <ol style="list-style-type: none"> a. The torque must overcome the coefficient of friction of UHMW against aluminum, ($\mu=0.1-0.2$) times the weight of the cart b. The forward drive signal must make the motor rotate the correct direction (without loss of generality) to move a latched cart away from the motorized end c. The reverse drive signal must make the motor rotate the opposite direction as the forward drive signal to move a latched cart towards the motorized end

Table A.3 HC-05 Bluetooth Module Requirement and Verification

Requirement	Verification
1. Receive an instruction that will be in the form of a sequence of bits	1. <ul style="list-style-type: none"> a. Using a power supply set to 5.0V and 50 mA, power the HC-05 chip b. Connect to the HC-05 through Bluetooth using a laptop c. Create the test circuit shown in [3] d. Follow the rest of the step outlines in [3] e. Ensure that the information f. Connect an oscilloscope to the Rx pin of the HC-05 chip g. Ensure that the sequence displayed on the oscilloscope matches the sequence sent
2. Transmit an instruction that will be in the form of a sequence of bits	2. <ul style="list-style-type: none"> a. Using a power supply set to 5.0V and 50 mA, power the HC-05 chip b. Connect an oscilloscope to the Tx pin of the HC-05 chip c. Send the ascii value “a” through the serial communication software d. Ensure that the sequence displayed on the oscilloscope matches the sequence sent

Table A.4 Master ATmega328P Processor Requirement and Verification

Requirement	Verification
1. Processor must be capable of utilizing serial port to write data	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> a. Mount ATM 328 processor to a base arduino launchpad. b. Power the device via usb connection to a computer c. Connect the Tx output line of the processor to an oscilloscope channel and ground the other lead. d. Utilizing the Arduino IDE, upload the tutorial serial test sample sketch[7] to the board. e. Run the application f. Open the Arduino serial interface inside the IDE g. Send the ascii char “a” through the interface. h. Verify that the output on the oscilloscope side displays 0100 0001 by observing the rising and falling edges of the data stream passed in.
2. Processor must be capable of utilizing serial port to read data	<ol style="list-style-type: none"> 2. <ol style="list-style-type: none"> a. Mount ATM 328 processor to a base arduino launchpad. b. Power the device via usb connection to a computer c. Configure a function generator to output a 3.3V, 1.65v offset, 1kHz, 50% duty cycle square wave. Connect the positive lead to the Tx line of the processor. d. Utilizing the Arduino IDE, upload the tutorial serial test sample sketch [7] to the board. e. Run the application f. Open the Arduino serial

	<p>interface inside the IDE</p> <p>g. Check to see that a sequence of 1's is received on the serial port followed by a sequence of zeros, representing the square waveform.</p>
--	---

Table A.5 Belt Control Requirement and Verification

Requirement	Verification
1. Move the belt away from the stepper motor	<p>1.</p> <ul style="list-style-type: none"> a. Use a power source set to 12 V dc to power the stepper motor b. Using a function generator, send a forward command to the stepper motor driver c. Ensure that the stepper motor moves in a CW motion d. Using a multimeter, probe the input channel A+ and A- with positive and negative leads respectively e. Repeat same instruction for channel B+ and B-. Voltage should be high in the CW direction
2. Move the belt towards the stepper motor	<p>2.</p> <ul style="list-style-type: none"> a. Use a power source set to 12 V dc to power the stepper motor b. Using a function generator, send a forward command to the stepper motor driver c. Ensure that the stepper motor rotates CCW

Table A.6 Track Low Voltage Bus Requirement and Verification

Requirement	Verification
<ol style="list-style-type: none"> 1. Provide uninterrupted power of 5 V dc ($\pm 5\%$) to the master ATmega328P processor 2. The system can provide atleast 500mA ($\pm 5\%$) 	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> a. Connect the low voltage bus to a power source set to 5 V dc. b. Using a voltmeter, verify that the bus is supplying a steady voltage of 5 V dc ($\pm 5\%$)

Table A.7 LM317T 12 V to 5 V DC-DC Converter Requirement and Verification

Requirement	Verification
<ol style="list-style-type: none"> 1. Convert 12 V dc ($\pm 10\%$) to 5 V dc ($\pm 10\%$) 2. Ensure the max current load is under 1.25 A 	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> a. Connect power supply set to 12 V dc to the input of the converter b. Using a voltmeter, verify that the output voltage is within 5% of the 5 V dc. c. Place a load resistor in series with the output and measure the Amps through the branch using a multimeter.

Table A.8 Collision Control Requirement and Verification

Requirement	Verification
<p>1. The Normally Open Limit switch is capable of opening and closing a circuit by switch from the open to pressed position simultaneously</p>	<p>1.</p> <ol style="list-style-type: none"> a. On a test breadboard connect the normally open limit switch in series with a 220 Ohm resistor b. Apply 5 V dc utilizing a bench power supply across limit switch and resistor c. Using a multimeter, check the output current on the resistor to verify that current is not passing through the resistor (I.E. Open Circuit) d. Push the limit switch in to complete the circuit and verify if there is 20 ± 5 mA flowing through the resistor e. Depress the switch and repeat steps c-d 2 more times to ensure repeated results

Table A.9 Cart Requirement and Verification

Requirement	Verification
<ol style="list-style-type: none"> 1. The latching servo motor(s) must be able to latch on command from its respective slave ATmega328P 2. The pan servo must be able to pan on command from its respective ATmega328P 3. The tilt servo must be able to tilt on command from its respective slave ATmega328P 	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> a. Send the signal from the ATmega328P to actuate the latching servo horns to 45° b. Verify that the servo horns have moved to the assigned position 2. <ol style="list-style-type: none"> a. Send the signal from the ATmega328P to move the servo to a specific position b. Verify that the pan servo has moved to the assigned position 3. <ol style="list-style-type: none"> a. Send the signal from the ATmega328P to move the tilt servo to a specific position b. Verify that the tilt servo has moved to the assigned position

Table A.10 ATmega328P Processor (Slave) Requirement and Verification

Requirement	Verification
1. Processor must be capable of utilizing serial port to write data	<ol style="list-style-type: none"> 1. Write Process <ol style="list-style-type: none"> a. Mount ATM 328 processor to a base arduino launchpad. b. Power the device via usb connection to a computer c. Connect the Tx output line of the processor to an oscilloscope channel and ground the other lead. d. Utilizing the Arduino IDE, upload the tutorial serial test sample sketch[7] to the board. e. Run the application f. Open the Arduino serial interface inside the IDE g. Send the ascii char “a” through the interface. h. Verify that the output on the oscilloscope side displays 0100 0001 by observing the rising and falling edges of the data stream passed in.
2. Processor must be capable of utilizing serial port to read data	<ol style="list-style-type: none"> 2. Read Process <ol style="list-style-type: none"> h. Mount ATM 328 processor to a base arduino launchpad. i. Power the device via usb connection to a computer j. Configure a function generator to output a 3.3V, 1.65v offset, 1kHz, 50% duty cycle square wave. Connect the positive lead to the Tx line of the processor. k. Utilizing the Arduino IDE, upload the tutorial serial test sample sketch [7] to the board. l. Run the application m. Open the Arduino serial interface inside the IDE

	n. Check to see that a sequence of 1's is received on the serial port followed by a sequence of zeros, representing the square waveform.
--	--

Table A.11 Pan Control Requirement and Verification

Requirement	Verification
1. Rotate the pan servo from [4] through the full range of motion [-180, 180] degrees (or [0,360]) ($\pm 5\%$)	1. <ul style="list-style-type: none"> a. Using a power source set to 5 V dc and 150 mA, power the pan servo b. Using a function generator, send a high signal to the servo control pin. c. Use a protractor to ensure that the servo rotates through its full range of motion: 360 degrees ($\pm 5\%$).

Table A.12 Tilt Control Requirement and Verification

Requirement	Verification
1. Rotate the tilt servo from [4] through its full range of motion [180, 90] ($\pm 5\%$) from vertical	1. <ul style="list-style-type: none"> a. Using a power source set to 5 V dc and 150 mA, power the pan servo b. Using a function generator, send a high signal to the servo control pin c. Use a protractor to ensure that the servo rotates through its full range of motion: 90 degrees ($\pm 5\%$)

Table A.13 Light Source Requirement and Verification

Requirement	Verification
1. Turn the LED light bulb on	1. <ul style="list-style-type: none"> a. Connect the LED light bulb to the test circuit b. Using a function generator, connect the gate pin of the FET to a constant voltage of 2.0 V dc c. Ensure that the light bulb turns on
2. Turn the LED light bulb off	2. <ul style="list-style-type: none"> a. Connect the LED light bulb to the test circuit b. Using a function generator, connect the gate pin of the FET to a constant voltage of 0.0 V dc c. Ensure that the light bulb turns off

Table A.14 Belt Latch Control Requirement and Verification

Requirement	Verification
1. The latch stepper motor is able to deflect the belt 20 degrees (± 5) to transmit the belt's motion to the cart	1. <ul style="list-style-type: none"> a. Using a power source set to 5 V and 100 mA, power the latching servo b. Using a function generator, connect the signal pin to 5 V dc c. Measure belt deflection with a protractor d. Iterate on the value in this requirement experimentally with a range of weights (0-8 lbs) hanging below the bearing to simulate carts of different weight

Appendix B: Figures and Reference Tables

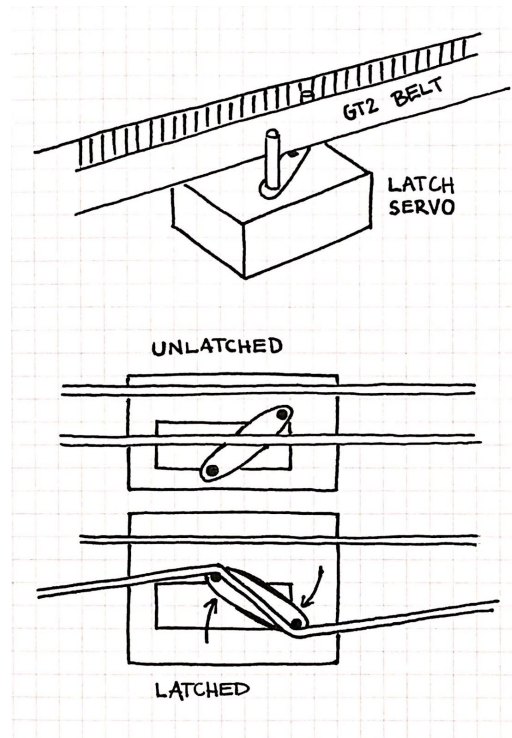


Figure B.1. Belt latching functionality

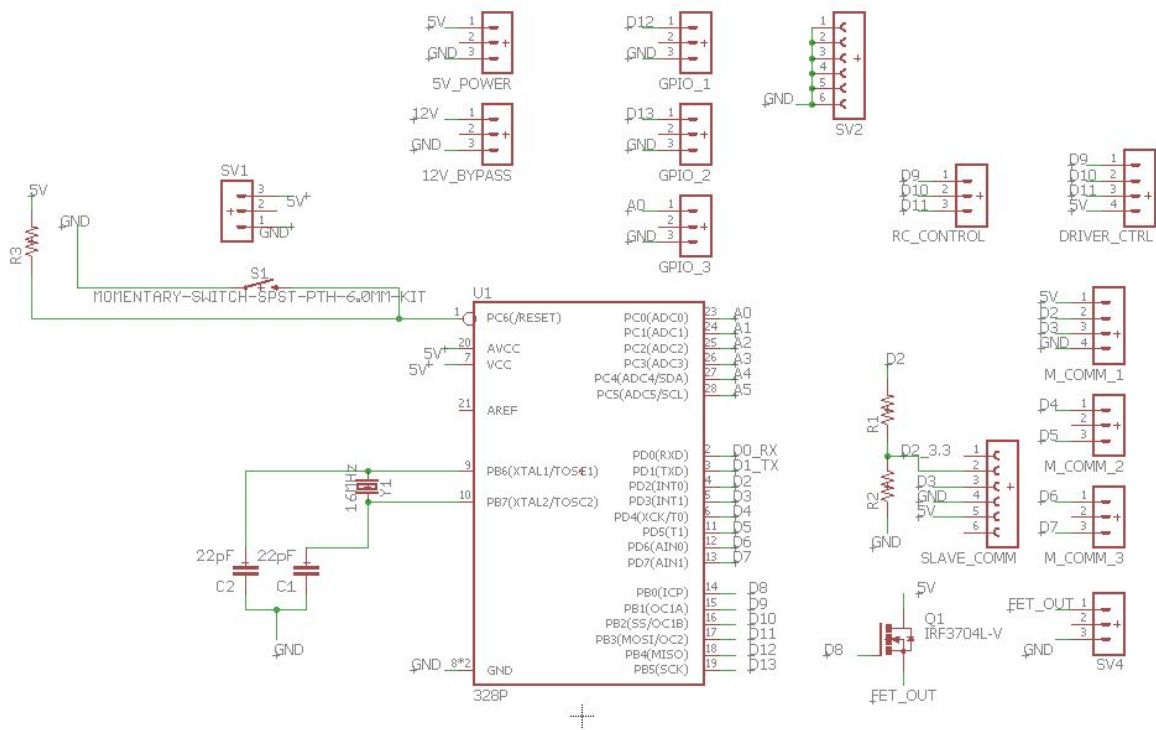


Figure B.2 Control PCB

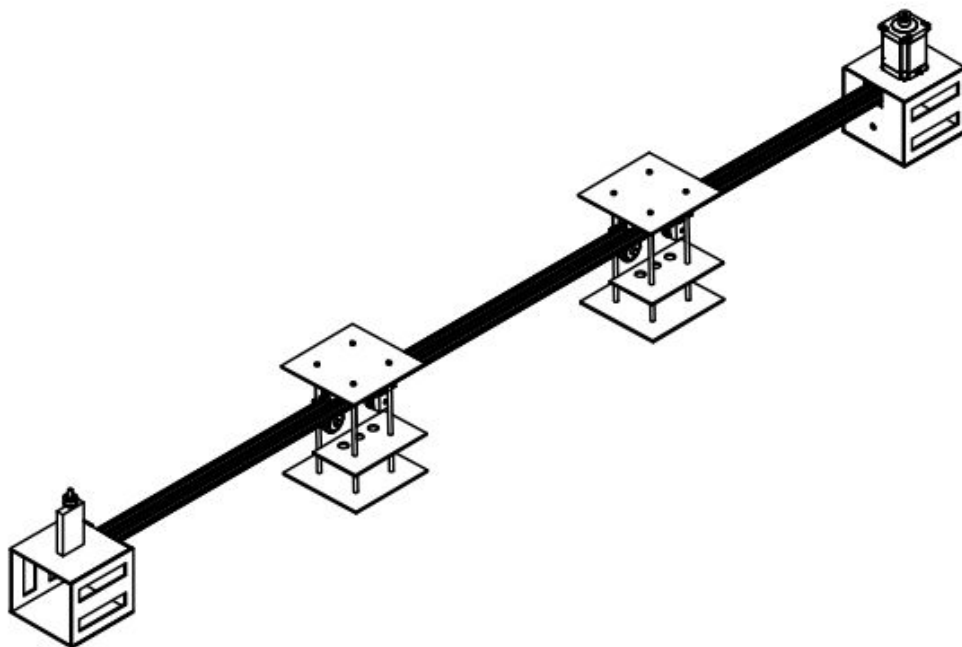


Fig B.3.1 3D render of the full track in Autodesk inventor

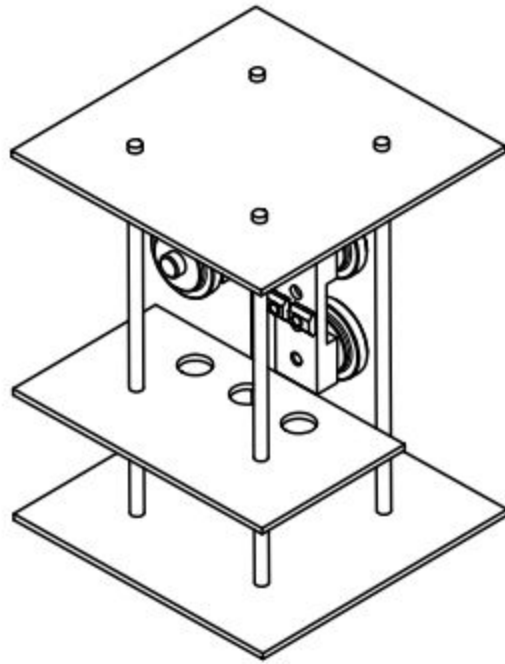


Figure B.3.2 3D render of the light cart module

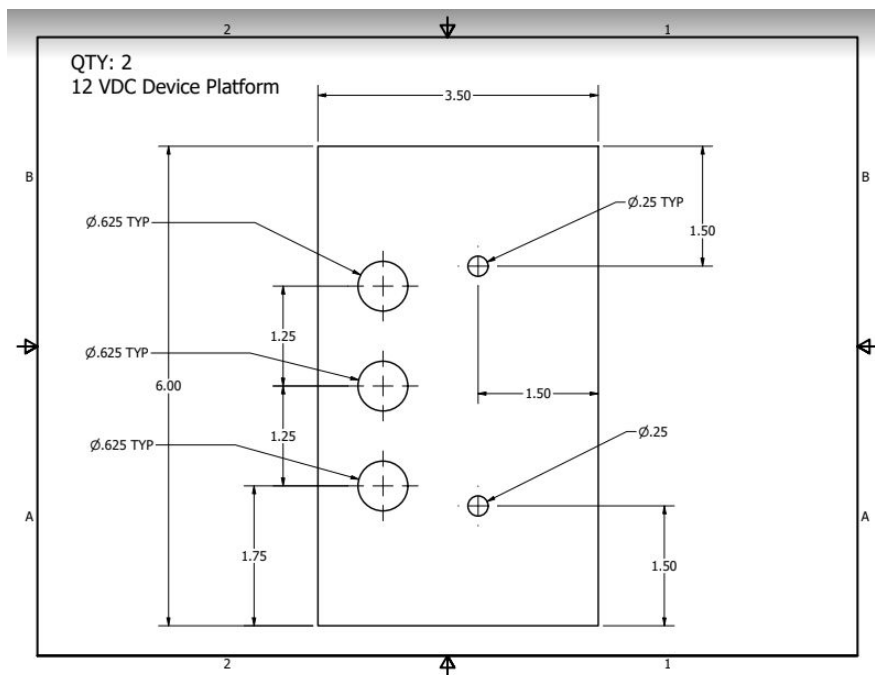


Figure B.3.3 A machine drawing example of a component from the light cart

Table B.1 App Interface Instruction Set

Instruction Sent	Action Carried Out
‘f’	Step cart forward 100 steps
‘F’	Drive cart forward
‘r’	Step cart reverse 100 steps
‘R’	Drive cart reverse
‘w’	Step tilt up one degree
‘W’	Drive tilt up
‘a’	Step pan one degree counter-clockwise
‘A’	Drive pan counter clockwise
‘s’	Step tilt down one degree
‘S’	Drive tilt down
‘d’	Step pan one degree clockwise
‘D’	Drive pan clockwise
‘L’	Latch
‘U’	Unlatch
‘o’	Turn light off
‘O’	Turn light on
‘b’	Drive belt clamp backwards
‘B’	Drive belt clamp forwards

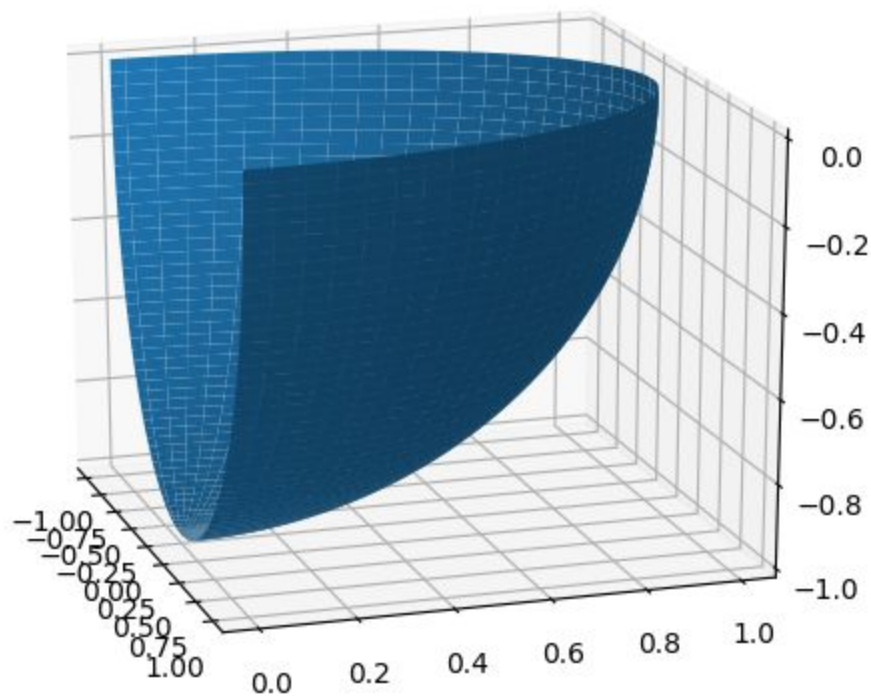


Figure B.4 Pan range 180, Tilt range 90

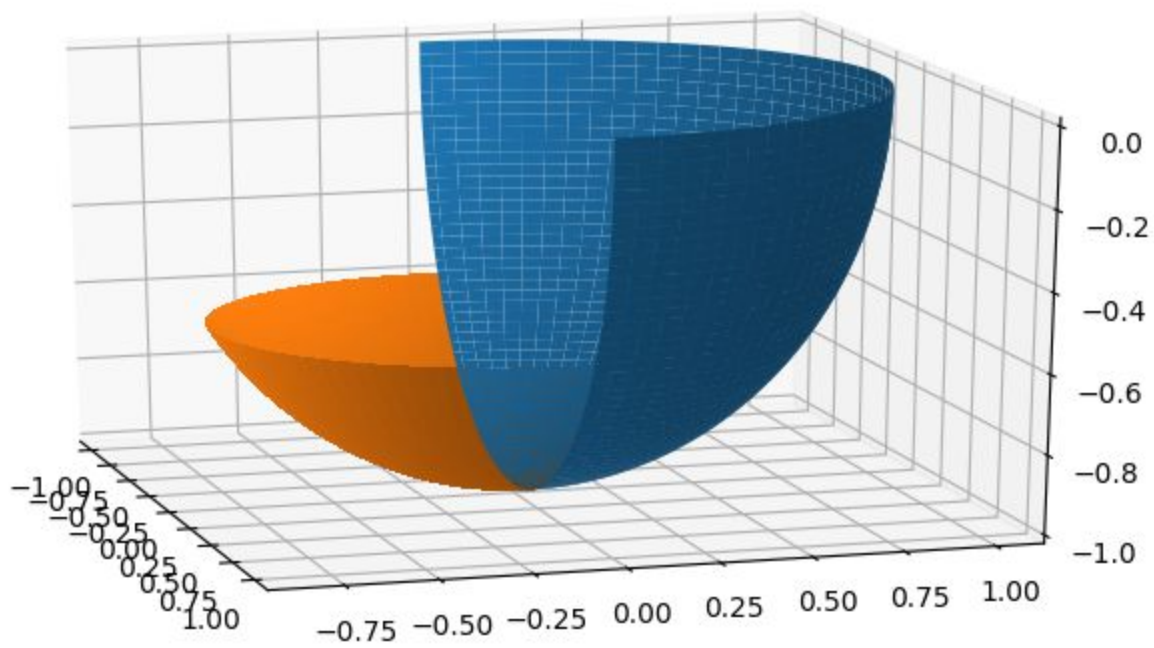


Figure B.5 Pan range of motion with software modification ($\frac{3}{4}$ View)

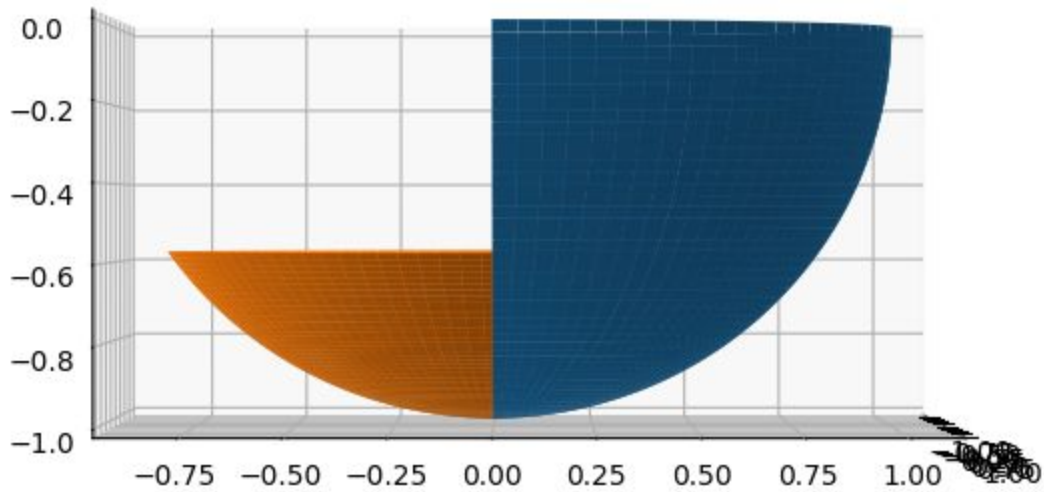


Figure B.6 Pan range of motion with software modification (side view)

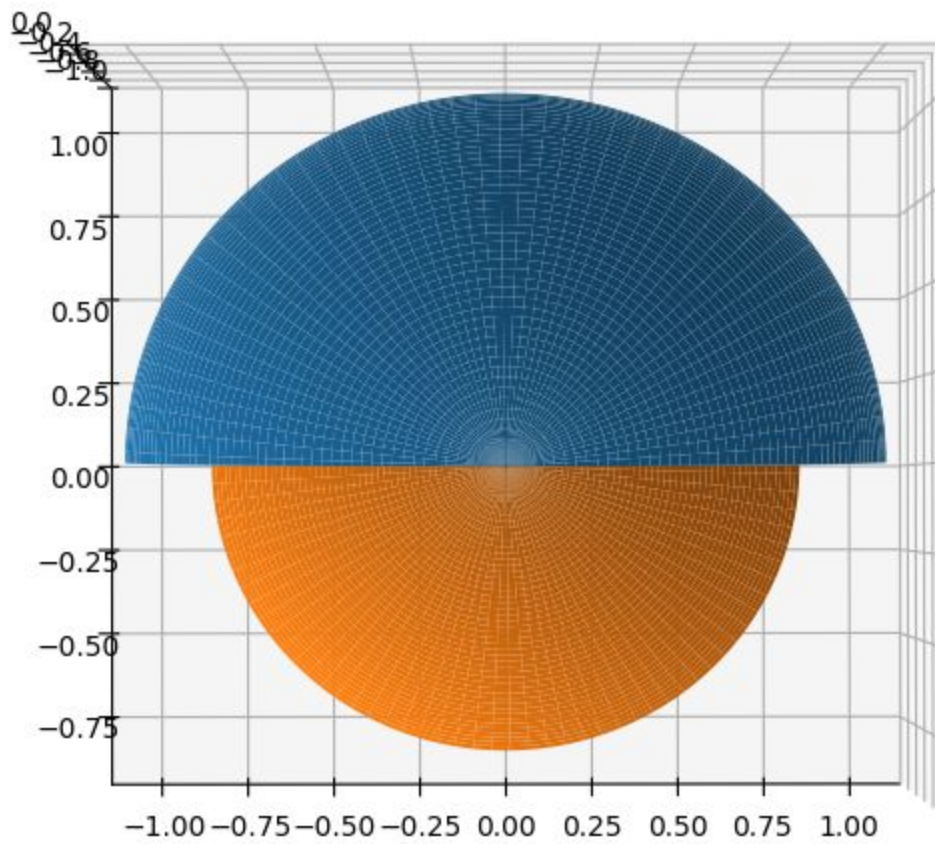


Figure B.7 Pan range of motion with software modification (top view)