UCF Senior Design

The AutonoMouse

Keep Your Pet Company While You're Away

(2021-4-27)

Group 14

Christopher Monks	**Electrical Engineering**
Nadja Solano Diaz	**Computer Engineering**
Jonathan Stolz	**Electrical Engineering**
Jing Wang	**Computer Engineering**

Table of Contents

Executive Summary *1*

1 Introduction........ *2*

2. Project Description........ *3*

2.1 Project Motivation and Goals **3**

2.2 Specifications **3**

2.3 Constraints **4**

2.4 Software Block Diagram **5**

2.5 Hardware Block Diagram **5**

2.6 House of Quality **6**

2.7 Finance and Budgeting **8**

2.8 Milestones **10**

3. Research Related to Project Description and Part Selection *11*

3.1 Existing Similar Projects and Products **11**

3.2 Relevant Technologies **13**

3.3 Part Selection **18**

3.4 Possible Architectures **41**

3.5 Final Part Selection **48**

4. Standards *51*

4.1 Search at www.nssn.org **51**

4.2 Design Impact of Relevant Standards **53**

4.3 Economic and Time Constraints **63**

4.4 Environmental, Social and Political Constraints **65**

5 Hardware and Software Designs *65*

5.1 Initial Design Architectures and Related Diagrams **65**

5.2 Mouse Subsystem **72**

5.3 Box Subsystem and Breadboard Test **74**

5.4 Charging Subsystem **76**

5.5 Additional Features **79**

5.6 Software Design **86**

6.0 Project Prototyping *99*

6.1 Initial Integrated Schematic 99

6.3 Initial PCB Vendor and Assembly 107

7.0 Project Prototype Testing Plan 113

7.1 Hardware Test 114

7.2 Software Test 120

8.0 Project Operation 126

9.0 Administrative Content 128

9.1 Milestones 128

9.2 Project Summary and Conclusion 132

Appendix 133

Citations: 133

Executive Summary

Pet owners often feel a sort of guilt when leaving their pets alone for extended periods of time. Cats are one of the main culprits of being left alone by their owners, sometimes for a couple days during travel. They're left food and water, and possibly the crinkle ball on the floor, or the battery-operated feather toy that shuts off automatically after ten minutes. The guilt adds up from leaving a pet alone unentertained. There are pet cameras to monitor these best friends while away and keep an eye on them to know they're safe, but it still leaves the pet bored. The solution to this is a toy that can be operated by the owner even while they're away.

Our goal with this project is to create a product that will leave pet owners guilt-free when leaving for days at a time while their pet stays home. The main objective in this is to satisfy the problem in every way possible. It's easy to create a laser pointer that can be monitored through your phone. But it's not as satisfying to play with a pet in that way. It's inactive and gets boring for the user as well. The way to fix this is to entertain both user and pet with an idea that combines fun with our initial goals.

The AutonoMouse was born from this idea, essentially a mouse shaped RC car that can be controlled through a phone or computer. The design is specifically shaped like a mouse to attract the predatory instincts of whatever pet may decide to play with it. It's an interactive way of playing with pets that will help any separation anxiety the owner might have being away from their pet.

1 Introduction

Many people own pets, and many of those pets are cats. These feline friends can be quite lovely companions, but they are also a hassle at times. With an innate instinct to hunt and test its abilities, a cat will get antsy if unable to perform. Usually, owners will get toys to play with and amuse the cat such as: string, feather on a stick, fake fish, and fake mice. While these are all nice, each one still requires some level of human interaction, movement is required to attract the animal's attention. Without someone being home, the cat has nothing to do.

Figure 2: Mouse Illustration

The AutonoMouse is a system that entertains a cat without the need for human interaction. It activates periodically throughout the day and travels about the room to entice the cat into interaction while the owner is away. Should the customer choose, it can also be controlled via web app.

The Mouse is a simple device with two wheels, a façade of a rodent to attract the cat. It is small, compact, and quick. The device will operate in three distinct modes:

1. Mode 1: The default mode. In this setting, the mouse travels around the room, avoiding obstacles in an attempt to get the cat's attention.

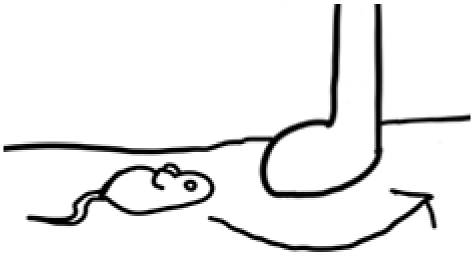


Figure 3: Mouse avoiding obstacle

2. Mode 2: When the battery is at low power, the mouse will automatically attempt to reach the charging station.

3. Mode 3: User Control Mode allows the customer to control the Mouse via a web app even when not at home. This allows them to play with the cat should they choose to use either a computer or phone. They can see what is going on by using the camera with a live feed.

Figure 4: Mouse Communicating with Box

The Box is a stationary section of the system. Its purpose is to charge and house the Mouse. It is plugged into a wall outlet for power so the device can run autonomously. While not activated, the Mouse component will remain inside the Box charging station, entering through a hole on the side. The Box will also act as an anchor or reference point for the Mouse to determine its position and navigate the surroundings.

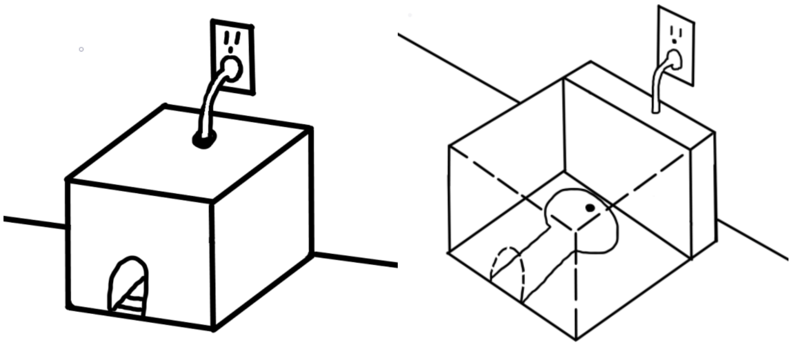


Figure 5: Simple illustration of outside and inside of the Box component

2. Project Description

2.1 Project Motivation and Goals

The motivation of this project is to create a functional autonomous cat toy using the engineering knowledge we have attained from attending the University of Central Florida. This device will test our skills in electrical and computer engineering as well as provide real world project, organization and time management skills that would not be attainable in a more confined lab environment.

2.2 Specifications

Specifications are requirements to be satisfied by a product, design, service, or material for it to be successful. This is a mix of engineering and marketing requirements that are made to meet the user's needs. This is what drives all the stages of development and what should be a vision of the design goals of the project. It is important for specifications to identify important details of the project while at the same time providing flexibility to develop innovative solutions. There are four important properties that should be considered when deciding the specifications for the project. It should be abstract, which means that the requirement should specify what the system will do but not how it will be implemented. This allows for flexibility when implementing a solution for the design. The condition should also be verifiable, which means that there should be a way to check that the requirement is accomplished in the final product. This is important because it allows for the final product to be tested and confirm that it is being built the right way. If a requirement is not verifiable, it should not be considered a requirement. Lastly, a requirement should be unambiguous and traceable so that it can be able to meet the user's needs. Elements like capability and condition should also be considered. Capability refers to what the final product must do, while conditions refer to measurable or testable attributes of the capability. It is also important to make sure that all the requirements or goals that are set are realistic and can be met. For this project, we have chosen the following specifications:

- The device has 3 different modes: Default, Find Box and User control.
- Box will recharge, 15 Watt, and give instructions to the Mouse, which can run for 2 hrs.
- The Mouse can rotate about its center axis 360 degrees
- The Box and Mouse must wirelessly communicate within a radius of 20ft
- The Box must be able to fully contain the Mouse unit.
- The Box dimensions must not exceed an area of 8 ft^3 and a weight of 30 lbs.
- The Mouse dimensions should not exceed 8 x 5 x 2.5 in and a weight of 2 lbs.
- The HD camera on Mouse will allow the user to see where they are going.
- The device is controllable via web app.
- The system can operate without human interaction.
- The device can entertain a common housecat.

2.3 Constraints

Constraints are also a different type of requirement which are caused by the environment or a stakeholder that impacts or limits the design. Sometimes constraints can violate the abstract property that was mentioned previously since they might specify how the system could be implemented. The following are the constraints for our project:

- Cost/Budget
- Construction Time
- Range that the mouse can travel
- Charging Time/Battery duration
- Motor power
- Size
- Weight

2.4 Software Block Diagram

Diagram 1 displays a general software logic diagram that shows how the three modes interact with each other, and how the User Web App and other hardware interact with the three modes. They are subject to change during the testing phase.

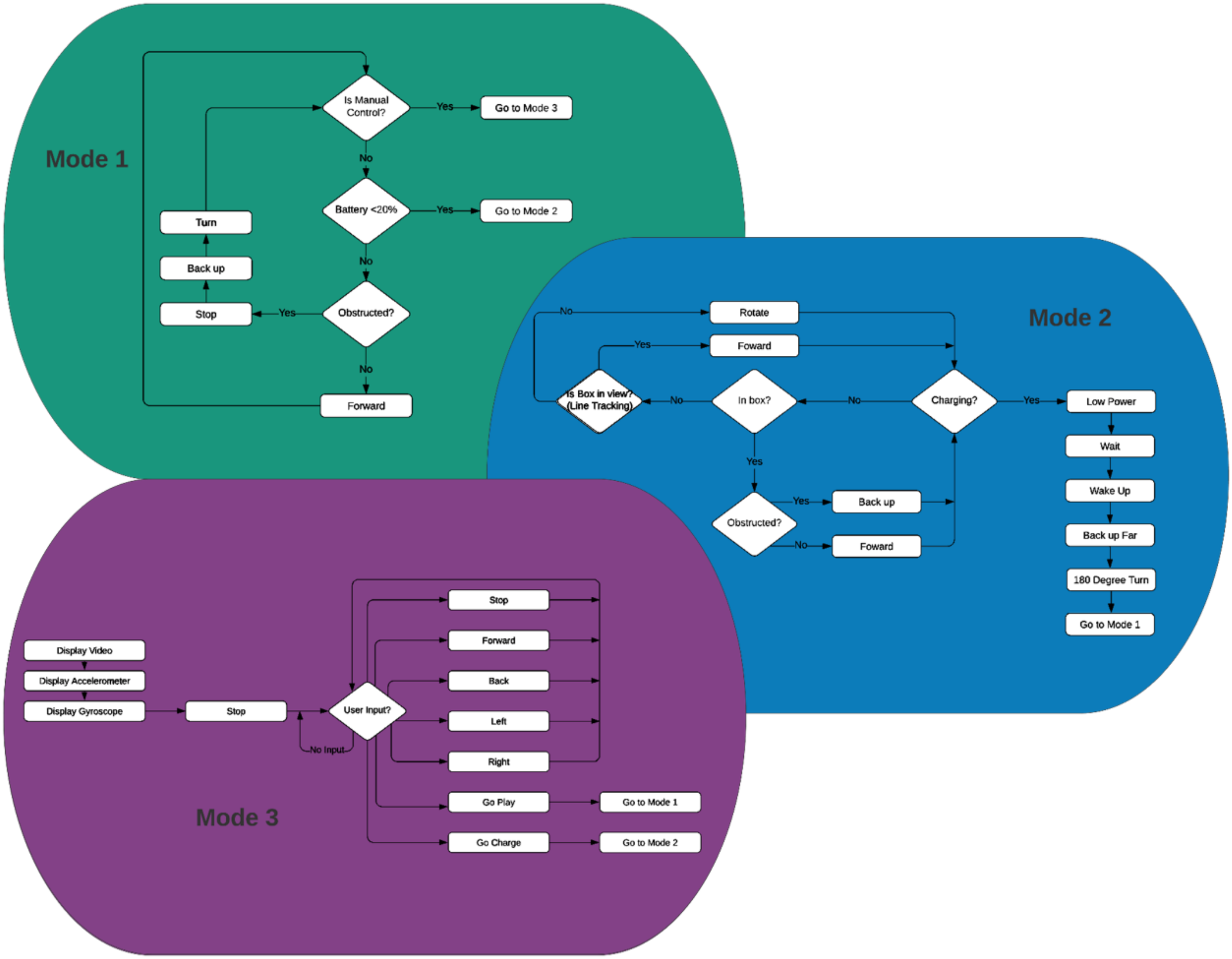


Figure 6: Software Block Diagram

2.5 Hardware Block Diagram

Figure 6 contains the required hardware, and some software routes involved in the project. The main objective of this diagram is to describe the communication between the box and the mouse, along with their respective components. To the right of the diagram is a legend with colored boxes to describe which person is designated to each task, along with a note to describe the terminology used in Figure 6.

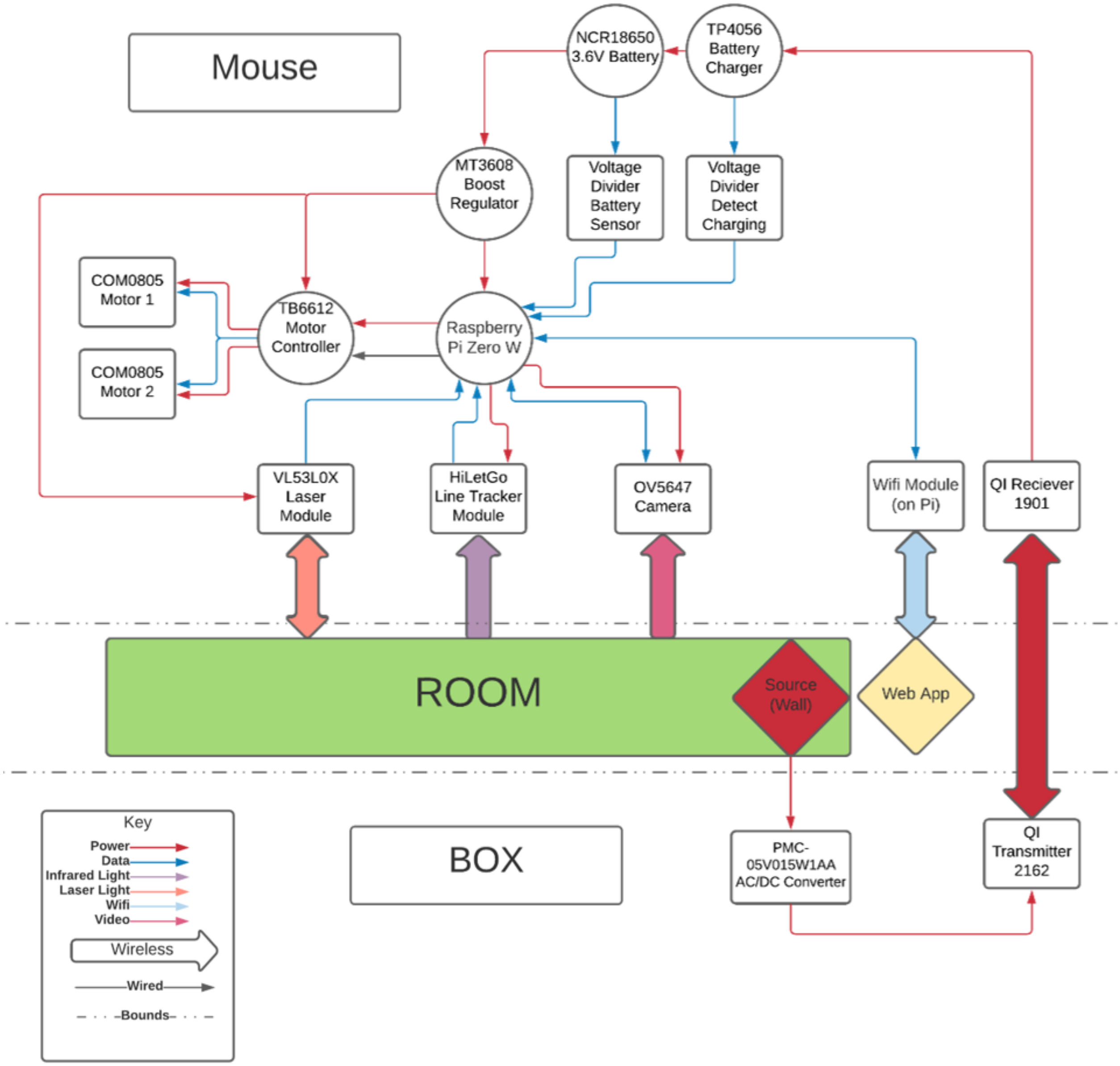


Figure 7: Hardware Block Diagram as of Senior Design I

2.6 House of Quality

HOQ, The House of Quality, is an excellent method for developing requirements. It is a design tool of quality function that identifies the customer's needs and their importance while correlating them with engineering requirements. Based on those correlations, objectives and priorities can be assigned. This process can be applied to a lot of systems that are in development. The output of this process looks like a matrix with the customer's needs in one dimension and the correlated requirements on the other dimension.

The HOQ is a part of a process called QFD, Quality Function Deployment. This focuses on the function of the execution of a quality plan and how to use resources to deploy that plan. They carefully listen to the voice of the customer and then respond to those needs and expectations. It is a structured method that identifies and prioritizes customers through the seven-management planning tool. Their methodology, besides the HOQ matrix, includes the most crucial product or service qualities.

The purpose of HOQ and QFD is to understand the customer's desires to find what they really need and provide an ideal solution. First, they need to understand the customer's priorities. This can be known after having an understanding of their needs and then breaking them down to find their priorities. Depending on how well the needs are accomplished is mainly how the customer will judge your solution. Sometimes there can be disagreements between the marketing and engineering on what is actually needed. In these cases, it is best to create a plan to decide all the priorities that both can agree on. The priorities of the customer should be ranked, and then a specific process and resource planning takes place.

The Engineering Requirements section displays engineering methods needed to measure and execute production. These address the technical needs that were created by marketing requirements to satisfy the customers need. These should be abstract, meaning they should explain what it does but not how it will be implemented. They should be verifiable, which means there has to be a way to measure and demonstrate that it meets the needs. These should be able to be traced to the original customers need. They should also be unambiguous and realistic.

The Marketing Requirements section to the left shows the priorities of the customer. They help express the customers wants and needs for the product or service. The Market Requirements describe the market opportunity, and they help understand market needs very fast before writing any details or investing a lot of resources. It is important to understand the problem we are trying to solve and what makes it unique. The size of the market and current alternatives should also be taken into account. Lastly, who you are solving the problem for and the functionality that must be included to solve their needs. Market Requirements can often be confused with Product Requirements, but it is not the same thing. While Market Requirements focus on the market opportunities, the Product Requirements focus on how the product will be built and to help teams understand what a product should do. Market Requirements should always be written before the Product Requirements.

In Table 1, we can see the technical needs of the project and the marketing requirements to meet the customer's needs. This helps us understand the tradeoffs between conflicting requirements. The "+" and "–" represent the polarity. The "+" being the goals we want to increase while "–" represents the ones we want to decrease. The "↑" represent a positive correlation, which means that both goals can be simultaneously met, while the "↓" represents a negative correlation where improving one will compromise the other goal. Finally, the targets for the engineering requirements are the final characteristics we would like to achieve when we build this project.

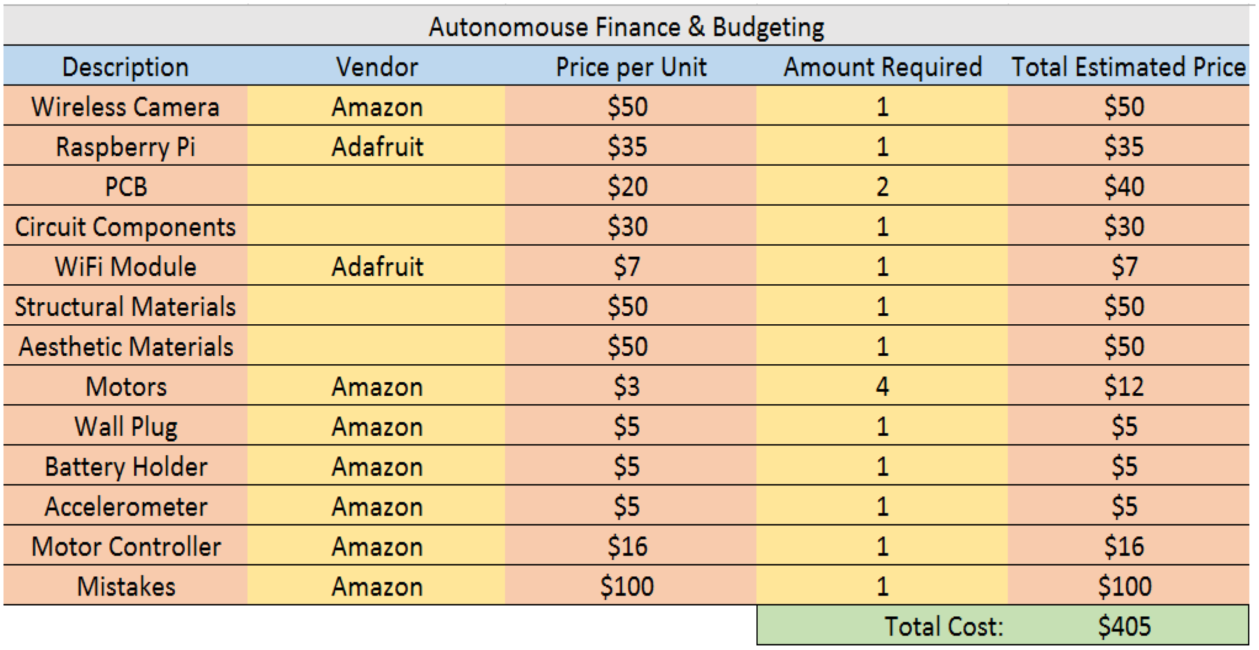
Table 1: House of Quality

↑ = Positive Correlation ↓ = Negative Correlation	Engineering Requirements	Size	Weight	Cost	Range	Battery Duration	Motor Power
Marketing Requirements		-	-	-	+	+	-
Size	-		↑			↑	↑
Weight	-	↑				↑	↑
Cost	-				↓	↓	↑
Range	+			↓			
Battery Duration	+	↑	↑	↓			↓
Motor Power	-	↑	↑	↑		↓	
Quiet	+			↓			
Stylish	+	↓	↓	↓			
Targets for Engineering Requirements		8 x 3 x 2.5 in	> 2 lbs.	$405.00	40 ft	4 hours	25 Watts

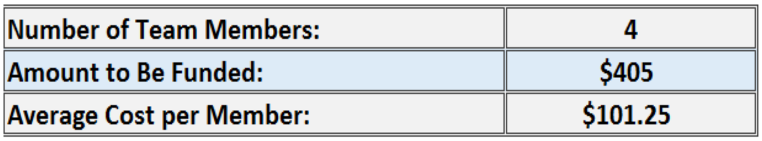
2.7 Finance and Budgeting

The following prices in Table 2 are a rough estimate of the average cost of these components on the market. Many calculations were rounded up in order to have some leeway with the budget. In an ideal situation, the total cost would be $20 - $40 lower than the amount shown. The prices are subject to change as the project continues, and innovations arise.

Table 2: Planed Finance and Budgeting

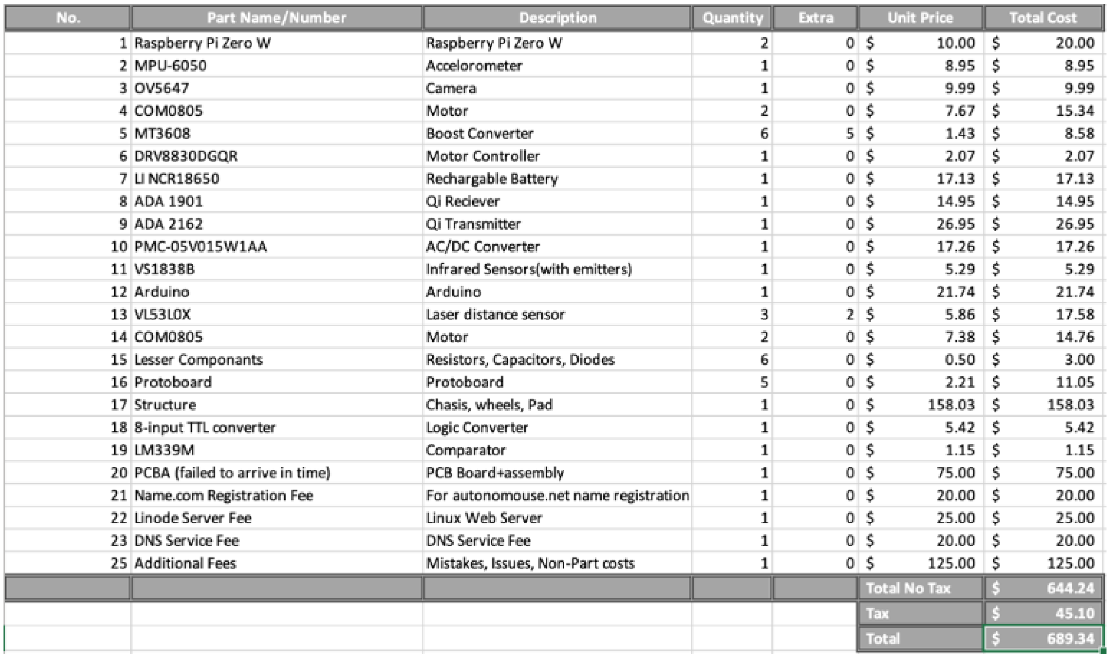


Autonomouse Finance & Budgeting				
Description	Vendor	Price per Unit	Amount Required	Total Estimated Price
Wireless Camera	Amazon	$50	1	$50
Raspberry Pi	Adafruit	$35	1	$35
PCB		$20	2	$40
Circuit Components		$30	1	$30
WiFi Module	Adafruit	$7	1	$7
Structural Materials		$50	1	$50
Aesthetic Materials		$50	1	$50
Motors	Amazon	$3	4	$12
Wall Plug	Amazon	$5	1	$5
Battery Holder	Amazon	$5	1	$5
Accelerometer	Amazon	$5	1	$5
Motor Controller	Amazon	$16	1	$16
Mistakes	Amazon	$100	1	$100
			Total Cost:	$405



Number of Team Members:	**4**
Amount to Be Funded:	**$405**
Average Cost per Member:	**$101.25**

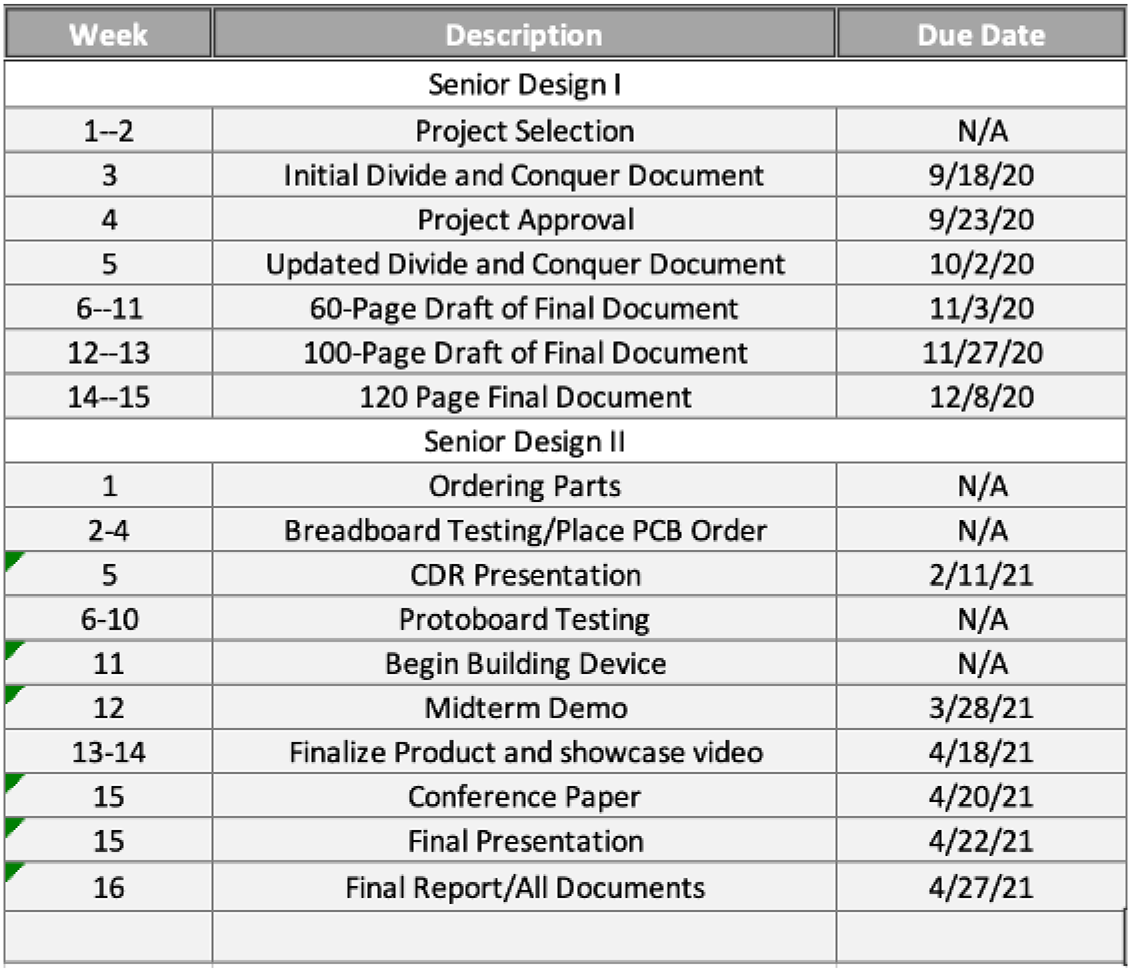
Actual Cost of the Project:



No.	Part Name/Number	Description	Quantity	Extra	Unit Price	Total Cost
1	Raspberry Pi Zero W	Raspberry Pi Zero W	2	0	$ 10.00	$ 20.00
2	MPU-6050	Accelorometer	1	0	$ 8.95	$ 8.95
3	OV5647	Camera	1	0	$ 9.99	$ 9.99
4	COM0805	Motor	2	0	$ 7.67	$ 15.34
5	MT3608	Boost Converter	6	5	$ 1.43	$ 8.58
6	DRV8830DGQR	Motor Controller	1	0	$ 2.07	$ 2.07
7	LI NCR18650	Rechargable Battery	1	0	$ 17.13	$ 17.13
8	ADA 1901	Qi Reciever	1	0	$ 14.95	$ 14.95
9	ADA 2162	Qi Transmitter	1	0	$ 26.95	$ 26.95
10	PMC-05V015W1AA	AC/DC Converter	1	0	$ 17.26	$ 17.26
11	VS1838B	Infrared Sensors(with emitters)	1	0	$ 5.29	$ 5.29
12	Arduino	Arduino	1	0	$ 21.74	$ 21.74
13	VL53L0X	Laser distance sensor	3	2	$ 5.86	$ 17.58
14	COM0805	Motor	2	0	$ 7.38	$ 14.76
15	Lesser Componants	Resistors, Capacitors, Diodes	6	0	$ 0.50	$ 3.00
16	Protoboard	Protoboard	5	0	$ 2.21	$ 11.05
17	Structure	Chasis, wheels, Pad	1	0	$ 158.03	$ 158.03
18	8-input TTL converter	Logic Converter	1	0	$ 5.42	$ 5.42
19	LM339M	Comparator	1	0	$ 1.15	$ 1.15
20	PCBA (failed to arrive in time)	PCB Board+assembly	1	0	$ 75.00	$ 75.00
21	Name.com Registration Fee	For autonomouse.net name registration	1	0	$ 20.00	$ 20.00
22	Linode Server Fee	Linux Web Server	1	0	$ 25.00	$ 25.00
23	DNS Service Fee	DNS Service Fee	1	0	$ 20.00	$ 20.00
25	Additional Fees	Mistakes, Issues, Non-Part costs	1	0	$ 125.00	$ 125.00
					Total No Tax	$ 644.24
					Tax	$ 45.10
					Total	$ 689.34

2.8 Milestones

Table 3 shows the general milestones and deadlines we hope to achieve with this project. No definite dates have been dedicated for building yet, but will be changed accordingly. Since Senior Design II will be more in-person heavy, online meeting will be held in conjunction with in-person meetings to build our project. However, if this is not a possibility, different project members may take on separate parts of the project on their own, and then plan to join their parts with the rest during the latter half of the building process. Currently for Senior Design I, weekly meetings are held to discuss goals and progress, along with any changes to plans. All opinions are shared on the project to best adapt to each possible situation, allowing for efficient use of each other’s time. Our main goal is to finalize our prototype early and add extra possible modifications with the remaining time.



Week	Description	Due Date
Senior Design I		
1--2	Project Selection	N/A
3	Initial Divide and Conquer Document	9/18/20
4	Project Approval	9/23/20
5	Updated Divide and Conquer Document	10/2/20
6--11	60-Page Draft of Final Document	11/3/20
12--13	100-Page Draft of Final Document	11/27/20
14--15	120 Page Final Document	12/8/20
Senior Design II		
1	Ordering Parts	N/A
2-4	Breadboard Testing/Place PCB Order	N/A
5	CDR Presentation	2/11/21
6-10	Protoboard Testing	N/A
11	Begin Building Device	N/A
12	Midterm Demo	3/28/21
13-14	Finalize Product and showcase video	4/18/21
15	Conference Paper	4/20/21
15	Final Presentation	4/22/21
16	Final Report/All Documents	4/27/21

Table 3: Milestones

3. Research Related to Project Description and Part Selection

3.1 Existing Similar Projects and Products

Robotics has always been a popular research topic; it combines computer science and engineering. There are many robotic research projects in various areas, such as medical, logistics, manufacturing, aerospace and etc. Robotic toys have been more and more popular in the recent years as technology advances, and the prices are becoming more acceptable for the general public. This is good for anyone that wants to study and having hands on experiences with the resources that are available at a reasonable price. There are also many similar products in the market with different designs and implementations, there are robotic toys for kids, adults, pets and etc. The following are a few examples that have some similarities in technology with our project, but different designs and implementations.

3.1.1 Devastator Tank Mobile Robot Platform

This is the Devastator Tank Mobile Robot Platform (Figure 8), a product developed by DFRobot. Tt is battery operated and fully compatible with popular microcontrollers on the market, such as Arduino, Raspberry Pi, Lattepanda and so on. It is sold as a kit with high speed motors, however the microcontroller and other sensors are not included the kit that is up to the user to customize accordingly. They offer instructions and sample codes for users to configure and assemble everything together. Depending on the configuration, it can do different things such as moving, changing directions, adding a camera for real time vision, etc. It can also be controlled by different platforms depending on the customization on configuration.

Figure 8: Devastator Tank Mobile Robot Platform by DFRobot (Amazon.com)

In our project, we will be using similar products such as a microcontroller, motors, various sensors, camera and etc. We will also implement a wireless charging station, which is different from this product.

3.1.2 SunFounder Smart Video Car Kit V2.0 PiCar-V Robot Kit

Figure 9: SunFounder Smart Video Car Kit (Amazon.com)

The SunFounder smart video car kit (Figurer 9) is similar to the Devastator platform, however, this version is only compatible with Raspberry Pi and it is not included in the kit. This kit also comes with a camera for real-time video transmission. Also depending on user’s configuration, it can move according to the controlling method.

3.1.3. JOEJOY interactive robotic cat toy

Figure 10: JOEJOY interactive robotic cat toy (Amazon.com)

The JOEJOY interactive robotic cat toy (Figure 10) is a 360 Degree self-rotating moving ball. It has 2 modes, automatic and manual. In manual mode, the user can interact with the cat by using a remote controlling the toy to move up, down, left or right. In automatic mode, the motion and speed are randomized, and it will automatically shut down after 30 minutes of work to save energy.

Our project has the same goal as this product, that is, to entertain a cat. Furthermore, we also want the user to be able to interact with the cat remotely over Wi-Fi while the user is away. Instead of using a manual controller, we will be implementing a Web Application, so that the user can use any mobile or computer platform to send instructions to the toy so that it can interact with the cat as if the owner is home.

3.2 Relevant Technologies

3.2.1 ARM vs. Intel x86 processors

ARM uses RISC (Reduced Instruction Set Computing) architecture, and x86 uses CISC (Complex Instruction Set Computing) architecture. ARM vs. Intel x86 is essentially RISC vs. CISC. As the name suggests, RICS only uses simple instructions that can be executed within one clock cycle, data has to be loaded into registers to perform an operation and then store it in memory, it cannot do operations from memory to memory, thus it requires more lines of code to complete a task in comparison to CISC, whereas CISC uses as few lines of assembly as possible, but this requires hardware that’s capable of understanding and executing the tasks. RISC also needs more RAM to store the assembly level instructions, and the compiler must also perform more work to translate high-level language. However, RISC require less transistors of hardware space than CISC, and all instructions execute in a uniform amount of time, it makes pipelining possible. Also since RISC requires fewer transistors, it in turn improves cost, power consumption, and heat dispassion. On the other hand, CISC offers many more instructions, and can execute multiple operations, thus it leads to better performance, but causes more power consumption. This explains why most mobile categories such as smartphones, laptops, tablets, and other embedded systems are dominated by ARM and most PCs, workstation computers and cloud computing are using x86, although ARM processors are also making their way into the PC market.

3.2.2 Microcontrollers

A microcontroller is a small computer on a integrated circuit (IC) chip. Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. A typical microcontroller includes a processor, memory and input/output (I/O) peripherals on a single chip. The following are a few popular microcontrollers on the market.

3.2.2.1 STM32F103C8T6

The STM32F103C8T6 (Figure 11) features a high performance ARM Cortex-M3 32-bit RISC core operating at a 72 MHz, an extensive range of enhanced I/Os and peripherals connected to two

APB buses, 64 or 128KB of flash memory, 20 KB of SRAM, 7 timers, and up to 9 communication interfaces such as USART, SPIs, CAN, and USB.

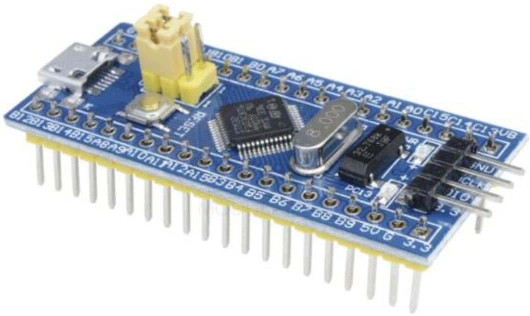


Figure 11: STM32F103C8T6

3.2.2.2 ATmega328

The ATmega328p (Figure 12) is good for people wo want to avoid the bulkiness of the Arduino boards, but still have access to the programming support, and other features associated with the Arduino development platform. It's an 8-bit AVR microcontroller based on RISC architecture. It features 32KB of flash memory, 2 KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, 3 flexible timer/counters with compare modes, internal and external interrupt, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, 6-channel 10-bit A/D converter.

Arduino is an open-source hardware/software programming platform based around Atmel microcontrollers. Arduino microcontrollers are pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory. It's equipped with sets of digital and analog I/O (input/output) pins that may be connected to other add-ons, it supports serial communication interfaces, some boards also have USB. There are many Arduino-compatible and Arduino-derived boards on the market, they differ in size, compatibility, processors and etc. Arduino and Arduino-compatible boards use printed circuit expansion boards called shields, which plug into the Arduino pin headers. Shields usually provide a dedicated function such as WIFI, Ethernet, motor control. Arduino boards have from 32 to 512K of flash memory that can be used for storage.

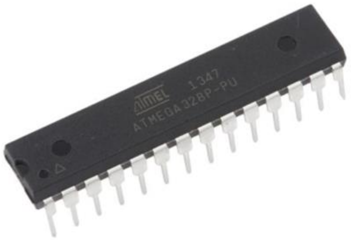


Figure 12: ATmega328

3.2.2.3 MSP430G2452

The MSP430G2452 (Figure 13) is a powerful, and relatively cheap microcontroller based on a 16-bit RISC CPU manufactured by Texas Instruments. This was board was used a lot in UCF laboratory studies. It features 8 comparator channels, 8 ADC channels, 16 GPIO's pins, 10-bit SAR, 1 timer, 8 KB of non-volatile memory, 256B SRAM, and supports 12C and SPI communication interfaces.

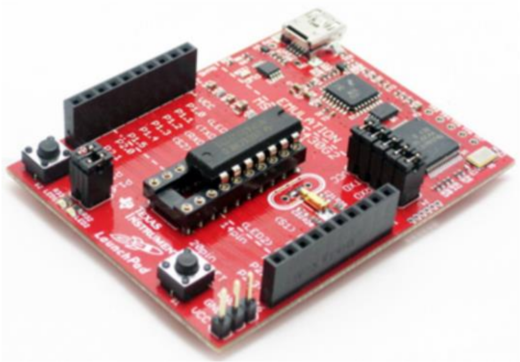


Figure 13: MSP430G2452

3.2.3 Raspberry Pi

Raspberry Pi is a credit card sized single-board computer. It has a Broadcom system on a chip (SoC) with integrated ARM-compatible central processing unit (CPU) and on-chip graphics processing unit (GPU). It is used in many applications such as tablets, robot control, lights, camera, and etc. It can do lots of tasks that a computer can do. Figure 14 displays a basic hardware layout of the Raspberry Pi.

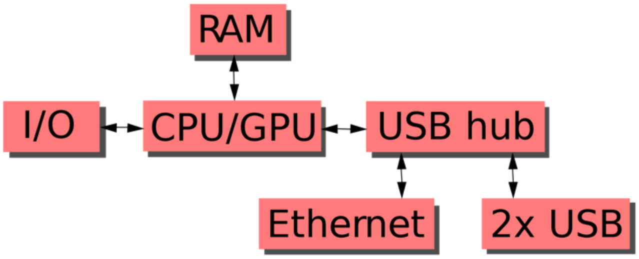


Figure 14: Raspberry Pi Basic Hardware Block Diagram

There are 5 generations in the Raspberry Pi family: Raspberry Pi, Raspberry Pi 2, Raspberry Pi Zero, Raspberry Pi 3, and Raspberry Pi 4. There are also different models within each generation. They differ in processing power, memory, size and weight, cost and input/output connectors. Model B has the "full size" boards which include Ethernet and USB ports. Model A are the square shaped boards with lower specifications and less USB ports than Model B, and no Ethernet, but at a lower price. Pi Zero (Figure 15) is the smallest in size, it has less processing power, no USB, no Ethernet, but very small, which can be very useful for a small smart toy.

Figure 15: Raspberry Pi Zero

3.2.4 Accelerometer

An accelerometer is a device that measures acceleration. The physics behind this device is similar to measuring the acceleration through a simple spring-mass system, where F = ma = kx, a = kx/m. Figure 16 displays a basic spring-mass system.

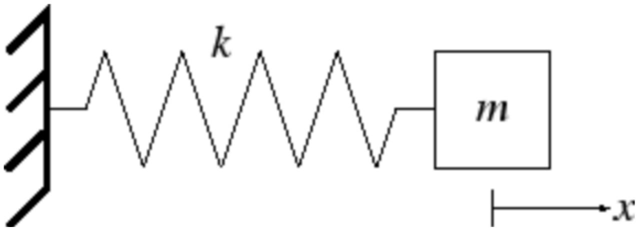


Figure 16: Spring-Mass System (source: CCRMA, Stanford University)

One method to sense the displacement x is by measuring the capacitance. The general equation for the capacitance of a parallel plate is C = ∑(A/d), where d is the distance between the plates. Let one plate to be the mass, and the other plate is fixed at its initial position. The capacitance across plates changes due to x, the displacement caused by acceleration a. By calculating C, displacement x can be found, which helps to infer the value of acceleration. Capacitance is easily measured electronically, but the spring-mass system is mechanical, so to achieve this on an Integrated Circuit (IC) level, a Micro Electro Mechanical System (MEMS) is implemented to keep the size small. Figure 17 displays the inside structure of a MEMS accelerometer.

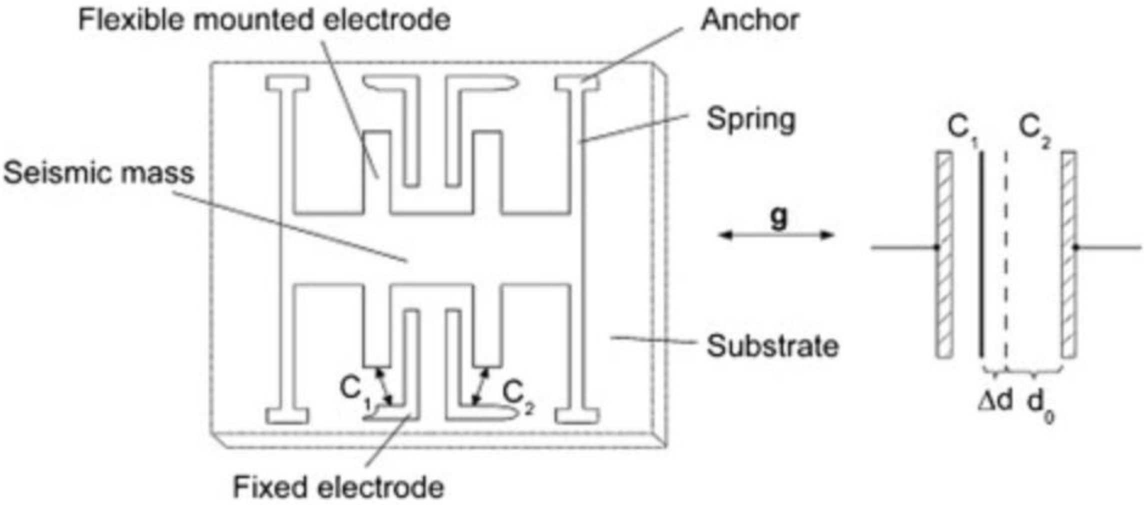


Figure 17: The structure of a MEMS accelerometer (source: M.N. Victorino, ... C. Menon, in Wearable Technology in Medicine and Health Care, 2018)

3.2.5 Wireless Charging

Wireless charging, also called inductive charging, is achieved by using an inductive coil to create an alternating electromagnetic field from the base, and with a second coil that is attached in the device, it then converts the energy back to current to charge the device. This is similar to how a transformer works, most of the wired small appliances in our homes work on relatively low voltage, so usually there’s a step-down transformer incorporated in the charging unit. In the inductive charging case, the primary coil is in the charging base, and the secondary coil and rechargeable battery are in the device.

3.2.5 Web framework

3.2.5.1 Web Browser

User's browser speaks http or https. The "s" version of this protocol uses certificate authorities and SSL certificates to encrypt the traffic. An HTTPS implementation of a project could use a cloud server such as Lets Encrypt, a free/public certificate authority (CA) that gives SSL certs to any Linux server that can verify they pointed a Domain Name Service (DNS) name to their server. The individual device would then connect to the cloud, which would also allow the user to run the device behind Network Address Translation (NAT) - the device can connect to the web server, and the user's device can communicate with the web server, so instructions could be sent as long as the user and the device both have internet access.

3.2.5.2 Web Server

The device's web server receives the web request. This may display the basic controls, show images, and provide web documents, but no interactivity at this level. The web server interprets

HTTP requests from the user's device and presents the links or controls that activate the device's behavior. Web servers include nginx or apache, both free and available for Linux.

3.2.5.3 Common Gateway Interface (CGI)

CGI defines a standard way in which information may be passed to and from the browser and server. the website executes CGI scripts which enable any code to run in php, java, or python based on user's inputs to the webpage. These CGI scripts are able to use Raspberry Pi's GPIO pins directly. If there are other pieces of software running on the device that manage the hardware, such as Robot Operating System (ROS), then the CGI scripts will not be able to exclusively access GPIO and hardware. Another middleware layer would be required. This makes the device more autonomous - ROS or the other software would be able to run without any user input, such as by timer, or as a reaction to sensors even when no users are connected. This middleware could listen to the CGI scripts over unix sockets for program-to-program communication, or monitor for a file being written to using inotify, which is a Linux faculty for low latency monitoring of filesystem changes. Figure 3.2.5.3 displays the process of using CGI.

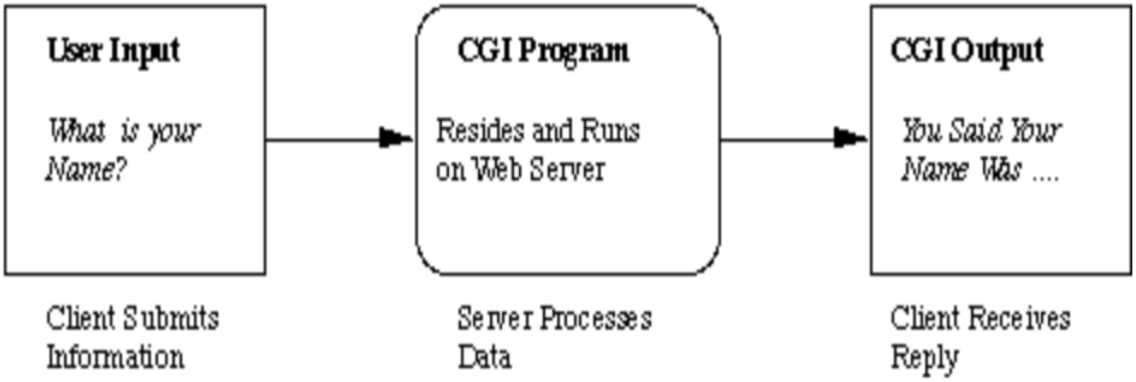


Figure 18: The Common Gateway Interface (source: dave@cs.cf.ac.uk)

3.3 Part Selection

3.3.1 Raspberry Pi

Figure 19: Raspberry Pi 4 Model B

The Raspberry Pi (Figure 19) is a complete computer built on a single board with many features required of a functional computer. These include microprocessors, memory, and input/output. It is widely used for many things like learning to code, building electronic projects, industrial automation, home automation, and commercial products. It has become a popular and economical alternative to more expensive commercial solutions due to its relatively low cost and high portability. The Raspberry Pi models that could be considered for this project are Raspberry Pi Zero W, Raspberry Pi 3 Model B+, and Raspberry Pi 4 Model B.

HARDWARE: The hardware interface of the Raspberry Pi depends on the model that is being used. It can have 40 or 26 dedicated interface pins, including a UART, an I2C bus, I2S audio, an SPI bus with two chip selects, 3v3, 5V, and ground. Through the SPI bus or the I2C, the maximum number of GPIOs can theoretically be indefinitely expanded.

PROCESSOR: The Raspberry Pi uses Broadcom SoC. A standard PC architecture that separates components depending on their function and connects them through a central interface circuit board. An SoC has a single integrated circuit that integrates all these functions and parts as if they were built into the motherboard. These include CPU, RAM, ROM, hard disk, USB connectivity, graphics, and memory interfaces.

RAM: The RAM on the Raspberry Pi depends on the model you decide to get. The earlier models only included 256 MB and then later 512 MB. The Raspberry Pi 4 model gives you the option to choose between 2 GB, 4GB, and 8 GB of RAM, but it cannot be upgraded after purchase, unlike the previous models.

PERFORMANCE: The Raspberry Pi 3 and all previous models have a GPU that provides 1080p30 H.264 high-profile encode and decode, OpenGL ES 2.0, and hardware-accelerated OpenVG. The Raspberry Pi and Raspberry Pi 4 use the VideoCore VI. It is a multimedia processor with two-dimensional DSP architecture that makes it efficient and flexible to decode and encode multimedia codecs in software while maintaining low power usage. It is faster than the ones used on previous models and has a PCIe that provides USB 2.0 and USB 3.0 functionality and overall provides a better performance. The Raspberry Pi can also be overclocked to 800 MHz and some to 1000 MHz.

POWER: The Raspberry Pi needs to be powered with a 5V power supply through a USB connector. The amount of current (mA) needed will depend on the model that is being used and what it is hooked up to.

SOFTWARE: The primary operating system for the Raspberry Pi is the Raspberry Pi OS. It is a Debian GNU/Linux based operating system, specifically designed and continuously being optimized for the Raspberry Pi line of compact single-board computers. For the desktop environment, it uses a modified LXDE with the Openbox stacking window manager.

NETWORKING: When it comes to Networking, the Raspberry Pi 1 Model B and B+, Raspberry Pi 2, Raspberry Pi 3 Model B and B+, and Raspberry Pi 4 all support wired ethernet. The Raspberry Pi 3,3+,4, and Raspberry Pi Zero W all have built-in wireless LAN connectivity, and a USB wireless LAN dongle can be added to any Raspberry Pi model. The Raspberry Pi 3, Raspberry Pi 4, and Raspberry Pi Zero W all have built-in Bluetooth.

STORAGE: Additional storage or SD cards can be used to have more space to install additional packages and make programs. The minimum size recommended for the SD card is 8GB. A USB stick or a USB hard drive can also be used for extra storage.

PERIPHERALS: Through the various connectors and pins, peripherals can be attached. Any device or component with USB capabilities can be used if the drivers are installed. Since the Raspberry Pi is configured to operate as a headless computer, it is possible to connect a USB computer keyboard and mouse to use the device.

CAMERA MODULE: The Raspberry Pi has camera modules to allow connectivity for a camera that can capture still images or video recordings. They are small PCBs that connect to the CSI-2 camera port by using a short ribbon cable. They connect to the Image System Pipeline in the SoC, where the entering camera data is processed and then turned into an image or video on the SD card or other storage.

VIDEO: The Raspberry Pi has a composite and HDMI port that supports the CEC Standard. The Raspberry Pi 4 Model B supports dual monitor up to 4k resolution through a micro HDMI.

AUDIO: There is a standard 3.5mm jack for audio out to an amplifier. Sound over HDMI and any supported USB microphone for audio will also work. The I2S interface can also be used for additional audio I/O.

Raspberry Pi Zero W

The Raspberry Pi Zero W is very good for tasks like driving lights, motors, or cameras. It is very small (2.6 in x 1.2 in x 0.2 in) and weighs 9g. It has a 1 GHz, single-core CPU, and 512 MB of RAM. It has a micro USB port but does not have a full-size USB port, but an adapter can be used. The GPIO pins are not attached, and they need to be bought separately and soldered. There is no ethernet, but it has built-in wireless LAN and built-in Bluetooth. The recommended PSU current capacity is 1.2 A and the typical bare-board active current consumption is 150 mA. The maximum total USB peripheral current draw is limited by PSU, board, and connector ratings. Out of the three models being considered, this one is the cheapest.

Raspberry Pi 3 Model B+

The Raspberry Pi Model B+ is slower and does not have a lot of the features that the Raspberry Pi 4 Model B has. It has a 1.4 GHz, quad-core processor, and it has 1GB of RAM. It has 1000BaseT wired ethernet, but the throughput is limited by its USB 2.0 connection to the SoC. It has a built-in wireless LAN and Bluetooth. It has 4 USB ports and HDMI ports. The recommended PSU current capacity is 2.5 A, the maximum total USB peripheral current draw is 1.2 A, and the typical bare-board active current consumption is 500mA. Compared to the Raspberry Pi Zero, it is more expensive. Compared to the Raspberry Pi 4 Model B, it can be cheaper, and since it uses less power, it generates less heat. However, it does not have a USB 3.0 and dual- monitor support.

Raspberry Pi 4 Model B

The Raspberry Pi 4 Model B is the latest and fastest model. It is powered by a 1.5 GHz, quad-core processor. You are able to choose between 2GB, 4GB, or 8GB of RAM, which is not possible with the other models. It is suitable for multitasking, programming, surfing the web, and having a lot of tabs open. It is very silent and energy-efficient. It has two USB 2.0 and two USB 3.0 ports and dual-HDMI out for multi-monitor support. They have a 1000BaseT wired ethernet, built-in wireless LAN connectivity, and built-in Bluetooth. The recommended PSU current capacity is 3.0 A, the maximum total USB peripheral current draw is 1.2 A, and the typical bare-board active current is 600mA. Compared to the other previously mentioned models, it is the most expensive one depending on how much RAM is chosen.

3.3.2 Wi-Fi Modules

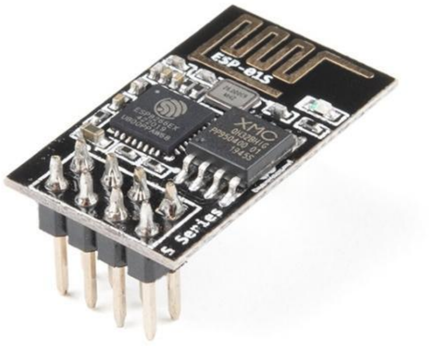


Figure 20: ESP8266

A Wi-Fi module simplifies the connectivity design process. The architecture can be a single solution where the MCU runs the Wi-Fi stack and host application in one chip. It can also be a host processor with Wi-Fi module solution where the wireless connectivity contains the Wi-Fi stack, and a separate processor runs the host application. Most Wi-Fi modules have I/O and peripheral interface support to suit different needs. They also support at least one of the various security standards like WPA, WPA2, WPA3, WPS, and others. The Wi-Fi modules that could be considered for this project are the ESP8266, ESP8285 and the ESP32.

ESP8266

The ESP8266 (Figure 3.3.2), is a self-contained SoC Wi-Fi Module with an integrated TCP/IP protocol stack that can give any microcontroller access to Wi-Fi network. It can be used to carry software application. It is also capable of offloading all Wi-Fi networking functions from another application processor. The ESP8266 module already comes programmed with an AT command set firmware. It has a built-in cache memory to reduce memory requirements for a better system performance. Through the GPIO port sensors and application-specific devices, the module on-board processing and storage capability can be integrated with minimal development and loading during runtime. Its highly integrated chip allows minimal external circuitry, which includes the front-end module designed to minimize the space occupied by PCB. Among the features of the ESP8266, it has APSD for VoIP applications and Bluetooth coexistence interfaces. It also has a self-calibrated RF that allows it to work under all operating conditions without requiring external RF parts. The ESP8266, specifically for mobile devices, wearable electronics, and networking applications design, can achieve the lowest energy consumption, together with many other patented technologies.

ESP8285

The ESP8285 is a highly integrated SoC Wi-Fi module that provides efficient power usage, compact design, and reliable performance. Designed with a lot of power management technologies, it can be used on mobile devices, wearable electronics, and applications. It can perform as a standalone application or as a slave to a host MCU. The cache is very fast and can increase the system's performance and optimize its memory. The processor has a maximum clock speed of 160 MHz and has very low power consumption. The CPU integrates the interfaces iBus, dBus, and AHB. The memory can be accessed through this interface. It integrates memory controller and memory units, including SRAM and ROM. The high-frequency clock is generated from internal and external crystal oscillator, which can have a frequency of 24 MHz to 52 MHz. ESP8285 implements TCP/IP and full 802.11b, 802.11g, and 802.11n, WLAN MAC protocol. The ESP8285 had 17 GPIO pins, which can be assigned to many functions by programming the appropriate registers. They can be multiplexed with other functions and can be configured internal pull-up, pull-down, or set to high impedance. It has one slave SDIO, two SPIs, one I2C, one I2S, two UART interfaces (UART0 and UART1), four PWM output, one infrared remote control, and a 10-bit precision SAR ADC. It integrates antenna switches, power amplifiers, low noise receive amplifiers, filters, and power management modules. It has a 1 MiB of built-in flash, which permits that devices with single-chip are capable of connecting to Wi-Fi.

ESP32

The ESP32 is a single 2.4 GHz Wi-Fi module and Bluetooth combo chip that integrates TSMC ultra-low power technology. It is designed to achieve the best power, versatility, reliability, and performance in various applications. ESP32 is designed for mobile devices, wearable electronics, and internet applications. It features low power chips, multiple power modes, and dynamic power scaling. ESP32 is a highly integrated solution for Wi-Fi and Bluetooth applications. It has around 20 external components and occupies minimal PCB area. The CMOS single-chip fully integrated radio and baseband allows it to remove external circuit imperfections or adjust changes in external conditions through advanced calibration circuitries. The operating voltage of the ESP32 ranges from 2.3 V to 3.6 V, but when using a single power supply, it is recommended to use 3.3 V. The recommended output current is 500 mA or higher. It has more GPIO pins than the ESP8266, and you can decide which ones are UART, I2C, or SPI through the chip multiplexing feature, which allows for multiple functions to be assigned to the same pin. It also contains more functions than the ESP8266, but it can be more expensive.

3.3.3 Micro Camera

Figure 21: Camera Module V2

A micro camera is a still or video camera that can be built into commonly used objects. They can be wired, where the camera will be connected by cable to a viewing or recording device, or wireless, where the camera can transmit the video signal to a receiver within a small radius where it can either be viewed or recorded. They might also send the video online so that it can be viewed remotely or transmit it to a receiver where it can either be recorded or viewed live. Some models that could be considered are ESP32-CAM, Camera Module VI, Camera Module V2, Camera V2 NoIR, ZeroCam FishEye, and ZeroCam NoIR.

ESP32-CAM

The ESP32-CAM is a small-size camera module that can be widely used in various applications. The ESP32-CAM is similar to the ESP32 module mentioned previously, but it has fewer I/O pins since most GPIO pins are used internally for the camera and the microSD card port. It also does not have a USB port, so an FTDI adapter will be required to program the device. Besides the two differences mentioned, all the other specifications are the same as the ESP32 module. It contains UART, SPI, I2C, and PWM interfaces. The adapter can be set for 3.3 volts using the 3.3-volt power pin or 5 volts using the 5-volt power pin. It includes an OV2640 camera module, which includes 2-megapixel sensor, array size UXGA 1622x1200, and image transfer rate of 15 to 60fps. The device can also support an OV7670 camera. Other specifications include up to 160 MHz of clock speed, 520 KB of SRAM, and 4 MB of PSRAM, and 9 GPIO ports. The Wi-Fi signal strength is sometimes not so great external antenna might be necessary as you move further away from the router or access point.

Camera Module V1 (OV5647)

The OV5647 is OmniVisions CMOS image sensor. It can deliver 5-megapixel photography and videos at a high frame rate 720p/80(HD). The camera module size is 8.5 mm x 8.5 mm x <= 5mm. The OV5647 is built on 1.4-micron OmniBSI backside illumination pixel architecture. It offers a lot of benefits compared to front side illumination technology. The OmniVision CMOS image sensor can reduce or eliminate lightning sources of image contamination to create a clean and stable color image. The low power OV5647 a two-lane MIPI interface and digital video parallel port.

Camera Module V2 (IMX219)

The Raspberry Pi Camera Module V2 (Figure 3.3.3) can be used to take photographs and high definition videos. It has a Sony IMX219 8-megapixel sensor with a resolution of 3280 x 2464 pixels and 3.68x 2.76 mm if image area. Its video modes are 30 FPS at 1080p, 60FPS at 720p and 90 FPS at 480p. It can be used for many things like CCTV, home security and surveillance, robotics, toys, motion detection. This camera works with all models of Raspberry Pi 1, 2,3, and 4.

Camera V2 NoIR

The Raspberry Pi Camera Module V2 NoIR is a no infrared camera module which does not possess the ability to see in the dark. It has a Sony IMX219 8-megapixel sensor, and it's pretty much the same as Camera Module V2 but without the infrared filter. It also works with all models of Raspberry Pi 1,2,3, and 4.

ZeroCam FishEye

ZeroCam FishEye is a small camera module for the Raspberry Pi Zero or Raspberry Pi Zero W, but by using an adapter cable, it can be used on models of Raspberry Pi 1,2 and 3. It has a 5-megapixel sensor with a resolution of 2592x1944 pixels. Its video modes are 30 FPS at 1080p, 60FPS at 720p and 90 FPS at 480p. The dimensions of the circuit and camera are 60 mm x 11.4 mm x 5.1 mm.

ZeroCam NoIR

ZeroCam FishEye is a small camera module for the Raspberry Pi Zero or Raspberry Pi Zero W, but by using an adapter cable, it can be used on models of Raspberry Pi 1,2 and 3. The ZeroCam NoIR has the same specifications in resolution, video modes, and dimensions as the ZeroCam FishEye. The only difference is that the Zero Noir camera has no InfraRed filter.

3.3.4 Chassis (Supporting Structure)

Figure 22: Aluminum 2WD TT Robot/Motor Chassis

The chassis (Figure 22) is the supporting structure of all the components. The chassis must be capable of supporting static and dynamics loads that will be applied during the operation. The rigidity of the chassis is one important parameter that must be considered. The chassis geometry and the construction materials properties determine the rigidity of the chassis. For this application, three materials are being considered, plastic, carbon fiber, and aluminum. Each material has different advantages and disadvantages. Plastic is light weighted and low-cost, but it has a low load capacity. Carbon fiber is light weighted and has a high load capacity, but it has a high cost. Aluminum is light weighted, has a high load capacity, and has an intermediate cost.

The relative rigidity of different material is established by comparing their Young's modulus (E). The Young's modulus is defined of the quotient of the stress over the strain. The stress (σ) is defined as the force (F, Newtons - N), divided by the cross-sectional area (A, Area – m^2).

$$\sigma = \frac{F}{A}\left(\frac{N}{m^2} = Pascal\ (Pa)\right)$$

The strain (ε) is defined as the quotient elongation (L-L_0) divided by L_0, where L_0 is the length before being deformed and L is the length after deformation. The units of strain are meter over meter or 1 (dimensionless).

$$\varepsilon = \frac{L - L_0}{L}\left(\frac{m}{m} = 1\right)$$

The Young's module is defined as:

$$E = \frac{\sigma}{\varepsilon}\left(\frac{\frac{N}{m^2}}{\frac{m}{m}} = \frac{N}{m^2} = Pa\right)$$

For the Young's modulus of the material considered in this project are 69 GPa for aluminum, 228 GPa for carbon fiber, and 2 GPa for plastic.

It is worth mentioning that when materials are being deformed, they go through two phases of deformation. The initial is considered elastic deformation, in this stage if the force causing the deformation is removed the material returns to its original form. The second stage is plastic deformation, in this stage the material will not return to its original form once the force is removed. Most materials go through the two deformation phases before failure. From the three material that are considered for this project aluminum and plastic have plastic zone, carbon fiber does not.

The rigidity of the chassis can also be increased by geometry. The geometry of the chassis will establish the moment of inertia (I). The moment inertia is resistance of a structural element to bending. The definition of the moment of inertia of a rectangular element is define as:

$$I = \frac{1}{12} bh^3$$

Where b is the width, and h is the height of the rectangular element. Although, it would be impractical to increase the entire chassis section height to increase the moment of inertia. For this reason, chassis stiffeners are used, which are structural elements that are attached to the chassis. The chassis stiffeners are elements that have a small width and a large height, giving them a large moment of inertia. The stiffeners also will increase the ability of the chassis to endure being drop or being struck.

The weight of the material is determined mostly by the volume of the material and the density of the material. For this project the density of the materials considered are 2.7 g/cm^3 for Aluminum, 1.93 g/cm^3 for Carbon fiber and 0.92 g/cm^3 for plastic.

An analysis for each of the material will be performed to determine which material is better suited for this project. The parameters that were discussed on this section will be considered to achieve the most efficient chassis design that meets all the requirements of the project.

3.3.5 Tail/External Fur Coating

Figure 23: Hexbug Mouse Robotic Cat Toy

When it comes to making a product, the appearance can influence the consumer’s choice. In this case, by adding the tail and the external fur coating, we can make the mouse look more realistic. This would not only get the attention of the consumer to buy the product, but it would also capture the attention and interest of the cat. Using a soft texture to cover the mouse will make it better and safer for the cat to play with. It is important to ensure the final product is safe. A lot of factors can contribute to this, and they all depend on the cat's size, activity level, and preference. This includes making sure there aren't any loose parts that could be ingested by the cat. It is also important to make sure the material that is used to make the tail and the fur coating is not toxic, has any chemicals or fillings that could potentially be dangerous for the cat. The Hexbug Mouse Robotic Cat Toy (Figure 23) is pretty similar to what we are trying to achieve. It looks realistic and the tail and ears are made out of soft rubber which makes it safe for the cat.

3.3.6: Accelerometer

The accelerometer is a device that measures acceleration. It is commonly designed using capacitive plates on very small springs. When motion occurs, the springs move causing a change in capacitance. These changes are quantifiable and are used to measure the acceleration the machine is experiencing. For accelerometers, the sensitivity is measured in mV/g where g is the acceleration due to gravity, 9.81m/s. Digital accelerometers will use LSB/g, least significant bit. Accelerometers used to only measure in one direction, but multidirectional systems are now much more common. These instruments are commonly found in many appliances, including some unexpected systems such as computers.

The purpose of the accelerometer will be to regulate the speed of the Mouse unit and to determine when the device is not moving or any sudden changes in speed. This will enable the device to detect when the cat has interacted with it and act accordingly. It can in theory also be used for simplistic navigation around objects, however this will more likely be left to the distance detection system.

MMA8451QR1

This accelerometer has a 1.95 V to 3.6 V supply voltage and interface voltage. It has triple axis, x,y,z capability with ±2g/±4g/±8g dynamically selectable full scale. The sensitivity can be from 4096(±2g) to 1024(±8g) LSB/g. The Output data rates are from 1.56 Hz to 400 Hz. The output is I2C and can be from 14-bit and 8-bit. It features three channels of motion detection: motion detection, pulse detection and jolt detection. It has a sleep mode. And a current consumption of 6 to 165 µA. The price of this device is $3.04.

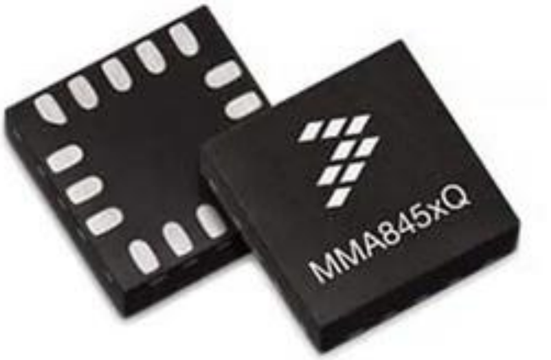


Figure 24: MMA8652FCR1

MMA8451QR1

The MMA8652FCR1 shares many features of the MMA8451QR1. This device however has an interface voltage range of 1.62 to 3.6 V. It only has a 12-bit digital output but does have four channels of motion detection: Freefall, Motion, Pulse, and Transient. The sensitivity of this device is 1024(±2g)-256(±8g) LSB/g. This part costs $2.06.

ADXL326BCPZ

The ADXL326BCPZ is an analog accelerometer. it operates at 1.8 to 3.6 V, has 3 axis sensing, draws 350 µA of current. This device has good temperature stability, a range of ±19g and is capable of bandwidth adjustment. The size of this device is 4x4x1.45mm. It has a price of $6.39.

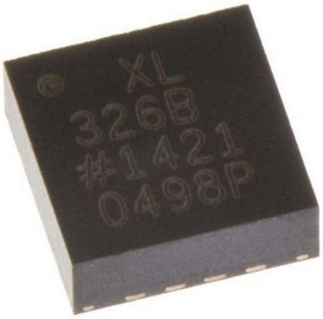


Figure 25: ADXL326BCPZ

BMA490L

This 12pin sensor, it has ranges of ±2g/±4g/±8g/±16g and an integrated FIFO with 1kB. This device is very small, 2mm by 2mm by 0.95mm, it can function with I2C or SPI and includes interrupt pins for any/no motion. The BMA490L runs on a voltage of 1.2 to 3.6 V. it has a very low current intake, with the total in performance mod being 150uA. This part costs $3.91.
It is possible that the accelerometer will have to not be directly attached to the PCB of the AutonoMouse. This is due to the need of carful placement of the sensor technology; the accelerometer must be placed just so to properly and accurately read the x y z coordinates. In such

a case, a modular PCB board containing the chip could be used in place of a direct chip, with the output nodes connected to the main PCB through proper wiring.

Figure 26: BMA490L

MPU-9250

This device is a self-contained accelerometer module. It contains a 9 axis MEMS sensor. This device is a combination of the MPU-6500’s 3 axis gyroscopic sensor and 3 axis accelerometer and the AK8963’s 3 axis magnetometer. It has been designed to be small with a size of 15mm by 25mm by 3mm. Its angular rate sensors can be set ±250, ±500, ±1,000 or ±2,000°/sec. The accelerometer functions can be programmed to range from ±2g, ±4g, ±8g and ±16g and comes with 16bit Analog to Digital converters for values. The input voltage can be 3.3 or 5V depending or preference, the standard operating currents for each section of the module are 3.2mA for the Gyroscope, 450uA for the accelerometer, and 280uA for the magnetometer. This part communicates using I2C protocol but can be configured for SPI. The price of this device is $8.95 for the entire module.

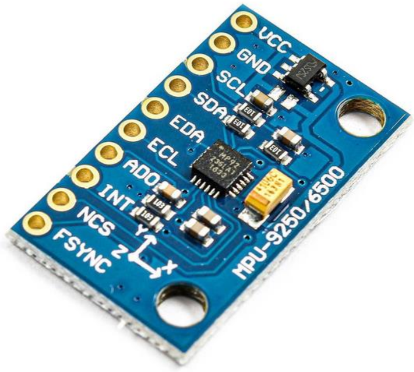


Figure 27: MPU-9250

474-SEN-12756

This accelerometer is another module, with the chip being preset up in the standard configuration. Unlike the MPU-9250, this device is only an accelerometer, still triple axis, with measurements of 2g, ±4g, and ±8g. It functions with a voltage source of 1.95 to 3.6 V. It can use I2C, SPI or UART communications. This part costs $9.50.

Figure 28: 474-SEN-12756

MPU-6050

The MPU-6500 is similar to the MPU 9250. It contains all the same statistics as the more advanced module but lacks the magnetometer. This chip is also smaller with a size of 16.5mm by 16.5mm by 2mm. It has about the same cost of $8.00.

Figure 29: MPU-6050

ADXL335

The ADXL335 is an analog accelerometer module. It operates with either 3.3V or 5V voltage source, is triple axis. The device is smaller than the MPU9250, with dimensions 19mm by 19mm by 3.14mm but not as small as the MPU6500. It has a significantly higher range on sensing that the other devices, ±200g, and has an adjustable bandwidth from 0.5Hz to 1300Hz in the XY and 0.5Hz to 1000Hz in the Z. Because this device is analog, it would require an external analog to digital converter to be attached between it and the raspberry pi zero w for data interpretation. The chip has a price of $5.35 and the PCB module version has a price of $14.95

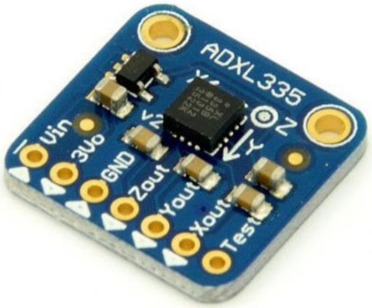


Figure 30: ADXL335

3.3.7 Motor

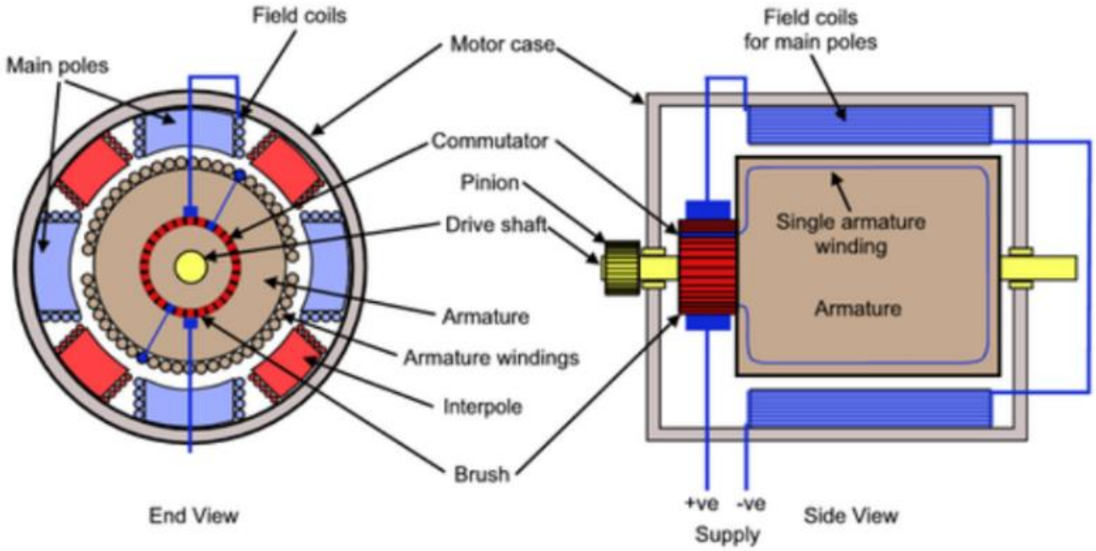


Figure 31: Motor Inner Workings

Motors are devices that move when powered by electricity. They provide rotational force, torque, and function in a manner opposite to generators. A motor uses magnetism, created by the current passing through it, to cause rotation in the shaft. The armature coils are given magnetic poles opposite to that of the field coils, this pushes/pulls the armature to rotation. When the shaft has turned, the poles switch, causing the system to again be out of balance and push the motor once again, keeping up the rotational force. Motors are needed in almost any device that has movement in it, robotic systems. They can need either AC or DC voltage supply. A simple motor only needs power to function. Altering the power, PWM, will change the rpm, speed, of the motor, and reversing the voltage polarity will reverse the motor. Motors will sometimes use a gear reducer to change the rpm or increase the torque capability without wasting power.

For many machines, motors require a motor controller. This device acts as an intermediate between the microcontroller and the motor. Microcontrollers often can only output small current ratings, 0.1 to 1 Amps, and most average sized motors may take up 2 amps or even more depending its statistics. The motor controller receives the signal from the microcontroller and then sends the required power to the motor directly from the supply, allowing for higher currents. It is likely that the AutonoMouse will not require a motor controller. Due to the size and power restrictions, the motors used will likely only use 0.1 to 2 Amps of current.

STSPIN250

This chip is used commonly in toys, it runs a voltage supply of 0 to 5V and can take a load of 1.8 to 10V. It has a max current output of 2.6 Amps and has programmable off-time current control. It is designed to operate in battery powered systems and has a standby consumption of less than

80nA. The part is readily available and has a unit cost of $1.99. If a motor controller is deemed necessary, it will likely be this one.

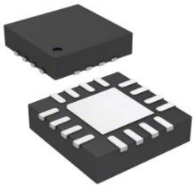


Figure 32: STSPIN250

For this project, size was a main restraint in considering motors. They had to be able to fit within the Mouse component and not cause considerable obstruction to the other parts. Since the max width of the Mouse must be 3 in, each motor cannot exceed 38mm length, anything under this is acceptable. In addition to being small they also needed to still have enough power behind them to move the device, since the estimated weight of the Mouse was 2lbs(0.9kg). Based on research, it was found that the average toy car, which is about the speed desired for the mouse, travels at about 8mph (3.57m/s). There is no specification for the time it takes to achieve this speed, assuming 5 seconds. This gives an acceleration of 0.714m/s^2 and a total force of 0.642N. Assuming a max wheel diameter of 2.5in(6.35cm), the maximum height of the Mouse, the total torque value required would roughly be 0.02 Nm. Arbitrarily deciding the minimum wheel size to be 0.5 in(1.27cm) the smallest torque would be 0.004Nm or 4mNm. These calculations do not take into account the resisting force of the wheel, however, due to the small scale of this project and the flexibility in wheel material, that force was determined to be inconsequential.

SE18K1ETY

This motor is within the desired range. It has an RPM of 4050 and a torque of 5mNm. This part has a size of 18mm x 18mm square with a shaft of diameter 2mm and length 9.9mm as well as a mounting hole spacing of 14mm. the motor has a weight of 31.75kg. However, this device requires 12V DC to operate. The SE18K1ETY has a price of $9.06.

Figure 33: SE18K1ETY

16G88-220P.1

With an RPM of 11000, the 16G88-220P.1 is an expensive little motor which provides 5.8mNm torque at a voltage rating of 3V DC. It is a round motor with a diameter of 16mm. The shaft has a diameter of 1.5mm and the length of the shaft and bearing is 7.50mm. This device can operate at temperatures from -30C to 85C. The motor costs an astonishing $81.45.

COM0805

COM0805 has an RPM of 420, a torque of 9.8mNm and operates at 6V DC. It is rectangular with a size of 10mmx12mm. The shaft of the COM0805 has a diameter of 3mm and the length of the shaft and bearing is 9mm. This motor costs only 7.67 per unit.

Figure 34: COM0805

PIS-0930

This motor has a cost of $22.39. It uses a voltage of 6V DC and has a torque of 12.75mNm. The diameter of the shaft is 3mm, the length of the shaft and bearing is 8mm, and the size of the motor is 20mm, rounded diameter.

Figure 35: PIS-0930

3.3.X AC/DC Converter

An AC/DC converter is a special device that converts ac power to a dc. The AutonoMouse will run using DC voltage, since the standard house outlet has an AC voltage of 120Vp, this will have to be converted. The desired power for the Mouse system is 25W and the highest voltage required is 12V DC. The output of this device will be attached to a recharging system for the mouse, contained within the Box section of the AutonoMouse. In addition to this adaptor a power cord will also be required to allow the Box component to be plugged into a wall outlet.

LS25-12

Figure 36: LS25-12

The LS25-5 is able to adapt AC voltages from 88 to 264 V to a stable 5V DC output. It also fulfils the 25W power requirement. The unit has a max current of 5A. This part's size is 78.7mm(3.10in) by 50.8mm(2.00in) by 28mm(1.10in) allowing it to easily fit within the specifications of the Box. This part does only have an efficiency rating of only 79% which is lower than average. Most AC/DC converters have a rating of 85% or more. This part is readily in stock with a price of $18.02 per unit. The LS25-5 has an enclosed frame, giving some extra protection.

EPS-25-5

Figure 37: EPS-25-5

The EPS-25-5 is able to adapt AC voltages from 85 to 264 V to a stable 5V DC output, it fulfils the 25W power requirement and has a max current of 5A. This part has a slightly smaller size at 3.00 x 2.00 x 0.94 inches. EPS-25-5 has a slightly higher efficiency than the LS25-5 at a rating of 81%, however it does not have an enclosed frame. This part has a unit cost of $14.15 making it significantly cheaper.

3.3.X Wheel

The wheels will be created using one of the many 3d printers at the team's disposal, it will be plastic, but might have an outer rubber coating for superior traction. The size of the wheel will be dependent upon the motor used; a stronger motor can use a larger wheel while a motor will lesser torque will use a smaller wheel. The device will have two motorized wheels at least.

3.3.8 Battery

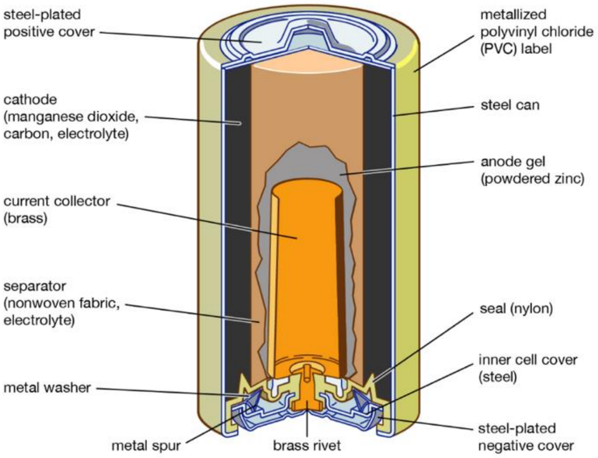


Figure 38: Inner Battery Workings

Battery cells are a power supply part that is used in many wireless devices. They use chemical energy to produce electrical power through a chemical reaction. Most are composed of zinc, manganese, potassium and graphite. Batteries come in various sizes and combinations. Batteries are rated with voltages and amp hours. The amp hour unit, Ah, is a measure of the max current a battery can provide for one hour. Batteries usually also have C rating which determines how much time it takes for the cell to discharge. The value of C is inversely proportional to time: 1C is equal to 60 mins, but 0.5C, or C/2, is equal to 2 hours.

Some arteries are rechargeable. These batteries use a reversible chemical reaction to be able to store power once depleted. Rechargeable batteries require a charging system that is capable of supplying a higher voltage than the rating of the battery, but not one so significant that it would destroy the cell. Rechargeable batteries are sometimes sold in packs, connected in series or parallel as they do not require the ability to be removed from the device, they are installed in.

The Mouse component of the Autonomous will be rechargeable. The goal is that the device will charge itself and thus the battery pack should not need to be removable. It must also be very small as to fit within the Mouse without obstructing any other parts. The power should be 25Wh, this is rather high and will likely involve using multiple cells to achieve if even possible. Due to the nature of batteries, finding a rechargeable battery which is both the proper size and power will be very difficult.

BGN800-5FWP-A800EC

This battery has a voltage rating of 4.8V and a max capacity of 1.8mAh. It has a length of 43mm(1.72in), a width of 27.4mm(1.08in) and a height of 48.8mm(1.92in) allowing it to fit quite well within the mouse, it is possible multiple might be able to fit within the Mouse. This device has an internal impedance of less than 25mOhms/Cell and a life cycle greater than 500. Each cell is rated at 1.2V. The standard charge is 180(0.1C) mA, 16 hours. They have a trickle charge of 0.03C-0.05C.

NH15BP-8

This is a simple AA rechargeable battery cell. It has a voltage rating of 1.2V and a current hour capacity of 2.3AH. The dimensions for a single cell are 14.5mm diameter and 50.5mm height. It has a weight of 27.22 g. This battery has a low voltage per cell, but thanks to its size, more than one could easily fit within the Mouse so a higher voltage can be obtained, probably about 3.6 Volts at max with 3 in series. However, to connect them together, a battery holder would be required, or the cells would need to be manually turned into a pack form. This may take up additional space. Each cell also has a high cost of $37.34 per cell.

Figure 39: NH15BP-8

ZB4.8V2X2SBSAA

This battery pack contains 2 nickel cadmium cells, it has a voltage of 4.8 and an amphour rating of 700mAh. The size is 51mm by 29mm by 29mm. This battery has lower current capacity than the _____ but the same voltage. Although it does not store as much power, if the system does need more power it may prove a more economic solution. The price of this device is $9.69.

Figure 40: ZB4.8V2X2SBSAA

HHR-450AB21

Another battery cell, this battery has a voltage of 1.2V but a large capacity of 4.2Ah. It is slightly larger than the NH15BP-8 at 18.2mm diameter by 67mm height. It also weighs about double, 60.78g. Should tow of these cells be placed in series format, it's high capacity rating would allow for little worry on the current flow. This part also has a nice cost of $8.82. It would definitely require a base voltage regulator to immediately convert the output to a more usable form.

Figure 41: HHR-450AB21

BG-621F1

This is a Sealed Lead Acid rechargeable battery. Although rather large, with a size of 41.3mm by 34.8mm by 76.2mm. It has significantly better voltage and capacity ratings, 6V out and 2A in. this device could directly connect to the motor controllers and most motors without requiring a voltage regulator. The having both a large current and voltage ensures that all devices would definitely have enough power available as all other parts would operate at a lower voltage rating, 5V or below. Despite it does have a large weight of 300g as well. This device has a cost of $29.85 per unit.

Figure 42: BG-621F1

Xeno XL-060F

Should rechargeability prove an unattainable demand or undesirable for mass production, the Mouse system could be constructed to use a primary battery. The Xeno XL-060F is a typical battery cell, it has a capacity of 2.4 Ah and a voltage of 3.6 V which is plenty to power all devices on the Mouse. The size of this part is AA and it would be of comparable size and weight to the NH15BP-8. With its voltage and capacity, a single Xeno XL-060F would probably be needed as compared to the HHR-450AB21 and NH15BP-8. It would require a voltage regulator to adjust its supply however for each device it is connected to. Of course this battery is not rechargeable so the entire system would need to be modified to allow for the removal and replacement of the power source. The price of this device is a mere $2.65.

Figure 43: Xeno XL-060F

3.3.9 Voltage regulator

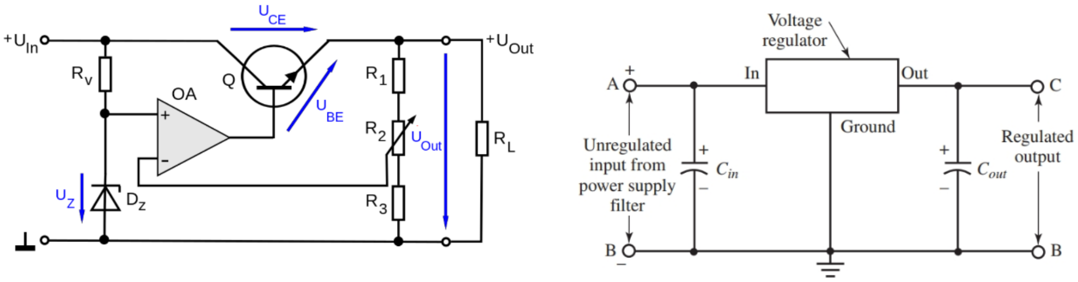


Figure 44: Voltage Regulator Circuit

Voltage regulators are an important addition to any electrical system. Their purpose is to maintain a steady, constant voltage source for components. Regulators output specific assigned voltage levels but can also be used to lower or increase the voltage at a node to ensure that the proper voltage value is attained. The output of a simple linear voltage regulator is always equal or less than the input voltage.

This is the internal diagram and circuit diagram for a linear voltage regulator. The design can work without the operational amplifier, but the output value will be less stable. Many set ups also include protection diodes.

It is possible to adjust a voltage regulator to attain a higher voltage output value using a combination of resistors by having the output feedback into the ground of the device. This works because the regulator measures its output value from its own ground pin, but not the input. It will thus change the output voltage such that the difference between the output and ground pin is the assigned voltage of the regulator. For example, if the regulator normally operates an output of 5V, and the ground is given 2V, the voltage of the output will be 7 volts since 7-2=5. This value cannot exceed the input voltage to the regulator.

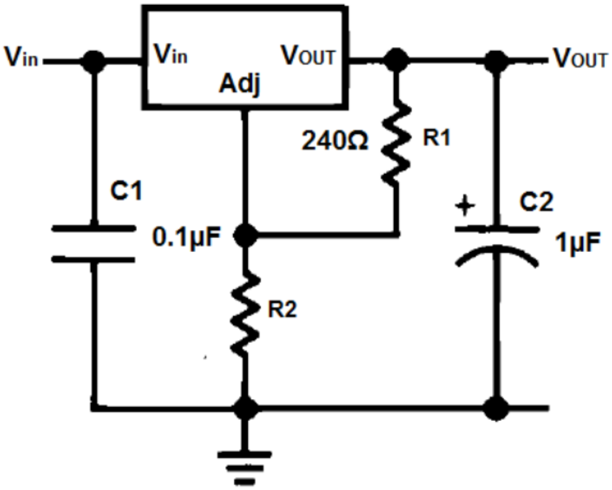


Figure 45: Voltage Regulator Circuit 2

The resistor values are key in deciding this new higher voltage. Assuming Vref to be the voltage across the resistor R1, the equation for Vout is found to be:

$$V_{out} = (\frac{R_2}{R_1} + 1)V_{ref}$$

In this linear setup, some current is lost to the R1 resistor system, in order to maximize efficiency, R1 must then be significantly larger than the load on the output. The ratio of R2 to R1 determines the increase in voltage Vout. Changing R2 will cause a change in Vout. R2 should always be greater than R1. The limitations of this design is that it cannot be used to lower the default voltage output, and it cannot be used to increase the output voltage past the input voltage.

$$V_{ref} \leq V_{out} < V_{in}$$

Switching Regulators:

There are however regulators that can increase voltage. Switching regulators, these type of regulators uses a more complex design. It involves using a switch control feedback system to create a pulse width modulation of voltage, turning a series of components on and off at a specific duty cycle. Because the circuitry is either on or off, less power is lost giving the device high efficiency. The sacrifice to this superior power usage is that switch regulators have higher noise than their linear counterparts due to the PWM. The design also allows for the generation of both higher or lower boost, buck, output voltages. Although this system has significant advantages, linear regulators are often still preferred for buck usage as they have more stable outputs and are a simpler, usually cheaper, design.

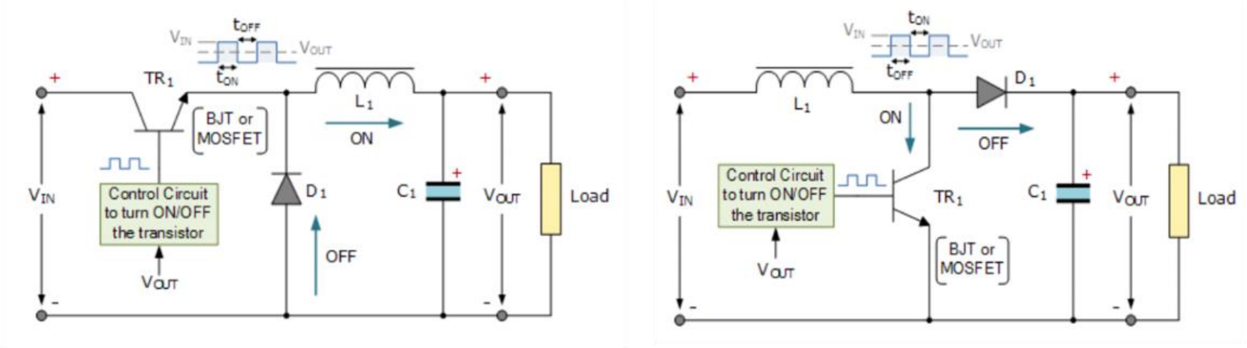


Figure 46: Switching Regulator

For the AutonoMouse, it is almost certain that voltage regulators will be used, many parts such as motors, accelerometers, microcontrollers, etc. will be running different voltages from the default battery voltage. The Mouse will likely contain at least one boost switch regulator to convert the battery power into a more useable voltage and current combination, then a series of buck regulators, each to stabilize the input to their respective components. The average price of a linear regulator is about 1-2 dollars while that of a switching regulator is 3-4 dollars.

TPS613222: This is a boost configuration switch regulator series designed to output a fixed 5V output voltage with +/-3% accuracy given an input of 0.9-5.5V. It has the possibility to output 1.8A max. This system has a roughly 90% efficiency. The regulator has thermal shutdown protection. Thanks to being a fixed regulator, it requires very little additional passive parts making it easy to implement. The TPS has a per unit price of $0.92.

Figure 47: TPS613222

MCP1700T-3302E/TT: MCP1700 is a series of linear regulator with an input operating range of 2.3 to 6 V and an output capability of 1.2 to 5V with a 0.4% tolerance. This regulator has a max output current of 250mA and a lot dropout of 178mV making it ideal as a power source to microcontroller units. It also comes with over current, short circuit and overtemperature protection. This part is available in a fixed 3.3V output configuration package. The MCP1700 has a price of $0.35.

LMR62014XMFE: The LMR is a variable boost step up regulator with the capability of reaching up to 20V output levels from an input range of 2.7 to 14V. This device has a high efficiency, 90% and can reach a max current level of 1.4A. Because it is not a fixed device, the Vout value must be up in the circuit using external resistors. The LMR62014 has a price of $4.75 per unit.

3.3.10 ADC Converter:

The AutonoMouse will have the ability to sense the charge of its battery. This is important to inform the device when it needs to return to the Box and recharge itself, in order for the Raspberry Pi unit to be able to interpret this analog value, an ADC voltage sensing device is required.

LTC2451: This ADC requires a 2.7 to 5.5V input with a supply current of only 400uA, it is able to sense voltages up to the value of the inputted Vcc with 0.02LSB RMS noise. It can be programmed to perform 30/60 conversions per second and has a 4LSB full scale error with a 1LSB offset error. This device has an internal oscillator meaning and no external components are required. The LTC2451 communicates with the microcontroller using I2C protocol. This chip costs $1.75.

Figure 48: LTC2451

3.4 Possible Architectures

The Autonomouse is a structure based around the concept of a roaming mouse that travels around a given room, and then returns to its hole, or home base. With this in mind, it is difficult to adjust the pictured concept without completely changing the project. However, various options on how to achieve a similar outcome and still maintain the whimsical aspect are possible.

3.4.1 Designs from Time/Budget Limits

When time and budget are reached, the project has to be adapted to become a product that can still deliver, but with lesser results than expected. This product has many different features, allowing changes to be made in many areas to fit constraints and specifications. A brief summarized list of what we want to accomplish with this project is listed below:

- Create a toy to entertain pets
- Be able to play with pet(s) while away
- Give user vision of pet(s)

This simple summary allows us to break the project down into tasks that we as engineers must accomplish. It is also ambiguous enough as to allow for modifications to the project as we see fit without neglecting our goals. This means that what we have designed for this project isn't a perfect solution to the problem, but rather our own solution devised after careful thinking. Below will be various options for the architecture of the design, and explanations deeming the positives and negatives of each choice.

3.4.2 Mouse Architectures

The mouse is an integral part of this project, defining its name and function. The ideal situation would be to have a mouse that actually resembles one. This might be difficult with the number of components we want to include in the mouse. To consider the ideal, the mouse would involve two wheels in the front, and a ball in the back to help it steer. This involves adding internal motors, which are going to add a large amount of size to the mouse. With each size increase, we stray further and further away from our ideal mouse. Without these additions though, we risk an inoperable mouse. To also include a motor controller, camera, and Wi-Fi enabled circuit as well requires Tetris level planning.

For the possibility that a normal mouse appearance can't be reached, or that it exceeds the size limit, very few options can act as a replacement. One of these is a gyroscopic ball structure. This would be a self-righting, spacious, and gyroscopic creation. Overall, it sounds the best in terms of providing the team with the most space to work with, but completely degrades the original animal-like characteristic of the project. It would also involve keeping the inner shell of the ball steady while the outer shell rotates and moves. This is to keep the camera steady. However, if fur is included, this becomes obsolete, completely obstructing the vision of the camera.

The motor configuration inside the mouse will affect the overall size, and possibly weight. Two options are to be explored in this case. The first being three motors that control each wheel separately. This is the option that requires the most space, and also applies the most weight to the mouse, a negative to our goal. However, this configuration is one of the most structurally sound. Having wheels directly connected to motors means there will be less parts to wiggle loose and disable the functionality of the mouse. It also offers the most power, not that we necessarily need a formula 1 racecar mouse. The second option is to run on two motors, and rely on gears and axels to drive the forward motion wheels, and another to turn. This applies less weight and mass, but also threatens structural stability if done incorrectly.

Lastly, circuitry inside the mouse will need to be placed as to not obstruct the motors, wheels, and camera, while also supplying power, and sending and receiving data to these components. There's also the matter of safety. Since this will be a toy, it's expected to experience some rough play, meaning all electrical components should be completely covered with no risk of electric shock.

3.4.3 Self-Charging

One of our requirements is that the mouse needs to self-charge. This will make sure it doesn't run out of batteries while the owner is away for long periods of time, thereby disabling it extremely easy to dock the mouse and charge. In the case that this isn't an option, and violates our constraints, a direct charge will be required by having a port in the mouse that hooks up to a charging cable in the home base. This requires much more accuracy, but will take up less space in the mouse, and is more easily accomplished.

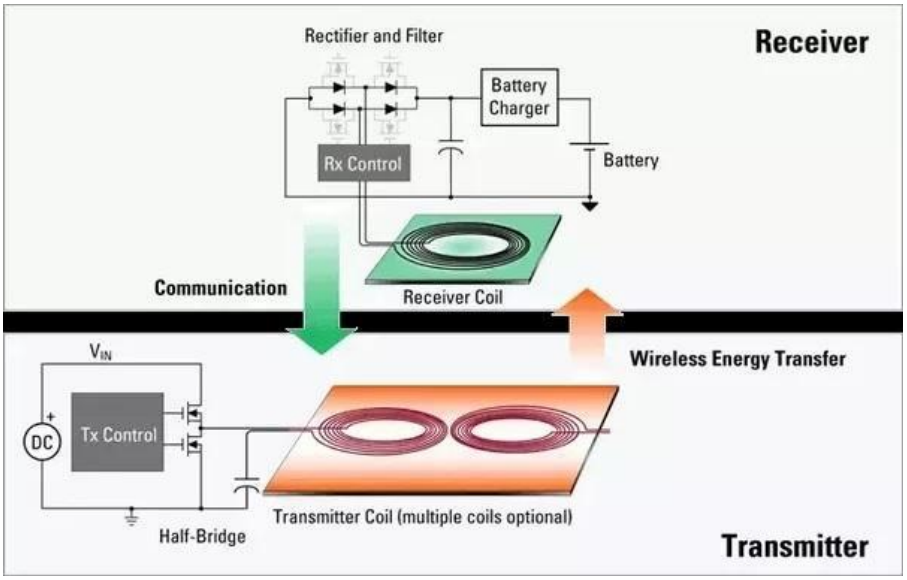


Figure 49: Self-Charging Receiver and Transmitter Diagram (Power Electronic Tips)

By using copper coils to oscillate a magnetic field between the mouse and the home base, the mouse can charge wirelessly. As for how this is accomplished in our design, there are a couple options to explore. The first design is simply a plate on the bottom of the home base that encompasses a wide area so that the mouse's movements don't need to be precise. It can simply crawl into the hole, charge, and leave again. Almost no extra calculations are needed for this, and makes the program a bit simpler. The downfall is that it is much more difficult to cover a large area with wireless charging than it is a small area. So even though the room for error is much smaller, the cost and time effort increases significantly.

Next is to have a small area for the mouse to land on. If the hole is narrow enough, this design shouldn't pose much of an issue. With the charging station set towards the back, the mouse should be able to detect where it is in an essentially two dimensional space, negating width for narrowness. Contradictory to the issues from the previous design, this should be more cost effective and less difficult to implement, but limits the area within the home base. For our design however, having a narrow hole allows for more space in the home base box for other technologies such as PCBs, sensors, laser pointers, etc. This would help to expand the multitude of functions given extra time, once the main goals of this project are finished.

3.4.4 Location Based Operations

The Home/Hole is responsible for most of the feedback and processing of information. Data is streamed directly from the home to the mouse, giving it directions on where to go and where to return. In return, the mouse gives information on location to assist the Home in processing and calculating paths. Many techniques have been devised for location based information to accurately report where a robot is at all times. The main strategy is centered on creating a map through landmarks identified through sensors.

Localization is the name given to the robot's position based on determined landmarks. As the robot travels, it identifies landmarks using sensors, then determines where it is based on those landmarks. In conjunction with this, we have mapping, which determines landmarks based on the robot's position. These two actions rely on each other to be successful since you can't have one without the other. Thus, there's the creation of SLAM, which stands for Simultaneous Localization and Mapping. By utilizing different algorithms, the mouse will be able to locate where it is, and create a map of the room it's stationed in.

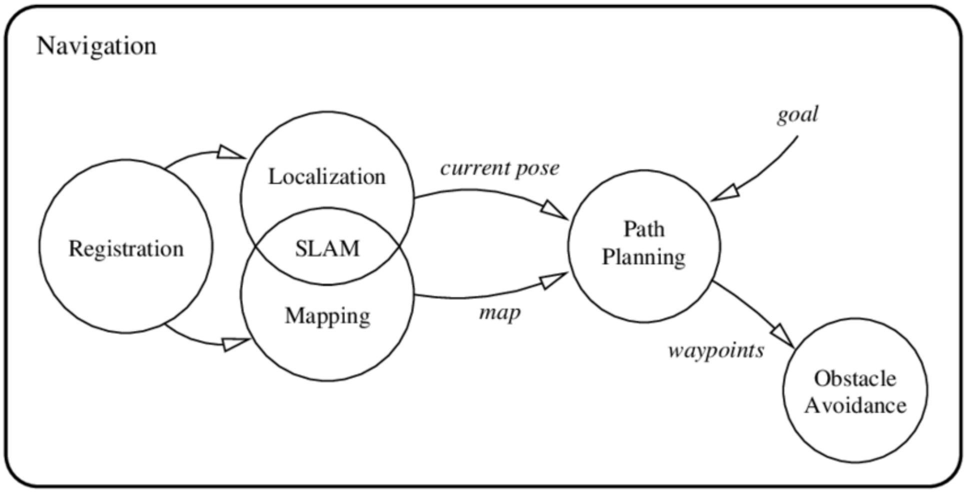


Figure 50: Combination of Localization and Mapping (Henrik Andreasson)

Distance can also be measure by essentially using odometry, or a program that can replicate it. By measuring overall wheel circumference and counting a single motor turn, the program can measure the distance travelled by the mouse. This will be useful in localization in conjunction with sensors. The robot can track its distance, determine an obstacle with a sensor, and mark that point as a landmark. This is just one of the options for mapping out a room that can be used in autonomous mode, and is unnecessary for manual mode where the user controls the mouse through a web-based app. The downside of this method is wheel-slippage and uneven turning. When using odometry, it is critical for the robot to follow a straight path when going forward. If the robot's mechanics tilt, then the localization will be thrown off since it won't have an accurate measurement of where it is.

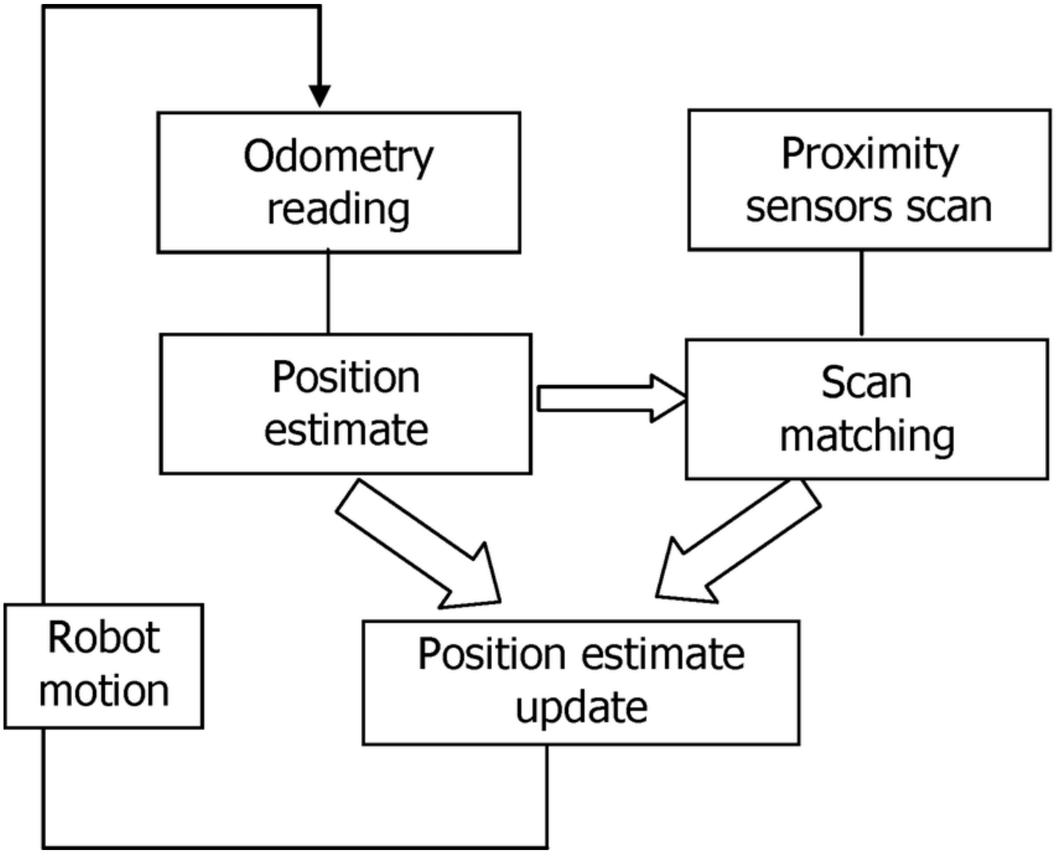


Figure 51: SLAM using Odometry and Sensors (Jiri Krejsa)

3.4.5 Video Recording and Streaming

If we use a Raspberry Pi in our mouse design, then it has a connection preinstalled that allows a video camera to be attached to it. Utilizing this would simplify the design greatly and allow for a much smaller build. It also allows for wireless streaming which is a goal of this project. By accessing a web server, the Raspberry Pi camera will directly stream the video of the camera to an app, showing the user what the mouse is currently doing. They can then use this video interface to help control the mouse's movements and play with their pet while they're away.

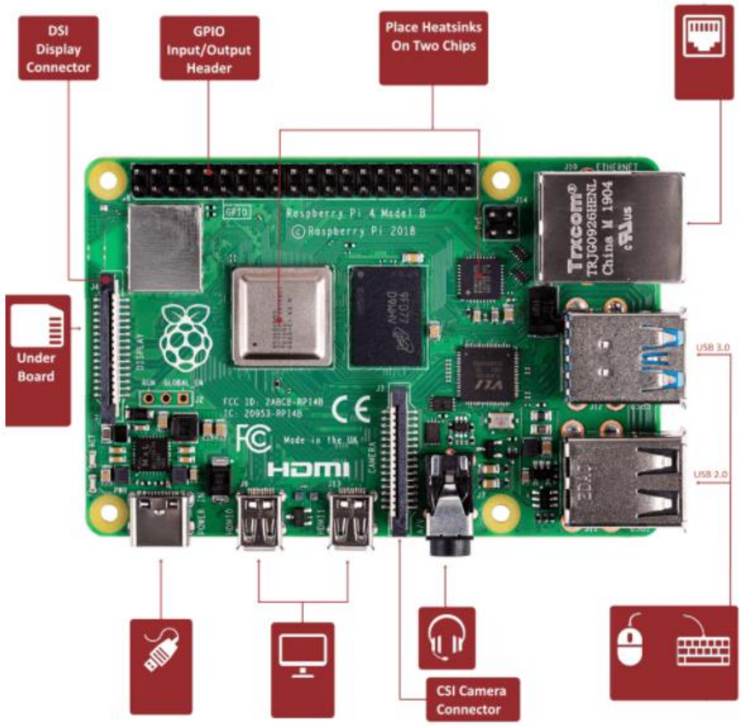


Figure 52: Raspberry Pi Configuration with CSI Camera Connector (Pi Shop)

In the off case that this camera attachment isn't an option, or the Raspberry Pi doesn't fit all our needed criteria, then an off-brand wireless "spy" camera can be used to accomplish the same thing. However, it will need access to a different board with a wireless connection to a receiver in order to stream the video elsewhere. This board could be an Arduino or other microcontroller enabled tech.

3.4.6 Wireless Connection

Since there needs to be a wireless connection for both the box and the user to communicate with the mouse, we're required to explore possible options for this. Bluetooth is one of the most common sources of wireless connection, is efficient with power, and supplies a simple short range connection for most technologies. It also doesn't require any sort of internet connection, so it should always be available. However, this is only for the communications between the mouse and the box since the communication between the user and the mouse will more often be at longer distances, making Bluetooth obsolete.

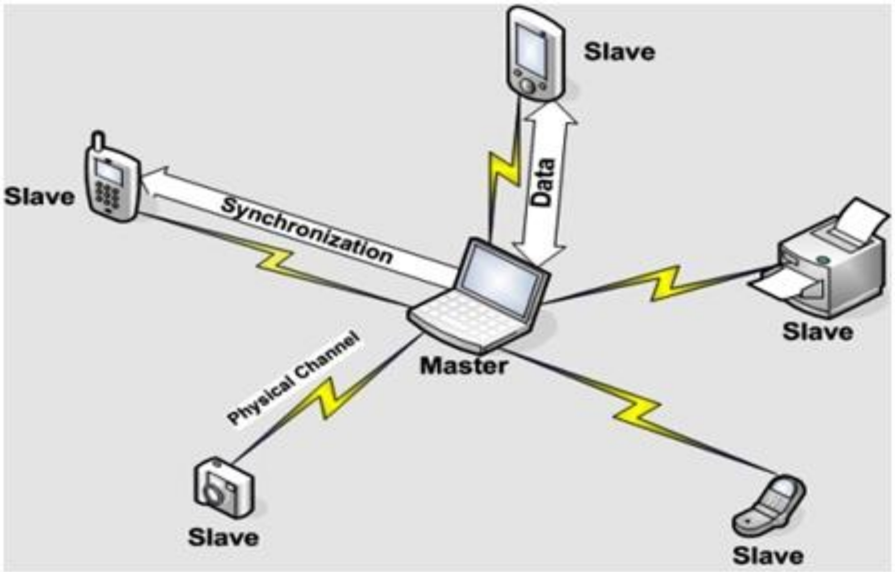


Figure 53.1: Bluetooth Connections (Vinnter)

It’s absolutely an option to add Bluetooth the mouse and the box, but also seems redundant when addressing the fact that the box would need to contain both Bluetooth and Wi-Fi in order to communicate with the user and the mouse. Wi-Fi is better at operating on full-scale networks, has fast connections, and contains many more options for security than Bluetooth, which is important for a product that is able to take video of the user’s house. Although it requires more battery to operate Wi-Fi over Bluetooth, the self-charging station in the design should allow that to not be an issue. Wi-Fi will also help for people with larger homes that may have lost connection after a certain distance operating with Bluetooth. The benefits seem completely one-sided, but all options need to be considered in the off-case one doesn’t pan out.

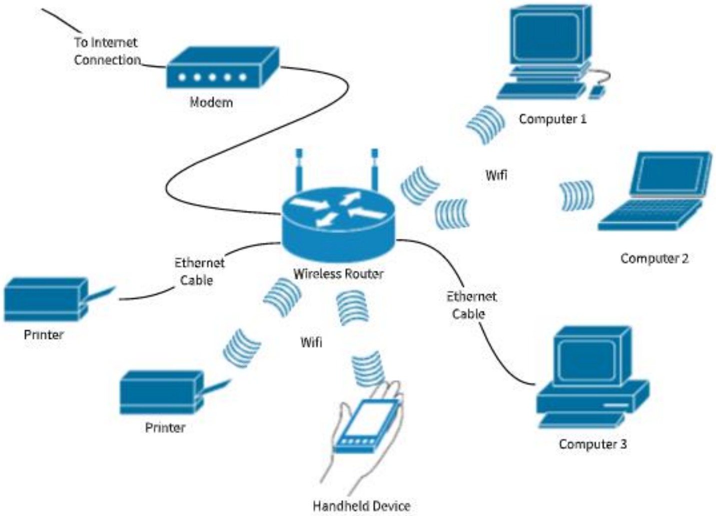


Figure 53.2: Wi-Fi Connections (LucidChart)

3.4.7 Box Design

The box in our project has been commonly referred to as the home base, or as the hole. There's not much we can do to the box without compromising its whimsical properties. The physical structure of the box, along with its internal structure are still up for debate as of this moment. To address this, we will go into details of the possible configurations for this box.

The initial design of the box is just that, a box. It would be a rectangular structure, with a mouse hole carved into it to house the mouse. This allows for plenty of compartment space to store the Raspberry Pi and other necessary components. The issue is that it may not be the most visually appealing. With more and more products coming out that focus on aesthetic, it's important to consider all the possibilities, while not compromising functionality. However in our case, aesthetic might play hand-in-hand with functionality and compactness.

A plane rectangular box laying on the floor may seem out of place and bulky. Redesigning this into a large half-oval shape will increase its appearance, still allow for room for the minimum amount of components we plan to put into the box, and possibly bring those components together to decrease wire length and overall minimize voltage drop from the length of the wire. This structure would house the hole almost in parallel with all points on the outside of the box, or in other terms, the box structure would be shaped like a common stereotypical mouse hole as shown in Figure 5 from the introduction. Since the charging station is on the bottom, no extra modifications are needed. In almost all shapes, the charging station will hold the same position as usual.

3.5 Final Architecture

By the end of the project, a final design was achieved through the combination of different elements discussed in this document. To start, what wasn't used was the location based operations. The mouse's small size made it difficult to fit in hardware that could support this technique. If we wanted to implement this, the mouse would need to have many more sensors that were highly accurate at detecting distances. Not only this, but implementing the program in our time frame would prove incredibly difficult since it is a complicated idea.

Second of what was not included is part of the box design. The box is still in the project, however, it does not include a raspberry pi inside it for communication. After taking out the IR homing as well as the idea of mapping, it was no longer necessary and would prove superfluous to our design. Instead, the only component inside the box is the wireless charger that hooks up to a wall outlet.

Other than these two, all the designs discussed in this architecture section were used in the final design. WiFi was implemented so that the user could drive the mouse while away from home. Wireless charging was included as mentioned in the previous paragraph. And video recording was also included into the final design successfully. Having these in the project allowed us to fit the specifications stated at the beginning of this section.

3.5 Final Part Selection

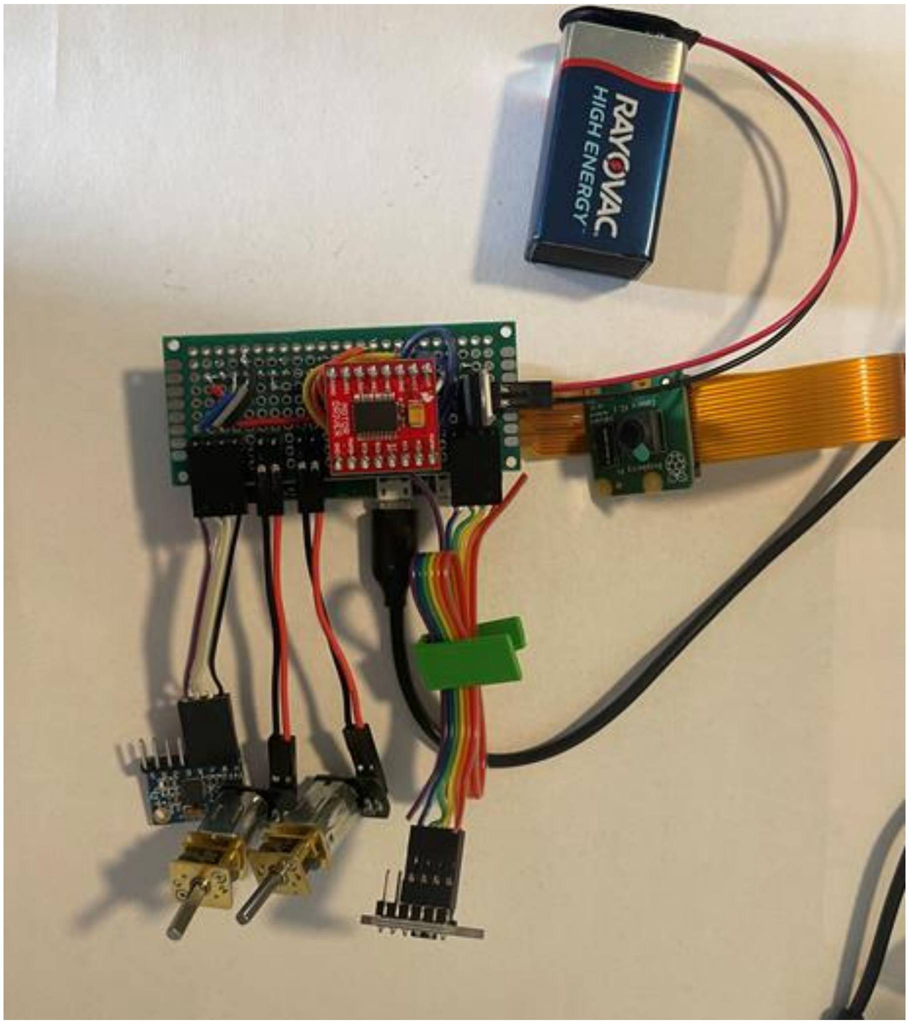


Figure 54.1: Final Parts

As we started to build and work on the project a lot of changes were made to the parts selection list (Figure 102). For the Raspberry Pi we chose the Raspberry Pi zero W because it was small, light and it had low electrical usage. It was also the best way to be able to provide a camera feed to the web app. For the camera we decided to go with the Camera Module V1 (OV5647). Since we chose a Raspberry Pi the Wi-Fi module was not needed since the Raspberry Pi already has built in wireless LAN and built in Bluetooth as well. We chose the motor (COM0805) due to its size, high speed, and torque. The motor controller (DRV8830DGQR) we chose was a H-bridge voltage controlled motor driver. The accelerometer (MPU-6050) was chosen because it was accessible and affordable. For the sensor we decided to pick a laser sensor (VL53L0X). Compared to the ultrasonic it was a lot smaller, had a wider field of view and provided a fast response since light travels faster than sound.

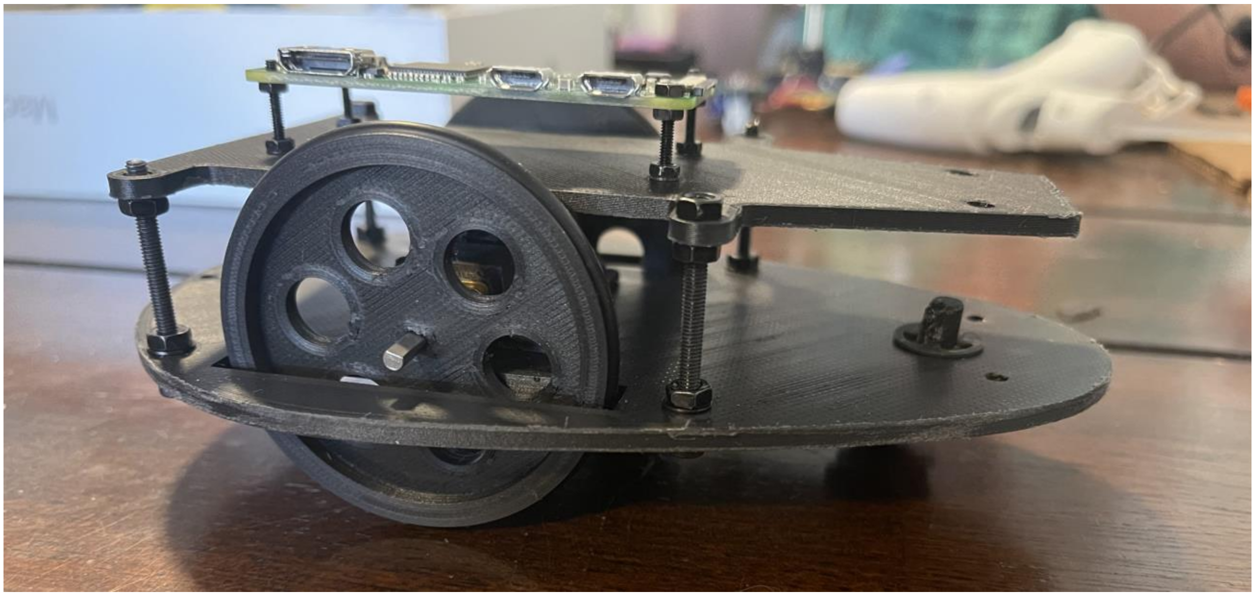


Figure 54.2: Chassis

Figure 54.3: External Fur Coating

The chassis (Figure 103) was designed in inventor and 3D printed using PLA plastic. The same thing was done for the wheels and they were wrapped with an O-ring for better traction. A bearing was added to the front wheel to provide smooth rotation. To create the external fur coating and the tail (Figure 104) we first 3D printed a case and then we wrapped it with yarn. All the materials used for this part were nontoxic and safe to use by the cats.

The charging system was originally going to be custom made, but it was later decided that it would be much easier to simply order a Qi compliant transmitter and receiver pair of devices and use those for the wireless charging system. Adafruit 1901 and 2162 were chosen as an ideal pairing, 1901 had a small size to allow it to fit well within the constraints of the mouse and 2162, to be placed on the charging bad, was designed specifically to be used with the 1901, making them a good matching pair. The battery was changed to an LI NCR18650 for its smaller size and higher capacity that the previously discussed batteries. Included with it was a TP4056 for properly charging the battery, which requires a steady 4.2V input when charging, the 1901 and 2162 both operate at 4.8-5.2V. Instead of using an ADC, a custom ADC was created using a series of voltage dividers, a LM339N, and a BOB-20009(see section 6.4). This is less accurate than a standard ADC but was easy to acquire and the mouse does not truly need much accuracy for the battery value as it will not be displayed to the user. A line tracker module was added to allow the mouse to properly station itself on the Box.

The final parts list was then as follows:

Raspberry PI Zero W: Chosen due to its small size, low power requirements and ability to handle video streaming, the PI is the brain of the Mouse. The Zero W comes with a preinstalled wifi chip allowing for communication between the Mouse and the webapp for the user control mode. The Pi requires 5V if utilizing wifi. It has a size of 60mm x 30mm. An additional Pi will be used inside the box working with an Arduino to regulate the infrared emitter diodes.

OV5647: The OV5647 is OmniVisions CMOS image sensor. It can deliver 5-megapixel photography and videos at a high frame rate 720p/80(HD). The camera module size is 8.5 mm x 8.5 mm x <= 5mm. The OV5647 is built on 1.4-micron OmniBSI backside illumination pixel architecture.

MPU-6050: This 3-axis accelerometer and gyroscope combo will let the mouse know its orientation and movement. Its angular rate sensors can be set ±250, ±500, ±1,000 or ±2,000°/sec. The accelerometer functions can be programmed to range from ±2g, ±4g, ±8g and ±16g and comes with 16bit Analog to Digital converters for values. The input voltage can be 3.3 or 5V.

COM0805: COM0805 has an RPM of 420, a torque of 9.8mNm and operates at 5V DC. It is rectangular with a size of 10mmx12mm. The shaft of the COM0805 has a diameter of 3mm and the length of the shaft and bearing is 9mm. Two of these motors are used to move the Mouse.

TB6612FNG: This motor driver IC controls both the COM0805 motors. It is capable of taking up to 15V for motor voltage and uses 3V for the logic system. The max current it can put into a motor channel is 1.2A.

TP4056: This is the ideal battery charger for an LI NCR18650 Battery, it takes a 5V input and regulates it to a stable and steady 4.2V charging level. TP4056 was chosen to be included in a modular form as to not restrict the type of battery being used, should the chosen capacity prove incapable of sustaining the system.

VL53L0X: This laser module lets the mouse detect obstacles as it approaches them using time of flight. It can operate with 3-5V and has a distance detection range of 50-1200mm. The VL53L0X uses I2C communication.

LI NCR18650: This rechargeable battery has a voltage rating of 3.6V, it has a total capacity of 3250mAh and is typically run at 0.2C making it capable of running the Mouse. This battery was chosen for its small size, 70mm long with a 20mm diameter.

MT3608: This boost regulator will turn the 3.6V from the 18650 battery into the required 5V that the motors, the Pi and the laser module require.

ADA 1901: A Qi standard receiver for wireless inductive charging, it outputs 5V, 1A. This component is 48mm x 32mm x 0.5mm and is on the underside of the Mouse.

ADA 2162: A Qi standard transmitter, it operates at 5V, 2A. Located on the Box charging station, this part has dimensions of 53mm x 53mm x 4mm.

PMC-05V015W1AA: A 5V, 3A ACDC converter, it will be used to power everything in the Box.

HiLetGo IR Module: This infrared device functions on 3.3V or 5V. It is ideal for line tracking systems and will be used to help the mouse find the box.

LM339N: A 4 comparator IC, it works with a series of voltage dividers and a logic shifter to create a simple ADC to tell battery level and charge state.

BOB-12009: The logic level shifter to be used with LM339N, it turns any 5V input into 3.3V.

Structure: The chassis, wheels, and charging station were all custom made on 3d printers.

4. Standards

A standard is a repeatable, established, and documented way of doing something to ensure interoperability, make life simpler, and increase the reliability and effectiveness of the goods and services we use. They contain a lot of precise criteria and technical specifications made by a group of experts in the field who provide a general agreement. They are meant to be used frequently as a guideline, rule, or definition. Standards protect consumers because they ensure the product or service is reliable, durable, and safe. They also make sure it is fit for its purpose and performs as specified, especially in situations where the consumer is not offered a lot of choices. Some standards help facilitate world trade by checking if a product or service follows regulations and legislation and providing certificates for consistency. Some examples of types of standards that should be considered for this project are safety, reliability, communications, data formats, design methods, programming languages and connector standards.

4.1 Search at www.nssn.org

The NSSN is an electronic network that serves as a search engine for the standards systems of many organizations that work in the development, production, distribution, and use of technical standards. It is free and is sponsored by ANSI, The American National Standards Institute. The NSSN serves as a single point to search for information and facilitate access to domestic and international approved standards. It connects people who develop standards for the people that need them. You are able to search by organization, title, document, or keywords. Institutes and organizations like ISO, IEC, ITU, IEEE, and IPC make it possible for standards to be developed and shared around the world to ensure consistency, quality, and safety of products and to facilitate world trade.

ANSI: The American National Standards Institute is a private non-profit organization that manages the development of voluntary consensus standards for goods, services, processes, systems, and personnel in the United States. Many companies, government agencies, trade associations, consumer groups, industries, and academics are members or representatives of ANSI.

They also represent the United States internationally to make standards that are universally accepted in different industries. They are the United States representatives to the ISO, International Organization for Standardization and the IEC, The International Electrotechnical Commission. ANSI does not create standards; instead they accredit the procedures of standards made by different organizations.

ISO: International Organization for Standardization, is an international organization made of representatives from various national standards organizations. It is the largest organization of international standards. They provide certifications to ensure the safety, quality, consistency, and efficiency of the goods and services. They cover a lot of areas in the industry, like technology, healthcare, food safety, energy management, and manufactured products. This is very important when it comes to world trade, as it makes the process a lot easier by giving consistent standards among the nations. A lot of industries will not buy goods and services that have not been certified by ISO.

IEC: The International Electrotechnical Commission is an international organization that takes care of all international standards related to electrical, electronic, and related technologies. They have members to represent each country, and they all get a vote on what goes into the IEC standards. This way, they are able to represent the needs of every nation. A lot of devices around the world depend on IEC standards to perform, fit and work safely together. IEC encourages the making of products, systems, and services that are safe, efficient, and environmentally friendly to promote world trade and economic growth. They also work closely with organizations like ISO and ITU to create standards.

ITU: The International Telecommunication Union is the United Nations specialized agency for information and communication technologies that develop standards to ensure networks and technologies seamlessly interconnect. ITU helps to create standards for the global use of satellites, space services, and wireless communications. They also allow telephones and internet access to work locally and globally.

IEEE: Institute of Electrical and Electronics Engineers is a professional association to develop and define electronic, electrical, and computer science standards. They started as AIEE, American Institute of Electrical Engineering's and then later changed to IEEE. They cover a lot of industries like consumer technology and electronics, biomedical, telecommunication, robotics, home automation, nanotechnology, transportation, power, and energy.

IPC: The Global Association for Electronics Manufacturing is a global association that creates standards for assembly and production requirements of electronic equipment. This includes OEMs, EMS, PCB, cable, and wire harness. They are accredited by ANSI while also being recognized globally for their standards. They drive the electronics industry's success globally by ensuring reliability, consistency, and quality.

4.2 Design Impact of Relevant Standards

4.2.1 Wireless charging standard

4.2.1.1 Qi

Qi is an open interface standard that defines wireless power transfer using inductive charging over distances of up to 4 cm (1.6 inches), with Extended Power Profile, it can support power transfer up to 30 watts, developed by the Wireless Power Consortium (WPC). Devices that operate with the Qi standard rely on electromagnetic induction between planar coils. The Qi specification can be downloaded freely after registration.

A Qi system consists of two parts (Figure 45), the Base Station and the Mobile Device. The Base Station is connected to a power source and provides inductive power; the Mobile Device consumes inductive power. The Base Station contains a power transmitter that comprises a transmitting coil that generates an oscillating magnetic field; the Mobile Device contains a power receiver holding a receiving coil. The magnetic field induces an alternating current in the receiving coil by Faraday's law of induction. Close spacing of the two coils, as well as shielding on their surfaces, ensure the inductive power transfer is efficient.

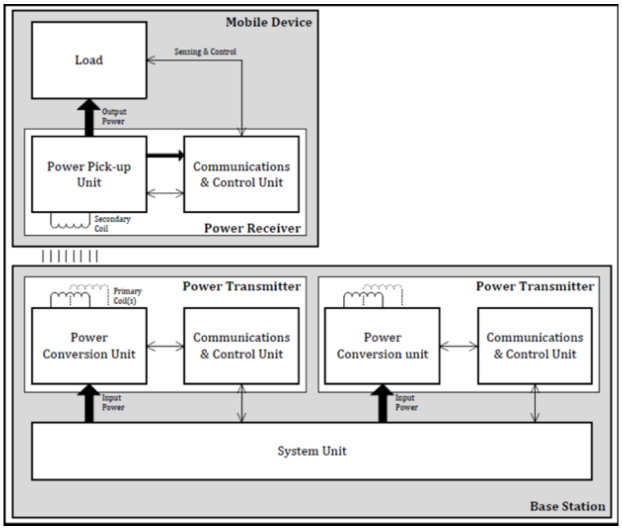


Figure 55: Qi System

4.2.1.2 Rezence

Rezence is another interface standard developed by the Alliance for Wireless Power (A4WP). It’s also using magnetic coupling, but with resonant inductive coupling, which is different from non-resonant inductive coupling. Resonant inductive coupling is where the maximum voltage is generated on the secondary coil when the primary coil is driven with a resonant frequency of the secondary side. This improves efficiency relatively by reducing the copper loss of the primary coil and heat loss.

A Rezence system consists of a single power transmitter unit (PTU) and one or more power receiver units (PRUs). The interface standard supports power transfer up to 50 watts and over distances up to 5 centimeters. The power transmission frequency is 6.78 MHz, and up to eight devices can be powered from a single PTU depending on transmitter and receiver geometry and power levels.

4.2.2 Design Impact of Wireless Charging Standard

Although the above two wireless charging standards are open interface standards, which are not part of ISO or ANSI standards, they are dominating the mobile devices in the industry.

Qi was release in 2008, and by 2019 the Qi standard was incorporated into more than 160 smartphones, tablets, and other devices. The companies that are working with the standard include Apple, Asus, Google, HTC, Huawei, LG Electronics, Motorola Mobility, Nokia, Samsung, BlackBerry, Xiaomi, and Sony.

Rezence was formed in 2012 to compete with Qi, it hasn’t really had a chance to release products that implement this technology yet. However we are slowly seeing places like Starbucks, McDonald’s are now using Rezence Powermats.

Due to availability, cost and manufacturability constraints, we will most likely to implement the wireless charging part of our project using the Qi standard.

4.2.3 Communication Standards and Protocols

4.2.3.1 IEEE 802.11-Wi-Fi Standard (Wikipedia)

IEEE 802.11 defines the physical layer protocols for implementing wireless local area network(WLAN). It specifies the bands that can be used in the electromagnetic spectrum for Wi-Fi. The 802.11 standard consists of a series of half-duplex over-the-air modulation techniques that use the same protocol. 802.11 g/n work in both the 2.4 GHz and 5GHz bands, 802.11 b only works in the 2.4 GHz band, and 802.11 a only works in the 5 GHz band. Because of the choice of the 2.4 GHz band, 802.11 b/g/n equipment occasionally suffer interference from microwave ovens, cordless telephones, Bluetooth devices and etc. 802.11 a/g control their interference by using orthogonal frequency-division multiplexing(OFDM) signaling method, and 802.11b uses direct-sequence spread spectrum (DSSS) signaling method. These are unlicensed ISM (Industrial, Science, Medical) bands, they are free for use at low power. Figure 46 shows the 802.11 bands.

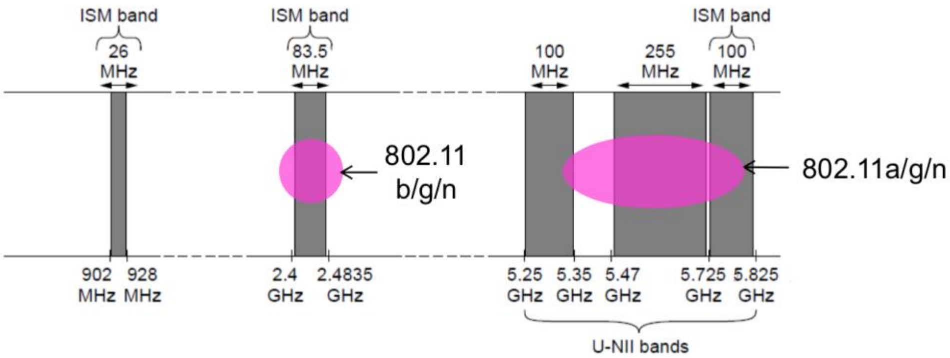


Figure 56: 802.11 Bands (source: UCF Communication Networks lecture)

4.2.3.2 Serial Communication Protocols

Serial communications is a modern communication method that transfers one bit at a time. There are two types of protocols, synchronous and Asynchronous. Synchronous is where the transmitter and the receiver share the same clock signal, and Asynchronous is where the transmitter and receiver each generate their own clock. Below are three commonly used serial communication protocols.

- **UART-** Universal Asynchronous Receiver/Transmitter. It can be setup as half-duplex or full-duplex, and uses one wire in each direction.
- **I2C-** Integrated Circuit. It uses Bus topology which means it uses shared wires and it is asynchronous. It's usually consist of one master and multiple devices. The master always initiates the communication such as read or write, and it drives the clock signal as well. It's setup as half-duplex, which means the data can be transferred in one direction only at a time, but it is bi-directional. Each device has a unique address.
- **SPI-** Serial Peripheral Interface. It is synchronous and it can be setup as full-duplex or half-duplex, it uses 4 wires.

4.2.3.3 Internet Protocol Suite

The Internet Protocol Suite defines a set of communication protocols for the layering stack used in the internet and similar computer networks. It's often referred to as TCP/IP because TCP and IP were the foundational protocols. It's comprised by the Internet Engineering Task Force (IETF), which is part of Standards Developing Organization (SDO). Protocol layering is the main structuring method used to divide up network functionality, and each layer talks to its peer using a protocol. Figure 47 displays the TCP/IP model which includes 4 layers: application, transport, internet, and link layer.

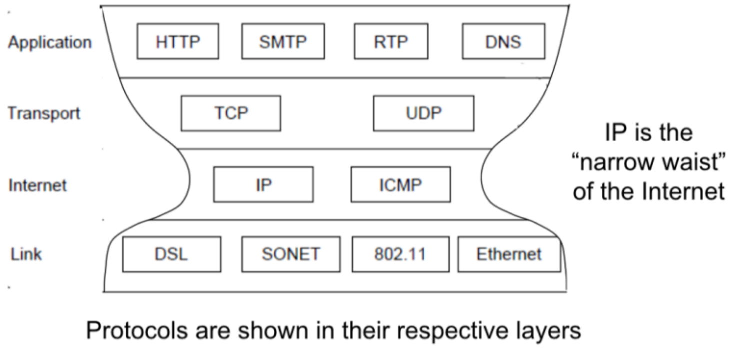


Figure 57: TCP/IP Model (UCF Communication Networks lecture)

- **Application Layer**: supports network applications, defines the types of messages exchanged, message syntax, message semantics and rules. protocols examples include HTTP (Hypertext Transfer Protocol), FTP (the File Transfer Protocol), SMTP (Simple Mail Transfer Protocol)
- **Transport Layer**: host to host, end to end data transfer, protocols include TCP (Transmission Control Protocol), UDP (User Diagram Protocol). TCP is a connection-oriented protocol, it offers reliability and error correction, whereas UDP is a connectionless protocol and it's typically used in streaming applications where time sensitivity is more important than reliability.
- **Internet Layer**: routing of datagrams from source to destination, protocol examples include IPv4(32-bit addressing), IPv6(128-bit addressing), and routing protocols.
- **Link Layer**: data transferring between neighboring network elements. Protocol examples include PPP (Point to Point Protocol), Ethernet.

4.2.3.4 Hypertext Markup Language (HTML)

HTML is the standard markup language for the Web browser. It was first publicly mentioned by Tim Berners-Lee as "HTML Tags" in 1991. It went through a few versions, and HTML 3.0 was proposed to the IETF in 1995, but expired five months later without any further actions. W3C and WHATWG organization then started working together on the specifications of HTML5. Finally in 2019, WHATWG became the sole publisher of the HTML standard. The WHATWG organization called it "Living Standard", which means it's always being improved, and new features can be added but functionality will not be removed.

HTML is a way of enhancing plain-text content. The content most familiar for HTML's use cases is "Hypertext", which was an exciting, futuristic term for what we now think of as ordinary, simple websites where underlined words lead to entirely separate pages of information. The markup consists of tag names in angle brackets, and can be nested, meaning the inside of any set of brackets can contain almost any other valid set of tags inside it. HTML is intrinsically linked to two other major technologies: CSS and Javascript. CSS (Cascading Style Sheets) changes the appearance of elements of an HTML document by relating position in the tag's structure to appearances or animations. Javascript allows HTML elements to execute code to self-modify or communicate with servers when the user interacts with certain elements with actions like clicking, scrolling, hovering, or drag-and-drop. HTML is a framework for turning basic text content into an interactive and appealing user interface.

4.2.3.5 Common Gateway Interface (CGI)

CGI defines a standard way in which information may be passed to and from the browser to the server. It's first developed by the National Center of Supercomputing Applications (NCSA) in 1993, and formally defined in RFC 3875 in 1997.

4.2.4 Design Impact of Communication Standards and Protocols

Internet communication is based on protocols, from the physical layer to the application layer, it is important to follow these protocols accordingly so that the devices can communicate with each other properly over the internet. An important part of our project is to develop a Web Application so that the user can send instructions over the internet to the device in order to interact with the cat. The following are some of the protocols that will be implemented, however, they are subject to change later during the testing phase of the project.

- Device must have Wi-Fi capability, which is included in a Raspberri Pi.
- HTTP (Hypertext Transfer Protocol) will be used as the Web's application layer protocol. HTTP uses client/server model, where the client (browser) requests, receives and displays the Web objects, and the Web server sends the response to the requests using HTTP protocol. HTTP also uses TCP connection from the transport layer to initiate and close a connection. HTTP has a stateless property which means the server maintains no information about client's past requests.
- CGI will be used as the middle tier protocol to transfer information to and from the browser and the Web Sever.
- HTML will be used as the presentation language for the browser.

Every electronic device runs on serial communication. In our project, serial communication protocols will be used to transfer data between the microcontroller and other peripherals.

4.2.5 Programming Languages

4.2.5.1 ANSI C

C is a procedural programming language that supports structured programming, lexical variable scope and recursion. It was originally developed to write operating system by Dennis Ritchie and Ken Thompson in the 70's. It became and still is popular since the 80's. ANSI C, ISO C, and Standard C are successive standards for the C programming language published by the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO).

C is the original "high level language" because it was invented to save programmers the trouble of re-writing programs in custom assembly language for each different brand of computer. Nowadays it's considered a very low level language because of manual memory management (malloc, free, and the use of pointers). It's very performant because it can be translated almost directly to machine code. It's considered perilous to program in because memory management is prone to leaks and illegal pointer exceptions. C should be used when performance is utterly paramount, and only if the time and resources for testing are available.

4.2.5.2 Python

Python is a high-level and general purpose programming language. It supports object-oriented, structured, functional programming and etc. One of the biggest advantage of Python is its standard library. Python has been in development since the early 90's, it's partly inspired by a language called ABC. It's popular as a learning language, back end web development language, and very popular in machine learning now. Its popularity is due to its compatibility with C, C modules can be written and exposed through Python, so highly performance-critical elements can be written in C, while less performance critical items can be easily and quickly written in Python.

4.2.5.3. Java

Java is a garbage collected, interpreted, object oriented language designed by Sun Microsystems to make application development faster for programmers. Garbage collection frees up programmers from the difficulty of allocating memory: variables are allocated when they are declared, and when they are no longer in-scope of any namespace, they are automatically culled. This has a performance penalty, though: garbage collection routinely pauses the execution of the program so that the Java Virtual Machine (JVM) can check counts of pointers to determine which memory can be released. Therefore, Java is a tradeoff for easier and faster development, at the cost of program runtime. The Java Virtual Machine is the interpreter for Java bytecode: Java programs are not compiled down to machine code. The bytecode produced by the Java compiler is truly cross-platform, meaning it can be transferred to a Windows, Mac, or Linux computer and run on those platforms JVMs without recompilation. Compared to C, where source code is portable but compiled programs are not, this is another tradeoff that reduces performance for the convenience and productivity of developers.

4.2.6 Design Impact of Programming Languages

Different devices or parts support different languages, it is necessary for us to "speak" the language these devices can understand, which is why standards and protocols are important. Finding the most suitable programming languages for the devices that we are using in our project is a major decision. We also have to consider which one is the best for understanding among group members since we will be working together and implementing different parts of the project.

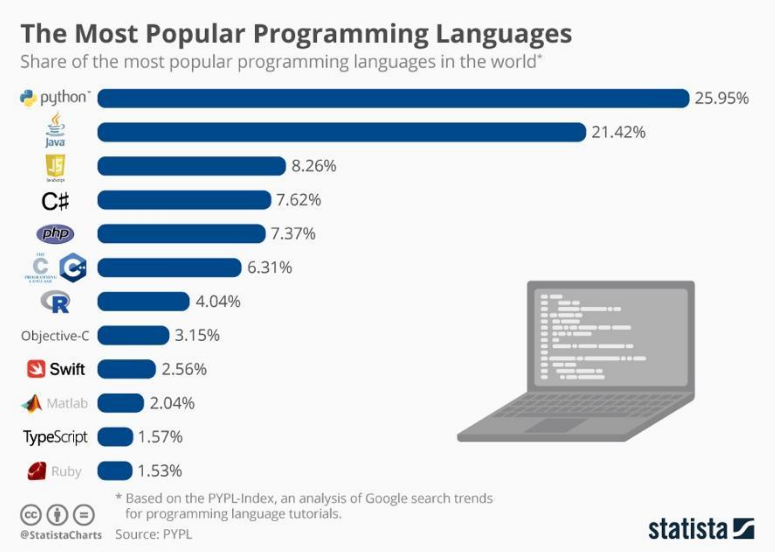


Figure 58: Most Common Programming Languages (Statista)

4.2.3 Safety Constraints

The Autonomouse has a responsibility to both humans and animals to be safe to use. This means that it shall abide by the safety standards set by ANSI and IEEE to fulfill the design requirements. In creating an electrical based project, it is important to consider the risk of electric shock, exposure to radio-frequency electric fields, and high heats as a result of short circuits or soldering acts.

The main issue that comes to mind with the Autonomouse is the mouse itself. As mentioned in Section 3.4, there is the restriction put on the design to protect the wires and other electronic devices so that the pet isn’t met with an electric shock if playing too rough with the mouse. To prevent this, we have to delve into quality control as well as safety. Each wire should be checked for frays and cuts along the wire so that no metal is exposed. All the wiring should be safely compartmentalized in the mouse, protected by difficult-to-break material that will allow rough play without worry.

Next comes the general size of the mouse, and the components in the mouse. If the mouse is too small, it becomes a choking hazard for pets and small children. This should be kept in mind. However, if it is too large, it might scare the pet, but also risks a tripping hazard to humans given its roaming nature. Finding the perfect medium is essential to the safety of people and animals. The material of the fur should also be taken into consideration for the overall safety of the product. It’s imperative to use a non-toxic faux fur that resists shedding in order to avoid choking the pet in unfortunate circumstances. It should also resist tearing as much as possible without compromising the texture.

Tripping hazards aren’t uncommon with moving objects on the ground. Size is a factor in this, but it can also be prevented through sensor input and visual recognition. Our specification for this is that the mouse should be able to detect the motion of humans or doors and immediately stop its movement, or set a path to avoid collision. Not only may the act of hitting the mouse cause injury to humans, but also to the mouse, possibly breaking it and ruining functionality.

Many of these constraints have taken the mouse in mind, but have not considered the data of the camera. The video from the camera is streamed to the phone of the user, allowing them to view whatever the mouse sees. However, for cyber safety, this data needs to be absolutely protected from any sort of hacking or viewing from outside sources, since it is a personal film of the user's home and personal belonging. The video should be encrypted or require an account to log in to. Otherwise, the design poses a risk to the owner's information and privacy.

4.2.5 Manufacturability Constraints

Being a product that is restricted to size, it is important to take into consideration the size of the possible parts that will be used from the design. For example, it is not viable to design a 3" by 2" mouse that requires a circuit of 3" by 3". Even if the design of the circuit is correct, it breaks the limits of the product and cannot be manufactured without redesigning the project. Therefore, manufacturability needs to be taken into account.

The maximum limitations for the mouse will be 3" in width, 5" in length, and 3" in height. So, the circuitry should be able to fit within those limitations, along with the motors, battery, and camera. This will make the system extremely crowded, but will be up to us as engineers to design it so that everything is able to fit within those dimensions. The outer shell of the mouse should also be thin enough to allow accurate streaming to receiver nearby. It's simple enough to add a Wi-Fi chip to the mouse, but it might obstruct the signal if there is too thick of a wall between the chip and the receiver. Options for this are to adjust our materials to those that do not absorb wireless signals. Metal absorbs signal to a great extent and should be eliminated as a build material at all costs except for simple fasteners such as screws and nails. This is an example of a metal that should technically not be used. For manufacturing requirements, the materials of the project should not be unobtainable or expensive. We do not want to require gold or uranium to be part of the project. This should be obvious, but requires stating.

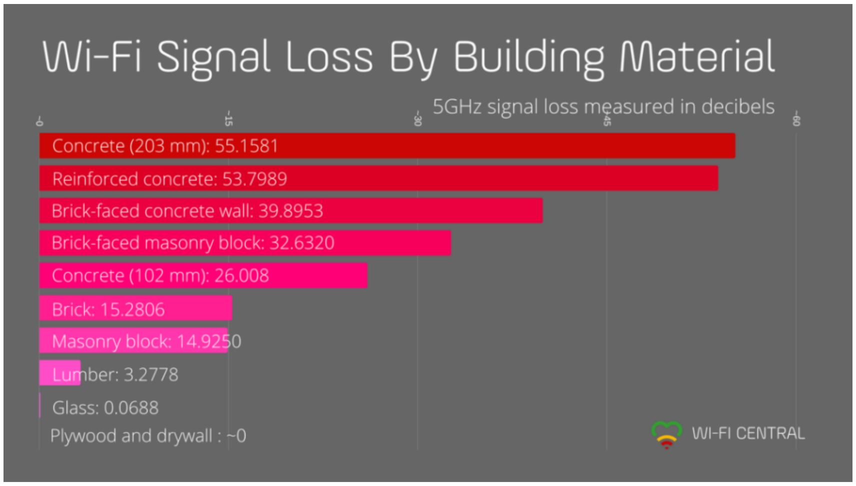


Figure 59: Wi-Fi Loss by Material (EyeNetworks)

There are currently no constraints placed on what parts will be used in our design. However, the available options consist of those that can adapt to a signal sender and a signal receiver. This is necessary to send video over distances of at least 20 feet. There shall be no obstructions that can limit the rate at which video is uploaded to less than 60 frames per second. Lastly, there shall be absolutely no overheating within the circuit. It shall operate from 0 to 45 degrees Celsius. This allows for a broad range of extreme household temperatures, and also discourages dangerous overheating within the mouse or box.

The last thing to consider in the mouse is that it retains some ability of water resistance. Cats don't particularly have an issue with exposing toys to spit, but it is a specialty of dogs. So, if a dog happens to get its mouth on the mouse, then there would be risk of exposing it to a water like substance that could destroy the mouse if not properly protected. It will also be a floor-based product, so the chances of running over a liquid spill are somewhat likely. It's possible to detect a spill with visual based programming, but is more difficult with clear liquids such as water. For this very purpose, the mouse must be able to drive over water, and accept a small level of surface moisture without affecting its function.

The second part of this project is focused around the box / home base / hole as mentioned throughout this document. The goal of the box is to house the mouse, charging station, and accept and send signal to and from the mouse. The dimensions of this box should be small enough as to not intrude on household function, and large enough to house the material. But all in all, the box should aim to be as small as possible while still achieving the design goals. To make manufacturing and sustaining more easily, there should be plenty of room for circuitry and the charging component in the hole. The goal is to make the box less than or equal to 1'x1'x1' to achieve the above goals.

4.2.6 Ethical Constraints

This is a section dedicated to the ethics that our team plans to uphold throughout the project rather than a section on the design of the product. We'll start by labeling our key characteristics we would like to maintain as a team, and then follow it with project ethics that will outline our responsibilities when creating the design.

Figure 60: General Ethics Outline (OZ Assignments)

The first topic we'll cover in transparency as team members. It is our goal to properly update our milestones and progress as time passes. Transparency in this sense means to tell the truth about our progress rather than lying to make it seem like there's more progress than is actually done. Not only does this inhibit the project as a whole, but might drag others into doing extra last minute work. In a corporate scenario, lying about work will bring more trouble than good. It's better to be honest about falling behind and requiring help than to lie and ruin the trust of fellow team members.

The second topic to be discussed will be the overall health of our team members. Under no circumstances will we jeopardize the health of each other for the project. This includes overworking and stress. To incorporate this into our project and assist in this goal, we plan to schedule times and due dates for things to be done, along with group meeting to check up on progress. This will allow us to use our time more efficiently in order to avoid sleepless nights spent writing reports that will make the due date in time. If anyone is overworked or has a job that takes most of their time, they will communicate with us and we can assist in splitting up the work accordingly.

Moving on to project ethics, we will prioritize safety, manufacturability, functionality, and realism. The constraints of safety and manufacturability are mentioned above, so this will quickly summarize the remaining two topics. The functionality of our project is the most important aspect as it defines a working product versus a product that wouldn't even make the shelves of a store. Efficiency will be our number one priority to make the project functionality a definite. We uphold our ethical responsibility to provide a viable product to any customers. The second idea to be discussed is realism. This is simply to create a design that doesn't use ideas that haven't even been made yet. For example, in an extreme scale, we wouldn't use time travel in our design goal since it is currently impossible. The second version of realism to cover is of the kind that exceeds our constraints and budget. Building a $3,000 part into our design just won't work for a small group of members working off their own money. Identifying realistic designs will help to eliminate ideas that may have taken time away from the project otherwise.

4.2.7 Final Ethics and Safety

As a team, we successfully accomplished our goals for ethics. We did not jeopardize the health and safety of individuals. During this pandemic, all team meet-ups were done while wearing masks to prevent the spread of disease to any members. We each did our parts of the project as to not burden any one person with too much work. All project designations were done in an equal manner to make sure each member had equal work responsibility as well. We also practiced the transparency of our work, keeping each other up to date with what was and what was not working throughout the project, and not promising anything that we couldn't accomplish.

We will now discuss the two types of safety precautions in this project. The first precaution we took was working with any sort of possibly dangerous equipment. When using soldering irons, a window would be open at all times to avoid harmful fumes. All wires were disconnected from their power source as well before being touched to reduce chances of being shocked.

For the mouse, obstacle avoidance was built in to reduce the possibility of tripping. When a person steps in front of the mouse, the mouse is set to avoid it. This is also shown in our demonstration videos. No dangerous wires are hanging outside of the mouse either, and it will not risk battery melting through a short since all wires are covered and spaced appropriately. For the mouse case, it is made of yarn, hard plastic, and non-toxic glue. So, any pets that play with the mouse do not risk harming themselves from possibly eating or trying to eat it.

4.3 Economic and Time Constraints

4.3.1 Budget Constraint

This project has a limited budget, as external funding was lost due to COVID-19, each student has agreed to set aside $125 of their own money to aid in the purchase of parts, a total of $500 dollars. This budget is flexible to a degree and going over budget, while not ideal, would not result in the cancelation or failure of the project. Nevertheless, cheaper parts will be considered over more expensive options. The current estimated cost of the AutonoMouse is $405, this is under the allotted budget.

4.3.2 Quantity Constraint

There must be a sufficient quantity of parts available in order to use them. An item that cannot be obtained in enough volume will be discarded for one that can be. If a component is not available in mass, it should be carefully considered before use. Part of the project is to create a replicable product, in order for that to be accomplished all parts must be attainable with relative ease.

4.3.3 Quality of Parts and Part Failure Constraint

A part is worthless if it is too cheaply made, the components must be of a decent quality in order to not waste money. Inexpensive part listings should be closely examined before purchasing. Even so there is a chance an item will have to be repurchased due to breaking in some unexpected event.

4.3.4 Exceptions

Certain parts are excused from economic constraints, such as the wheel, which will likely be 3d printed at virtually no cost. Wire, and other basic circuit components will be largly inconsequential on price or quantity.

4.3.5 Overall Time Constraint

Starting December 10th, this project will have one semester to be completed. There will be no extensions on this constraint, it has no flexibility and going overtime will have serious repercussions.

4.3.6 Part Arrival Constraint

Construction of the Autonomous will begin around January 11th, at least some parts will need to be capable of arriving before that date. In addition, all parts must be attained before early March in order to assure enough time to properly build, test and code the device. If a part cannot be acquired within the given time span, it must be replaced by one with a more optimal delivery time.

4.3.7 Testing Constraint

The machine must be finished by late March early May in order to be properly tested. Programming of the Autonomous will also take time, at least some assembly must take place before programming can begin to be tested, but the early basic code design can begin anytime once the Raspberry Pi is obtained.

4.3.8 Additional Class/Work Constraint

Some students on the team will be taking additional classes as well as senior design, not only will the classes themselves take up valuable time, but study, homework, and projects will also affect their availability to work on the AutonoMouse. Exams and Quizzes will be a distraction that might impact time further. Some team members also have jobs which take priority over working on this device.

4.3.9 Sleep Constraint

Sleep is an important and necessary part of the human biological system, 8 hours sleep per day is considered a healthy for the average person. Deprivation of sleep can be dangerous for elongated periods of time. As such, sleep will be a necessary loss of time.

Age Group	Recomended Hours of Sleep
0–3 months	14–17 hours
4–11 months	12–15 hours
1–2 years	11–14 hours
3–5 years	10–13 hours
6–13 years	9–11 hours
14–18 years	8–10 hours
18–25 years	7–9 hours
26–64 years	7–9 hours
65 +	7–8 hours

Figure 61: Required Sleep per Age Group (Medcor)

4.3.10 Food Constraint

Similarly, to sleep, food and water consumption must be maintained, and this process takes up time. Especially, for on campus activities, it may take up to 3 hours total of the day to prepare, eat and drink.

4.3.11 Lab Unavailability Constraint

There is a limited amount of lab space available at the University of Central Florida allocated to senior design. Obtaining the Lab for extended periods of time may prove difficult, and there may be wait time before use.

4.4 Environmental, Social and Political Constraints

4.4.1 Environmental Safety Constraint

The AutonoMouse not cause any lasting damage to the environment. Proper disposal instructions will have to be included to avoid littering or improper disposal of toxic substances that might be contained in some of the parts.

4.4.2 Social Constraint

The autonomous features a built in Wi-Fi capable camera that could be seen as invasive to the privacy of the customer. It is very important that the system be secure and private so that the buyer has nothing to fear.

4.4.3 Political Constraint

This project must adhere to the laws of the United States government. Proper copyright and patents must be obtained for sale of this product should it enter marketability.

5 Hardware and Software Designs

5.1 Initial Design Architectures and Related Diagrams

After discussing the possible design architectures in section 3.4, a conclusive outcome was reached for the prototype of our design. These designs are subject to change over time, but this section will accurately discern the architecture that the project will be centered around.

5.1.1 Initial Mouse Architecture

The wheels of the initial mouse design will use two motors to control one wheel each. This takes up a reasonable amount of space, but is the most efficient design to allow power and turning within the mouse. One motor would not allow the mouse architecture to move omnidirectionally. To help with stability, a loose ball or caster wheel will be provided in the back of the mouse. The caster or ball allows turning without requiring something to power them. They simply go with the flow of the mouse’s movements, and require minimal space.

A Raspberry Pi Zero W will be inside the mouse as the main controller. The Pi Zero W has Wi-Fi compatibility and allows for 1080p video recording, the most crucial parts in our design. It also is an incredibly small size of 30 mm by 65 mm, easily fitting within our mouse size constraint of approximately 2” by 3”, or 50.8 mm by 76.2 mm (height being neglected for the time being). The Raspberry Pi Zero W acts as a controller for the mouse, receiving and sending signals to the box that assist in its movements, mapping, and localization.

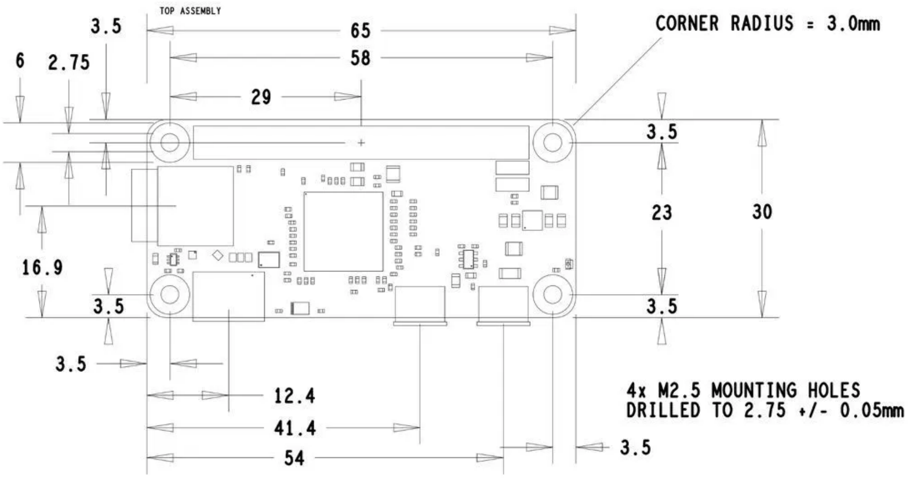


Figure 62: Raspberry Pi Zero W Dimensions (Electronics-Lab)

The fur of the mouse will be a short faux fur that’s non-toxic and soft. The best color will be grey to represent a mouse as accurately as possible. This can be derived from a toy store mouse with the same kind of appearance, or something of our own creation. Adding eyes to the mouse would add to its realistic nature, but isn’t entirely necessary if it can’t be done without creating a choking hazard in the process. So, this will be a possible addition to the future design with a focus on attempting to implement it. The tail will be made of a similar non-toxic fabric, no wire involved as it could pose as a danger after enough rough-play.

5.1.2 Initial Box Architecture

The materials of the box will mostly be wood based as it allows Wi-Fi signals to pass through with ease. If the box was made of metal, it would heavily impede the signal and render the signal obsolete. It will be a 1’ x 1½’ x 6” box that will have a hole that houses the charge station and the mouse. The extra ½ foot protects from curious cat paws trying to reach in and grab the mouse straight out of its home. In a separate compartment, there will be the other brains of the operation, a second Raspberry Pi that will be able to transfer information with the Raspberry Pi located in the mouse. This will be doing most of the processing of information such as discerning sensor and video information to help path the mouse. It will also be the initial receiver of all input coming from the web-based app that the user can use to control the mouse. That data is then pinged to the mouse.

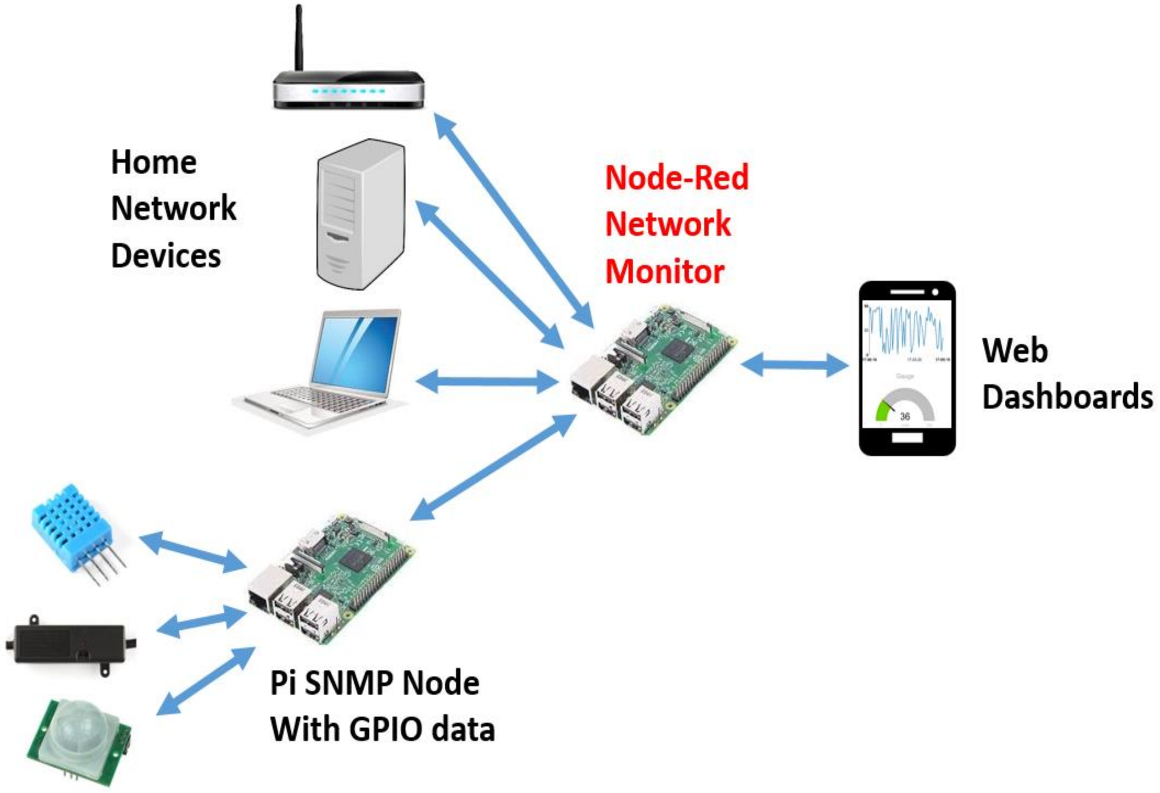


Figure 63: Raspberry Pi Communication between Devices (Fun Tech Projects)

As mentioned previously, the hole houses the charging station. In this case, a flat and small area wireless charging station will be used. The mouse enters the hole and parks on the charging station after reaching the end of the hole. It uses the copper coil technology to wirelessly charge itself, for hassle-free charging, and longevity while the user is away. This is one of the possible features that might be downgraded depending on constraints, including time and budget. It is a stretch goal, and may be transferred over to direct wired charging later.

For the Raspberry Pi in the box, the Raspberry Pi 3 will be used. It provides both Wi-Fi and Bluetooth options. Since Bluetooth between the mouse and the box is a viable option, it's great to have a board that can accomplish both Bluetooth and our main option, Wi-Fi. This should allow for versatility within our project and open us up to more options given the opportunity. One of these being allowing the user to play with their pet while they're home using Bluetooth instead of Wi-Fi, since they'll more than likely be in range. Another additional option within this board is a second camera slot. If chosen, the user would be able to not only view the room from the point of the mouse, but the box as well.

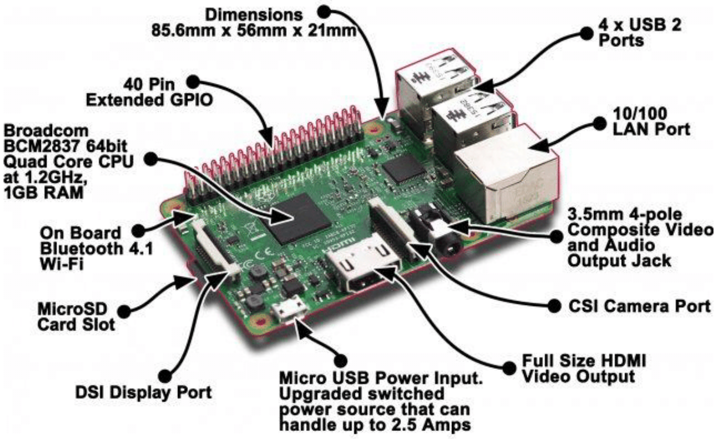


Figure 64: Raspberry Pi 3 Outlay (Agrawal)

5.1.3 Initial App Design

The web-based app will be able to show the video of the mouse while simultaneously allowing control over the mouse movements. For this to work, the Raspberry Pi and the phone / computer need to have access to the same web server to stream the video. The goal is to have a small control overlay on top of the video feed much like that of a video game to move the mouse around via motors. The phone / computer will send the actions over to the box, which will then relay those messages to the mouse. To make sure the mouse is receiving the signals, it will ping back when it receives a command. If not pings are received by the box, then the boards will attempt a reset, or use another pre-preprogrammed fail safe.

Either Python of Java will be used to program our app and the respective technology it accompanies. Not only are they widely used throughout the world, but Python retains the ability to translate most languages, allowing it to crossover from program to program with ease. These languages provide a lot of options for our project, and can allow for deep-learning into object identification. Having the program be able to identify humans and cats is a secondary goal of this project, but would upgrade its performance and possibilities exponentially.

5.1.4 Deep Learning

Deep learning is probably the largest and most intimidating stretch goal of this project. It seeks to identify objects and use an appropriate command in response to recognizing it. It could be as simple as highlighting it, or for our use, addressing the motors to avoid it. Deep Learning is an ever expanding field that currently holds a lot of technology's attention for its abilities to adapt and respond to numerous situations without actually being programmed to react in that exact way.

For example, through trial and error, a program mimicking a human can learn to jump over an obstacle without the programmer directly saying to “jump”.

Our deep learning machine will accept an input layer, pass it through weighted neurons, and attempt to give the correct output. If it is wrong, it will adjust the weights of those neurons to adapt its understanding. After enough training, the neural network will be able to identify cats, humans, and possibly other pets by utilizing training data. The figure below shows the equation our deep learning algorithm will use to train its identification skills.

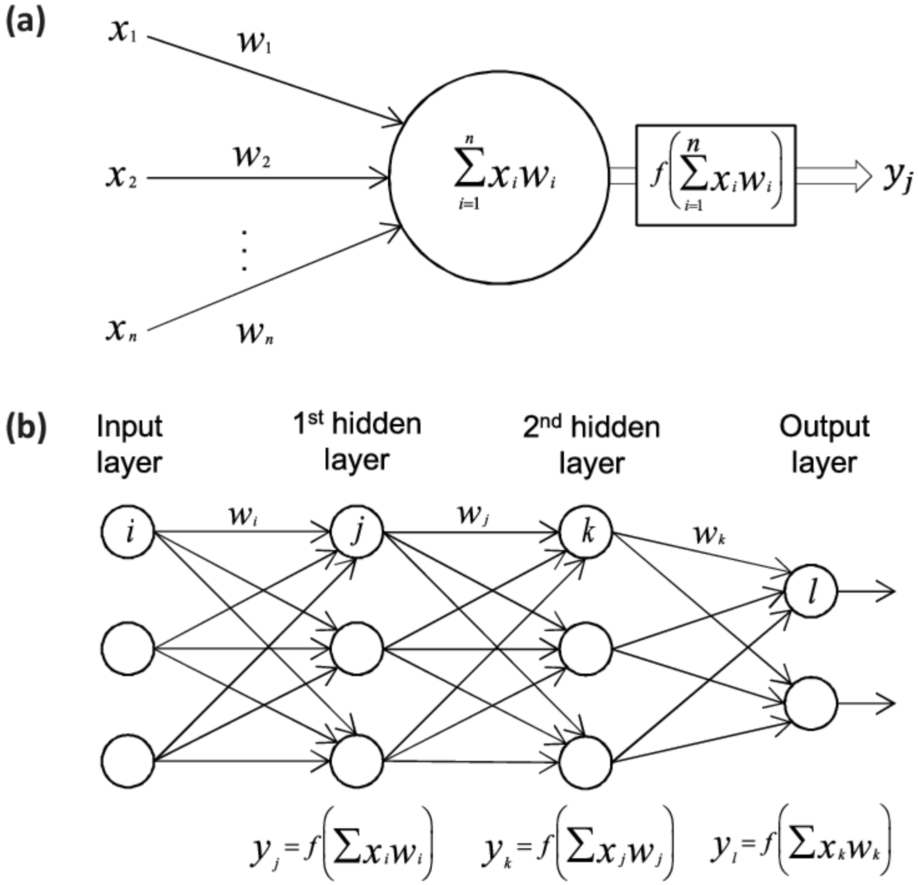


Figure 65: Neural Network with Corresponding Equations (Research Gate)

5.1.5 Controlled Responses

Two settings were decided on for the AutonoMouse; a user-controlled mode, and a general autonomous mode guided by visual mapping. Both of these modes are necessary in order to accomplish the project goals and provide sufficient entertainment to the pet.

Autonomous mode is the general mode for playtime at any time. It can activate throughout the day without any action demanded by the pet or the owner. The mouse will leave its hole and run around for a little bit before running back to its hole. To make this even better with time, the mouse should be able to detect obstacles along the way and steer away from them. This will be done through either visual sensors or ultrasonic sensors. Visual sensors like cameras are much more accurate, but require an extreme amount of processing in order to render information. Ultrasonic sensors can constantly send out signals that they then wait for to return. This allows them to locate objects directly in front of them. However, this doesn't do as well on curved or angled objects as the signals bounce in different directions rather than back at the sensor. Ultrasonic sensors would also be difficult to incorporate into the aesthetic mouse design we are aiming for, and therefore should only be considered as a last minute option.

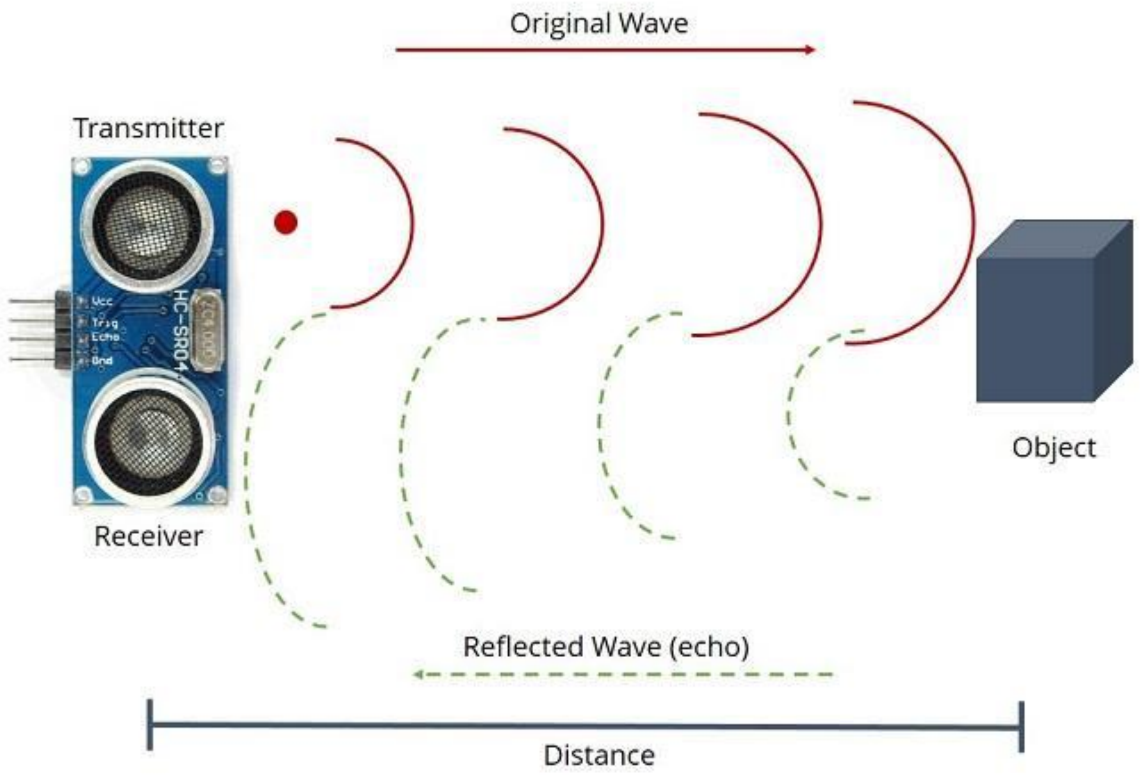


Figure 66: Workings of an Ultrasonic Sensor (Random Nerd Tutorials)

Using a vision based sensor works to map out the area as well as provide video to the owner. This is the idea we will be using for our mapping and localization. By taking in visual shots of the surroundings and applying various filters over them, we can discern the general layout of the room as well as the robot's current position in it. The downside is that this is much more difficult to implement than ultrasonic. It's the difference between letting the sensor do all the work with ultrasonic, and letting the program do all the work with visual mapping.

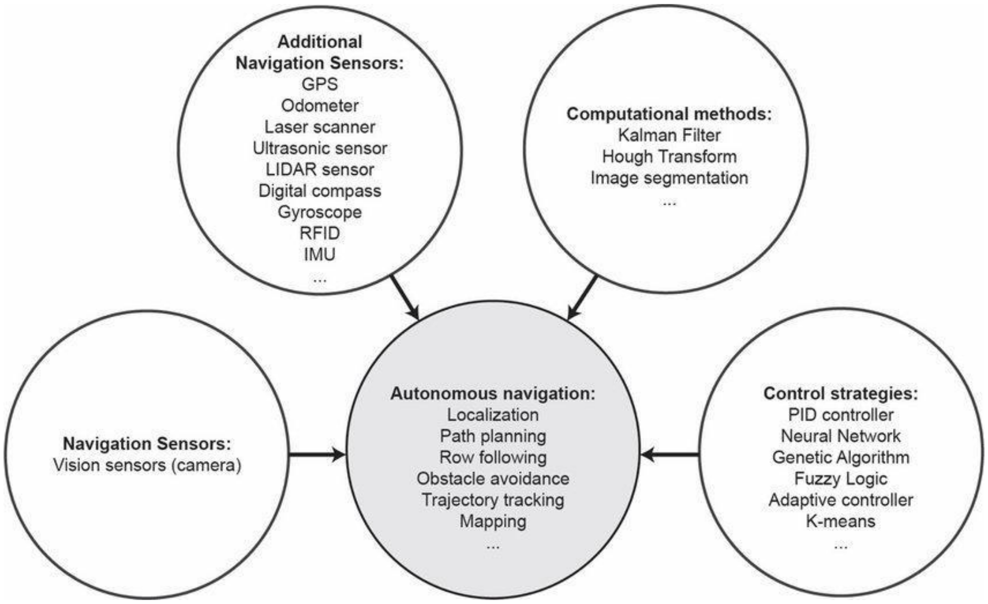


Figure 67: Visual Mapping with Filters (ResearchGate)

The user-controlled mode has been stated previously in this document, but will be summarized in this paragraph for the sake of continuity. This requires minimal processing of information, with absolutely no mapping involved. The vision and movement is completely based off of the user and what they do with the mouse. A web-based application is used to provide the controls, that each respectively control a direction in which the mouse goes. Forward will drive both motors forward, left will drive the right motor forward, right will drive the left motor forward, and reverse will drive both motors backwards. The goal is to make it as easy to use as possible without any steep learning curve. Through the app, the user will also be able to switch back to the normal autonomous mode as desired. After a certain amount of allotted time, it will automatically switch back to being autonomous.

5.1.6 Final Design Architecture Continued

This is a continuation of the Final Architecture note in section 3.5. Our app design was consistent with our initial design, controlling the mouse with a video feed and video game like controls. The mouse buttons are labeled as left, right, forward, back, and stop. However, the last two sections of Deep Learning and the sensors stated in Controlled Responses were not included in the final design. Deep Learning was going to be used to detect an image of the cat in order to play with it or avoid it. It was also going to possibly be used for pathing. However, this was too ambitious for us, so we could not include it in the project. As for the controlled responses, neither Ultrasonic or the camera were used for editing the mouse's movements. Instead, a laser sensor was used for the detection of obstacles. In the latter section of controlled responses, the user mode is discussed slightly. This part remained exactly the same as described.

5.2 Mouse Subsystem

The mouse is one of the main two components of the project, with the other being the box from which it leaves. Inside the mouse houses numerous components to assure it runs properly. The parts and explanations listed below are part of the initial design and are subject to change as trials continue.

Our mouse subsystem originally consisted of two SE18K1ETY motors with an RPM of 4050, a starting torque of 27.72[mN•m], and rated load of 5[mN•m]. This would provide our mouse with substantial enough power to overcome terrain obstacles like carpets and ground unevenness. However, including this in our design would have increased size substantially, as well as forcing a larger battery voltage of 12V DC. Therefore, we decided on using the COM0805 motors with a much smaller RPM of 420, a rated torque of 1.388[mN•m], and only a requirement of 6V DC. This version is much cheaper than the SE18K1ETY and only sized at a thickness of 10mm x 12mm, compared to the 18mm x 18mm thickness of the SE18K1ETY. The lower voltage also provides the design with lower space consumption. The main issue is that the torque and RPM rating largely decrease with our decision. In the future, our design might change if this motor choice fails to work on common house terrains.

The COM0805 will be powered a BGN800-5FWP-A800EC BatteryGuy battery pack. This pack is capable of supplying a 6V and 900 mAh power to our motors, which is exactly as required for the COM0805 motors. The battery pack will also be used in conjunction with a voltage regulator to manage the output to the other components in the mouse. This voltage regulator will be a L7805CV used to provide a stable 5V output to the input of the Raspberry Pi Zero W, which also then converts that voltage into a 3.3V output by itself.

The Raspberry Pi Zero W will be used as a median for most technologies used in this mouse. This includes the camera, Wi-Fi capabilities, motor control, and receiving battery input. The Raspberry Pi Zero W includes a CSI camera connector that will attach a compatible camera. Our choice for this design is the PiCamera since it is already designed for use with the Raspberry Pi. The camera module attaches to the Raspberry Pi via a ribbon cable. The camera can then be enabled by the Raspberry Pi's configuration menu when hooked up to a computer. How this works will be further discussed in a later section involving software design, along with the built-in Wi-Fi card.

Attached to the Raspberry Pi is going to be an MMA8652FCR1 accelerometer that can detect changes in the mouse movements. When the cats plays with the mouse, the accelerometer will react to the abrupt change in pace and notify the mouse to respond to this. It will then pace itself back to the hole. The accelerometer is also useful in justifying its current location by working in tandem with the odometry to provide extra directions. For example, if the odometry suspects that the mouse has turned a full 90 degrees, but the accelerometer reads different, the calibration for mapping can be adjusted accordingly.

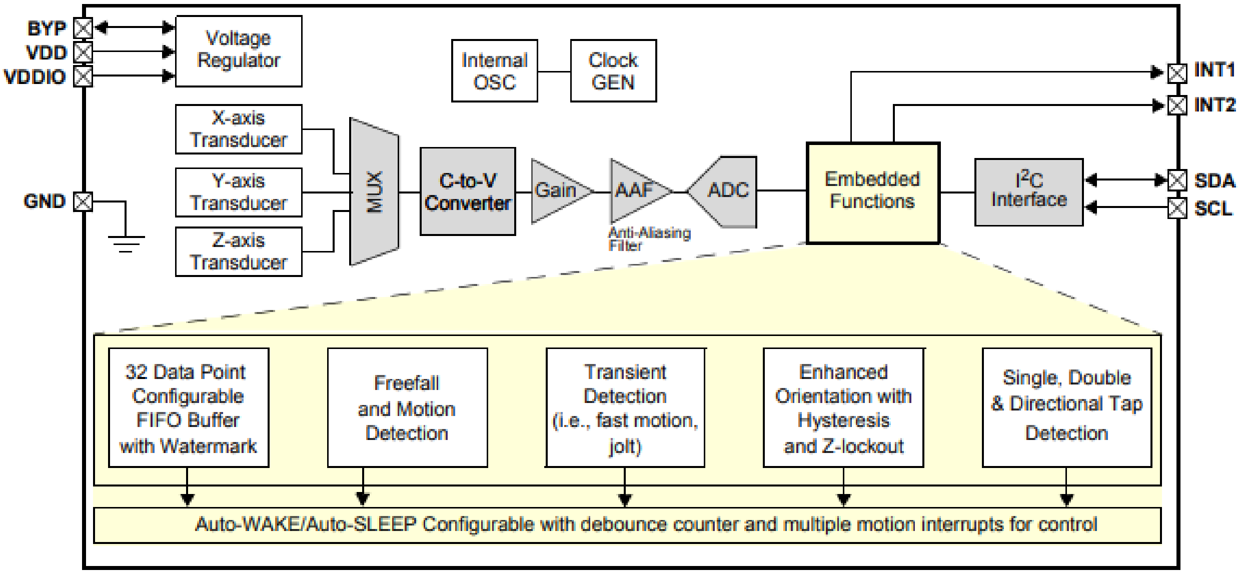


Figure 68: MMA8652FC Block Diagram (NXP)

The mechanical parts of the mouse are the wheels, axles, and wheel bearing. The bearing we will be using is a TruePower 5/8" roller ball transfer bearing. The roller ball will be equipped at the back of the mouse attached to the bottom bracket of the mouse. This has no electrical parts attached to it. It is a completely free rolling ball that will allow the mouse to easily turn, while maintaining stability. The second option for this was caster wheels. However, due to their occasional disagreeable turning nature, the idea was scrapped. A simple ball bearing produces much more stability, ease of use, and turning radius. Thirdly, a second motor and wheel could be used, but would disagree with the turning of the mouse. However, it would give the mouse much more power if our current design doesn't have enough to cross specific house terrains.

For this design, it is important to have wheels large enough as to not allow any carpeting to reach up and hit the bottom of the mouse as it moves. This would impede its movement and possibly render it incapable to drive. So this project will contain Yeeco plastic toy car wheels, with an outside diameter of 1.7", width of 0.5", and shaft hole diameter of 0.12". This fits perfectly with the shaft diameter of the COM0805 motors as well, so there is no extra need to supply more axels unless there are extra complications. The main issue with these wheels is that they are plastic, so there is the possibility of slipping on surfaces with low traction, such as hardwood floors. This can be amended by applying a rubber coating around them if they do not work as intended. The wheels will also be partially hidden inside the mouse, and may further decrease the length between the floor and the base of the mouse. However, larger wheels will increase the size of the mouse, which is a constraint we need to keep low at all costs. Further test runs will check to see if this is an active problem with our design.

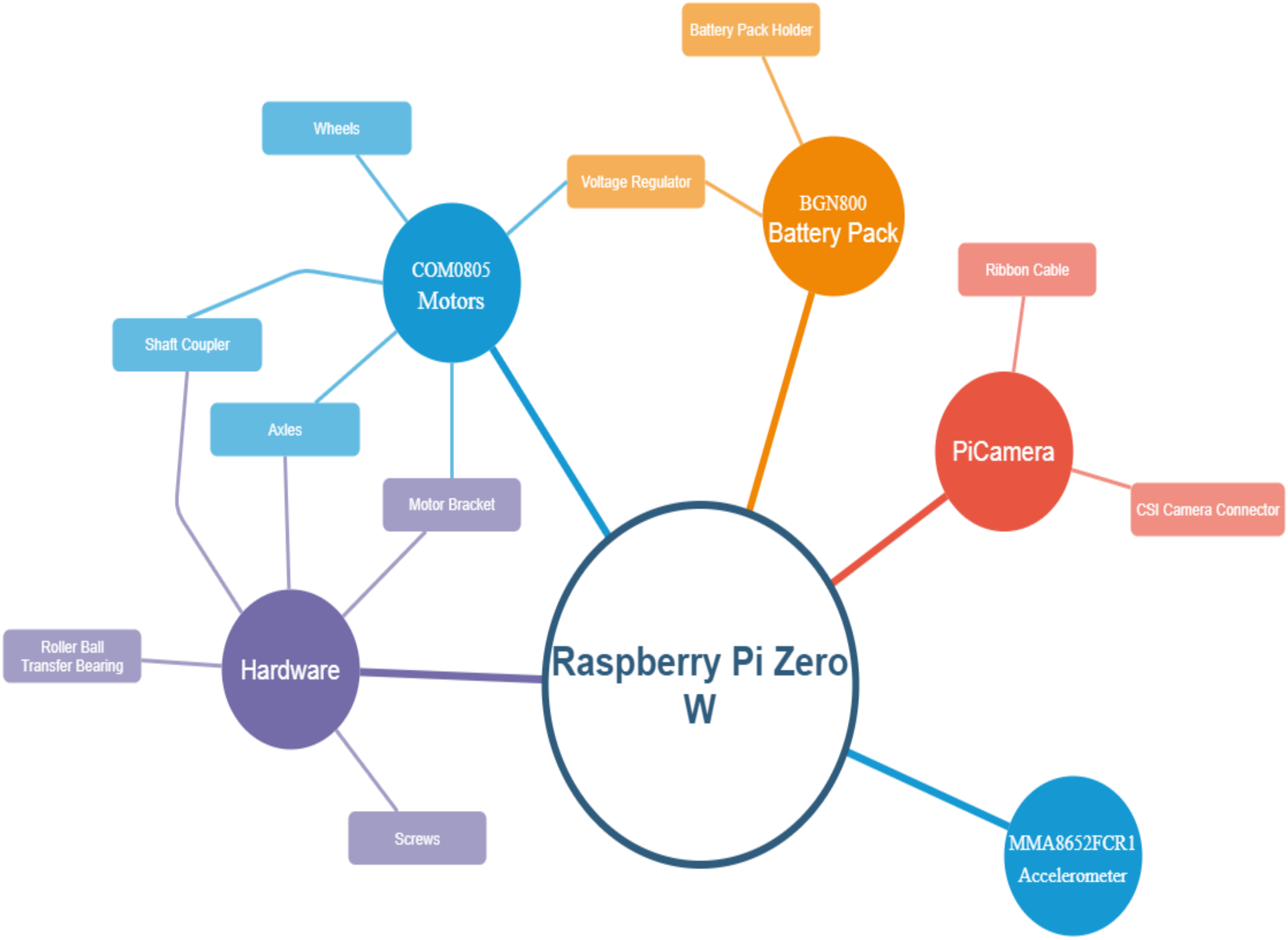


Figure 69: Mouse Part Diagram

The following is part of a just-in-case scenario where the axles are not long enough, or are having unexpected issues that inhibit the design. To extend the shaft of the motor or replace it with a new axle, the Twidec 3mm-6mm brass shaft coupler will couple the motor shaft with a new axle. The 3mm hole will fit the COM0805 motor shaft, and the 6mm end will adapt to any further axle extension. This will also be useful to consider if for any reason the keying in the motor shaft causes any issues for the design.

5.3 Box Subsystem and Breadboard Test

The box is the main control system of our design. While the mouse roams around, the box controls it and runs various algorithms to test its location and map the room. The box is much more software oriented than the mouse due to its controlling nature, however, assumes the most parts as well that will be discussed in the following paragraphs.

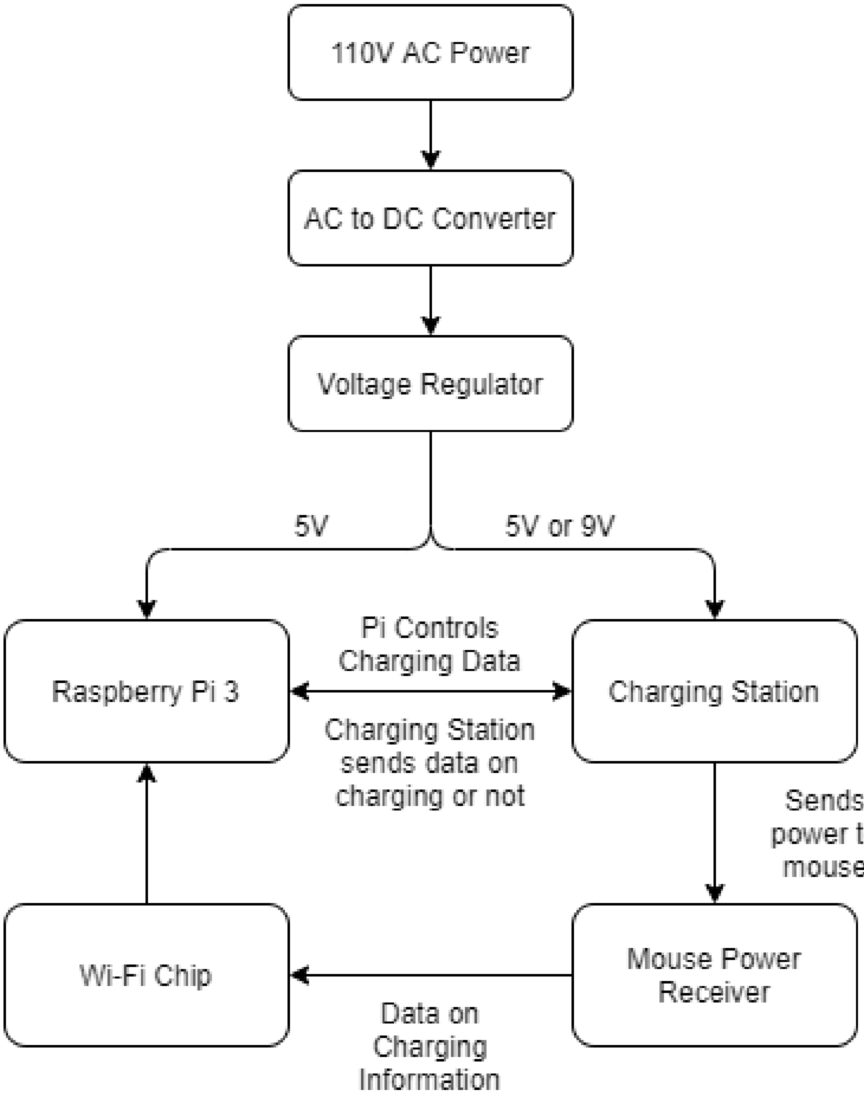


Figure 70: Simple Power Diagram for Box Subsystem

This subsystem holds a Raspberry Pi 3 that will be the basis for most of the calculations done for this design. This Raspberry Pi also accepts a 5V input to produce a 3.3V output. It will be used as a controller for the charging station, location and mapping of the mouse, as well as the main communication link between the user and the mouse. However, instead of a battery pack previously used in the mouse, there will be a direct connection to a wall outlet, which supplies a consistent 110V AC. This requires that the box is hooked up via an AC to DC converter that will be able to supply a stable supply of power to our box. The two major power components to consider are the Raspberry Pi 3 and the charging station included in the box. The Raspberry Pi will require a 5V input, while the charging station will require a voltage heavily reliant on the size of the charging station we decide to use in this project. This will likely require a consistent 5V or 9V AC to DC converter. However, with the addition of the Raspberry Pi, only one AC to DC converter will be needed, and therefore will require a voltage regulator to change the voltage if 9V is required for the charging pad.

5.4 Charging Subsystem

For the sake of ease, our design will attempt to use a 5V requirement charging station in order to supply the mouse with 7.5-10W. To design the charging station, we will require a wireless charger PCBA, a wireless charger receiver, and soldering materials to connect it all. Underneath the box will be a connection branching from the Raspberry Pi and a separate power source. This is for both control, and supplying the necessary power to the wireless charger PCBA respectively. The PCBA will be underneath a base in the box, and will be constantly active in order to charge the mouse when it docks. For maintenance purposes, the mouse will send a signal to the box that it is charging when it is supposed to be. This will activate a light on the box, as well as send a message to the user to show that it is either charging or not. This will be completely for maintenance and problem identification purposes.

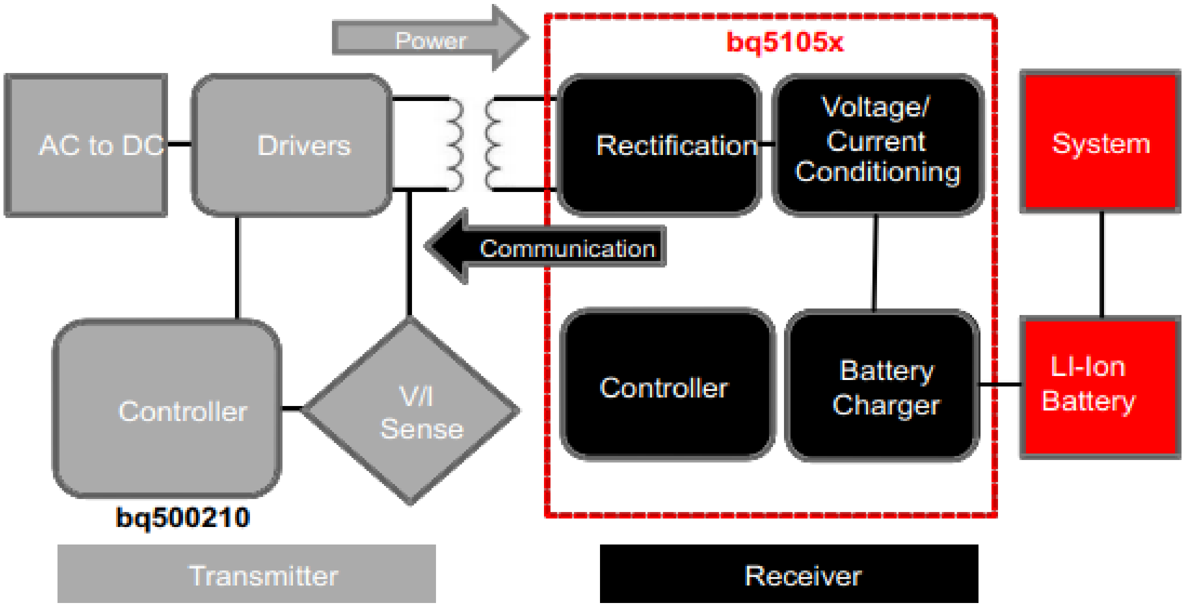


Figure 71: Block Diagram for TI bq5105xB Wireless Power Receiver and Battery Charger

The mouse will contain the wireless charger receiver that will utilize a USB C type connection to recharge the battery. Between the charging station and the battery pack will be a current sensor circuit that tells the Raspberry Pi when it is and isn't charging. This schematic uses an LM741 Op Amp, a 1KΩ resistor, and a 470Ω resistor. The 1KΩ resistor will connect to the positive terminal of the Op Amp. Before that will include the current signal coming from the charger. At the output terminal of the Op Amp will be the 470Ω resistor to produce an output current. This is the current that the Raspberry Pi will receive and detect to warn the user of charging or lack thereof. The point of adding the Op Amp is to reduce the possibility of not detecting smaller current signals. If our wireless charger ends up sending a small current into our receiving coils, but isn't detected, then we would assume that the design isn't working at all, as opposed to working, but receiving low power.

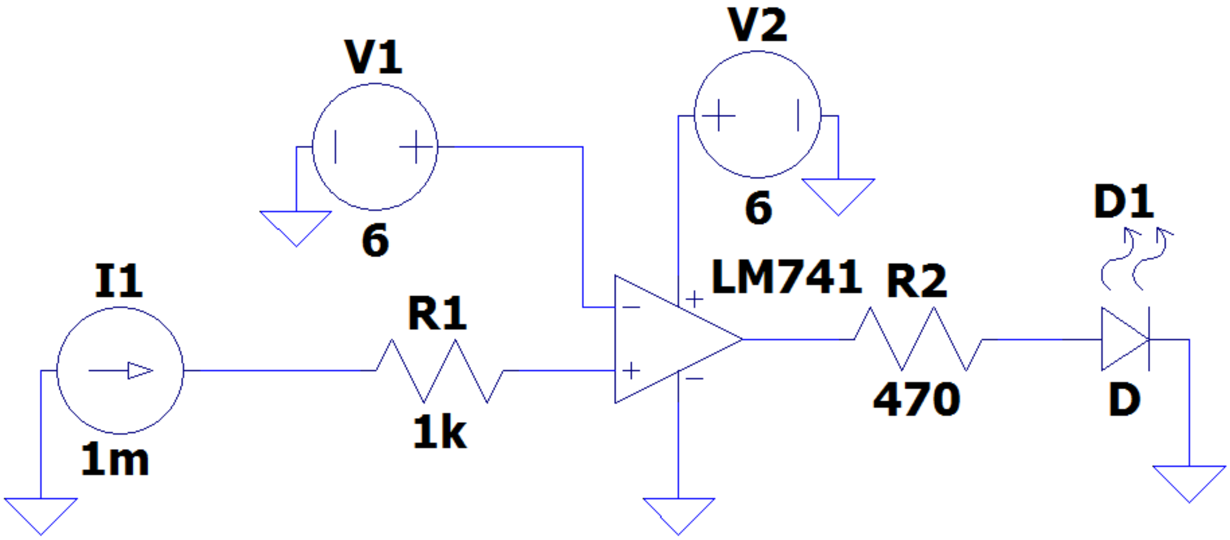


Figure 72: Current Sensor

5.4.1 Testing

LTspice XVII was used to test many of the circuit designs stated above. One of the results of this is shown in part 5.2.3. Since not all components for these designs are readily available, breadboard testing was not used as of now. This may change in the future. For parts that were not on LTspice XVII, similar substitutions were used.

The first test was for that in Figure (1), which showed the current sensor circuit. The theory was that no matter how small the current was, it would get boosted high enough to easily be sensed by any secondary tool. In this case, it needed to be sensed by the Raspberry Pi. To test this, a 1 uA current source as opposed to the 1m current shown in Figure 62.

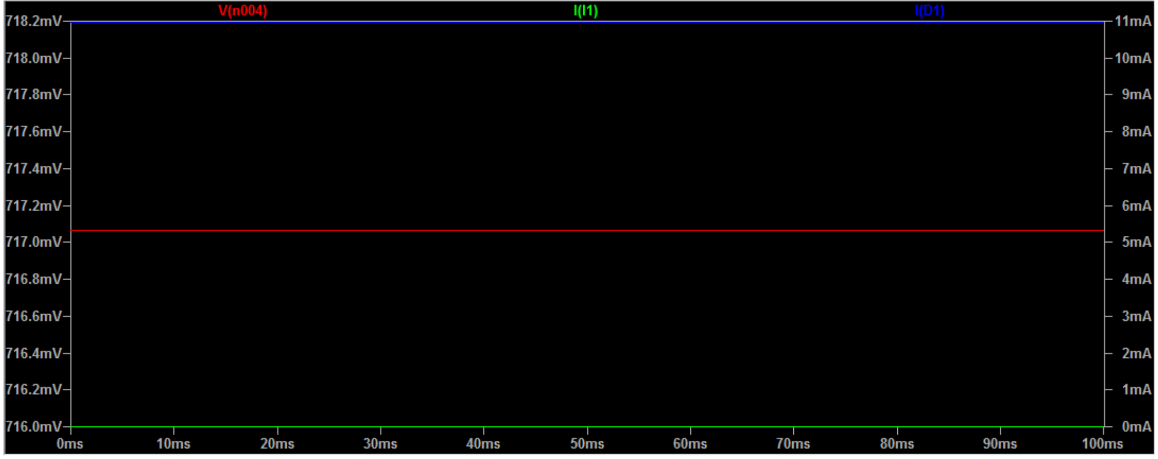


Figure 73.1: Current Sensor with 1uA Input

This was the received output of the circuit. The red line refers to the voltage after the 470Ω resistor, which produced a 717mV output. The green line is the input current of 1µA, which is extremely small. However, it produces a current through the 470Ω resistor of 11mA. This is substantially larger than the input and much more easily detectable by the Raspberry Pi. When a current of zero is entered as an input, the current value appears in fA, which is practically negligible. So, there will be no error with a current that is practically zero, but current that is substantial enough will notify the Raspberry Pi.

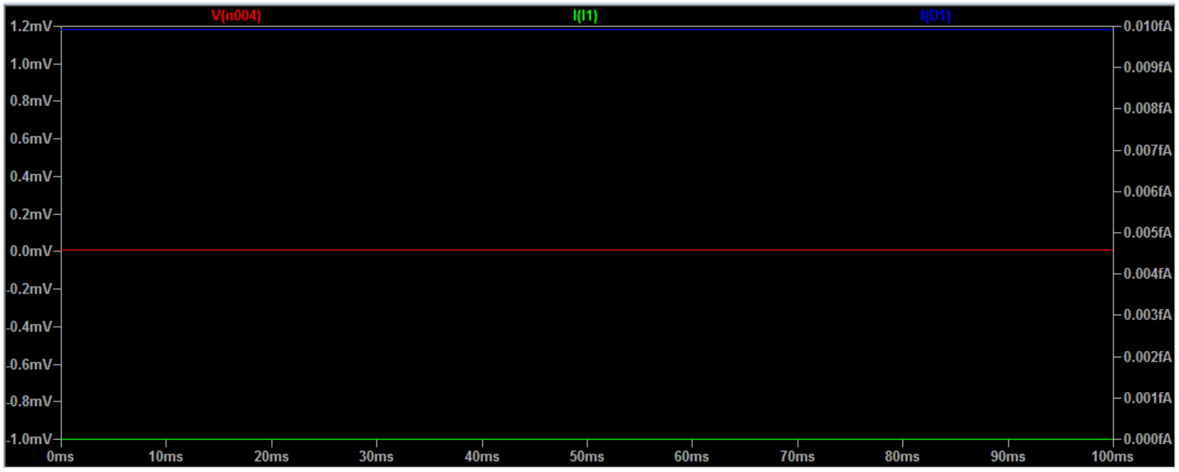


Figure 73.2: Current Sensor with 0A Input

5.4.2 Final Charging System Edit

The charging subsystem remained relatively the same throughout the process. However, it was not controlled by the raspberry pi, since that was removed from the box design. Instead, it is always on and waiting for the mouse to dock in order to supply power to it. In order to have some idea of the battery level on the mouse, a comparator was introduced to the mouse design. The comparator would check the battery against certain voltage levels and send through the voltage when the battery dropped below a certain designated voltage. The voltage input is the 5V output from the Raspberry Pi Zero W. It is used with a voltage divider circuit to change the voltage values in the comparator to the voltages we want to check the battery against. Shown below is the schematic for representing the battery in conjunction with the comparator.

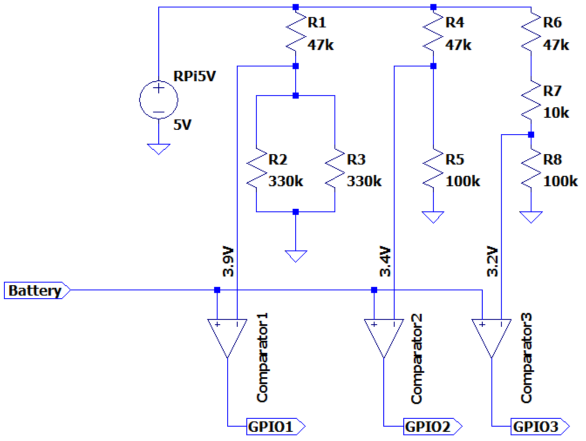


Figure 74: The implemented battery state sensing comparator circuit

5.5 Additional Features

There are certain features that would elevate the effectiveness of the AutonoMouse as an entertainment device. These features would require completely new software and hardware design, but would definitely benefit the project if implemented. For now, these design ideas are considered stretch goals, as they are welcome with enough time, but not necessary for proper function. Some of these ideas might be completely different from our original design, but work off of the use of Wi-Fi compatibility and follow the same general structure involving the Raspberry Pi as the main form of communication.

5.5.1 Laser Pointer

One of the tried and true toys for cats to play with is the laser pointer. It operates on the cat's predatory instincts to hunt it as it flashes around the room. Playing with a cat with a laser pointer is one of the most entertaining activities around for both humans and cats, and could easily upscale this project into a full-on entertainment center. When considering the laser pointer, it would be best to attach it to the box rather than the mouse, for extra space and control.

Shining a laser pointer across a room would be motion heavy. In this sense, it would be optimal to require at least two different motors to control the direction and speed of the laser. To hit different angles of the room, one motor has to control the angle of the pointer, and another would control the direction. These motors would hook up to the Raspberry Pi 3 in a very similar manner that the motors in the mouse were attached to the Raspberry Pi Zero W.

The goal is to have the laser pointer have enough range to move around a room freely, without and rigidity. If all the laser pointer can do is point at walls, then it essentially loses its function as an entertainment device. The software required for the laser pointer would be simple, just clarifying a joystick for the movement of the laser. The main issue is getting the laser to adjust fast enough. Since cats will be hunting it, a stagnant laser will provide no benefits to our design. With just the motors stated above, we might hit a plateau in range and speed.

5.5.2 Treat Launcher

Cat treats are one of the most basic ways to train a cat. When they do something correct, they're rewarded with a treat. Owners also often like to see their pets happy when they're away, so there's a direct correlation with adding a treat launcher to this design. One of the main purposes is also to attract the cat to the idea of this project, rewarding them for their time of playing. A treat launcher is more mechanical than anything, and requires very little coding to get it working properly. It is, however, also very motor heavy, launching and resetting multiple times over.

Launching anything requires a high instant force, rather than a steady force over time. Meaning, we can't just attach motors to a launching pad and expect it to work. There are two common designs that should be used to implement this idea into our project. The first design is a spring loaded wind-up mechanism. The motor brings back a spring, holding a tray for the treat, then releases the spring once activated. This will create an instant force in the spring the launches the treat forward and into the air. The required hardware component for this will likely be a 3D printed launch setup, a compression or extension spring, a motor capable of carrying the max spring force, and fishing line to bridge between the motor and the launch pad. Any 3D printed items stated in these ideas will be designed using computer-aided design (CAD) software.

The motor would be the SE18K1ETY motor stated previously in section 5.2 since it holds plenty of power to compress a 3 lb. extension spring. The main reason to go with the extension spring over the compression spring is because launching with a compression spring might overextend the launch pad, and reach further than it is supposed to. In this case, it could collide with another part of the launcher and cause damage to the system. However, the extension spring has the limit of only compressing as far as its materials allows.

The second option is to use friction and motor speed to solve the problem. Instead of using springs to launch, a rubber wheel spun by a motor with high RPM would be able to send the treat a sizable distance. The SE18K1ETY motor would also be used for this as it boasts an rpm of 4050. This is less reliable as the previous option though. Since it requires the gravity of the treat feeder to slide the treat into the wheel, there are complications that can occur with the accuracy of the treat positioning.

5.5.3 Treat Feeder

For this design, a treat feeder can be its own singular project, or be paired with one of the treat launchers shown above. The launchers have the job of adding entertainment value to the project. Automatic treat feeders already exist, but again, this one would work through an app, so the owner will be enabled to feed treats to their pet at their discretion.

Since everything in this project is automatic, motors are required in most places to move our mechanical parts around. The treat feeder is no different. Similar to a bubblegum machine, treats will be slotted in a holed circular plastic plate, below a container full of treats. The metal plate will be rotated by a COM0805 motor, which will then dispense the slotted treat onto a surface. For cases of the treat launcher, the treat will be placed into the launch pad. In the wheel-guided launcher, there is a different technique. Since gravity will be in play, the treat will need to be fed into a steep tube, the exits out next to the entrance of the spinning wheel.

5.5.4 When-Away Syncing

It would be tedious if the owner wanted to play with their cat while they’re away from home all the time, work being the main contributor fort this. Our design has talked plenty about autonomous mode and controlling the mouse through an app. However, a possible situation arrives where the owner may not want the AutonoMouse running 24/7, or that they possibly forget to shut it off. This brings about the When-Away Syncing. Much like the Nest Learning Thermostat, the design would be able to detect when the owner is away from home, and when they are not.

By using a motion detector such as the mini PIR motion sensor by Seeed Studio, the project can track motion around the house to determine a routine for activating the AutonoMouse. The motion detector would communicate with the box, tracking movement at an adjustable height. The height adjustment is required to track humans instead of other animals. If the motion sensor is too low to the ground, it might track cats and create a routine exactly opposite to what we want to integrate.

This could also work the same for large dogs. Optimally, the sensor would be at a height closest to the maximum height of the shortest person in the household. It will not be directly connected to the box structure since the height would be impossible to adjust. More than likely, it would be attached through wires connected to the Raspberry Pi, and placed on a high shelf in the household.

The motion detecting architecture could be transformed into a camera that recognizes human faces. This would eliminate the possibility of other things triggering the motion sensor, and improve the design overall without a doubt. However, now we face a matter of privacy. The AutonoMouse is already essentially a roaming camera, but at least operates at the owner's whim. However, adding a camera to the main living space of a home or apartment intrudes on the user's privacy to an almost extreme extent. Add this on to the fact that our cameras are connected to servers in order to stream to the app, and the entire project turns into a spying mechanism that can go incredibly wrong if the security isn't high enough.

To discuss the motion sensor, we first need to address its physical connections and hardware. The motion detector will be mentioned earlier has a pin connection, which requires it to be directly connected to the Raspberry Pi or other intermediary. In this case, we will use the latter. To promote stability, the motion sensor will be attached to a miniature breadboard. The voltage will be supplied from the Raspberry Pi, through solid core wire, and into the motion sensor, then looped back around to a ground. A third wire will be routed into the feedback from the motion sensor and into the Raspberry Pi for data processing. Since the motion sensor provides its own circuit and resistance, no extra component will be needed in series or parallel with it. Provided below is a simple diagram for how it will be connected to the box.

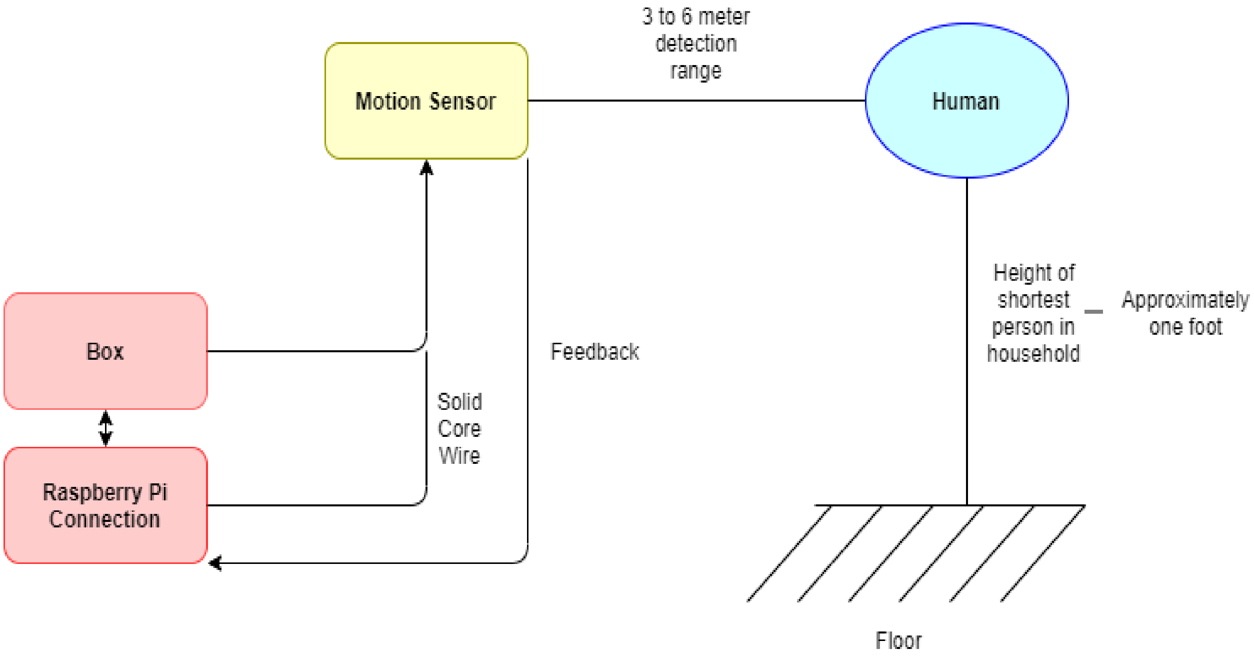


Figure 75: Motion Sensor Function Diagram

Next is the software aspect of the design. Given will be a simple summary of how the code will be implemented, and not a full detailed explanation, as this is only a possible architecture of the project. Data is fed from the motion sensor into the Raspberry Pi as shown in Figure 75. The motion sensor sends this data as a digital output, meaning the code will only have to process data as a high or low input. For our program, a timer would be placed to start a 24 hour cycle. This cycle would continue as motion is logged throughout the day from the motion sensor. After each 24 hour cycle, the data from the log will be processed. After each day, the logs will be compared to each other, and fix any holes in their data. These holes refer to motion that wasn't logged from previous days, but appeared on other days. To account for weekends, each cycle will also include a 7 day log as well, assuring that holes aren't filled in for motion as a result of a day off. This is the basic programming summary, and can be adjusted in more detail if acquired into the project design.

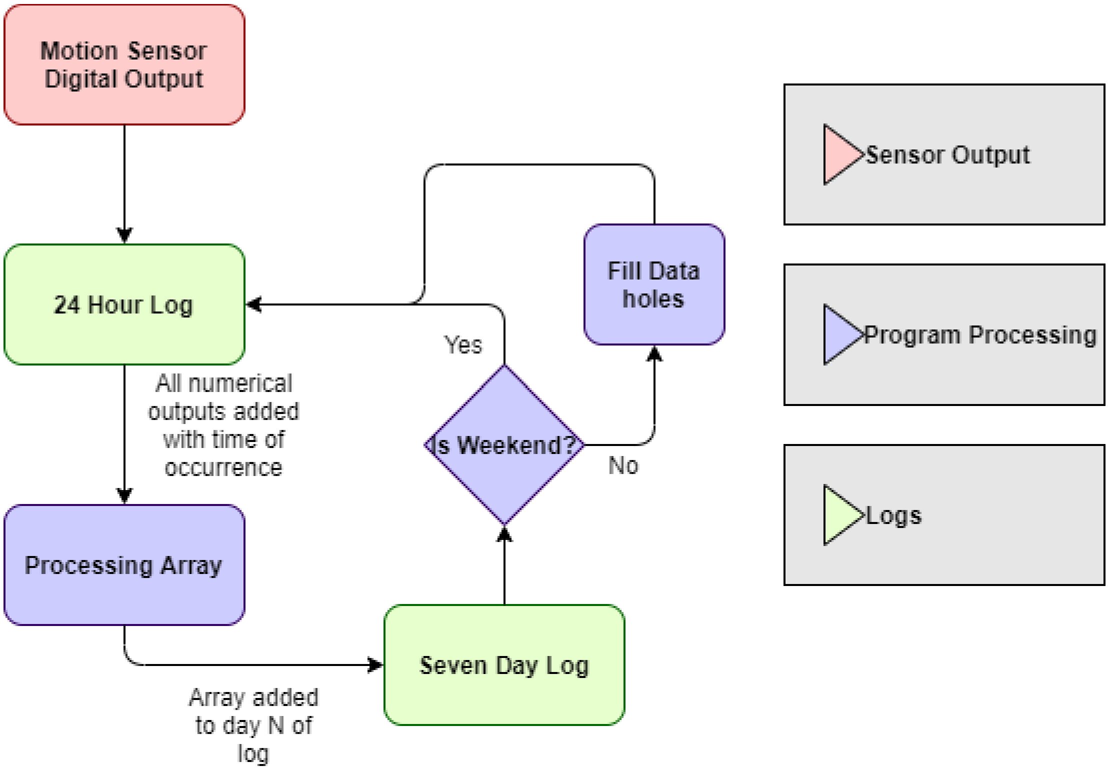


Figure 76: Motion Sensor Programming Diagram

5.5.5 Maintenance Signals

Many components in this project will need to be tested and evaluated to assure quality control. Examples of this are the mouse wireless charging, the app to mouse movement control, and room mapping. Since these are such important features, it would be ideal to include sensors in our designs that will notify both us, the creators, and also the owners of anything that isn't working. These notifications can be sent by either app, or through LED lights placed on the box. Both would work best if possible.

The Raspberry Pi will need to receive a signal from each of these sensors to determine where the problem lies. The mouse wireless charging will send a signal when the current detector does not detect a current when it is supposed to be on charge. The mouse movement is easily recognizable by the owner by the lack of mouse movement with the camera, or by in-person observation.

However, to recognize this electrically, the accelerometer in the mouse can track the force from the mouse's acceleration. If one of the movement buttons in the app is being pressed, but the accelerometer doesn't detect anything, then it can be concluded that the mouse movement applications aren't functioning properly. Lastly, the room mapping relies on the camera, odometry, and the accelerometer to map the room. The number of applications involved in this makes it difficult to test. Tracking map progress is almost impossible since there's little to no way to test for map completion. However, by checking the camera, odometry, and accelerometer connections, we can essentially test the working order of the mapping. If one of these factors isn't functioning, then the mapping shouldn't be working.

No extra sensors are required to test these functions, so the signals will be entirely software based. However, a simple circuit can be devised to send the outputs signals from the Raspberry Pi to the LEDs on the box for a simple maintenance notification.

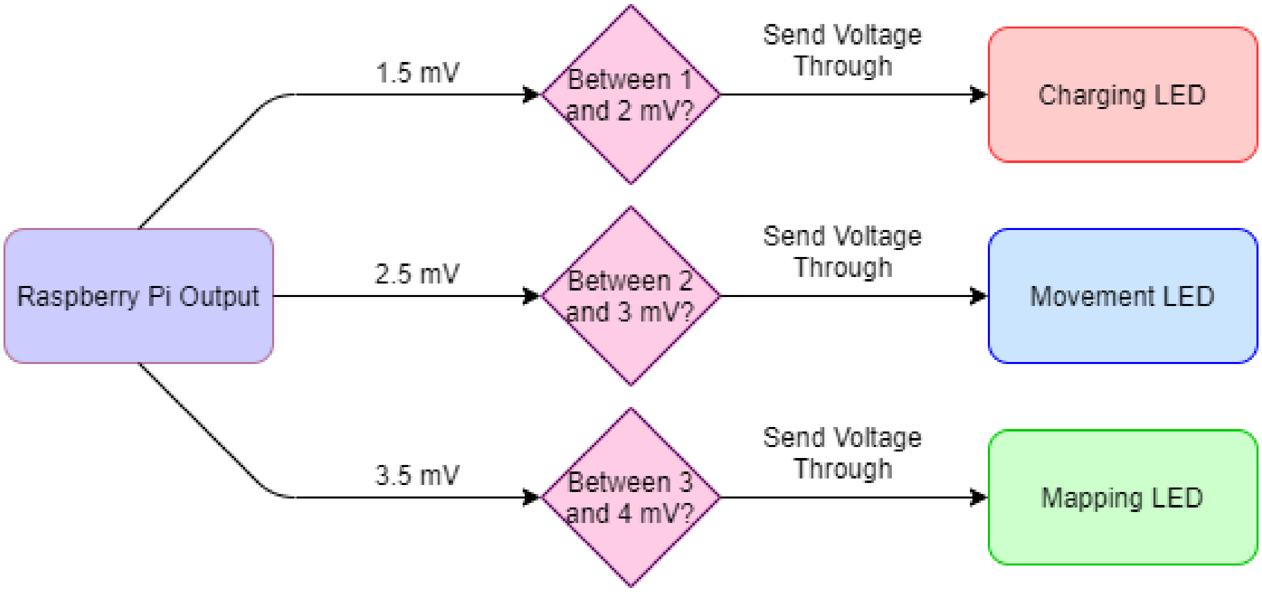


Figure 77: Flowchart Diagram for Maintenance Circuit

The diagram shown above is shown to create a circuit that powers all the above LEDS with one Raspberry Pi output. By varying the voltage of the Raspberry Pi, we can essentially minimalize the voltage output of the Raspberry Pi, as well as save output slots for it. Since sustainability is one of the goals of this project, including this simple maintenance alert would be a huge step up given enough time at the end of the project. Even implementing this midway throughout the building process would help identify key issues with our design.

As stated, the above scenario is only if we are hard-pressed for output slots on the Raspberry Pi. The easiest approach to maintenance lights that need to light completely separately from one another is to drive three inputs from the Raspberry Pi into the standalone LEDs. This only leaves a simple circuit with minimal circuit materials required. The LEDs can be connected to the power with only a simple 233 Ω resistor in series for each one. They then connect to their own ground.

5.5.6 Additional Interface

Originally everything in this design was supposed to be controlled through an app, but adding more options for changing the movement functions of the mouse would increase the project's versatility. This can be as simple as adding a four position selector switch to the box that can switch between controlled mode, autonomous mode, intermittent mode (running infrequently throughout the day), and off. This would be wired into the Raspberry Pi's inputs, with a direct connection to power, and would be located in the box subsystem. This idea would be especially useful if the ideas from section 5.5.1 – 5.5.3 were implemented, so they could be used separately from the main function of the project.

5.5.7 Additional Maintenance

Many projects are ailed by a lack of sustainability. Products are bound to break at some point, but it is a matter of whether they can be fixed, or if they have to be completely replaced. RAM, motherboards, and CPUs are all products that often end up in the trash when they're recognized as broken, or faulty. One goal of our project is to not trap any parts in a container where they can't be maintained. This is apparent in both our box design, and mouse design, two containers that make access to parts rather difficult.

To start off with, there are certain parts in our project that have a low sustainability rate, meaning that if they break, it is unlikely they will be able to be fixed without replacing. These parts are as follows: motors, Raspberry Pi, accelerometer, camera, Wi-Fi module, and wireless charging coils. When broken, they will have to be replaced with a new part since they are not within our realm of maintenance. Some of these parts are easy to identify when broken, but some are harder to notice without further tests.

Almost all parts in the build are connected to the Raspberry Pi. The first step in maintenance should be to check for any loose connections within the structure. A motor not working is a large issue, but doesn't necessarily mean the motor is broken. There may be a faulty connection, or the program has an error that doesn't allow movement within the motor. To combat this ahead of time, all parts will be tested separately as they are added into the design to avoid prolonged searches for errors. In this way, we can address all the working parts in our project to assure that they are likely not to blame for errors.

Section 5.5.4 mentioned maintenance signals, which is an easy way to visibly document errors for both user and designer alike. To take this one step further, in every connection between the Raspberry Pi and another utility, there can be a power feedback to a separate input of the Pi (given that there are enough slots to hold them). The power can then be analyzed in the software to provide insight to which parts are properly receiving power, and which are not. This will greatly help in troubleshooting the project. For the user, each part can be labeled in their interface, and then if there seems to be an issue with power, the interface will mark the part in red. This way if they are in need of assistance, maintenance can quickly and efficiently be provided as the knowledge of what isn't working is already aware to the producer.

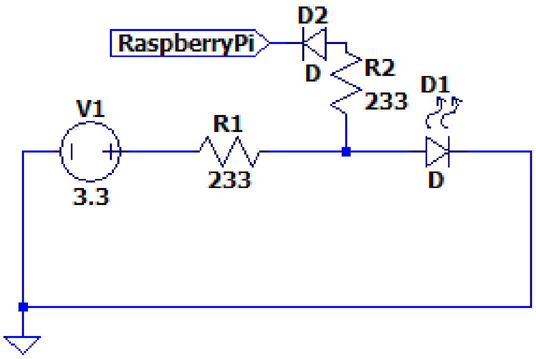


Figure 78: Power Feedback Circuit

To address the issue of accessing all of our parts, it takes into account the general structure of our design. The box is the simplest to provide access to. Simple hinges at the top of the box that allow it to open will make immediate access to most of the parts viable. The bottom of the box will be screwed into place since it is a safety concern as it houses the wireless charger. The inductance from the coils may be hazardous if tampered with. For maximum safety, it will be very difficult to get to, but in the prototype, a sliding configuration that acts almost as a drawer would suffice for taking out and fixing the charging station.

The mouse is a design that's made to be small, compact, and durable. This combination raises various issues for maintenance. On top of that, it is coated in a fur that's not meant to be taken off either. In the mouse, from inside to out, lies the hardware, the chassis, the rubber protection, and the fur coating. A slit will be cut into the fur, which will then be lined with Velcro. The rubber protection will surround only the top half of the chassis, while the bottom remains as the chassis. This way the rubber remains as protectant, but also removable. The hardware components are fastened to the chassis, but are in no way an issue to remove and examine. This satisfies all the needs from the first statement, while addressing sustainability.

5.5.8 Final Thoughts on Additional Features

Many ideas were stated in this section that could benefit the project in a variety of ways. In this final subsection, the plausibility and overall priority for this project will be discussed. This way if extra time and budget is had at the end of Senior Design II, then we can incorporate these ideas in order of importance.

Maintenance will take priority over all other add-ons in the design. Being able to buff the design we already have will improve its longevity and function. The power feedback mentioned in section 5.5.6 will become the first addition at the end of the project, if not already incorporated into the current design. It provides substantial error checking, and may increase efficiency of project building if we run into problems consistently. As a smaller outlook on this, the maintenance signals from 5.5.4 will be a slightly less complicated version of the power feedback, while still supplying helpful information on error research. It is both easy to incorporate, and budget friendly.

The third maintenance feature is manipulating the shell of both the mouse and the box to have their parts easily accessible. This is mentioned in the final two paragraphs of 5.5.6. Although this is one of the most likely features to be added into our project, the design is likely to go through many changes as the building process continues. So, although useful, may not even be relevant by the time of the final design.

Lastly, the laser pointer, treat launcher, and treat feeder from sections 5.5.1, 5.5.2, and 5.5.3 respectively, are stretch goals for the end of the project. They have the last priority, and are in no particular order better than the others. Not only would they take plenty of bonus time to integrate, but would also increase the budget by a fair amount, since they are in themselves almost complete individual projects if added with complexity.

5.5.9 Senior Design II Outlook on Additional Features

Disappointingly, time did not allow any of these extra features to be added into our project. However, there is a solid framework for any of these designs if we were to add them in the future. A lot of these designs could be set up the same way the web app was set up to connect to the raspberry pi. In other words, by using the website to communicate with these additional features, we could add them rather easily. Time did not allow for this though, and we decided to focus on the overall function and quality of the mouse rather than these additional features.

The most relevant additional feature however are the maintenance lights. Although not included in the mouse, unless we count the blue light that shines when the mouse has successfully docked on the charging station, there were many maintenance features added in the process of creating the project. Many breadboard prototypes for the design included LEDs on the circuit pathway to make sure electricity was getting to certain parts of the circuit successfully. If the LEDs did not light up, then we were able to easily track where the problem was in the prototype. It would have been fun to include these into the actual final product, but the space constraint made it much more difficult.

5.6 Software Design

Software is the link that ties everything together. The software design of the AutonoMouse includes two major parts: Hardware Interaction design and Web App design. The Hardware Interaction design describes component interfaces, communication and interaction between components. The Web App design describes the Web frame work and user interface.

5.6.1 Hardware Interaction Design

All hardware devices need a way to communicate with the central hub, and that is through software. There are 6 major hardware components involved in software design: Raspberry Pi, Motors, Camera, Laser sensor, 6-axis Accelerometer/Gyroscope, and Battery.

5.6.1.1 Raspberry Pi

Raspberry Pi is the central hub of the AutonoMouse, it runs, administers every piece of software. All the other hardware components are communicating through Raspberry Pi’s GPIO (General Purpose Input/Output) ports.

5.6.1.1.1 GPIO Programs

The Raspberry Pi’s GPIO programs are in charge of the following:

- Sensor (Ultrasonic sensor and 9-axis Accelerometer/Gyroscope/Magnetometer) programs poll data periodically and stores the data.
- Motion-a special program for camera (more details in the camera section).
- Output programs send outputs to motor controllers, wireless charging controller, and Wi-Fi.

5.6.1.1.2 Data Collection

Data is collected by the Raspberry Pi from the other individual devices that are connected to Raspberry Pi’s GPIO. Data collection involves event, polling, and trends.

- An **event** is something we care about happening such as an obstruction comes in range, an obstruction goes out of range, or the mouse stopped suddenly and etc. The devices don’t provide events, they provide values, with no insight or as to what current value means in the stream of changing values.
- So instead we use **polling.** The Raspberry Pi requests the value from the devices repeatedly, 20 times per second, 100 times per second, and etc. we will determine experimentally what value is the best use of CPU time during testing phase. We will also store a number of values and determine experimentally what size of history is enough data to detect events, but not too much which would waste memory.
- We have a history of data, we look for **trends** to determine when an event has occurred. We can use slope, rolling averages, residuals, or other mathematical concepts to determine when a number indicates a change in a sequence. When a trend shows something has happened, we can send the event a single time to some higher-level decision making part of our robot's software.

We plan on designing the User Interface more like a developer interface, so that all of these collected data will be programmed to display on the User/Developer Interface. (If this project becomes a commercial product, we could easily take out the data displaying for the developers).

5.6.1.1.3 Serial Communication

I2C protocol will be used for serial communication between the Raspberry Pi and other devices. This can be setup by using the “wiringPiI2C” library. First the library needs to be imported by using “#include <wiringPiI2C.h>”, and then the following basic functions are available to use:

- To initialize: int wiringPiI2CSetup (int ID), ID here needs to be the I2C number of the device, and this can be found by using “gpio i2cdetect”.
- To read: int wiringPiI2CRead (int fd);
- To write: int wiring PiI2CWrite (int fd, int data);
- To write 8 or 16 bit data into specific register:
    - ⇒ int wiringPiI2CWriteReg8(int fd, int reg, int data);
    - ⇒ Int wirngPiI2CWriteReg16(int fd, int reg, int data);
- To read 8 or 16-bit data from specific register
    - ⇒ int wiringPiI2CReadReg8(int fd, int reg);

⇒ nt wiringPiI2CReadReg8(int fd, int reg);

5.6.1.1.4 Web Server/Interface

The Raspberry Pi communicates with the Web server, User Interface and Backend Interface:

- Linux Web server nginx receives all the requests from the User Interface, and hosts the user interface files (in HTML).
- CGI script runs GPIO programs directly, or communicates with "mouse daemon" (more details in the Backend Interface section).

5.6.1.2 Motors

There are two motors connected to the two wheels, left and right. Two things will need to be setup for the motors: differential drive and motor controllers.

5.6.1.2.1 Differential Drive

The mouse is going to run differential drive, which is the theory behind "why" signals are being sent, it is using the difference between the left and right wheels to make the mouse moving in different directions. The specific mechanisms are as follows, all of them will be implemented in software.

- The left and right wheel should have its independent speed values range from 0 to 255 or -127 to 128 depending on the motors, where the greater the value the greater the speed.
- When the difference is 0, which means the left and right wheels have the same value for the left and right, the mouse would either go straight forward or straight backwards. (Figure 66)

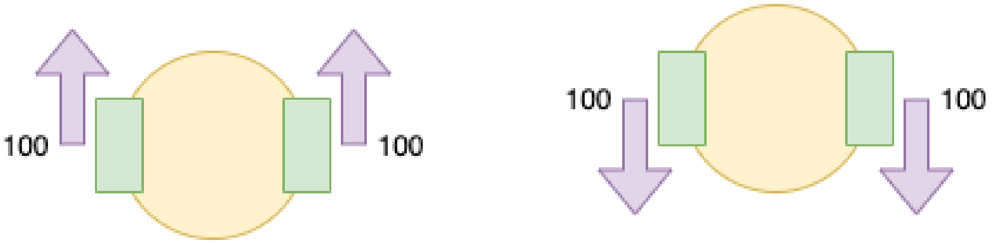


Figure 79: Straight Forward or Straight Backward

- When one side of the wheels has a lower value, it's still going forward but it's going to curve in that direction and turn in that direction. (Figure 67)

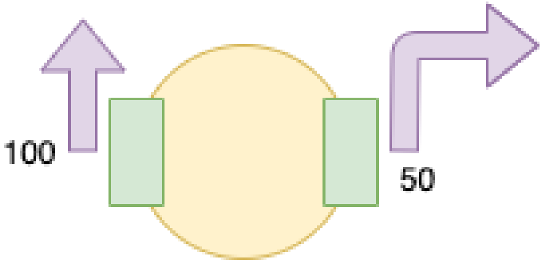


Figure 80: Turn Right

- When it’s a positive value on one side, and negative on the other, the mouse should turn in place. (Figure 68)

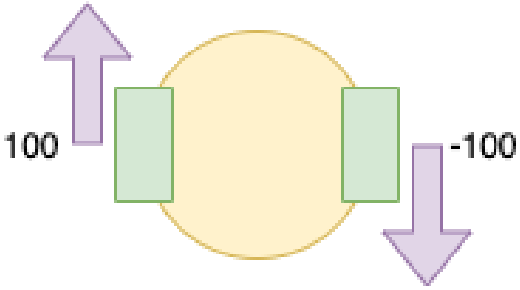


Figure 81: Turn in Place

5.6.1.2.2 Motor Controller

The motor controllers are used to control the wheels and to communicate with the Raspberry Pi. This will be setup through the I2C serial communication protocol.

5.6.1.3 Camera

There are two main purposes of the camera in our project: Streaming and Object detection.

5.6.1.3.1 Streaming

A free Linux program called Motion will be used for camera streaming. The Motion programs allow users to run security cameras. When a security camera is setup, Motion would record to file or perform instructions such as “turn on the lights” in home automation. What’s useful about Motion is that it allows the user to connect to it at any time through a Web compatible protocol, so the user can enter certain URL in the browser and connect to the Motion daemon (a daemon is a computer program that runs as a background process), and it would stream its JPEG image that continuously updates. This would work perfectly with what we are trying to achieve in our Web App as far as streaming goes. There are two things need to be done for this: Raspberry Pi camera setup and “Run as service” setup.

According to Motion’s basic setup guide: “Raspberry Pi cameras can be set up two different ways. If Motion is installed by using the apt packages (e.g. apt-get install motion), then the camera must be set up using the bcm2835-v4l2 module which creates a v4l2 device for the camera. Users will need to install this module using the command *sudo modprobe bcm2835-v4l2*. This will set up the camera as a normal v4l2 device and it can be accessed via a standard /dev/videoX device. If Motion is built from source or installed via the deb packages on the project release page, then an additional option is to set up the camera using the mmalcam_name parameter or using the bcm2835-v4l2 module. When Motion is installed via apt, the mmalcam option is not available."

We will also be running Motion as a service in the background to respond to requests, it will automatically start whenever the Raspberry Pi is started. To setup this service, we will be following the Motion's basic setup guide again, the following are a few basic instructions form the guide: "To set up to run as a service, first edit the file */etc/default/motion* and revise the line to indicate *start_motion_daemon=yes*. Next, edit the main *motion.conf* file and specify daemon as on When the computer is restarted, Motion should now be running. The following commands control the Motion service.

- Start the Motion *servicesudo service motion start*
- Stop the Motion *servicesudo service motion stop*
- Restart the Motion *servicesudo service motion restart*".

More details will be followed from the guide:
https://motion-project.github.io/motion_config.html#basic_setup

After Motion is installed and configured, the following steps will take place to display the streaming on the user interface:

- Via HTTPS, mouse registers via API. It receives back SSH details.
- Via SSH, mouse connects to server. Reverse forward the Motion camera feed port.
- The autonomouse.net website shows the image in a specific port which forwards to the Raspberry Pi over SSH port forwarding
- <img> embedding is used to connect webpage to Motion

5.6.1.3.2 Object Detection

Object detection/recognition involves deep learning, which is part of machine learning. Deep learning and Machine learning are based on Artificial Neural Networks (ANNs) which is a network of data related to each other like a human brain neuron network, and the computer learns and produces outputs based on experiences from the data.

"Object recognition is a general term to describe a collection of related computer vision tasks that involve identifying objects in digital photographs."-Machine Learning Mastery by Jason Brownlee.

Due to time, cost and knowledge constraints of our project, we might not be able to develop our own neural networks. We might be able to re-use already trained neural networks such as Region-Based Convolutional Neural Networks (R-CNN) or You Only Look Once (YOLO). However, this still might not be feasible due to our constraints.

Another option for us is to use a sensor combined with the camera to detect objects. Once an object is detected. figure out which side of the object that the mouse is on, and set it to steer away from it.

5.6.1.4. Laser Sensor

The Laser sensor is used to detect obstruction. It outputs either an integer value or an analog voltage value, which gives the distance to obstruction, and the Raspberry Pi is setup to read this value from its GPIO. Once a value is given, we have a few options to program the mouse to steer away, either left, right or backwards. The data will be collected by polling at max speed the device says is possible.

Opensource code from Github was used to get the laser sensor readings:
https://github.com/pimoroni/VL53L0X-python

5.6.1.5 Accelerometer/Gyroscope

The 6-axis device is used to determine the position of the mouse, check for crash, and polling data. For the position, it will require some trial and error for us to figure out the position of the accelerometer, and then we can determine which way is down, so that we can interpret if the mouse is upside down. If the mouse is upside down, we can program it to flip the camera feed, flip left/right in our software, and we will determine if it should continue to operate or shut itself down. To check for crash, we could look for a sudden deceleration. It will also be setup for data collection by polling possibly at 100 times per second by the Raspberry Pi. we could set it up to poll 100 times per second and store the last 1000 data points. After the data is collected by the Raspberry Pi, we are going to look at the trends of the data to see if the mouse's orientation different, if the mouse going at the direction as expected.

I2C communication protocol will be setup for the serial communication. A tutorial with open source code is available from "Raspberry Pi Tutorials" to set this up with Raspberry Pi.
https://tutorials-raspberrypi.com/measuring-rotation-and-acceleration-raspberry-pi/

5.6.1.6 Battery/wireless charger

To get information from the battery, it needs to be connected to a charge controller so that the charge controller can communicate with the Raspberry Pi. The charge controller will be connected to Raspberry Pi's GPIO as well. The charge controller allows us to tell if we are on a power supply and how much battery we have. For this we want the state information such as "is charging (1/0 or Y/N)", and current charge state (%) from the data collection.

The wireless charger would verify the battery's state and start or stop charging accordingly. We would also like to set it up to send an alert to the user's interface if it's not charging the battery otherwise it would die if it's not charging and the user wouldn't know anything about it if it's lost somewhere in the house.

5.6.1.7 Web App

The Web App is where the user can send instructions from the user's phone or computer to the AutonoMouse over the internet. The purpose of the software design of this part is to get communication instructions from the user's phone or computer through/across the internet to where the mouse is. We are going to take the user's input, and go through the user's local network, and then go to the cloud server. The cloud server is then going to look up the user's information such as login and account information, and it will then look up locally inside the web server to see what device the user is registered. Once the cloud server looks up the user's login information and knows who the user is, and knows what end device the user is trying to communicate with, it then uses its own separate communication channel to take the instructions from the user via application and repeat that over its communication to the device so that the device can perform that instruction. (Figure 69)

Figure 82: Web App Communication

The following are the 4 key parts of the Web App design: Wi-Fi configuration for the mouse to have internet access; Cloud for the user to be able to communicate with the mouse anywhere; the User Interface for user control and sending request for processing; the Backend Interface to receive requests and interpret it to send out commands to the mouse.

5.6.1.7.1 Wi-Fi

Ideally, in order to make the AutonoMouse commercially ready and user friendly, we would like the user to be able to open the box, select a Wi-Fi access for the device and then connect to it. This part is about how to configure the device initially and also once the device is configured, its Wi-Fi connection is going to be the basis of how it gets to the internet, how it gets to the cloud server, and how it receives instructions. The following three phases will take in place. (Figure 70 illustrates this process)

- **Phase 1:** Phone connects to Raspberry Pi's access point. First we need to configure the mouse to be a Wi-Fi access point. The mouse is going to broadcast its SSID (Service Set Identifier). SSIDs are network names of the available wireless networks nearby. At this point, the Raspberry Pi is acting as a router, and the user's phone or computer would connect to it. The user connects to the mouse's access point and go to its default gateway. A default gateway is an IP address the user gets when connecting to a network, it acts as an access point to another network. The access point and default gateway will be configured in the Raspberry Pi. For example, we could configure the mouse's default gateway to be at address 192.168.0.1, and then the first device connects to it can be an IP address from this pool such as 192.168.0.100, and go up from there if more things need to connect later.

- **Phase** 2: User has configured access and Raspberry Pi has internet access. After the user's phone or computer connects to the mouse's access point and puts in the address 192.168.0.1, it'll pull up a local user interface that's running on the mouse. It'll ask the user which Wi-Fi to connect to and what the password is.
- **Permanent phase:** User can reconnect over any Wi-Fi that has internet access. Once the Wi-Fi info is entered in the mouse, it's going to take that info, save to its configuration and stop being an access point. The mouse is no longer an access point, and the user's phone/computer is going to be dropped off of the Wi-Fi because the mouse is no longer hosting it, and the mouse is going to connect to the Wi-Fi that the user told it to connect to. If this fails because the password entered was wrong, then there will be a loop taking the user back to the first step to configure the mouse to be an access point again.

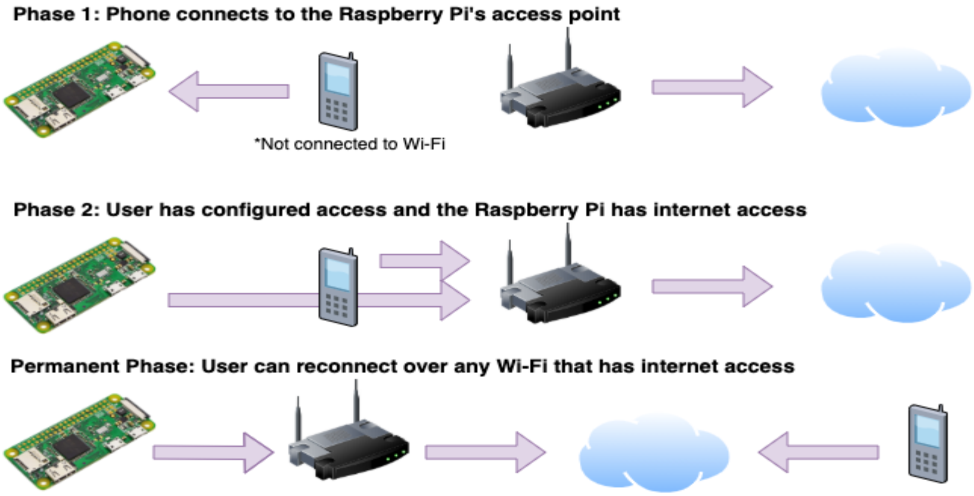


Figure 83: Wi-Fi Configuration

The whole point of the above steps is so that the user can select a Wi-Fi for the mouse to connect to without having to edit any text or pulling out a USB cable to program the mouse.

5.6.1.7.2 Cloud

How a device connects to a cloud server is using the same network procedure as the concept of Reverse Shell. "A reverse shell is a type of shell in which the target machine communicates back to the attacking machine. The attacking machine has a listener port on which it receives the connection, which by using, code or command execution is achieved"((29 et al. *ICMP Reverse Shell*). Reverse Shell is a hacker term, however here our AutonoMouse acts as the attacker, the target machine is the cloud server and the mouse's base station is going to initiate a connection to the cloud server. All of the Wi-Fi configuration steps are needed because the mouse has to be on the Wi-Fi and have internet access before it can connect to the cloud server. The mouse then handshakes with the cloud server, which basically logs in and gives the cloud server its device ID or serial number to register. For the whole cloud setup, the following three services are required: Domain Registrar, Domain Name Service (DNS), Application Hosting.

Domain Registrar. For a commercial product, it is important for us to have our own unique domain name for our website, it would look more professional and more user friendly. To do this, we would need to purchase a domain name by registering it at a registrar such as www.name.com. For example, we could register it as "autonomouse.com". This will put our information on WHOIS ("WHOIS is a query and response protocol that is widely used for querying databases that store the registered users or assignees of an Internet resource, such as a domain name, an IP address block or an autonomous system, but is also used for a wider range of other information".- Wiki). Domain Name Service (DNS). The domain registrar only creates the website's name, it doesn't respond to requests for the IP address. So we would also pay a service fee to have a DNS so we can program the domain name to point to our IP address and vice versa. We believe this step is very necessary because we wouldn't want the user to having to type out the whole IP address each time the user wants to use it.

Application hosting. To host the in the application in the cloud, we need a server. One good option is to pay for a Linux server service called Linode, which charges about 5 dollars a month for 1GB RAM. We'll host a login page that takes requests, and that login page is going to get the user an account and associates the user with the individual device. So the device has a reverse shell to talk to the cloud server, the cloud server is in control of the device, and when the user logs in to the web side of the server, the user can log in with the account and it knows which device the user should have control of. It's like having a frontend and backend interface, the user is going to be connected to some frontend that shows some controls, the backend server is going to take those controls, lookup the specific mouse, and send the information through the mouse's reverse shell. (more details to follow in the User Interface and Backend Interface sections) Figure 71 shows a basic structure of the application hosting.

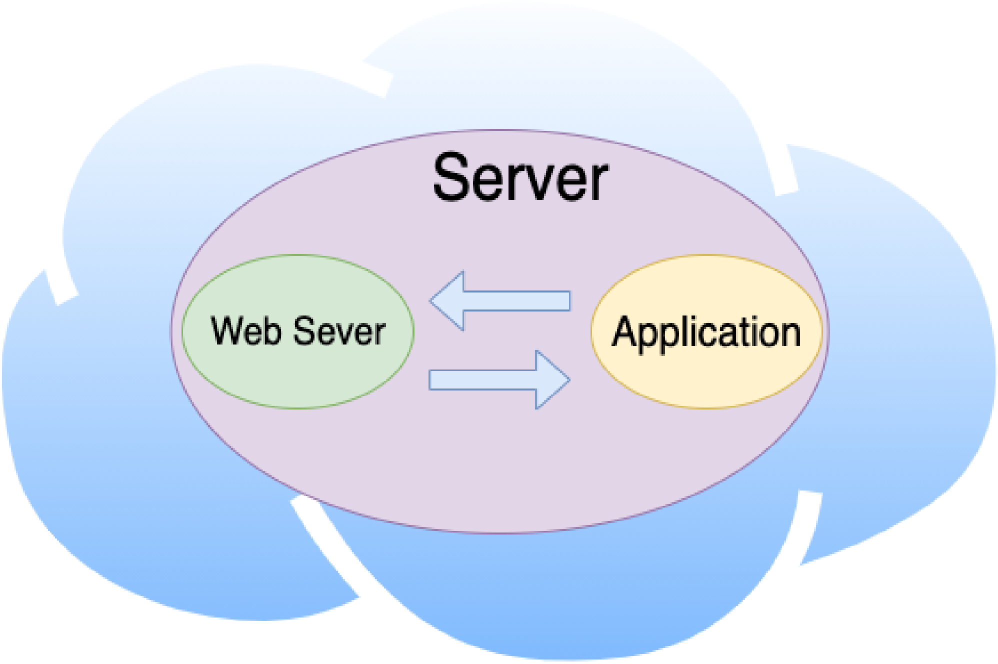


Figure 84: Application Hosting

5.6.1.7.3 User Interface

We are going to design the user interface more like a developer's interface instead of a commercial interface so that it would display all the sensor data that's coming from the mouse through the server to the user's phone/computer, and we would like to visualize and do calculations on all of these data. It will also have buttons for the user to press to control the mouse manually, and the user should be able to send the mouse to charge, or send the mouse to play on its own. The following are some more details about each of the elements that's on the user/developer interface page.

Accelerometer/Gyroscope data. We would like to use three.js (a JavaScript library for 3D graphics) to display the x, y, z axes of the accelerometer, gyroscope, and magnetometer vectors. They would be shown as a straight line from the origin to the x, y, z values in 3D space. This is very useful information for us to know which way is down, which way is north and which way it's spinning. The sooner we can get this interface setup the better, because once we start to move the mouse around, we are going to see the shift in the accelerometer, we would need to think about the outputs from here as we write the rest of the code about how the mouse moves around.

Laser sensor data. The output from the ultrasonic sensor will be either a integer value or an analog voltage value. To display this, we could show a min/max bar graph from the data collection.

Camera feed. This is done by embedding Motion stream with <img> and port forwarding. <img> is a box on a website that has another website inside of it. The mouse's Motion web page will be in that box, appearing on the control website.

Battery level. To display the battery level, we first need to retrieve the battery information, and this is done through the backend API (Application Program Interface). With the API's charging property, we can use it to indicate the charging state such as low, medium, high. We will then embed this in our HTML for display. We will also add information in our CSS (Cascading Style Sheet) to customize the battery bar's colors and percentage, such as red for low, yellow for medium, and green for high.

Buttons to press. Buttons will be setup through HTML <button> element, and we'll write custom JavaScript as a program to process the user input. The following are some key buttons will be displayed:

- Go Charge
    - ⇒ The mouse would go back to the base station and charge himself. We would also like to program it to send the user an alert when the battery is fully charged, and when the battery is low

- Go Play
    - ⇒ This is where the mouse performs a loop of actions such as going forward, backward, left, right , and turn in place. We would also set a timer for this so that it doesn't drain all the battery all at once. Once the time is up, it would go back to its base station unless the user takes over the control.

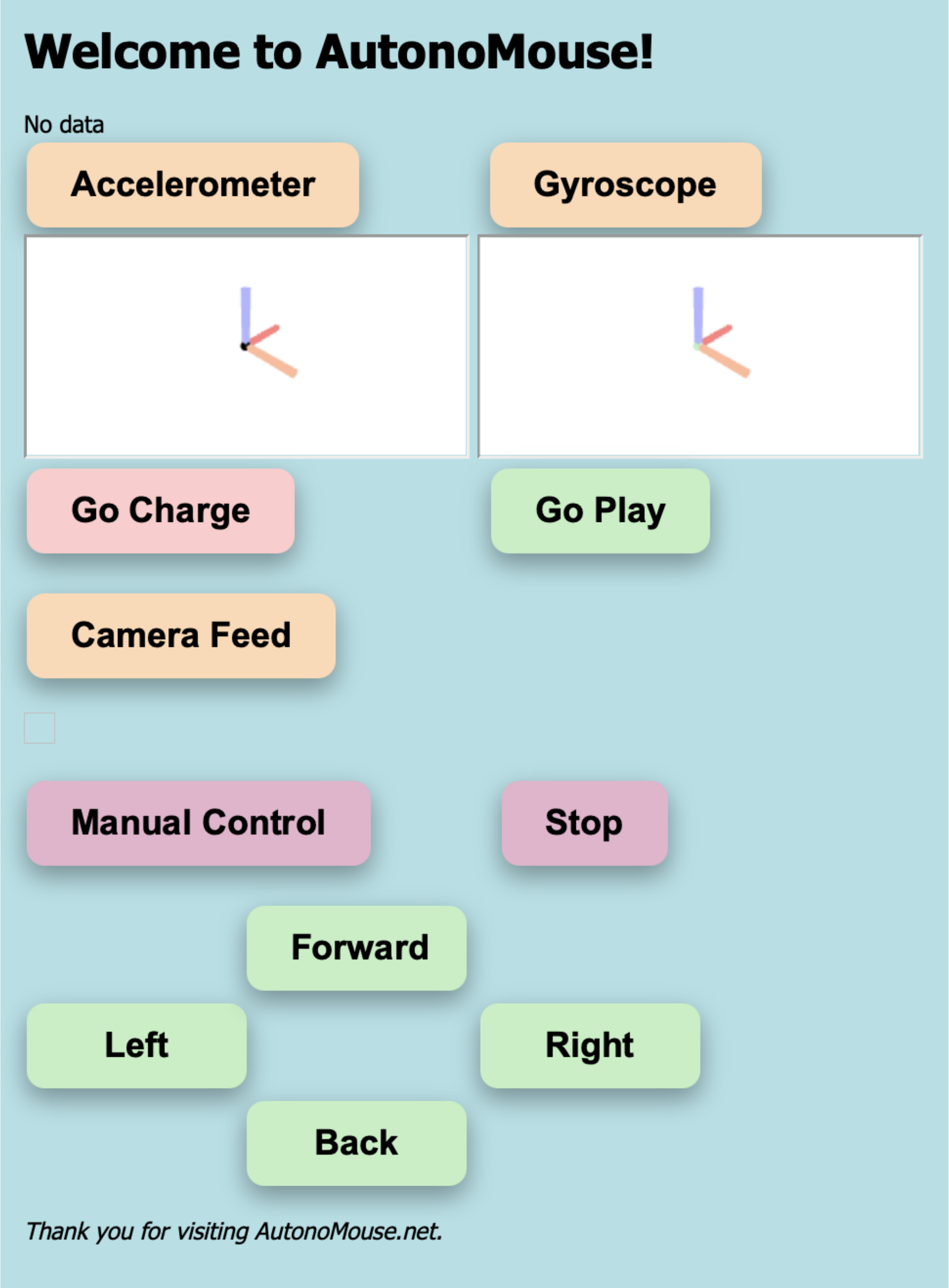
  - ⇒ Obstacle detection is also implemented in this mode using the laser sensor. The threshold we used was 5cm, when the laser sensor detects a distance of 5cm from an obstacle, the mouse would back up, wait for one second, and then turn left slightly.

- Manual Control:
  - ⇒ Forward
  - ⇒ Backward
  - ⇒ Turn Left
  - ⇒ Turn Right
  - ⇒ Stop

Interface languages will be used are HTML, JS (JavaScript), CSS (Cascading Style Sheets):

- HTML is used to describe the presentation of the web page with some basic structures of the page. Tags are used as commands.
- JavaScript is used as a program embedded in the web page, it runs at the user's tier to add functionalities to the presentation. JS elements send requests to the backend server for processing.
- CSS describes the style of the web page such as the background color, font size, page margin and etc.

We believe "less is best" for the controls of the user interface, so that it would also be children friendly if the child is the cat owner and would like to control the mouse to play with his/her pet. Figure 72 on the next page gives a visual of the planned User/Developer Interface page layout. Again we would like to display all the data information for developing purpose, if this project were to become a commercial product, all the sensor data would be hidden.



Welcome to AutonoMouse!

No data

Thank you for visiting AutonoMouse.net.

Figure 85: User/Developer Interface Page Layout

5.6.1.7.4 Backend Interface

There are three important parts for the backend interface: Web server, CGI (Common Gateway Interface) application and API (Application Programming Interface) structure.

Web server. We plan on installing the free Linux Web server called nginx on the Linux Linode server. It receives all the requests, and hosts the user interface files. For example, the HTML file for the user interface is going to be in a folder that nginx shares it with on the server.

CGI application. We will be using nginx reverse proxy for the CGI application. "When NGINX proxies a request, it sends the request to a specified proxied server, fetches the response, and sends it back to the client" (docs.nginx.com). When the web server receives the requests, it's going to realize that these requests are for control input instead of for user interface resources, so it's going to pass that on through reverse proxy to the CGI application. We have to define this protocol. For example, the user presses buttons on the user's interface, nginx knows those buttons point to locations that are not in the web server, they are in the CGI application, and then a script code would be run for that button to execute on the CGI side. The following are a few key things need to be addressed here:

- JSON is going to be used as a format/protocol for transmitting data in web applications. It would look something like this {"user_input": "left", "value": 15} (15 is the degree it would turn). This is sent from the user interface.
- Upon receiving the request, the CGI application would interpret that to send GPIO commands. Should we send commands directly or over a "mouse daemon" ?
    - ⇒ In the direct communication, the user's web request leads to the GPIO code being executed with no intermediary. For example, when the web browser sends this data back {"user_input": "left", "value": 15}, it would turn that into something like: the left motor is on GPIO pin 6, so send this signal to GPIO pin 6.
    - ⇒ with the daemon, it runs in the background and does GPIO actions one at a time, and pulls them off a queue. The web request just loads up the queue.
    - ⇒ In the web version, multiple browsers could send multiple, conflicting hardware commands simultaneously. In the daemon version, the web requests will be serialized when they write to the queue.

API structure. We have to write our own API protocol to communicate with the mouse software depending on the commands we create. For example, if we type in autonomouse.com/api, the nginx would recognize that as we are trying to talk to the mouse software. The following are two examples:

- autonomouse.com/api/accelerometer
  In this case, we would use the GET method and it would return the "[x, y, z]" values such as "[0.008, -0.004, 9.8]"
- autonomouse.com/api/move
  In this case, we would use the POST method. For example, if the instruction we got was "direction = left", we would set it up to return True/False, if Ture, then it would execute the action depending on the method we choose to use: directly or "mouse daemon".

6.0 Project Prototyping

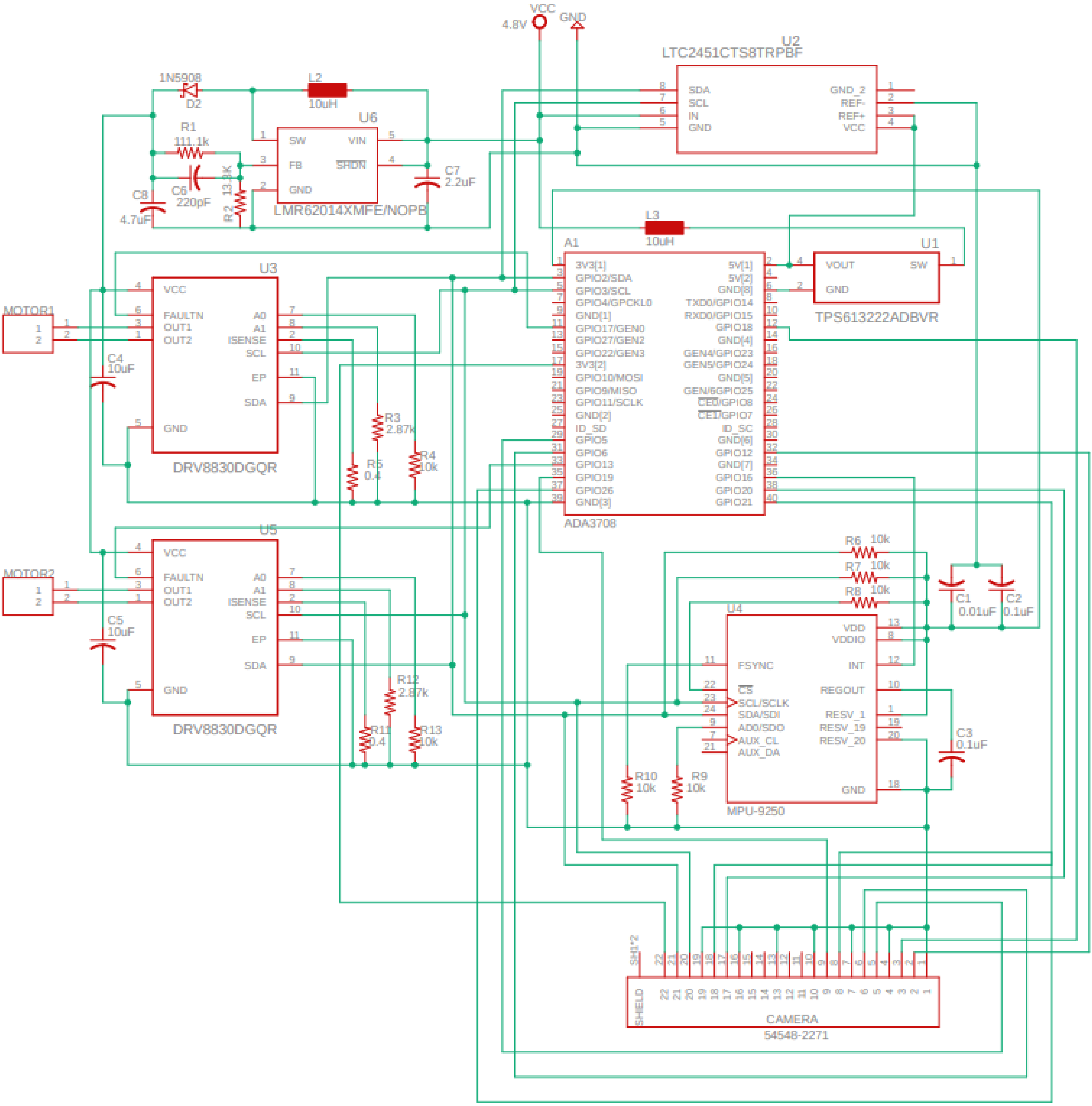
Section 6.0 details the creation of the project in terms of the whole, combining all electronic parts and subsystems outlined previously into one fluid design. This design will be used to create the PCB which will be purchased and used.

Theoretical prototyping of the AutonoMouse system was performed to ensure the device will function as planned and to further define the machine. Integrated schematics of each system, Box and Mouse were designed, as well as the schematics in this section do not reflect the eventual PCB of this device but are a representation of all parts, modules and PCB that could be used in this project. It is likely that these designs will change during the building of the mouse and box units. An example PCB footprint was created based on the schematics presented in this section, and several manufacturers of PCB boards were considered to prototype the board. The cost for each supplier was analyzed and compared to that of the other options. Some companies offer discounts, these were taken into account when comparing the prices. Quality of manufacturing was also looked at for each business as well as the capabilities of each, how many layers they can produce, the lead time it takes and overall period in which an order is processed. Reviews from other clients were also looked into for customer input. There is also the question of weather to order the parts and then place them on the PCB, or to instead order PCBA, printed circuit board with assembly. Using a full assembly service may require the sending of part materials to the company or having the company acquire them at an additional charge.

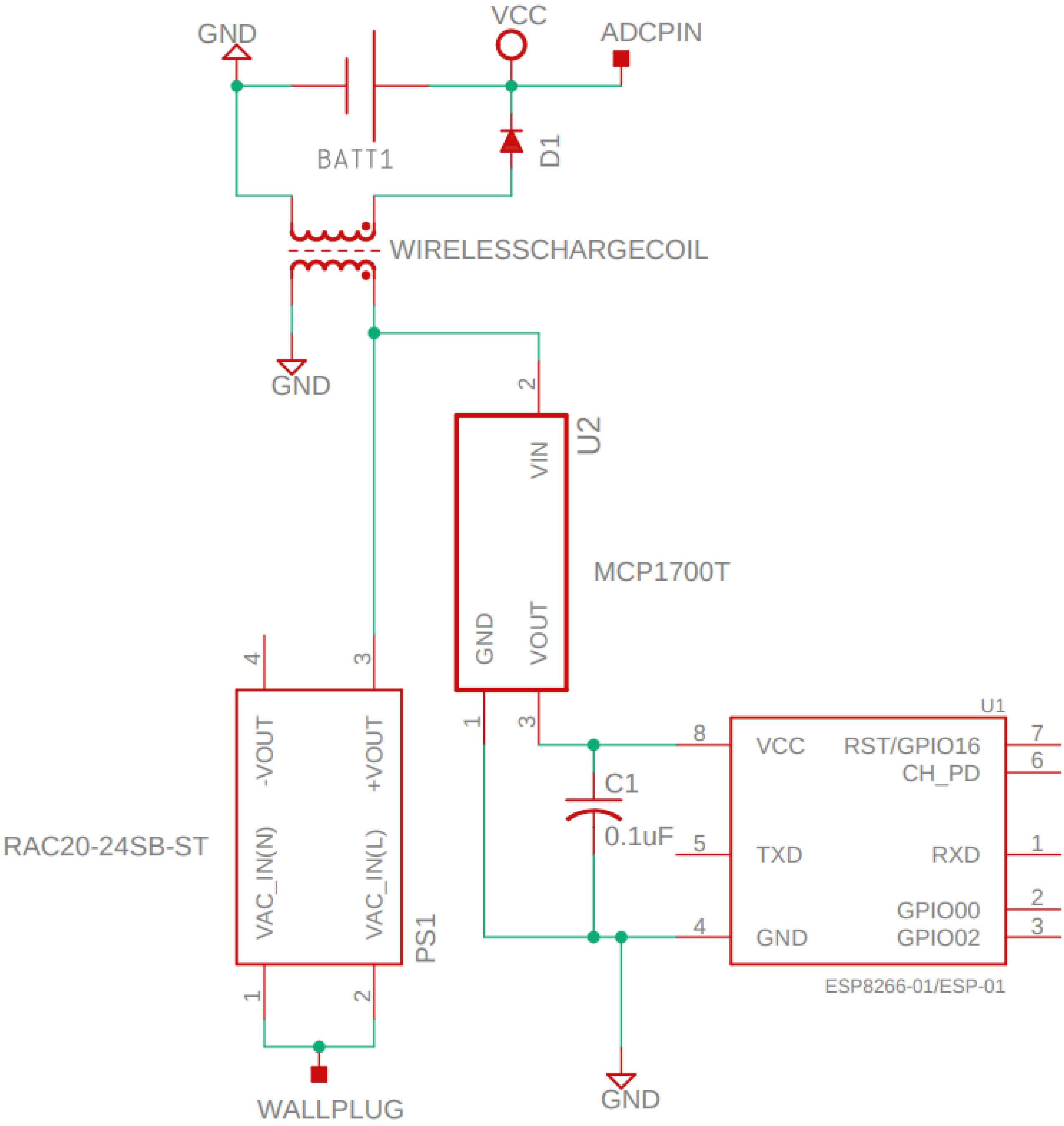
6.1 Initial Integrated Schematic

This part will go over all schematics relating to the AutonoMouse, the following diagrams schematics outline the Mouse and Box electronic circuits as well as the charging system. It will also go over exactly how each component will be placed into the external device, and how much room each part will take up. Most parts have been decided from those listed in the parts section, it is possible that some may be replaced in the future for a more successful final product. This is not because the schematics in this part are viable, using them will definitely create a functional product. But some parts may be difficult to obtain within the given timeframe or prove hard to program when presented as just a chip component. Both the Box and Mouse parts of the AutonoMouse are covered by these schematics, but the block diagram outlining the combination of all parts and their placement within the system is only done for the Mouse. This is because the box has less drastic restrictions to its size.

Schematics in this section were created using the Autodesk Eagle software for PCB design. Parts libraries where downloaded from either the supplier of the device or from SnapEda. All resistor values are part R-US_0204, all capacitor values are C-USC0603, all inductor values are part SD8. Diode used was a 1N5908 Schottky diode. Some circuit components were tested in Multisim to ensure their functionality was applicable to the desired values.



VCC
4.8V
GND
U2
LTC2451CTS8TRPBF
SDA
SCL
IN
GND
GND_2
REF-
REF+
VCC
1N5908
D2
L2
10uH
U6
SW
VIN
FB
SHDN
GND
LMR62014XMFE/NOPB
R1
111.1k
C6
220pF
C8
4.7uF
R2
13.3K
C7
2.2uF
L3
10uH
A1
3V3[1]
GPIO2/SDA
GPIO3/SCL
GPIO4/GPCKL0
GND[1]
GPIO17/GEN0
GPIO27/GEN2
GPIO22/GEN3
3V3[2]
GPIO10/MOSI
GPIO9/MISO
GPIO11/SCLK
GND[2]
ID_SD
GPIO5
GPIO6
GPIO13
GPIO19
GPIO26
GND[3]
5V[1]
5V[2]
GND[8]
TXD0/GPIO14
RXD0/GPIO15
GPIO18
GND[4]
GEN4/GPIO23
GEN5/GPIO24
GND[5]
GEN6GPIO25
CE0/GPIO8
CE1/GPIO7
ID_SC
GND[6]
GPIO12
GND[7]
GPIO16
GPIO20
GPIO21
ADA3708
U1
VOUT
SW
GND
TPS613222ADBVR
U3
MOTOR1
VCC
FAULTN
OUT1
OUT2
GND
A0
A1
ISENSE
SCL
EP
SDA
DRV8830DGQR
C4
10uF
R3
2.87k
R5
0.4
R4
10k
U5
MOTOR2
VCC
FAULTN
OUT1
OUT2
GND
A0
A1
ISENSE
SCL
EP
SDA
DRV8830DGQR
C5
10uF
R12
2.87k
R11
0.4
R13
10k
R6 10k
R7 10k
R8 10k
C1
0.01uF
C2
0.1uF
U4
VDD
VDDIO
FSYNC
INT
CS
REGOUT
SCL/SCLK
SDA/SDI
AD0/SDO
AUX_CL
AUX_DA
RESV_1
RESV_19
RESV_20
GND
MPU-9250
C3
0.1uF
R10
10k
R9
10k
SH1*2
SHIELD
CAMERA
54548-2271


GND
VCC
ADCPIN
BATT1
D1
WIRELESSCHARGECOIL
GND
2
U2
VIN
MCP1700T
GND
VOUT
1
3
4
3
-VOUT
+VOUT
RAC20-24SB-ST
VAC_IN(N)
VAC_IN(L)
PS1
1
2
WALLPLUG
8
C1
0.1uF
5
4
GND
U1
VCC
RST/GPIO16
CH_PD
TXD
RXD
GND
GPIO00
GPIO02
7
6
1
2
3
ESP8266-01/ESP-01

6.1.3 Mouse Schematic:

This schematic show in Figure 73 outlines the interconnectivity and power flow for the AutonoMouse's wireless component. There is a Wifi Module, the Broadcom/Cypress BCM43438 or CYW43438, installed within the Raspberry Pi Zero w not shown in this schematic. The Camera module, OV5647, connection port is shown in the schematic, but the internal schematic of the camera is not shown. The motors, 1 and 2, are also connections, the motors themselves will not be directly on the board. all resistors, capacitors and resistors are standard. Two step up boost regulators were used, one LMR62014, U2, set to deliver a 6V output from a given 4.8V input and one TPS613222, U1. U2 is used to as the source for both the loads and power supply of the two DRV8830 motor controllers, this is because neither motor is able to draw more than a 0.70A Stall, 0.27A load, current, and device is capable of supplying 1.4 A max, which is identical to the max delivery of 1.4A. The other regulator, U1, connects only to the Raspberry pi's 5V power input pin, with the remaining devices being powered by the 3.3V input pins of the microcontroller.

The battery has been attached to the microcontroller via an ADC, LTC2451, this will give the Pi the ability to tell what percentage of power the battery has remaining to aid in decision making and to notify the user during control mode. The MCU will be set to return to the Box should the voltage drop below 4.6V, however, it is unlikely that the device will ever reach a low battery state due to the periodic resting of it within the Box at the end of Mode 1.

The accelerometer resistor and capacitor combinations have been set up with the standards given by the datasheet of the module. The RES_V19 Aux_CL and Aux_DA have been left open as they are not important to this system. The MPU-9250 contains an interrupt pin connected to GPIO 16 of the Pi. This will interrupt the MCU when the Mouse stops suddenly, triggering Mode 2 of the Mouse. Mode 3 is user controlled and will ignore this pin. An accelerometer device can have difficulties with placement in relation to other parts. It must be in the proper position to accurately sense movement on the x, y, z plane. As such, it will likely be tested separately from the rest of the design to determine proper location before being attached to the circuit.

Although not present on the schematic, the CYW43438 wifi/Bluetooth module is an important part of the system. In Mode 1, it will be responsible for sending out a signal to the ESP8266 located on the Box, timing how long it takes for the signal to be returned. This is key for the pathing ability of the Mouse. This component will also be used for mode 3 to receive data from the mobile webapp for user control and send out the camera's video data to the app.

The two motor controllers, the camera, the ADC converter, and the accelerometer, are all using I2C communication protocol. Therefore, each must have unique addresses, if the addresses are not unique, the systems will need to be reprogramed so that they are. They are attached as slaves to the master microcontroller. The I2C connection contains pull up resistors R6, for the SDA data line and R7, for the SCL clock line. Each has a value of 10k Ohms.

The charging system of the mouse will contain a specialized inductor coil designed to receive transferred energy from the Box through mutual inductance. To prevent the inductor from drawing current away from other devices during the noncharging state, a diode will be placed in series with it. The hope is to design the coil with just the right amount of inductance such that charging will come to a halt once the battery is full, however if this proves to be impossible to achieve, a switch will be placed to cut off the inductor circuit once charging has reached max capacity to prevent overcharging of the battery, this switch will require intercommunication with the between the Box and Mouse during the charging state should it be implemented. There is also concern for the current draw from the system pulling too much amps away from charging the battery, the system can take up to 1.8 Amps of current and the planned inductor would be able to deliver about 2 amps. If the Mouse operating circuits take too much, it is possible the battery will not charge at all or too slow to matter. Testing will be done to ensure a proper induction charging circuit is created by the time a prototype is completed.

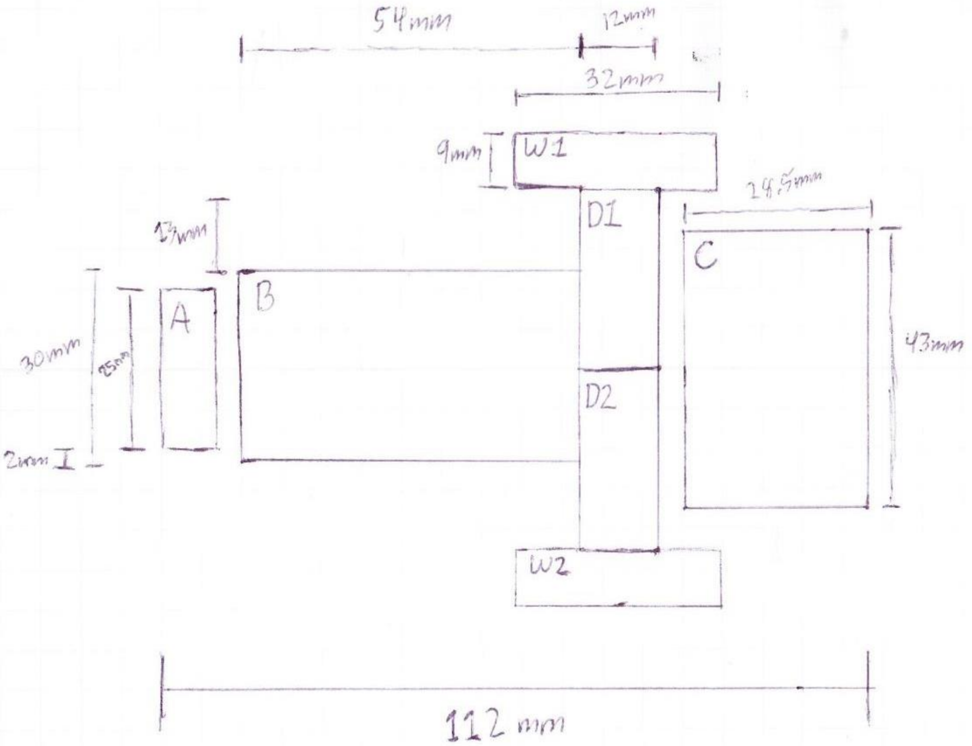
6.1.4 Box Schematic:

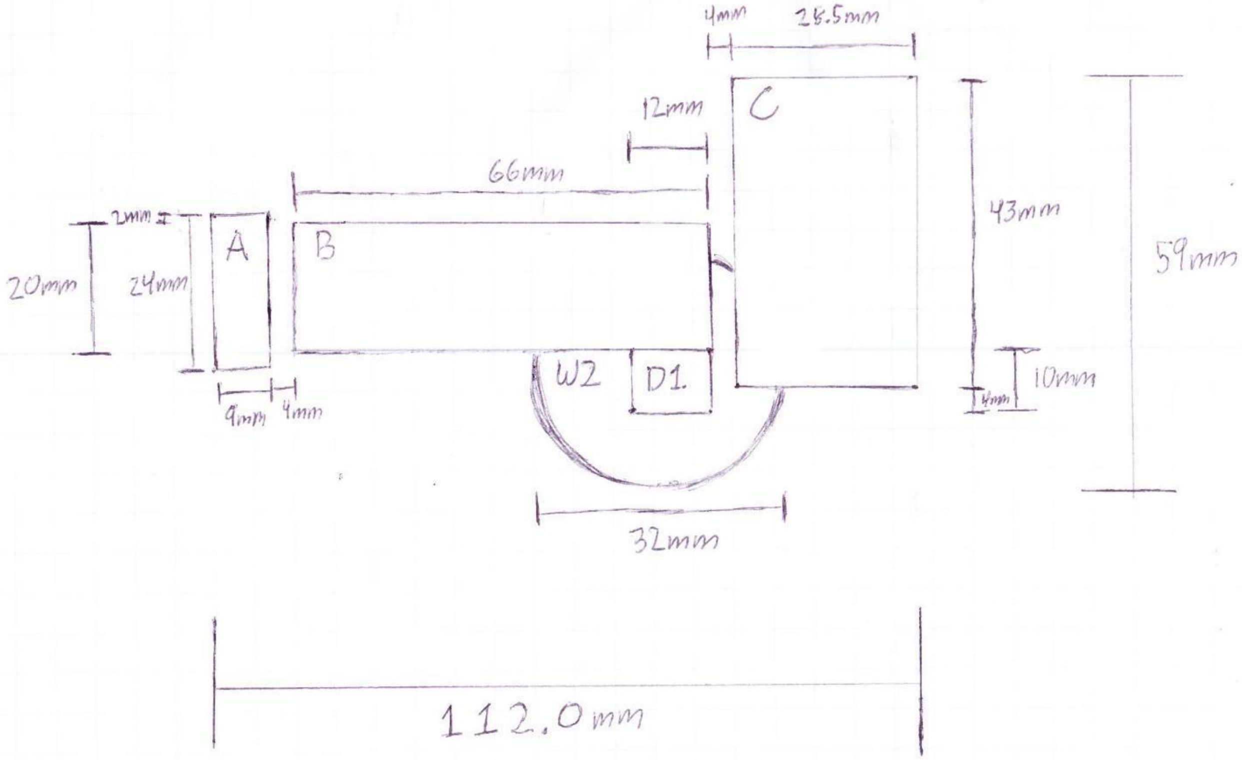
The box is a simple device when compared to the Mouse. As can be seen from Figure 74, it contains an AC/DC converter, EPS-25-5, a 3.3 fixed voltage out regulator MCP1700T, an inductive coil to be paired with that of the Mouse, and the ESP8266 wifi/Bluetooth device. The ESP8266 device was chosen due to its advanced system, it contains a microcontroller capable of performing tasks other than simply retrieving data for wifi/bluetooth management. In the box, this wireless communication device will be programmed to echo back the signal given to it by the mouse to aid with distance determination. Because it does not need to directly communicate with a microcontroller The ESP8266 Rdx, Tdx, and CH_Pd pins have been left open.

This device will require a wall plug, represented in the diagram as a pin. The MCP1700 regulator used in the Box is the only step-down linear regulator in the system, as linear regulators tend to be less efficient that switch regulators, some power loss will result from using this device. The max power dissipation is roughly 900mW under certain conditions.

The wireless charge system, which is represented as two coupled inductors within the schematic, Figure 74, will be a unique design for the mouse, it will be incompatible with other charging systems. The induction coils will draw/deliver 2 amps of current to the Mouse. The ESP8266 is set in parallel with the induction coil, this is because the device takes very little current, roughly 20mA, resulting in an insignificant power loss to the charging system 0.066W. It is possible that, to enable the Mouse to more easily attain a proper charging state, multiple inductors will be set up in parallel.

The current system allows for the addition of one extra wireless charging coil without requiring a change to the AC/DC converter however, if more are designed, the device will have to be swapped for a component with a higher wattage. It is important to note that the AC/DC converter shown in the schematic is not the desired part, but is instead RAC20, this is because the ESP25-5 and the LS25-12 both do not have schematic representation. RAC20 is a close equivalent to the ESP25-5 so it was selected to represent the converter in this schematic.


54mm
12mm
32mm
9mm
W1
D1
28.5mm
13mm
C
B
A
30mm
25mm
43mm
D2
2mm
W2
112 mm


4mm
28.5mm
12mm
C
66mm
43mm
59mm
2mm
A
B
20mm
24mm
W2
D1
10mm
4mm
9mm
4mm
32mm
112.0mm

6.1.5 Structure of Circuitry Within Mouse:

This block schematic shows the location of the structure of the parts and modules within the Mouse as well as the space each component takes up. The diagram features the lengths widths and heights of all components represented in a Block format. Block A is the camera module, which is located at the front of the device to allow for, B represents a block containing multiple circuit parts, see Figure 75, C represents the battery. Part D1/D2 and the corresponding W1/W2 represent the Motors and the wheels attached to them. These wheels were arbitrarily given a value of 32mm. The Mouse will contain a third wheel for stabilization, but as this part will be purely mechanical in nature, it is not featured in Figure 76. Most Blocks have been given a small amount of clearance between one another, with the exception of the motors, D section, which are directly connected to the B section and the W, wheels. It is possible that during construction, these gaps will increase in size due to the addition of supporting structure for the circuitry, leading to Mouse elongation. The block diagram also does not take into account the additional size issues that will be presented by the wires used to connect each of the blocks together.

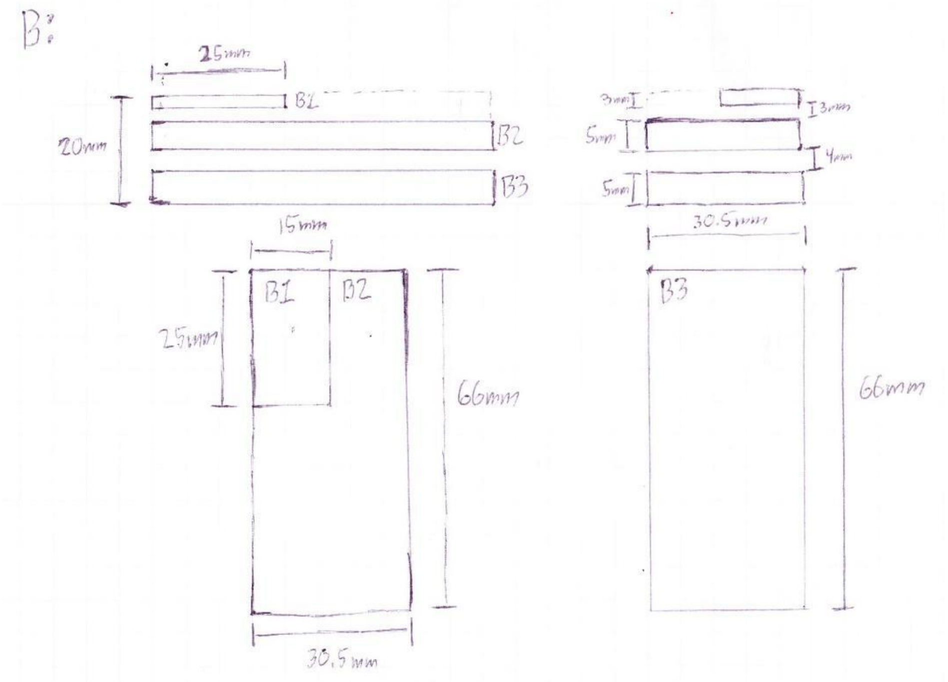


Figure 89: Internal of Block B

The component B is in actuality a block made up of the custom PCB, B3, Raspberry Pi Zero W, B2, and the accelerometer, B1. These parts have been grouped together as a single unit for ease of design and to save space. As electronic devices, it is better that there be some space between each of the boards. Currently it is estimated at 3-4mm, but this will likely change due to mounting conditions and necessity. The placement of the accelerometer is arbitrarily in the corner of the B block, its position may change in need be. It must however be to the side of the Raspberry Pi to enable easy contact between microcontroller and the PCB.

6.2 Tools

Certain tools and machines will be required for proper assembly of the AutonoMouse. Many of these tools are already at the disposal of the team in excessive amounts and thus do not need to be included in the pricing of the device. UCF labs will be used to aid in building of the AutonoMouse. Other facilities will include: The house of the student responsible for production of the structural parts, the house of the student testing the power regulation, the house of the student programming the Raspberry Pi and other parts and the house used to test the device when completed.

Standard Construction Tools:

Average construction tools will be needed to assemble components. These tools include, drills, screwdrivers, cutting tools, etc. Some type of adhesive glue will be required to attach the external covering of the Mouse. Most of the team should have easy access to these types of tools and materials, no additional costs need be applied.

Basic Measurements:

Instruments used to measure will be required to verify certain specifications have been fulfilled. A simple ruler will be needed to ensure the proper length, height and width of the Mouse and Box. A scale will be used to make sure the weight fits the desired amount.

Soldering:

Soldering skills and respective tools such as a soldering iron will be required for properly putting together the PCB as well as attaching the external modules and wiring. All group members have had at least some soldering experience. In addition to member having these materials, the University of Central Florida grants students access to solder and soldering irons.

3D Printing:

Some parts of the Mouse and Box will be constructed using 3D printing technology. These pieces include the chassis of the mouse, the wheels, the internal structure of the box, the supporting structure for the external covering of the Mouse, the platform for the charging station within the box. Each member has direct access to a 3D printer at no additional costs outside the plastic required to print the item. On average, a 1000g reel of plastic costs about $20, this should be enough to construct all supports of the system. To go along with this, modeling software will be used to pre-construct the parts before printing.

Electrical Measurements:

A digital multimeter will be required to properly test the outputs and input to electronic devices and ensure functionality. UCF has many multimeters available to use in their labs and some of the team has access to their own personal multimeters. In addition to a multimeter, an oscilloscope may be required to ensure the I2C interface is working properly.

6.3 Initial PCB Vendor and Assembly

An important part of this project is creating a printed circuit board, PCB, to go within the Mouse. This board will contain many of the parts from the schematic section. Creating a PCB requires special tools, since there is no quick access to these machines, production of the PCB will be outsourced to a company

6.3.1 PCB

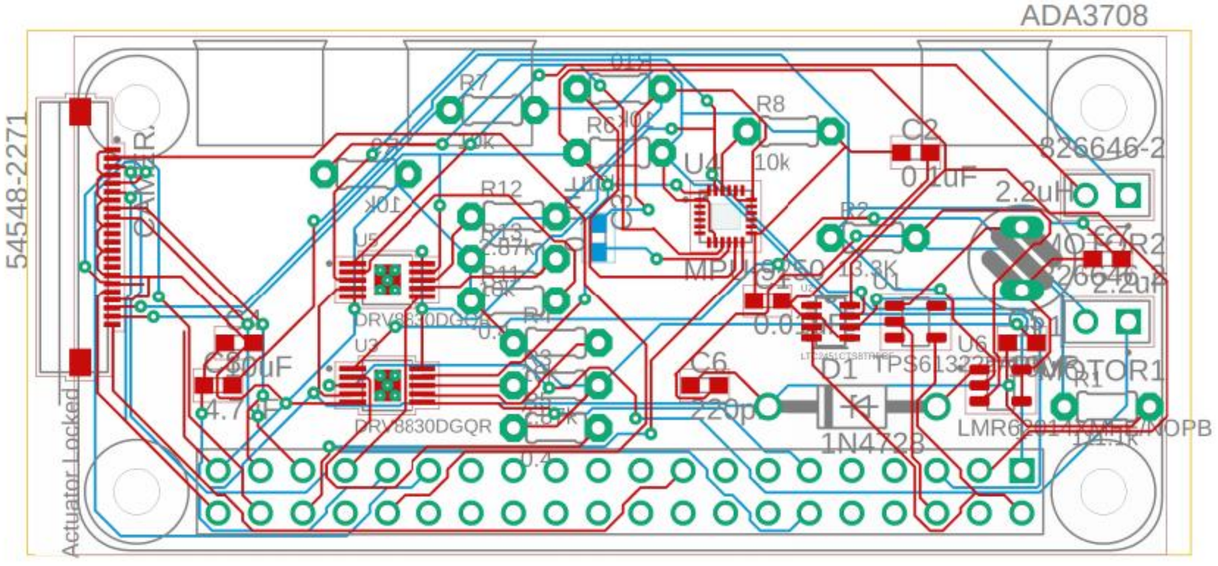


Figure 90: PCB Design

The above diagram shows the theoretical layout of the PCB. It is highly unlikely that all components will be present on one PCB board for the Mouse section. The accelerometer and camera will likely be external modules that simply connect to the PCB which will house the other parts. It is unfortunate that this must be done, as having all devices on one PCB would be significantly helpful in reducing the size of the AutonoMouse. However, it would be very difficult to program all the individual pieces of equipment, or test their viability, if all parts are located on one board. The accelerometer specifically has an added hindrance of being difficult to place, having separate modules connect to the PCB may also be beneficial for the replacement of broken parts, should a fatal error occur in construction and testing.

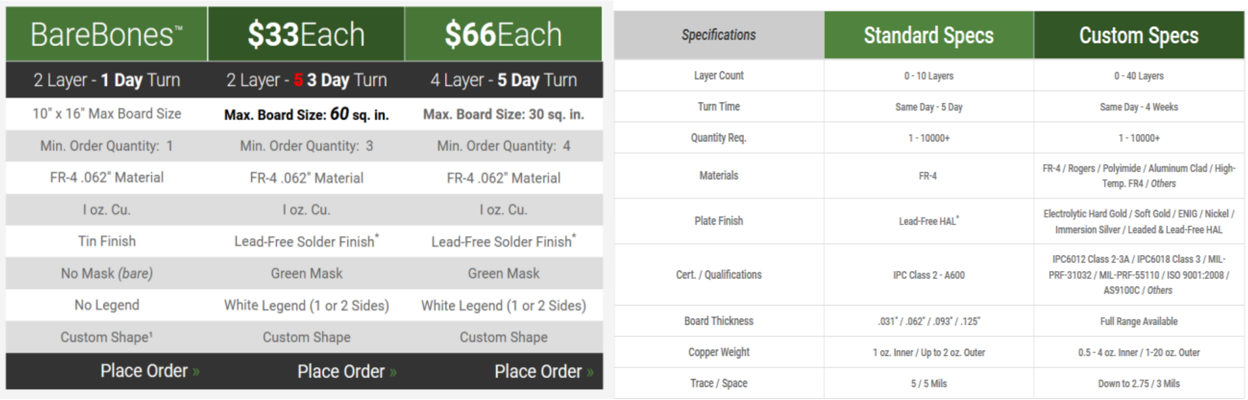
The PCB will be sure to include both motor controllers, the ADC device, the voltage regulators, and the charging system. Resistor and capacitive components will also be present where needed, but some modular devices may come with these already present, causing they layout to change. Any module not directly connected to the PCB will be soldered on or connected through wires. The goal of the PCB is to be the same size or smaller than the Raspberry Pi Zero W such that they can be layered on top of one another given enough space between them to prevent unwanted interference. The Box PCB will easily be able to contain all its components, it may in fact prove unnecessary for the device to have a PCB at all. Given its ample size and few parts, it will likely be more efficient to simply place the parts strategically within the area of the box.

6.3.2 PCB Vendors

Designing a PCB and printing one are two very different things. It would be very hard and costly to create a PCB alone. Thankfully there are many businesses built around supplying and assembling these boards for those who need them. With so many options it was important to look into these companies in order to find the best possible supplier to order from.

Advanced Circuits 4PCB:

Table 4: PCB Purchase Specs



BareBones™	$33Each	$66Each
2 Layer - **1 Day** Turn	2 Layer - 5 **3 Day** Turn	4 Layer - **5 Day** Turn
10" x 16" Max Board Size	**Max. Board Size: 60 sq. in.**	**Max. Board Size: 30 sq. in.**
Min. Order Quantity: 1	Min. Order Quantity: 3	Min. Order Quantity: 4
FR-4 .062" Material	FR-4 .062" Material	FR-4 .062" Material
I oz. Cu.	I oz. Cu.	I oz. Cu.
Tin Finish	Lead-Free Solder Finish*	Lead-Free Solder Finish*
No Mask *(bare)*	Green Mask	Green Mask
No Legend	White Legend (1 or 2 Sides)	White Legend (1 or 2 Sides)
Custom Shape[1]	Custom Shape	Custom Shape
Place Order »	Place Order »	Place Order »

Specifications	Standard Specs	Custom Specs
Layer Count	0 - 10 Layers	0 - 40 Layers
Turn Time	Same Day - 5 Day	Same Day - 4 Weeks
Quantity Req.	1 - 10000+	1 - 10000+
Materials	FR-4	FR-4 / Rogers / Polyimide / Aluminum Clad / High-Temp. FR4 / *Others*
Plate Finish	Lead-Free HAL*	Electrolytic Hard Gold / Soft Gold / ENIG / Nickel / Immersion Silver / Leaded & Lead-Free HAL
Cert. / Qualifications	IPC Class 2 - A600	IPC6012 Class 2-3A / IPC6018 Class 3 / MIL-PRF-31032 / MIL-PRF-55110 / ISO 9001:2008 / AS9100C / *Others*
Board Thickness	.031" / .062" / .093" / .125"	Full Range Available
Copper Weight	1 oz. Inner / Up to 2 oz. Outer	0.5 - 4 oz. Inner / 1-20 oz. Outer
Trace / Space	5 / 5 Mils	Down to 2.75 / 3 Mils

4PCB is a US based company with many options for inexpensive PCBs. Their quickest option, barebones, offers a quick, 1 day turn for PCB design, at the cost of having certain limitations, such as a 2-layer max. Some purchases, although less expensive also require a minimum order quantity to be fulfilled, making them cost more than they appear. However, the company does offer discounts and free PCB construction tools to students as well as a coupon for 50% first customer purchase of under $250.

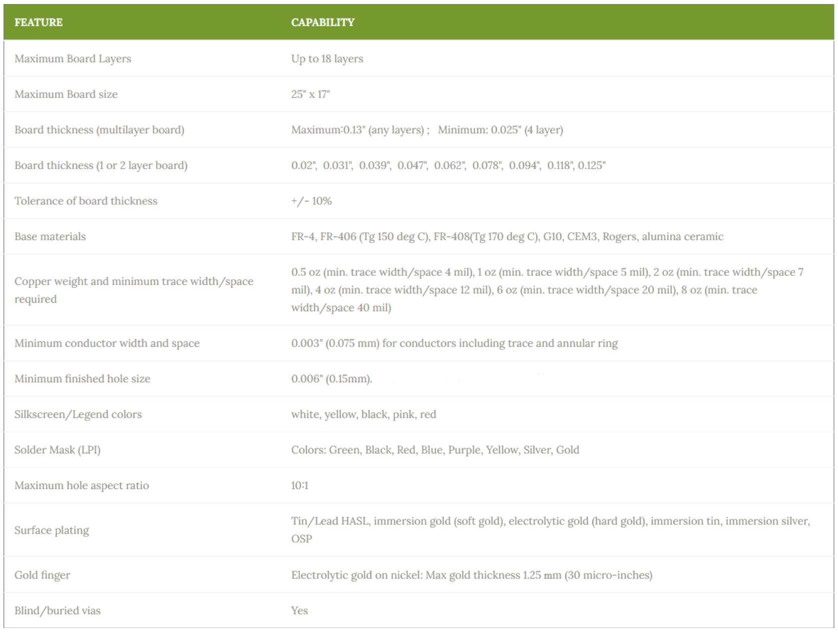
Advanced Circuits does not outsource any of their manufacturing processes, which are all USA based. In addition to PCB, they also offer assembly. For the PCBA they use the latest MYDATA PCB assembly equipment. This company also offers their own free to use design software, PCB artist, for creating custom PCBs. The average lead time is 10 days. Advanced Circuits retains very favorable reviews online, with many claiming it as the best option for prototyping.

TEKNET:

Started in 2001, this company offers a 2 day turn production for prototypes and small orders of bare board PCB. Unlike Advanced Circuits, this business doesn’t seem to offer any type of discounts for students or first-time customers, however they do have some promotional coupons for extra services such as free stencils on PCB orders and offer a single use money back guarantee code if your order is delayed. They also appear to have relatively inexpensive pricing, with the 2-layer product starting at $19.89 with significant limitations. Although they began in Arizona USA, where the headquarters is located, their manufacturing takes place outside the states in one of their two china-based facilities. Teknet has received generally good reviews but not as exemplary as Advanced Circuits.

The standard bare board PCB specifications for this company:

Table 5: PCB Board Specs



FEATURE	CAPABILITY
Maximum Board Layers	Up to 18 layers
Maximum Board size	25" x 17"
Board thickness (multilayer board)	Maximum:0.13" (any layers) ; Minimum: 0.025" (4 layer)
Board thickness (1 or 2 layer board)	0.02", 0.031", 0.039", 0.047", 0.062", 0.078", 0.094", 0.118", 0.125"
Tolerance of board thickness	+/- 10%
Base materials	FR-4, FR-406 (Tg 150 deg C), FR-408(Tg 170 deg C), G10, CEM3, Rogers, alumina ceramic
Copper weight and minimum trace width/space required	0.5 oz (min. trace width/space 4 mil), 1 oz (min. trace width/space 5 mil), 2 oz (min. trace width/space 7 mil), 4 oz (min. trace width/space 12 mil), 6 oz (min. trace width/space 20 mil), 8 oz (min. trace width/space 40 mil)
Minimum conductor width and space	0.003" (0.075 mm) for conductors including trace and annular ring
Minimum finished hole size	0.006" (0.15mm).
Silkscreen/Legend colors	white, yellow, black, pink, red
Solder Mask (LPI)	Colors: Green, Black, Red, Blue, Purple, Yellow, Silver, Gold
Maximum hole aspect ratio	10:1
Surface plating	Tin/Lead HASL, immersion gold (soft gold), electrolytic gold (hard gold), immersion tin, immersion silver, OSP
Gold finger	Electrolytic gold on nickel: Max gold thickness 1.25 ㎜ (30 micro-inches)
Blind/buried vias	Yes

Teknet also offers assembly of PCBs. They require a minimum passive component size of 0201, and a minimum pitch size on the BGA of 0.4mm. This manufacturer also requires a test fixture or test design. The company can supply the parts used at an additional cost or the customer may supply the parts for assembly. They use either tin/lead reflow or lead free reflow solder types and offer stainless solder steel stencil, frameless or framed. The average lead time for Teknet is 10 days.

The Gold Phoenix Printed Circuit Board Co:

This company specializes in both PCB and PCBA manufacturing. In addition to normal PCB, Gold Phoenix offer a variety of board types, such as FLEX PCB and aluminum boards and PCB for high frequencies. The base price for PCB boards from this group appears to be $75-$100. They offer more board customization than most companies and use very high grade equipment in manufacturing, they have PCB with heavy copper, gold plating, immersion gold, hot air leaving, PCB with gold fingers, Blind and buried holes on PCB, both extra thin and extra thick PCB, and impedance control PCB. Like Teknet, Gold Phoenix is based in China. This business has a base lead time of 5 days but offers up to 1-day lead time for additional costs, $60, however, they offer a special 8-hour lead time for prototyping. Most reviews for this business are positive.

6.4: Finalized Hardware Design

Many alterations were made to the PCB design after initial planning to incorporate changes to the part list, the final PCB schematic was as follows:

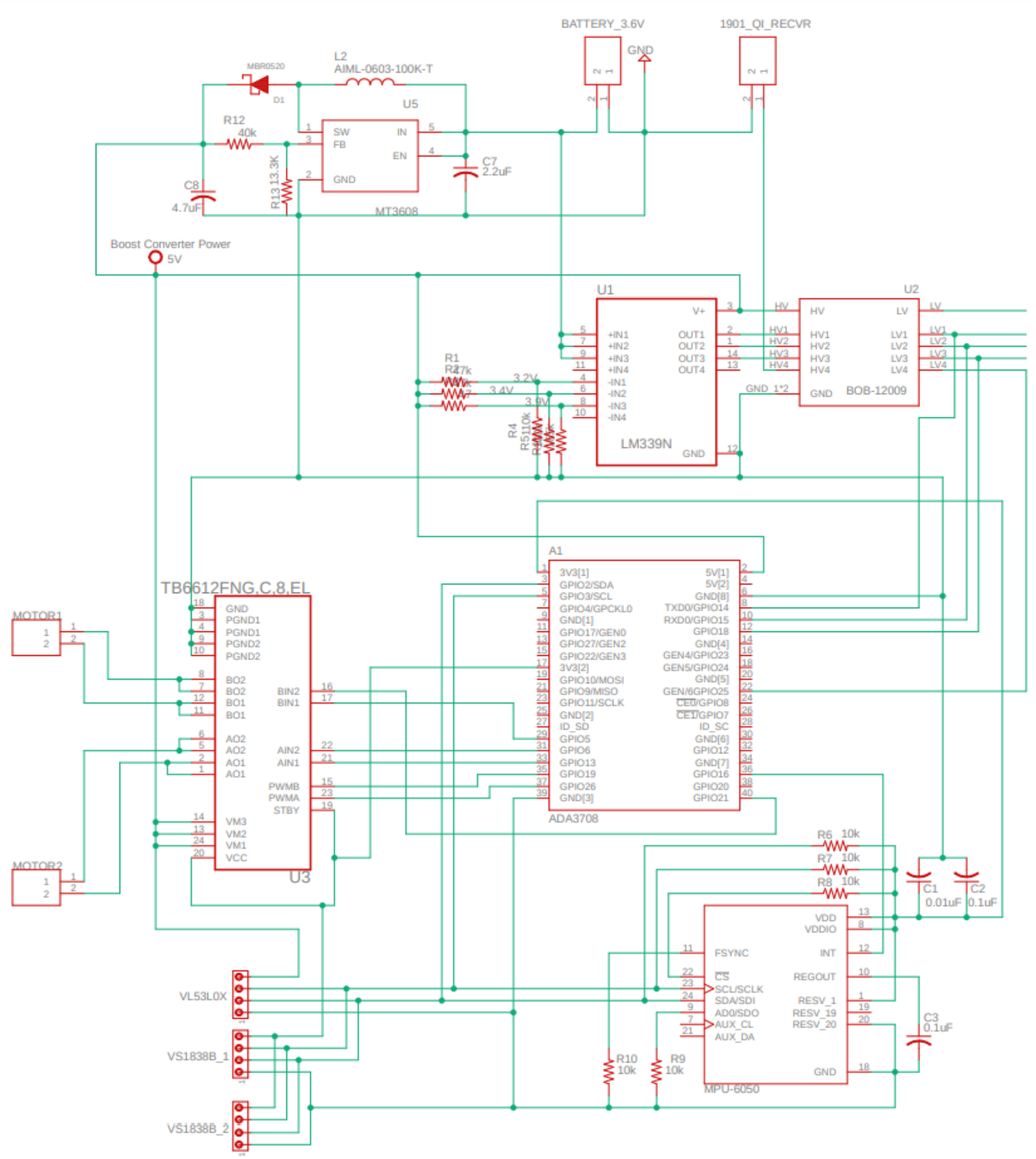


Figure 91: Second Generation PCB Design

The Vendor for the PCB was changed to PCBway, this china based company offered similar deals to the previously discussed printed circuit board companies, but with an easier to understand ordering system for PCB and Assembly options. The total for 5 circuit boards and one assembly board was around $75. Unfortunately, due to COVID-19 there were significant delays in the manufacturing process, certain parts were unacquirable by the company and the estimated time of completion shifted past 4/18/2021 which was the day of presentation. A few other PCB companies were looked at but none promised superior speed of manufacturing. In the end the PCB ended up being a sunk cost and the protoboards were used for final presentation.

The first generation protoboard, created before the PCB delay, contained the Raspberry Pi, motor controller, and connections for the accelerometer, motors and laser module. This allowed the device to accomplish the “go play” and “manual control” modes. The 5V required for operation was satisfied by using a buck converter to step down a common 9V battery as the desired 3.6V rechargeable battery had not yet arrived. This board was designed to go above the motors at the center of the mouse.(see figure 92)

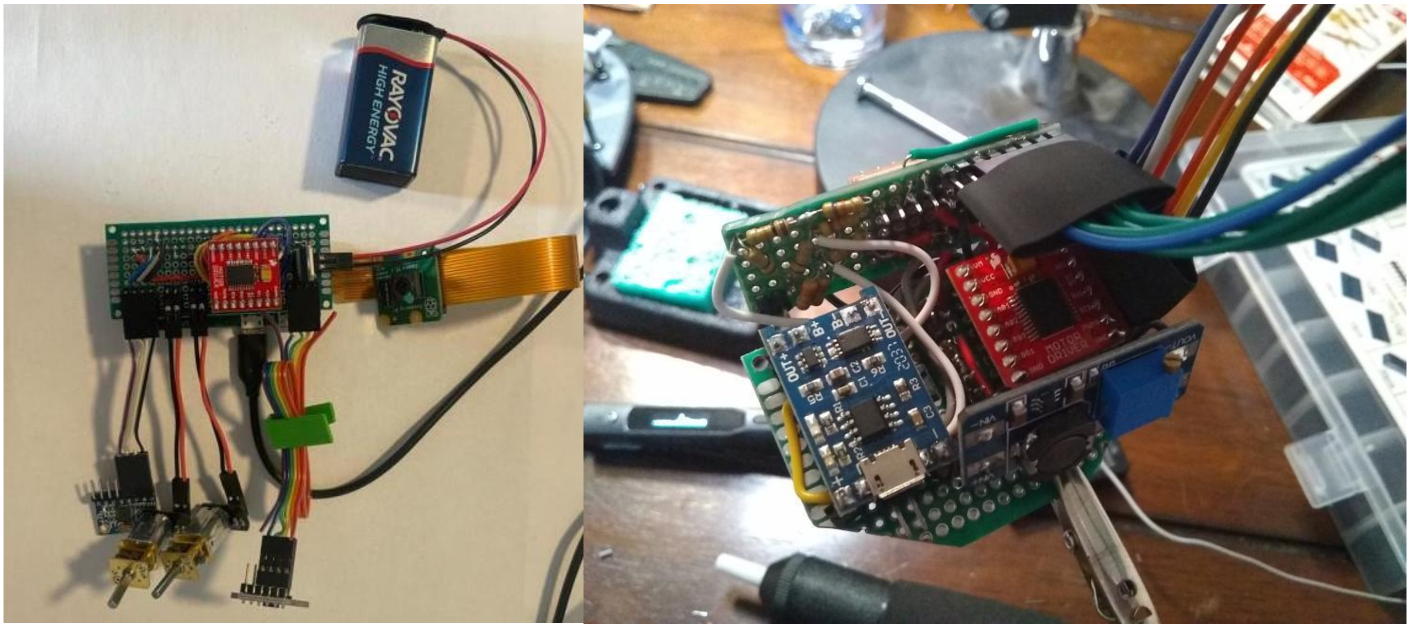
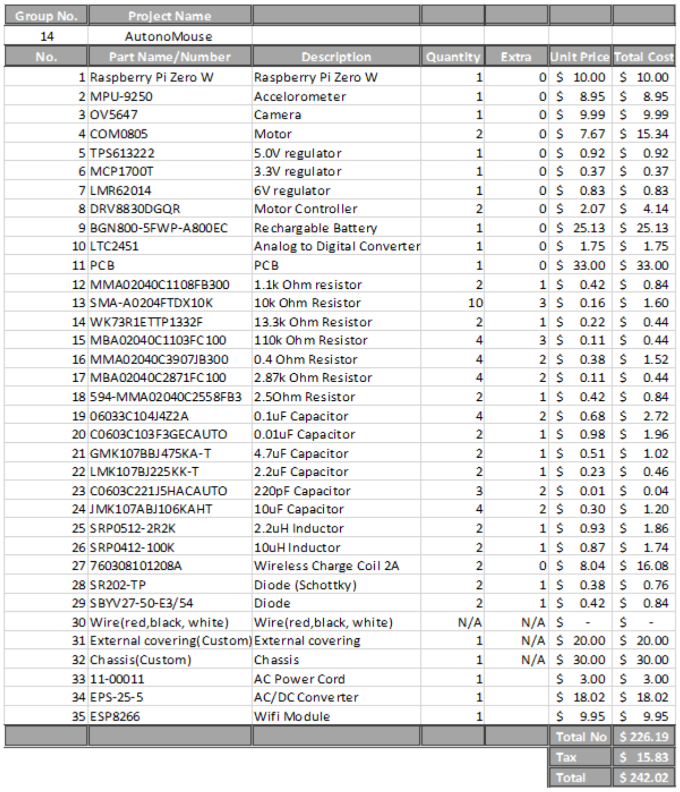


Figure 92: First generation protoboard(right) and second generation protoboard(left)

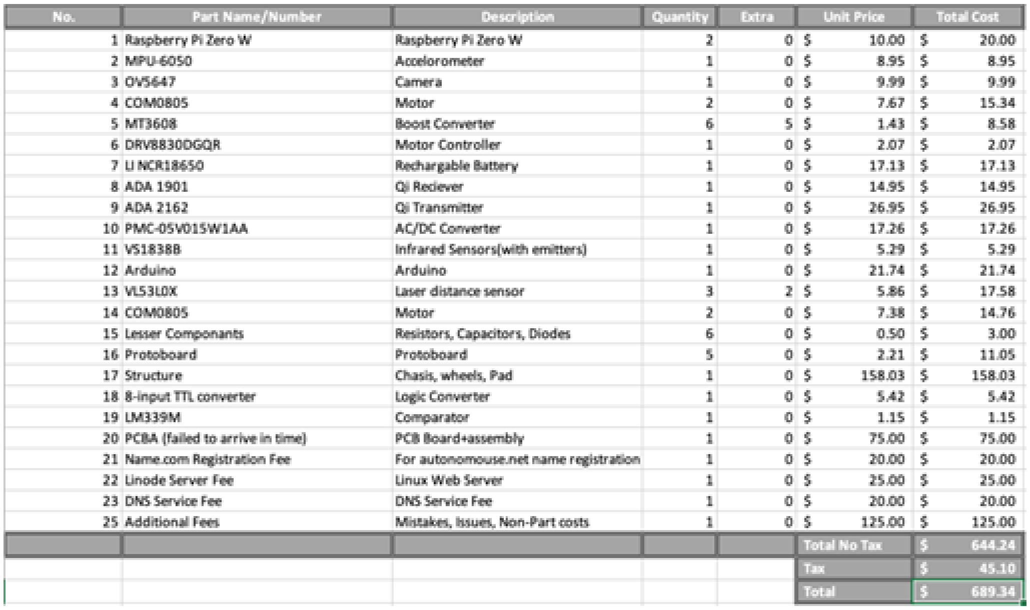
In order to incorporate the charging system and infrared sensors, a second generation protoboard was built. This board was designed to be situated at the back end of the mouse behind the motors, instead of above them. In addition to the previous parts included in the first generation board, this protoboard also housed the battery charging module, the boost converter to convert 3.6V into 5V, connections to the battery, 1901 Qi receiver and infrared sensors, and a simple custom ADC made out of an LM339N comparator. The crude ADC would compare reference voltage levels, created by 3 voltage dividers, to the current voltage on the battery, and output a signal when the value dropped below 3.9, 3.4 and 3.2V, which roughly equated to 80, 50 and 20% respectively. (see figure 91 and 92)
The Box component of the mouse was heavily dumbed down, and in the end it was mostly just a charging pad with a box shape that included the 2162 and a PMC-05V015W1AA.

6.5: Bill of Materials

The actual cost of creating Autonomous was over that of the estimated, this is due to both unforeseen costs, such as the purchase of a website to run the webapp and implementation of infrared as well as underestimating costs of some parts: charging coils ended up costing about 40 dollars, PCBA was in actuality 75 dollars and became a sunk cost when it failed to arrive in time.



Group No.	Project Name					
14	AutonoMouse					
No.	**Part Name/Number**	**Description**	**Quantity**	**Extra**	**Unit Price**	**Total Cost**
1	Raspberry Pi Zero W	Raspberry Pi Zero W	1	0	$ 10.00	$ 10.00
2	MPU-9250	Accelorometer	1	0	$ 8.95	$ 8.95
3	OV5647	Camera	1	0	$ 9.99	$ 9.99
4	COM0805	Motor	2	0	$ 7.67	$ 15.34
5	TPS613222	5.0V regulator	1	0	$ 0.92	$ 0.92
6	MCP1700T	3.3V regulator	1	0	$ 0.37	$ 0.37
7	LMR62014	6V regulator	1	0	$ 0.83	$ 0.83
8	DRV8830DGQR	Motor Controller	2	0	$ 2.07	$ 4.14
9	BGN800-5FWP-A800EC	Rechargable Battery	1	0	$ 25.13	$ 25.13
10	LTC2451	Analog to Digital Converter	1	0	$ 1.75	$ 1.75
11	PCB	PCB	1	0	$ 33.00	$ 33.00
12	MMA02040C1108FB300	1.1k Ohm resistor	2	1	$ 0.42	$ 0.84
13	SMA-A0204FTDX10K	10k Ohm Resistor	10	3	$ 0.16	$ 1.60
14	WK73R1ETTP1332F	13.3k Ohm Resistor	2	1	$ 0.22	$ 0.44
15	MBA02040C1103FC100	110k Ohm Resistor	4	3	$ 0.11	$ 0.44
16	MMA02040C3907JB300	0.4 Ohm Resistor	4	2	$ 0.38	$ 1.52
17	MBA02040C2871FC100	2.87k Ohm Resistor	4	2	$ 0.11	$ 0.44
18	594-MMA02040C2558FB3	2.5Ohm Resistor	2	1	$ 0.42	$ 0.84
19	06033C104J4Z2A	0.1uF Capacitor	4	2	$ 0.68	$ 2.72
20	C0603C103F3GECAUTO	0.01uF Capacitor	2	1	$ 0.98	$ 1.96
21	GMK107BBJ475KA-T	4.7uF Capacitor	2	1	$ 0.51	$ 1.02
22	LMK107BJ225KK-T	2.2uF Capacitor	2	1	$ 0.23	$ 0.46
23	C0603C221J5HACAUTO	220pF Capacitor	3	2	$ 0.01	$ 0.04
24	JMK107ABJ106KAHT	10uF Capacitor	4	2	$ 0.30	$ 1.20
25	SRP0512-2R2K	2.2uH Inductor	2	1	$ 0.93	$ 1.86
26	SRP0412-100K	10uH Inductor	2	1	$ 0.87	$ 1.74
27	760308101208A	Wireless Charge Coil 2A	2	0	$ 8.04	$ 16.08
28	SR202-TP	Diode (Schottky)	2	1	$ 0.38	$ 0.76
29	SBYV27-50-E3/54	Diode	2	1	$ 0.42	$ 0.84
30	Wire(red,black, white)	Wire(red,black, white)	N/A	N/A	$ -	$ -
31	External covering(Custom)	External covering	1	N/A	$ 20.00	$ 20.00
32	Chassis(Custom)	Chassis	1	N/A	$ 30.00	$ 30.00
33	11-00011	AC Power Cord	1		$ 3.00	$ 3.00
34	EPS-25-5	AC/DC Converter	1		$ 18.02	$ 18.02
35	ESP8266	Wifi Module	1		$ 9.95	$ 9.95
					Total No	$ 226.19
					Tax	$ 15.83
					Total	$ 242.02



No.	Part Name/Number	Description	Quantity	Extra	Unit Price	Total Cost
1	Raspberry Pi Zero W	Raspberry Pi Zero W	2	0	$ 10.00	$ 20.00
2	MPU-6050	Accelorometer	1	0	$ 8.95	$ 8.95
3	OV5647	Camera	1	0	$ 9.99	$ 9.99
4	COM0805	Motor	2	0	$ 7.67	$ 15.34
5	MT3608	Boost Converter	6	5	$ 1.43	$ 8.58
6	DRV8830DGQR	Motor Controller	1	0	$ 2.07	$ 2.07
7	LI NCR18650	Rechargable Battery	1	0	$ 17.13	$ 17.13
8	ADA 1901	Qi Reciever	1	0	$ 14.95	$ 14.95
9	ADA 2162	Qi Transmitter	1	0	$ 26.95	$ 26.95
10	PMC-05V015W1AA	AC/DC Converter	1	0	$ 17.26	$ 17.26
11	VS1838B	Infrared Sensors(with emitters)	1	0	$ 5.29	$ 5.29
12	Arduino	Arduino	1	0	$ 21.74	$ 21.74
13	VL53L0X	Laser distance sensor	3	2	$ 5.86	$ 17.58
14	COM0805	Motor	2	0	$ 7.38	$ 14.76
15	Lesser Componants	Resistors, Capacitors, Diodes	6	0	$ 0.50	$ 3.00
16	Protoboard	Protoboard	5	0	$ 2.21	$ 11.05
17	Structure	Chasis, wheels, Pad	1	0	$ 158.03	$ 158.03
18	8-input TTL converter	Logic Converter	1	0	$ 5.42	$ 5.42
19	LM339M	Comparator	1	0	$ 1.15	$ 1.15
20	PCBA (failed to arrive in time)	PCB Board+assembly	1	0	$ 75.00	$ 75.00
21	Name.com Registration Fee	For autonomouse.net name registration	1	0	$ 20.00	$ 20.00
22	Linode Server Fee	Linux Web Server	1	0	$ 25.00	$ 25.00
23	DNS Service Fee	DNS Service Fee	1	0	$ 20.00	$ 20.00
25	Additional Fees	Mistakes, Issues, Non-Part costs	1	0	$ 125.00	$ 125.00
					Total No Tax	$ 644.24
					Tax	$ 45.10
					Total	$ 689.34

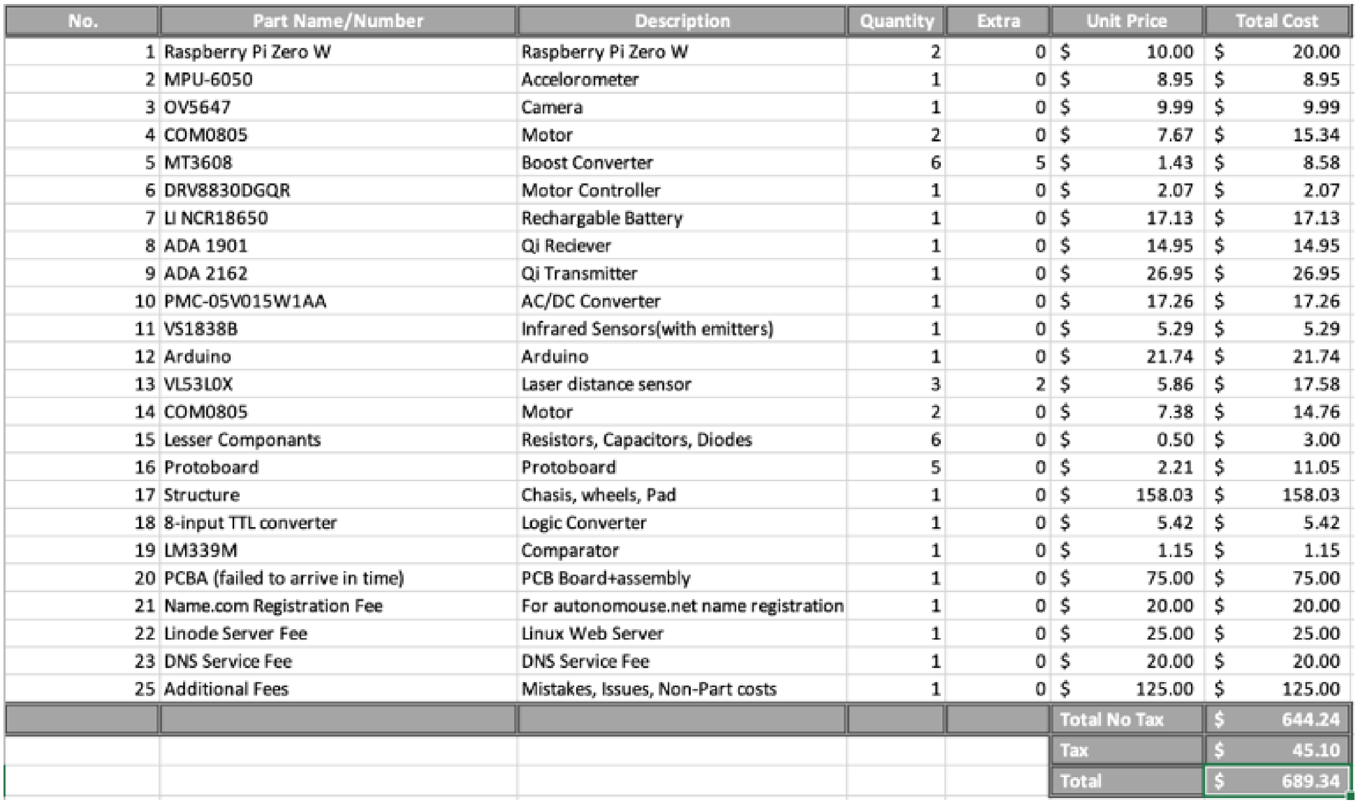
Figure 93: Bill of Materials Estimated(top) and Actual(bottom)

7.0 Project Prototype Testing Plan

A prototype is a model or early sample of a product built to test a concept or process. It is a good way to provide specifications for a working system rather than a theoretical one. It is used in a lot of areas like electronics and software programming. Testing a prototype is a very important step in this project as it will confirm that the final product will work as it is supposed and it will meet the consumer’s needs. Performing these tests early on can also help avoid very unfortunate scenarios.

There are a lot of different prototyping methods that can yield different results. These results will depend on the type of prototype that is made since it will impact the way that users will interact with it. It is important to consider testing goals since this will help develop many questions and testing scenarios. Lastly time constraints or knowing by when the product needs to be ready is very important to take into consideration when picking a prototyping method.

A prototype will differ from the final product for various reasons. The first one is the material that is going to be used. Sometimes the material of the final product can be very expensive, or it can take a long time to be developed. In this case the prototype will be made from different materials than the ones that will be used in the final product. The process or fabrication of the product could also be very expensive like for example expensive custom tooling. This may lead to using a different fabrication process for the prototype that will vary from the final product. Prototypes



No.	Part Name/Number	Description	Quantity	Extra	Unit Price	Total Cost
1	Raspberry Pi Zero W	Raspberry Pi Zero W	2	0	$ 10.00	$ 20.00
2	MPU-6050	Accelorometer	1	0	$ 8.95	$ 8.95
3	OV5647	Camera	1	0	$ 9.99	$ 9.99
4	COM0805	Motor	2	0	$ 7.67	$ 15.34
5	MT3608	Boost Converter	6	5	$ 1.43	$ 8.58
6	DRV8830DGQR	Motor Controller	1	0	$ 2.07	$ 2.07
7	Li NCR18650	Rechargable Battery	1	0	$ 17.13	$ 17.13
8	ADA 1901	Qi Reciever	1	0	$ 14.95	$ 14.95
9	ADA 2162	Qi Transmitter	1	0	$ 26.95	$ 26.95
10	PMC-05V015W1AA	AC/DC Converter	1	0	$ 17.26	$ 17.26
11	VS1838B	Infrared Sensors(with emitters)	1	0	$ 5.29	$ 5.29
12	Arduino	Arduino	1	0	$ 21.74	$ 21.74
13	VL53L0X	Laser distance sensor	3	2	$ 5.86	$ 17.58
14	COM0805	Motor	2	0	$ 7.38	$ 14.76
15	Lesser Componants	Resistors, Capacitors, Diodes	6	0	$ 0.50	$ 3.00
16	Protoboard	Protoboard	5	0	$ 2.21	$ 11.05
17	Structure	Chasis, wheels, Pad	1	0	$ 158.03	$ 158.03
18	8-input TTL converter	Logic Converter	1	0	$ 5.42	$ 5.42
19	LM339M	Comparator	1	0	$ 1.15	$ 1.15
20	PCBA (failed to arrive in time)	PCB Board+assembly	1	0	$ 75.00	$ 75.00
21	Name.com Registration Fee	For autonomouse.net name registration	1	0	$ 20.00	$ 20.00
22	Linode Server Fee	Linux Web Server	1	0	$ 25.00	$ 25.00
23	DNS Service Fee	DNS Service Fee	1	0	$ 20.00	$ 20.00
25	Additional Fees	Mistakes, Issues, Non-Part costs	1	0	$ 125.00	$ 125.00
					Total No Tax	$ 644.24
					Tax	$ 45.10
					Total	$ 689.34

sometimes will not have all the requirements that the final product will have. The final product will usually go through a lot of quality assurance tests to verify conformance with specifications.

There are many reasons why prototype testing is done. It can help identify potential faults and allows the consumer to give their opinions so that improvements can be made. It can be tested for safety issues and relevant regulations. Specific tests can also be carried out at this stage. Sometimes it can be assumed that the entire product is failing and needs to be redesigned when it might be a component. At this stage it is a lot easier to test each component separately then to redesign them. Testing similar designs or products of competitors can also be very helpful in this phase. Overall prototype testing can allow us to plan a more cost effective and efficient final product.

A prototype will only represent some compromise of the final product due to the differences in materials processes and design. It is also possible that the prototype may fail to perform in areas where the final product would perform acceptably. This does not mean the design of the prototype is flawed. While prototypes do reduce the risk that the design will not performed as planned it cannot eliminate all risks.

7.1 Hardware Test

There are different types of test to perform at each stage of the hardware product development. The main reason for hardware testing is to be able to exercise the design across multiple dimensions to validate that it will perform as it is supposed to once its built. This will help us save a lot of time and money. Some examples of these are the voltage, temperature, vibration and load. Another thing to consider are the limits of the design. Operating the Unit Under Test to failure with different reliability, durability and environmental tests can help with this. For now, we are focusing on hardware testing but later we will also cover software testing since a lot of the hardware products that will be used are driven by some amount of software.

At early stage testing you should consider usability test to help refine the industrial design of the product. Material and finish tests are also important to assess the many options for durability. To analyze the different mechanisms in isolation before a prototype is ready, critical component tests are also very useful. These tests will give you good feedback to create a working prototype.

At pre-production testing your prototype will be closer to the real thing. The focus is now switched to verifying a subset of the Unit Under Test performance and functionality. At this stage system-level stress tests, environmental tests, life tests and compliance certification tests can be performed. Here testable requirements become important and the focus is on verifying that the product is of enough quality before being used by the consumer.

Testing in the field is for situations were a product was tested when it was made but later need to be retested. This could be because it got damaged or there is a concern of damage, maintenance check. Testing will always keep going for as long as the assembly line is still running. It is very important to budget ample time and manpower for testing like any other development.

Testing methods will be different depending on the Unit Under Test but there is commonality in the test flow. These testing methods will often be stimulating the Unit Under Test either electrically or mechanically and sequencing through various test steps. Taking various physical property measurements like for example vibration, temperature, pressure, load, voltage and current. Analyzing the acquired data to calculate if the test either passed or failed and then reporting the results. For this project some of the parts we will be testing are the motors, battery, chassis, wheels and the voltage regulators. These may also include the power usage, power quality and voltages.

7.1.1 Voltage Regulator

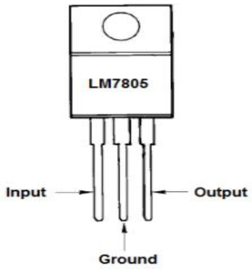


Figure 94: LM7805

As mentioned before a voltage regulator is a device that takes in an input voltage and then regulates it down to the voltage that is rated for. In this project we will be using voltage regulators since many parts will be running on different voltages. To test a voltage regulator, we must perform a voltage test. We will check the voltage input into the voltage regulator and the voltage output from the voltage regulator. If the voltage of the regulator read at the input pin and output pin is correct, then we can confirm it is working correctly otherwise it might be a defective voltage regulator. If we look at Figure 78, the voltage that needs to be regulated goes into pin 1 (Input) and the regulated output voltage comes out through pin 3 (Output).

We will start by measuring the input voltage. The first thing we need to check is that the enough voltage is being supplied to the regulator. If the regulator is not receiving enough voltage it will not be able to output the regulated voltage. We can do this by checking the voltage from the input pin to ground by using a multimeter placed in DC voltage setting. The voltage that is read should be higher than the voltage the regulator is rated to output.

We will also need to measure the output voltage. This can be done by checking the voltage form the output pin to ground by using the multimeter placed in DC voltage setting. If the multimeter reads a voltage that is near its rated output voltage, that means the voltage regulator is functional and working as it is supposed to.

7.1.2 Rechargeable Batteries

As mentioned before a rechargeable battery is a type of electrical battery that can be recharged many times as opposed to a disposable one which is supplied fully and discarded after use. For this project we will be using rechargeable batteries. These will be tested in the following categories:

- **Reliability:** A lot of batteries die prematurely for many reasons. It is important to make sure the batteries don't suffer from reliability problems.
- **High-Drain Performance:** Devices that need a lot of power quickly are referred to as high drain.
- **Capacity:** A rechargeable battery is measured by capacity which is the amount of electrical charge stored inside the battery. Having more charge in the battery will increase the electrical current it can deliver and the longer it can power your device.
- **Run time:** The run time of the battery should be good for at least 4 hours before having to recharge it.
- **Self-Discharge:** Self-Discharge means that the battery loses charge just by sitting around, unused. It would be good to have a battery with low self-discharge rate. The only problem with low self-discharge batteries is that they tend to have lower capacity so in the end you must choose between longer shelf life or higher capacity.

- **Voltage**: A high initial voltage can make battery life in high-drain devices last longer. There can be downsides to a high initial voltage but since we will be using a voltage regulator this problem should be avoided.
- **Voltage Drop:** A lot of rechargeable batteries maintain their voltage over the whole charge and then suddenly plummet
- **Charging:** Over charging can reduce the cycle life of the battery. Some chargers run on a timer. These will either overcharge the battery or fail to fill it up completely. It is important for the charger to stop charging when the battery is full to prevent this. It is also important for the battery to charge as fast as possible.
- **Cycle Life:** The batteries should be good for over a hundred charge cycles. To achieve this, we must prevent overcharging or waiting until the batteries are completely run down before charging them.
- **Recycling:** When the battery no longer useful because it won't hold a charge or its capacity, it would be good if it could be easily recyclable
- **Memory Effect:** The memory effect is when you repeatedly discharge a battery to the same level before charging it again. The battery will remember the discharge level and then it will always fail to allow the full capacity of the battery to be used. The battery can also suffer decreased capacity due to repeated discharging or overcharging.

7.1.3 Chassis

Since chassis provides the support to all the other element that compose the project, and it will be subjected to static loads and dynamics loads it's important that is tested to verify that it will not fail during operation. The drivetrain, that includes the wheels, wheel shaft, bearings, gears, and motor, is responsible for providing the forward/reverse motion and the direction control of the robot. The test will be divided into static load and dynamic load testing. Since the chassis and the drive train are the system that will be affected by the proposed tests, the test exemplar should be a chassis fitted with the drivetrain system.

The static load tests are designed to verify the performance of the system when subjected to a static load. For this test the chassis will be loaded on a flat surface with weights starting with two pounds and increasing in two pounds increments up to ten pounds. The weight range from two to ten pounds, is because this is the weight range of typical domestic cat. Although in real life applications the loads will be applied in a dynamic manner, the static load test results will provide information of weak points in the system.

The dynamic test will be divided in four categories: drop test, slope test, flat surface test.

- **Drop Test:** The drop test (Figure 79) will simulate a condition in the mouse as accidentally push over and edge and falls a predetermined vertical distance. The initial test will be dropping the robot from a height of 1 ft and increasing the height by 1 ft up to 4 ft. The robot shall be pushed in a very soft manner in a way the initial velocity is almost zero. The results from these tests will be documented and special interest will be given to any areas of failure.

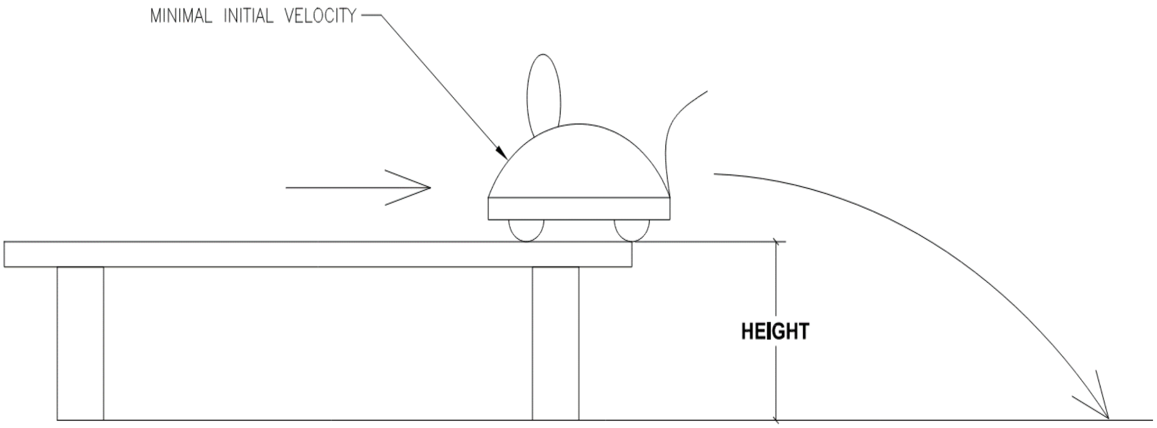


Figure 95: Drop Test

- **Slope Test:** The slope test (Figure 80) will measure the capability of the drive train to move loads on a slope. This test will also serve to measure the effective of the wheels used for the project, as the drivetrain may have the torque required to move the load, but the wheel cannot develop the required friction to go up the slope. For this application wheels with rubber covering the tread side is typically used. The test will be to make the robot go up a 48 inches long sloped surface. The test will start a 15° slope and will go up to 45° in 15°

increments. The slope test matrix will be repeated without load a then with two pounds and four pounds. This test will operate the chassis and the drivetrain on extreme conditions that will expose any weakness in the drivetrain system.

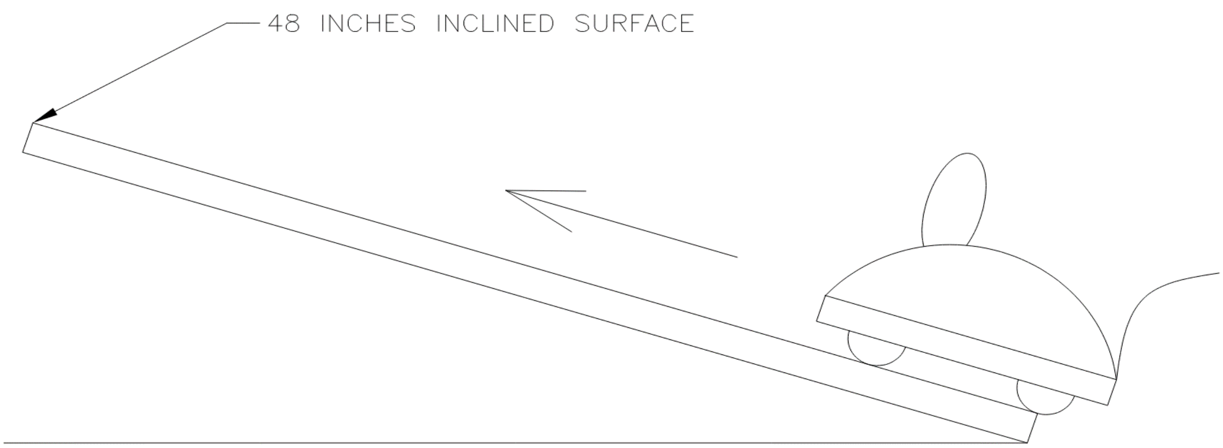


Figure 96: Flat Surface Test

- **Flat Surface Test:** The flat surface test (Figure 81) is designed to test the directional control of the robot and the load carrying capacity. For this test 96 inches by 48 inches figure track will lay out on a flat surface. The mouse will run the track completing tens laps at the maximum possible speed. The test will be repeated without any load and then load starting with two pounds up to ten pounds, in two pounds increments. The drivetrain should be able to maneuver the track with additional. The additional load simulates a condition in which the cat is standing on top of the robot. The temperature of the drivetrain components shall be monitored during the testing using an infrared thermometer.

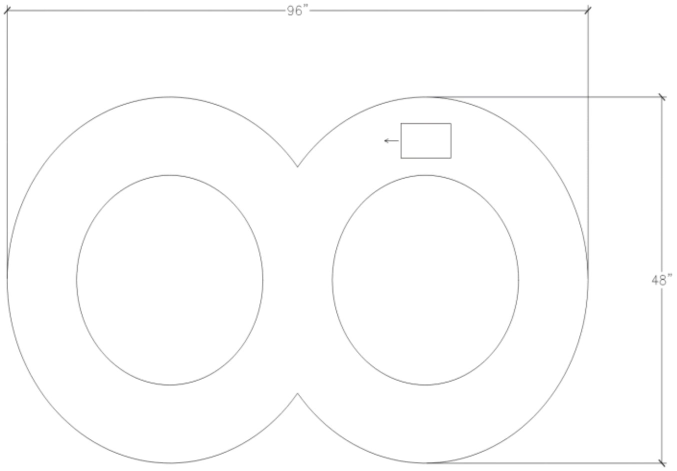


Figure 97: Flat Surface Test

Although the previous test will be executed to validate our design, it doesn’t mean that these are the only tests required to fully verify the operation of the chassis and drivetrain. We may find out during testing the several materials for the wheel treads will have to be evaluated to determine the most suitable for our application. Another condition that may arise during testing is an overheating condition in the drivetrain, which may require further testing of gear ratios or reselection of the electric motors.

Finally, another important aspect is safety, could the robot release small part that can become chock risk, or could some part start to overheat causing a fire hazard. Although the test protocol presented in this section are not so exhaustive to make a definitive conclusion about the safety of our product, some deficiencies will maybe be observable.

7.1.4 Power Test

Power supply is the foundation for electronic devices, and because of this testing, its performance is essential. This ensures a high quality and reliable product. Not testing it can lead to potential problems after the final product is made. Power supplies can operate fine under typical conditions but could be on the edge of regular operation. A lot of things like being heated or cooled or aging can potentially cause a power supply to fail.

System conditions and environment can vary greatly from application to application. To perform the test, it is important to understand which tests are needed and how to perform them. Test specifications and a test plan should be established for the power supply. These should include all acceptable operating limits and operating conditions under which the final product must operate. This will ensure the final product meets the test specification. Some of the test that could be considered for this project:

- **Output Voltage Accuracy and tolerance -** As voltage levels in a system decrease, the output accuracy requirements will become more demanding. Sometimes circuits require a tight tolerance, and this can require a combination of the initial accuracy as well as other contributions. Output accuracy is easy to measure, but it does not show what will be the worst-case accuracy overproduction variations in component values. This can be a constraint that is best determined by simulation or calculations.
- **Start-up time and Overshoot –** It is important to consider the amount of time the power supply takes to start providing a clean output voltage. Sometimes the power supply can be designed to start after the input rises above a specific voltage level. The high inrush current will be higher the faster the startup of the power circuit is. This can result in a drop in the system voltage, and it can be worse the power is limited.
- **Current Limit –** A lot of power supply designs have a method to limit current in case of extreme conditions or failure. Current limits are mostly used for safety and component protection when fault conditions occur. It can also be used to limit the current during normal conditions. We can obtain the current limit of our design by increasing the load until the output voltage drops by a specific amount, which can vary depending on the intention of the current limit.

- **Efficiency** – The efficiency of a power supply is considered to be the amount of energy that goes out from the circuit divided by the amount of energy going into the circuit. Accurately measuring the efficiency of a power supply is not hard, but mistakes can always occur and result in significant inaccuracies.
- **Stability -** Power supply stability is a very important and there are many methods to easily measure it.

7.2 Software Test

Software testing is a process to evaluate the functionality of a software application with an intent to find out if the developed software met the specified requirements and to identify were it is failing. This involves the execution of system components to evaluate one or more properties of interest. This is done to identify errors or missing requirements early on and before the delivery of the final product.

According to ANSI/IEEE 1059 standard, a process of analyzing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item. The main reasons why software testing is important are to ensure product quality, security, customer satisfaction and cost-effectiveness.

There are many benefits to software testing. The first one is that it is cost effective. Testing the software early on can help you identify errors and in the long run it will cost less to fix them. Security is also very important and a sensitive benefit of software testing this is because people want to use products they can trust. The quality of the final product is also very important, and testing will ensure that a quality product is delivered to the customer. Finally, customer satisfaction is the main aim of the product. Software testing will ensure that the user has the best experience.

There are also important strategies in software testing. Unit testing helps the developers know if an individual part of the code is not working as it should. Integration testing helps with the construction and design of the software. With this you can see if the integrated units are working without errors. System testing is where your software is compiled as a whole and then tested completely. This will check for functionality, security, portability and other things.

7.2.1 Web App

For this project a Web Application will be used to allow the user to control the mouse and see a video stream of the camera that will be placed in the mouse. The Web App needs to be tested before going live to make sure that it is functioning properly and that it can be accepted by real time users. The things that need to be considered are functionality, usability, interface, compatibility, performance and security.

- **Functionality Testing:** For the functionality test there are a few things that need to be checked starting with all the links in the web page. The test forms on all pages should also be tested. These are used to receive information from the user and interact with them. It is important to check all validations and default values in the fields. Wrong inputs and options to modify, create, vie or delete the forms should also be checked. Cookies are small files stored in the user machine that are used to maintain the session. The Web Application should be tested by enabling or disabling them from your browser options. It should also be checked if they are encrypted before writing to the user machine and the effect on the web application when they are deleted. Another thing to consider is validating HTML and CSS because it is good for optimizing the Web Application for search engines. Testing the database for consistency is also very important. Checking for errors and integrity while doing any database related functionality or modifying any of the forms. It is also important to test that all the data is retrieved and updated correctly. The queries in the database should also be tested to check they are executing correctly.

- **Usability Testing:** Usability testing is a process by which we can test how easy it is to use something with real users. This is usually done by asking users to complete tasks and evaluating where they get confused or encounter problems. The Web Application should be easy to use, and the provided instructions should be very clear. There should be a main menu provided in each page and a sitemap should be available with all the links on the website and a tree view navigation. The colors, text, fonts and pictures should be placed properly and clearly.

- **Interface Testing:** Interface Testing verifies if the communication between two different software systems is working correctly. It is important to check that the communication is done properly and that it is compatible with the software, hardware, network and database. The main interfaces are web server and application server interface and application server and database server interface. All interactions between these servers should be tested. Errors should be handled properly. If an error message for any query by the application server is returned, the application server should catch and display the appropriate message to the user.

- **Compatibility Testing:** The compatibility of the Web Application is very important. This includes browser, operating system and mobile browsing. A lot of Web Applications depend on browsers. Depending on the browser the configurations and settings will be different and sometimes they might not be compatible with a Web Application. To avoid this the code that will be used for the Web Application has to be cross browser platform compatible. To confirm that this works the Web Application should be tested on different browsers like Google Chrome, Internet Explorer, Safari, AOL and other browsers. It should also be able to run on mobile devices. To do this we need to make a responsive web design which can either be a conventional website or a single page application that is viewable on small screens that work well with touchscreens.

  Another important thing is the compatibility with the operating system. A lot of times Web Applications are not compatible with all operating systems. This can be due to the development like graphics designs or interface calls with different APIs which may not be available in all Operating Systems. Therefore, it is important to test the Web Application on different Operating Systems like Windows, MAC and Linux. Lastly the Web Application should be able to run in mobile web browsers.

- **Performance Testing:** For the performance tests the web application should be able to sustain a heavy load. The test that should be performed are a web loading test and a web stress test. For the web loading test the Web Application should be able to handle many user requests, large input data from users, simultaneous connection to the data base and heavy load on specific pages. The web stress test is performed to crash the Web Application by giving it a lot of stress and check how the system reacts and recovers from it. In other words, we are stretching the system over its specified limits. Testing the Web Application functionality on different hardware platforms and different operating systems is done to check for memory leak errors. The performance test can also be used to get a better understanding of the Web Applications scalability. It can also be used to benchmark the performance on third-party products. A connection speed test should also be performed on different networks.

- **Security Testing:** Security Testing is performed to check for potential vulnerabilities and fix them. Some of examples are network scanning, vulnerability scanning, password cracking, log review, integrity checkers and virus detection. We can test the security by writing the internal URL into a browser and make sure that internal pages are not opened. Check how the system reacts to all invalid inputs and make sure web directories and files are not accessible. All error messages and breach attempts should show up on the files in the webserver.

7.2.2 Obstacle Avoidance

The obstacle avoidance technology is an important requirement for this project, and it is used for detecting obstacles and avoiding collision. This will require the help of a lot of sensors and there are different types used for obstacle avoidance. Some examples are IR sensors, proximity sensors and ultrasonic sensors.

The IR sensors can be used for obstacle detection. IR sensors can detect objects by emitting a short ultrasonic burst and then listening for eco. While under the control of a microcontroller it will emit an explosion that travels through the air, hits the obstacle and returns to the sensor. The sensor will then provide an output that terminates when an echo is detected. In other words, the IR sensors work by stopping the mouse immediately if any obstacle placed inline of the IR sensor fails to receive the light rays. After one minute it will check if the obstacle is still there. If it is the mouse will return to its starting place and if it’s not the mouse will keep moving forward.

A proximity sensor is a device that can detect the approach or presence of objects nearby without the need of physical contact. These sensors give guidelines and the device follows them by going on a straight path. There are different kinds of proximity sensors like inductive, capacitive, photoelectric and magnetic. They are used for path detection and they work by making the device turn left when a signal from the right sensor is not detected in the curve line. If the signal from the right sensor is detected, then it would be activated to go forward. When the line ends the device reverses at 180 and turns back to the same place.

For this project we will most likely be using an ultrasonic sensor since they are one of the best for obstacle detection. They also have a high range capability and are very low cost. The ultrasonic sensor transmits ultrasonic waves continuously from its sensor head whenever it is going in the right direction. If an obstacle gets in the way the ultrasonic waves are reflected from the object and that information is transmitted to the microcontroller. This will then control the motors and the direction based on the ultrasonic signals. It can enable the mouse to virtually recognize an object, avoid obstacles and measure distance.

After wiring the sensor and uploading the code the mouse should be able to move in a direction while avoiding all obstacles in its path. It should be able to measure the distance between the mouse and the obstacle. It should also be able to detect all objects inside a home regardless of their size or material. Once it detects an obstacle it should be able to calculate an alternative path. If it detects an obstacle in the middle of its path is should change directions until the sensors can no longer detect an obstacle.

7.2.3 Interface Testing

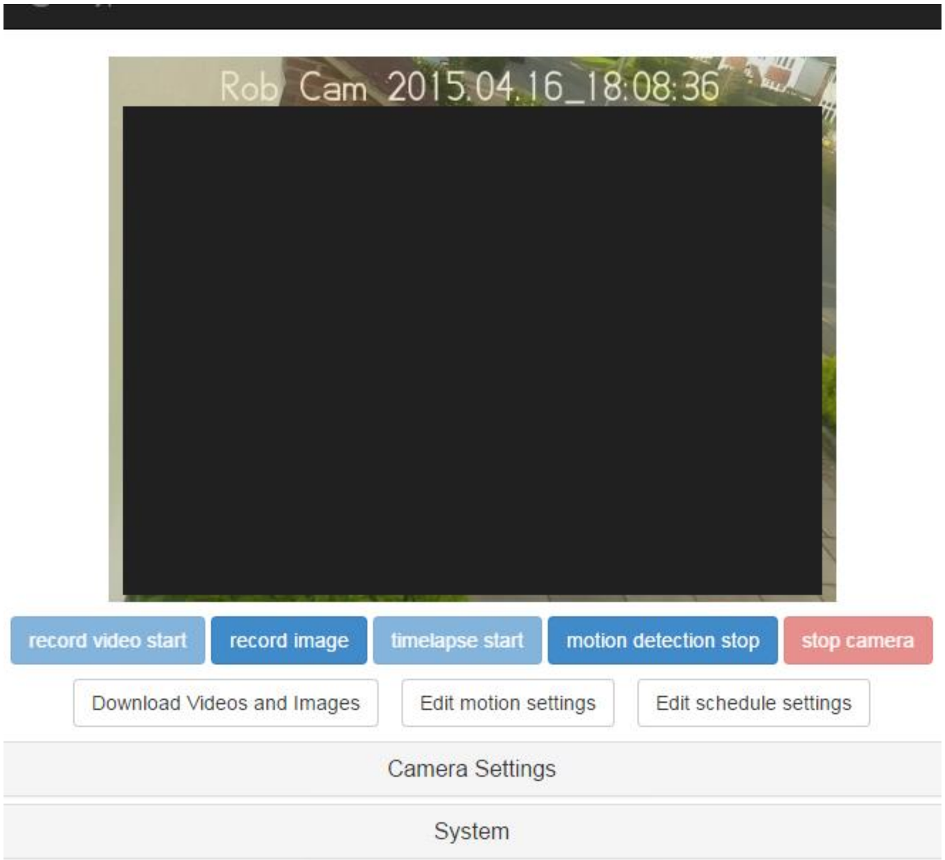


Figure 98: 7.2.3

As mentioned before, interface testing is a type of software testing that verifies the communication between two different software systems is working as it is supposed to. Any connection that integrates two components is considered an interface. These can consist of a set of commands, messages, and other things that enable the communication between a device and a user. For this project, an interface will be used for the camera and motor control. This will allow the user to be able to control the mouse and see where they are going through the web application.

Interface testing can be very important for various reasons. It can help us identify which application areas are usually accessed by end-users and check how easy to use they are. It can help us verify security requirements while communicating between two systems at the same time. It can also check if a solution is capable of handling network failures between an application server and a website.

Interface testing can be performed by checking that the servers are executed properly. Making sure that errors are also handled properly, and an error messaged is returned for any query made by an application. Checking the outcomes when resetting the connection to a web server. These are very important for web server and application server interface and application server and database server interface.

Abstract test cases can be used to create concrete instances of test cases for each implementation of interface testing strategy. This can be done without worrying about the implementation. Abstract test cases can perform implementation neutral tests. Concrete tests can instantiate objects to test and perform implementation specific tests. There are different tests to consider when doing interface testing. Some examples of these are:

- **Workflow:** This will make sure that the interface engine can handle the standard workflows as expected.
- **Edge Cases/ Unexpected Values:** This is used when testing requires a date, month, or day reserved.
- **Performance, load, and network testing:** Depending on the interface and connectivity infrastructure, a load test could be required. This es especially necessary with high volume interfaces.
- **Individual Systems:** This means that each system should be tested individually.

7.2.3.1 Raspberry Pi Camera Web Interface

The Raspberry Pi camera web interface is a web interface for the Raspberry Pi Camera module. It works on all Raspberry Pi models that are supported by the camera model. The web application is designed to run on the Raspbian Linux distribution. It is one of the most versatile and powerful tools that can turn the Raspberry Pi and the camera module into a platform for all different kinds of projects.

When opening the main web site (Figure 7.2.3), we will be able to see a medium resolution view of the camera. We will also have access to the camera setting control buttons and system controls. Some of these basic controls allow for capturing still photos, videos, or time-lapse sequences. A motion detection state, where video recording activity will be triggered when motion is detected, can also be used. Streaming mode is also possible through the system control bar. To ensure that this is working properly, the user should be able to see the stream through the web application.

7.2.3.2 Motor Control Interface

The motor control interface needs to be able to control at least two motors. This will help control the direction of the mouse. To ensure that this is working properly, the user must be able to control the mouse from the web application. This should work regardless of how far away the user is from the mouse.

7.2.4 Box Test

For this project, besides the user, the box will also be providing instructions to the mouse. Because of this, the mouse and the box need to be able to communicate wirelessly. Creating a testing plan might be a little bit complicated since it involves. The following are tests that should be considered to ensure that the box will be working as expected.

- **Connectivity Test:** Testing the connectivity as early as possible is very important to ensure the final product will meet the users ends. For this project, Bluetooth and Wi-Fi are being considered.
    - **Bluetooth Test:** Bluetooth uses radio frequencies to send signals between devices. If the wireless connection is blocked, this can weaken or drop the connection. This can sometimes be a hardware issue and other times a software issue. This could be due to many reasons like other Bluetooth or Wi-fi devices getting in the way of the connection, interfering radio frequencies, physical barriers, or channel hopping. It is important to make sure the software and hardware of the device are compatible with Bluetooth. A test device can be used to run a full diagnostic test to make sure the device is operating correctly. Then we can start looking for other factors that might be interfering with the signal.
    - **Wi-fi Test:** The Wi-fi can be tested by sending AT commands to the Wi-Fi module and checking that it responds as expected. The commands can be sent from any serial terminal program. If the module returns unreadable characters, the problem could be fixed by changing the baud rate.
- **Output Power Test:** All products that rely on wireless connections have a transmitter with a specific power level and threshold. It is important to know the power in the transmitter connection.
- **FCC Regulation Test:** There are specific frequency ranges at which it is legally allowed to transmit otherwise, it could interfere with other devices. It would be better if the Wi-Fi module being used id FCC certified.
- **Electromagnetic Interference Test:** Electromagnetic interference can happen when one electromagnetic field comes in contact with another. This can cause transmitting problems within the device.
- **Thermal Test:** Since the Box will remain indoors, overheating could be a potential problem in poor ventilated rooms or hot climates. This can ruin the electrical circuit

components or, even worse, lead to a fire. To test this, the temperature change of electrical board parts in different environments should be recorded. Materials with overheating problems should be replaced with more heat resistant alternatives. Another option could be to place overheating prevention tools like, for example, fans or heat sinks.

8.0 Project Operation

The AutonoMouse is a robotic toy that's used by the user to interact with his/her cat. The design of this device allows the user to control it over a Web app. As long as both the user and the AutonoMouse has internet access, the user can control it from anywhere.

8.1 Safety Precautions

We know that you are very anxious to start playing with your AutonoMouse, but it is very important that you take the time to read the manual before operating. The AutonoMouse is an overall safe device to operate, however, it is an electrical device, some precautions are recommended as follows:

- Although the AutonoMouse is a toy, but it's not intended for children under 5 years of age, unless closely supervised by an adult.
- The AutonoMouse is not waterproof, please keep it away from wet areas to avoid damage to the device.
- Please do not run AutonoMouse on public roads, congested arear, or anywhere you may encounter pedestrian or vehicle traffic.
- Before turning on the switch, please check if the device is fully closed without any wiring exposure. If there is, please contact us for a replacement.

8.2 General Information

The AutonoMouse has the following functionalities:

- The user can control the mouse from anywhere as long as both the user and the mouse has internet access.
- Auto-play: The mouse comes out of its house and drives around periodically to attract the cat.
- Manual Control with "Vision". The user will be able to see what the mouse is seeing from the camera feed.
- Self-Charge: the user can tell the mouse to go home and charge itself when the battery level is low.

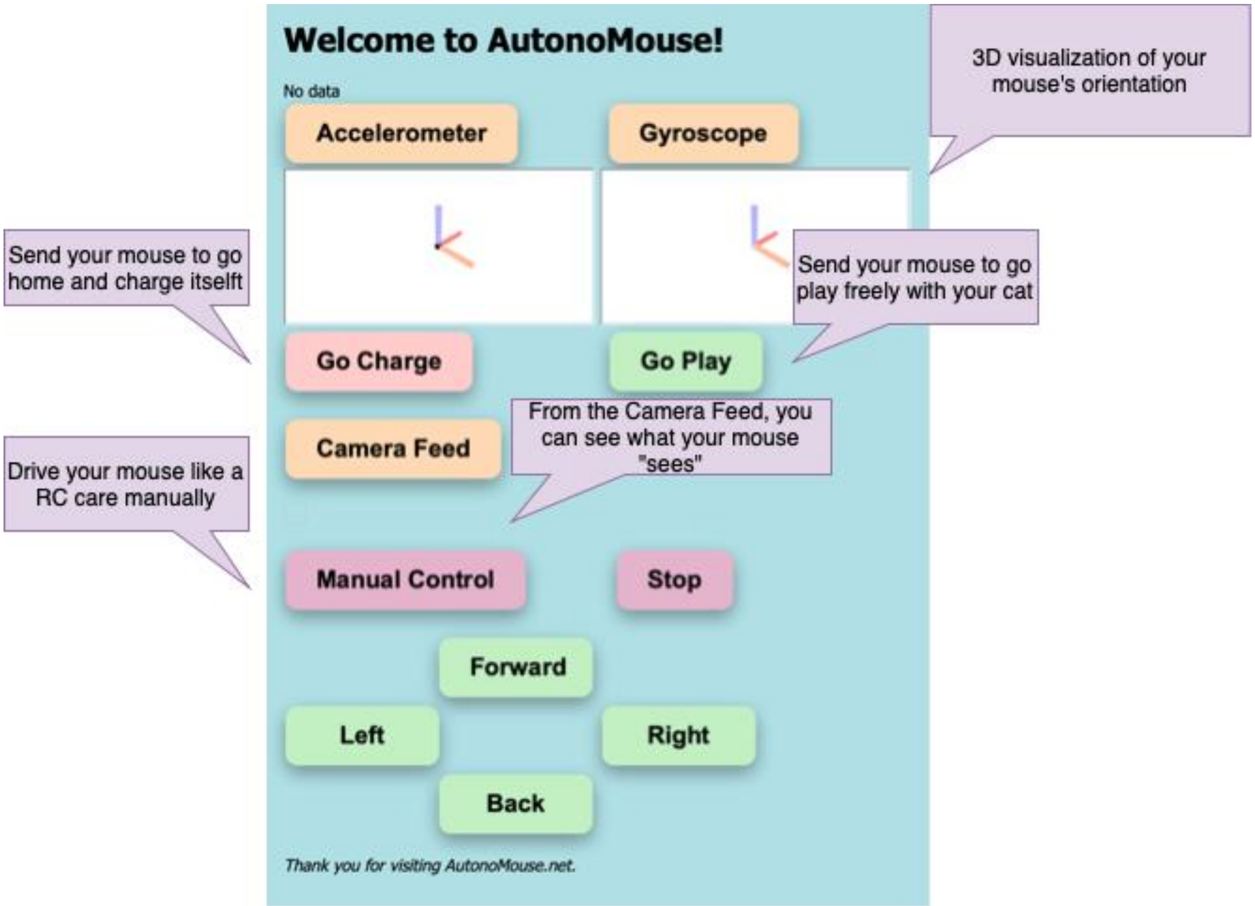
8.3 Initial Setup

This part is to get your AutonoMouse connect to your local internet so that you can control it from anywhere that has internet access.

Turn the AutonoMouse's switch on and please go to 192.168.0.1 on a phone or computer, select a WIFI for your AutonoMouse. You will only need to do this step once if the mouse is staying at the same home, but if you intend to move the mouse to different places with different internet access, then this step will need to be repeated each time an WIFI access is changed.

8.4 Getting familiar with the Web App

Go to http://autonomouse.net and you should see the Web App for your AutonoMouse. Below is a diagram shows you all the functions.



3D visualization of your mouse's orientation

Send your mouse to go home and charge itselft

Send your mouse to go play freely with your cat

From the Camera Feed, you can see what your mouse "sees"

Drive your mouse like a RC care manually

Figure 99: Web App Instructions

8.5 Trouble Shooting

We don't recommend that you disassemble the AutonoMouse on your own in any situation, as it might cause further damage to the device, for that reason, we cannot provide a replacement for a malfunction device that is disassembled. If for whatever reason the AutonoMouse is not moving at all, please check and see if the following two situations apply:

- Did you connect your AutonoMouse to your WIFI? If not, please follow the previous instructions on "9.3 Initial Setup".

- Is the battery low? Place the mouse on its charging station and wait for two hours and try again. (Be sure to connect the mouse to your WIFI if you haven't done that)
- If neither of the above worked, turn the device off, wait for at least two minutes, and then turn it back on, check again.

If the device is still not responding after the above steps, then please contact us for further troubleshooting or a replacement. Thank you for your patience!

9.0 Administrative Content

Milestones, project summary and conclusion will be discussed in this section.

9.1 Milestones

The entire project is separate in two parts, Senior Design 1 in Fall, 2020 and Senior Design 2 in Spring, 2021. Senior Design 1 is more focused on documentation of the project, and Senior Design 2 is more focused on the implementation of the project. Table 3 on page 10 gave a general idea of the milestones with timelines of our project. More details on each milestone is explained as follows.

9.1.1 Senior Design 1

The milestones for Senior Design 1 are mostly about research, design, and documentation. We have four members in our group, each of us is required to write 30 pages average with a total of 120 pages for the final document.

9.1.1.1 Project Selection

Many ideas were shared among the group members during our many brain storming sessions. We eventually narrowed down to the following 5:

- Pianist's Gloves. The motivation of this idea was to help an injured pianist. The main feature of the tool is to direct the fingers to play the piano along the music that's loaded into the program. Depending on the severity of the injury, this tool could help the pianist to "feel" again if his/her hands are completely unfunctional; it could help a partially injured pianist to practice for rehabilitation purpose. This could also be used as a learning tool for anyone that's interested.
- Smart Ring Alarm. The motivation of this idea was to eliminate disturbance towards partners while sleeping, the ring would produce a very soft vibration to act as an alarm or any type of notifications depending on the customized settings. With all the smart jewelries on the market, this could be the next hot item!
- Power Controlling Outlet. With all the smart home installations, this could be a very useful added feature. The motivation behind this was actually from finding a way to monitor a teenager while the parents are away. For example, if the teenager misbehaves, his/her gaming system power outlet could be turned off.

- Remote Garbage Can. Having robots to do all our chores would be a dream come true, that is the motivation behind this idea. Some people often forget to take the garbage cans to the street before the collection day, with a pre-programmed remote garbage can with vision, life will be easier.
- AutonoMouse. We decided to go with this idea because it's more feasible at our knowledge level, also more fun. The pianist's gloves might be too complex since it would be difficult to place the "gloves" with hands correctly where they would press the keys correctly. The smart ring might be too software heavy since it needs to extremely small. The power controlling outlet might be too dangerous to test on, and the remote garbage can might just be too big of scale for the parts. So we went with a fun and furry robot friend to help a cat owner in need.

To finally select the AutonoMouse idea for our project was our first milestone. This was quite difficult because this was new to all four of us, and we never actually "built" something from scratch.

9.1.1.2 Initial Divide & Conquer Document

This was the next difficult thing to do. After we finally decided on the project idea, we still weren't clear on the exact details of the project. This initial document required us to provide specifications, constraints, software and hardware diagrams, house of quality, finance and budgeting. This is almost the entire project in a 10-page document. We did our best on researching and put together the initial document, but we believe many things are subject to change.

9.1.1.3 Meeting with Professors about the Project

This purpose of this meeting was mainly to approve or disapprove of our proposal of the project. Luckily, we got approved.

9.1.1.4 Update Divide & Conquer Document

Since our project proposal was approved, we didn't have to make major changes to the initial document, only a few modifications in some parts.

9.1.1.5 60-Page Draft

This part was also very difficult. We didn't even know where to begin with this document, none of us has written such a lengthy document before. The guidelines of what must be included was somewhat helpful, but it still didn't give a good structure until we saw the example table of contents. From there, we distributed work among us based on the example table of contents since it also gave number of pages in the example.

9.1.1.6 100-Page Draft

With following the example table of contents, most of the 60-page draft did not start talking about the details of our project until Part 5, which is towards the end of the 60-page draft. From there on, more exact details of our project were included in the 100-page draft, such as the schematics of the hardware, the BOM, and the software design since this is a software heavy project.

9.1.1.7 Final Report Senior Design I

We are expected to write a 120 page document for the final report. It is very hard to believe we are so close to being done with this first chapter of our project. We understand that we might have some flaws in our design, and certain features we are hoping to achieve might not be feasible. We are looking forward to actually build our project in Senior Design 2 and many things are subject to change from this final report.

9.1.2.1 Build Prototype ("Basic Prototype")

The first milestone is to build the prototype, and this includes both hardware and software. It is important for us to get the Cloud part working as soon as we can because every piece of hardware needs software to communicate, we would need to test on individual part with software before assembling everything together. We hope to achieve the following 4 sub milestones in this phase.

- Cloud Prototype. For this milestone, Virtual Private Server (VPS) will respond to its domain name via DNS and hosting a website via nginx or Apache. This will probably not be the final user interface, but the VPS is ready to host any frontend or user interface code.
- Hardware Communication Prototype (Raspberry Pi, GPIO, soldering). All motors, motor controller, and sensors are electrically connected to the Raspberry Pi. This part might not include the camera, as it is a simple USB connection. Individual device protocols from their respective datasheets will be implemented. Simple demonstration programs will be written that send instructions or poll sensors with a command-line interface. The reason we want to test with a command-line interface first is so that we can make sure it works before implemented it in the frontend interface, this way we don't have to keep on modifying the frontend interface each time there's an error. Motors are speed controllable, and 'raw' values from distance sensors or accelerometer can be shown.
- Frontend Interface Prototype (Javascript, HTML, CSS). This part is to test web interface, using dummy data, simulate sensor data and appropriately display it on the screen. Buttons for all features are on-screen, but may not control anything yet.
- Backend Interface Prototype (Scripting language like Python, supports CGI). This will be an Application Program Interface (API) for all functions that the mouse can perform, and all requests the frontend can make. Dummy data to a suitable, custom protocol will be used. Authentication may or may not be implemented at this stage.

9.1.2.2 Testing and Modifying ("Integrated Prototype")

Assuming we've achieved the first milestone, now we can test the prototype as a whole. Lots of trial and error will be done during this phase. We need to make sure our mouse is "listening" to the commands that are sent from the user interface. We will be troubleshooting any errors that might come up during this phase, and we will also modify our documentation accordingly.

- CGI Scripts, The middleware:
    1. Cloud server runs the prototype back-end code. This means nginx or Apache (the web server) correctly forwards API-type requests to the back-end. For example, we will make sure all requests from website "autonomouse.com" go through the

web server and API sub-directory such as “autonomouse.com/api/” goes to the back-end.
2. Back-end API calls hardware scripts or re-uses the code for those hardware commands.
3. Back-end API calls use authentication, which is critical to prevent attacks against the mouse’s hardware from across the internet.

- Cloud Control. The hardware connects to the cloud server and opens a “reverse shell” so that Cloud Users can control the device over the internet.
- API Integration. The front-end API runs on the cloud server and talks to the real backend API and uses the reverse shell to send commands to the mouse, even over different local networks across the internet.

Below is a critical path diagram of above two milestones:

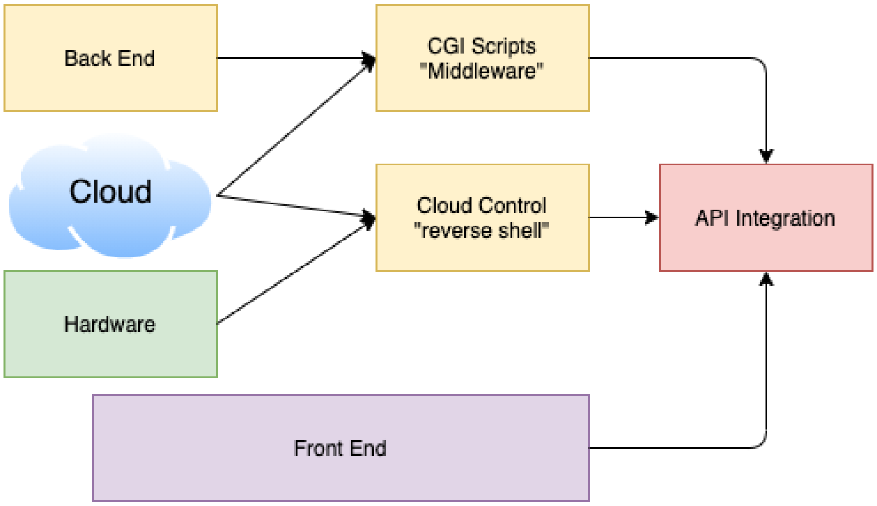


Figure 100: Critical Path

9.1.2.3 Finalize Prototype (Presentation-ready Prototype”)

“Full stack” test will be done in this phase, All commands from the HTML/Javascript frontend execute via the cloud and are performed by the hardware. Commands sent by the user are being executed on hardware irrespective of local network. Below is a figure of one possible “Full Stack” we might be implementing:

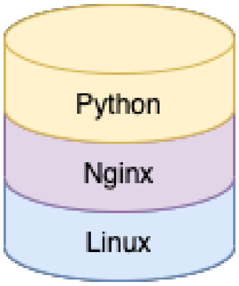


Figure 101: "Full-Stack"

Please note that usually the “Full Stack” also includes a database part, but since this is a prototype, we won’t be using the database. However, if this were becoming a commercial product, we would then need to implement a database to keep track which user controls which AutonoMouse. This is also the “beautifying” phase. This is the part where we make sure that the product is not only functional, but also attractive to the users.

9.2 Project Summary and Conclusion

Project AutonoMouse was in totality a successful device. It was capable of entertaining a household cat whether user controlled or in an autonomous state as was the goal. There were some issues with getting it to accomplish the second mode, finding the charging pad, but it is strongly believed that with some additional time, as well as a true PCB board, that task could be accomplished. There were a number of hardships when implementing and prototyping this device. Creating a webapp on top of the additional programming already required for the other two modes proved very challenging to do. Many parts were changed by the end of the project due to either unavailability or inadequate performance, leading to redesigns of the hardware and software system to accommodate these changes. The lack of a PCB, which was delayed past the completion date due to the aftermath of COVID-19 led to the whole system being put on a protoboard, which was quite a daunting task and resulted in many fixes needing to be performed. The Box component of AutonoMouse went from being a highly intelligent device to a more simplistic charging station. Finding a sensor system to allow the mouse to properly detect and seek the Box when low power became a rather difficult task.

Overall, we accomplished what we needed to in order to make a product that would help pet owners feel less guilty about leaving their pets alone. Many of the ideas included in this report were not added into the project, but instead were replaced by ideas we thought were better for both our constraints, and in general. Looking back, time was the biggest issue for this project. There were so many features we wanted to add, and didn’t consider how long building, testing, and implementing some of these features would take. However, what we did include were designs that were crucial to the mouse’s function and towards accomplishing our goals. If we were to do Senior Design II over again with the same goal, we think we would take a step-by-step approach instead of what we attempted to do, which was separating our tasks without knowing what was going to work, and what wasn’t. By actively working on single components, we could more confidently go forward with implementing pieces into the project. This was a great learning tool for all of us, and we feel much more educated in our fields after working together and gaining new knowledge to accomplish a task.

Appendix

Citations:

1. DFrobot. “Devastator .” *Amazon*, Amazon, https://www.amazon.com/DFROBOT-Devastator-Tank-Mobile-Platform/dp/B014L1CF1K/ref=sr_1_1_sspa?dchild=1&keywords=dfrobot+devastator&sr=8-1-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUFUWVMxVlhLRkZOODkmZW5jcnlwdGVkSWQ9QTAyNTUyMTMzNTA0R1A5N0IzTVBOJmVuY3J5cHRlZEFkSWQ9QTEwNDY0NTIxRDVLV0FGTDNKUzU3JndpZGdldE5hbWU9c3BfYXRmJmFjdGlvbj1jbGlja1JlZGlyZWN0JmRvTm90TG9nQ2xpY2s9dHJ1ZQ==
2. SunFounderDirect. “Smart Video Car Kit” *Amazon*, Amazon, https://www.amazon.com/SunFounder-Raspberry-Graphical-Programming-Electronic/dp/B06XWSVLL8/ref=sr_1_5?crid=3UYDZ3D0HVN02&dchild=1&keywords=sunfounder+smart+video+car+kit+v2.0+picar-v+raspberry&sprefix=sunfounder+%2Caps%2C353&sr=8-5
3. JOEJOY. “JOEJOY Cat Toy” *Consent*, Amazon, https://www.amazon.com/JOEJOY-Interactive-Automatic-Irregular-Rechargeable/dp/B08D6C6V1W/ref=sr_1_3?crid=2DQLOQVVVZ1T2&dchild=1&keywords=joejoy+cat+toys&sprefix=joejoy+cat%2Caps%2C286&sr=8-3
4. “ARM Architecture.” *Wikipedia*, Wikimedia Foundation, 13 Nov. 2020, https://en.wikipedia.org/wiki/ARM_architecture
5. Chen, Crystal, et al. “Risc vs Cisc.” *RISC vs. CISC*, Crystal Chen, Greg Novick and Kirk Shimano, 2000, https://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/risccisc/
6. Simons, Hadlee, et al. “Arm vs x86: Instruction Sets, Architecture, and All Key Differences Explained.” *Android Authority*, 11 Nov. 2020, https://www.androidauthority.com/arm-vs-x86-key-differences-explained-568718/
7. “Microcontroller.” *Wikipedia*, Wikimedia Foundation, 12 Nov. 2020, https://en.wikipedia.org/wiki/Microcontroller
8. Rouse, Margaret. “What Is a Microcontroller and How Does It Work?” *IoT Agenda*, TechTarget, 7 Nov. 2019, https://internetofthingsagenda.techtarget.com/definition/microcontroller
9. Staff, Maker.io. “Raspberry Pi Comparison: Which Pi Is Right for My Application?” *Digi*, Maker.io, 21 Feb. 2019, https://www.digikey.com/en/maker/blogs/2018/how-to-pick-the-right-raspberry-pi
10. “Raspberry Pi.” *Wikipedia*, Wikimedia Foundation, 10 Nov. 2020, https://en.wikipedia.org/wiki/Raspberry_Pi
11. “Raspberry Pi Models.” *The Pi Hut*, 21 Feb. 2017, https://thepihut.com/blogs/raspberry-pi-roundup/raspberry-pi-comparison-table
12. Raspberry Pi. “Buy a Raspberry Pi Zero.” *Raspberry Pi*, https://www.raspberrypi.org/products/raspberry-pi-zero/?resellerType=home
13. “Arduino.” *Wikipedia*, Wikimedia Foundation, 27 Oct. 2020, https://en.wikipedia.org/wiki/Arduino

14. Brennan, Eugene. “What Is a Microcontroller? - Programming an Arduino Board - TurboFuture - Technology.” *TurboFuture*, 27 Jan. 2014, https://turbofuture.com/misc/What-is-an-Arduino-Programming-Microcontrollers
15. “Top 10 Popular Microcontrollers Among Makers.” *Electronics*, 4 Aug. 2020, https://www.electronics-lab.com/top-10-popular-microcontrollers-among-makers/
16. “Qi (Standard).” *Wikipedia*, Wikimedia Foundation, 9 Nov. 2020, https://en.wikipedia.org/wiki/Qi_(standard)
17. “Rezence (Wireless Charging Standard).” *Wikipedia*, Wikimedia Foundation, 14 July 2018, https://en.wikipedia.org/wiki/Rezence_(wireless_charging_standard)
18. “Resonant Inductive Coupling.” *Wikipedia*, Wikimedia Foundation, 13 Sept. 2020, https://en.wikipedia.org/wiki/Resonant_inductive_coupling
19. “IEEE 802.11.” *Wikipedia*, Wikimedia Foundation, 29 Oct. 2020, https://en.wikipedia.org/wiki/IEEE_802.11
20. “Internet Protocol Suite.” *Wikipedia*, Wikimedia Foundation, 7 Nov. 2020, https://en.wikipedia.org/wiki/Internet_protocol_suite
21. “HTML.” *Wikipedia*, Wikimedia Foundation, 29 Oct. 2020, https://en.wikipedia.org/wiki/HTML
22. “Common Gateway Interface.” *Wikipedia*, Wikimedia Foundation, 9 Nov. 2020, https://en.wikipedia.org/wiki/Common_Gateway_Interface
23. “Python (Programming Language).” *Wikipedia*, Wikimedia Foundation, 12 Nov. 2020, https://en.wikipedia.org/wiki/Python_(programming_language)
24. “C (Programming Language).” *Wikipedia*, Wikimedia Foundation, 13 Nov. 2020, https://en.wikipedia.org/wiki/C_(programming_language)
25. “ANSI C.” *Wikipedia*, Wikimedia Foundation, 3 Nov. 2020, https://en.wikipedia.org/wiki/ANSI_C
26. “Java (Programming Language).” *Wikipedia*, Wikimedia Foundation, 27 Oct. 2020, https://en.wikipedia.org/wiki/Java_(programming_language)#Implementations
27. “MEMS THIN FILM PIEZOELECTRIC ACOUSTIC TRANSDUCER FOR COCHLEAR IMPLANT APPLICATIONS” - Scientific Figure on ResearchGate. https://www.researchgate.net/figure/MEMS-capacitive-accelerometer-prototype-architecture-28_fig7_328902293 [accessed 13 Nov, 2020]

28. “16G88-220P.1 Portescap: Motors, Solenoids, Driver Boards/Modules.” *DigiKey*, www.digikey.com/en/products/detail/portescap/16G88-220P-1/5232858?s=N4IgTCBcDaIIwDYDiAOFBaMYAMAFAdHAAQgC6AvkA.
29. “ADXL326BCPZ Analog Devices Inc.: Sensors, Transducers.” *DigiKey*, www.digikey.com/en/products/detail/analog-devices-inc/ADXL326BCPZ/2042674.
30. B. Bais, BY. Majlis, et al. “Optimization of MEMS Capacitive Accelerometer.” *Microsystem Technologies*, Springer-Verlag, 1 Jan. 1970, link.springer.com/article/10.1007/s00542-013-1741-z.
31. “BGN1800-4DWP-A800EC BatteryGuy: Battery Products: DigiKey Marketplace.” *BGN1800-4DWP-A800EC BatteryGuy | Battery Products | DigiKey Marketplace*, www.digikey.com/en/products/detail/batteryguy/BGN1800-4DWP-A800EC/12702761.
32. “BGN800-5FWP-A800EC BatteryGuy: Battery Products: DigiKey Marketplace.” *BGN800-5FWP-A800EC BatteryGuy | Battery Products | DigiKey Marketplace*, www.digikey.com/en/products/detail/batteryguy/BGN800-5FWP-A800EC/12702768?s=N4IgTCBcDaIEIHEByAOADGgtAVgGIHUAFTAQXTQFEBhE

AXQF8g COM0805 Pimoroni Ltd: Motors, Solenoids, Driver Boards/Modules." *DigiKey*, www.digikey.com/en/products/detail/pimoroni-ltd/COM0805/6873671?s=N4IgTCBcDaIMIHkCyAGAHCgrCAugXyA.

33. "EPS-25-5 MEAN WELL USA Inc.: Power Supplies - External/Internal (Off-Board)." *DigiKey*, www.digikey.com/en/products/detail/mean-well-usa-inc/EPS-25-5/7703160.
34. "LS25-12 TDK-Lambda Americas Inc.: Power Supplies - External/Internal (Off-Board)." *DigiKey*, www.digikey.com/en/products/detail/tdk-lambda-americas-inc/LS25-12/2057033.
35. Ltd, PRC Rail Consulting. "The Railway Technical Website." *Electric Traction Control | The Railway Technical Website | PRC Rail Consulting Ltd*, www.railway-technical.com/trains/rolling-stock-index-l/train-equipment/electric-traction-control-d.html.
36. "MMA8451QR1 NXP USA Inc.: Sensors, Transducers." *DigiKey*, www.digikey.com/en/products/detail/nxp-usa-inc/MMA8451QR1/2508916?s=N4IgTCBcDaILJwIIA4AsBWAjARQEqYAIQBdAXyA.
37. "MMA8652FCR1 NXP USA Inc.: Sensors, Transducers." *DigiKey*, www.digikey.com/en/products/detail/nxp-usa-inc/MMA8652FCR1/3831432?s=N4IgTCBcDaILJwIIA4BsBWMAxAwgJQEYQBdAXyA.
38. "PIS-0930 Pi Supply: Motors, Solenoids, Driver Boards/Modules." *DigiKey*, www.digikey.com/en/products/detail/pi-supply/PIS-0930/9522084?s=N4IgTCBcDaIAoEkDKBaADATgMxpAXQF8g.
39. "Raw Materials Company Inc." *What's Inside A Battery*, www.rawmaterials.com/page/education/battery-inside/.
40. "SE18K1ETY NMB Technologies Corporation: Motors, Solenoids, Driver Boards/Modules." *DigiKey*, www.digikey.com/en/products/detail/nmb-technologies-corporation/SE18K1ETY/6021449?s=N4IgTCBcDaIMoFECMAOA0khAVAmiAugL5A.
41. "Storage Battery." *Encyclopædia Britannica*, Encyclopædia Britannica, Inc., www.britannica.com/technology/storage-battery.
42. "STSPIN250 STMicroelectronics: Integrated Circuits (ICs)." *DigiKey*, www.digikey.com/en/products/detail/stmicroelectronics/STSPIN250/6579060.
43. "Switch Mode Power Supply Basics and Switching Regulators." *Basic Electronics Tutorials*, 2 Mar. 2018, www.electronics-tutorials.ws/power/switch-mode-power-supply.html.
44. "Voltage Regulator." *Wikipedia*, Wikimedia Foundation, 12 Nov. 2020, en.wikipedia.org/wiki/Voltage_regulator.
45. "Voltage Regulator: Working Principle & Circuit Diagram: Voltage Regulator in Power Supply." *Electrical Academia*, 9 Oct. 2018, electricalacademia.com/electric-power/voltage-regulator-working-principle-circuit-diagram-voltage-regulator-power-supply/.
46. *What Is an Adjustable Voltage Regulator?*, www.learningaboutelectronics.com/Articles/What-is-an-adjustable-voltage-regulator.
47. Dave, Murali. "DIfferential Drive Mechanism - Mlees-Robotronics." *Google Sites*, 2012, sites.google.com/site/mleesrobotronics/home/robotics/differential-drive-mechanism.

48. wiringpi.com/reference/i2c-library/.
49. “Motion Guide.” *Motion*, motion-project.github.io/3.4.1/motion_guide.html.
50. “Configuration.” *Motion*, motion-project.github.io/motion_config.html.
51. “HTML <iframe> Tag.” <i>HTML Iframe Tag</i>, www.w3schools.com/tags/tag_iframe.ASP. <div></div></body></iframe>
52. Brownlee, Jason. *A Gentle Introduction to Object Recognition With Deep Learning*. 5 July 2019, machinelearningmastery.com/object-recognition-with-deep-learning/.
53. Martin McMerkin 12-17-2017 at 20:42, et al. *Using a Raspberry Pi Distance Sensor (Ultrasonic Sensor HC-SR04)*. tutorials-raspberrypi.com/raspberry-pi-ultrasonic-sensor-hc-sr04/.
54. 0:56, Jamil 3-12-2018 at, et al. *Measuring Rotation and Acceleration with the Raspberry Pi*. tutorials-raspberrypi.com/measuring-rotation-and-acceleration-raspberry-pi/.
55. *HC-SR04 Ultrasonic Range Sensor on the Raspberry Pi*. 22 Oct. 2013, thepihut.com/blogs/raspberry-pi-tutorials/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi.
56. 29, October, et al. “ICMP Reverse Shell.” *Infosec Resources*, 14 Oct. 2020, resources.infosecinstitute.com/topic/icmp-reverse-shell/.
57. “NGINX Docs: NGINX Reverse Proxy.” *NGINX Documentation*, docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/.
58. Wikipedia contributors. "Daemon (computing)." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 7 Nov. 2020. Web. 21 Nov. 2020.
59. Wikipedia contributors. "Motion (surveillance software)." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 8 Sep. 2020. Web. 18 Nov. 2020.
60. Wikipedia contributors. "Deep learning." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 18 Nov. 2020. Web. 18 Nov. 2020.
61. Wikipedia contributors. "Service set (802.11 network)." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 3 Nov. 2020. Web. 20 Nov. 2020.
62. Wikipedia contributors. "Default gateway." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 31 Oct. 2020. Web. 20 Nov. 2020.
63. Wikipedia contributors. "WHOIS." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 21 Nov. 2020. Web. 21 Nov. 2020.
64. Wikipedia contributors. "Unix domain socket." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 20 Oct. 2020. Web. 22 Nov. 2020.

65. Raspberry Pi Foundation. “Raspberry Pi Documentation.” *Raspberry Pi*, www.raspberrypi.org/documentation. Accessed 6 Nov. 2020.
66. Raspberry Pi Foundation. “FAQs - Raspberry Pi Documentation.” *Raspberry Pi*, https://www.raspberrypi.org/documentation/faqs/. Accessed 6 Nov. 2020.
67. Hübschmann, Ida. “How to Choose the Best IoT WiFi Module.” *Nabto*, 23 Oct. 2020, www.nabto.com/how-to-choose-best-wifi-module-for-iot. Accessed 6 Nov. 2020.
68. “ESP8266 - NURDspace.” *NURDspace*, nurdspace.nl/ESP8266. Accessed 6 Nov. 2020.
69. *The Internet of Things with ESP32*. esp32.net. Accessed 6 Nov. 2020.
70. “ESP32 Series Datasheet.” *Espressif*, www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. Accessed 6 Nov. 2020.

71. Shawn. “Arduino Wifi Modules: Which One to Use?” *Latest Open Tech from Seeed Studio*, 13 Nov. 2019, www.seeedstudio.com/blog/2019/11/05/arduino-wifi-modules-which-one-to-use. Accessed 6 Nov. 2020.
72. Piltch, Avram. “Which Raspberry Pi Should I Buy?” *Tom’s Hardware*, 29 May 2020, www.tomshardware.com/how-to/raspberry-pi-buying-guide.
73. “ESP8285 Datasheet.” *Espressif*, www.espressif.com/sites/default/files/documentation/0a-esp8285_datasheet_en.pdf. Accessed 6 Nov. 2020.
74. “OV5647 Datasheet.” *SparkFun Electronics*, cdn.sparkfun.com/datasheets/Dev/RaspberryPi/ov5647_full.pdf. Accessed 6 Nov. 2020.
75. “Products.” *Raspberry Pi*, www.raspberrypi.org/products. Accessed 6 Nov. 2020.
76. “WiFi Module - ESP8266 (4MB Flash).” *SparkFun Electronics*, www.sparkfun.com/products/17146. Accessed 6 Nov. 2020.
77. “Aluminum 2WD TT Robot/Motor Chassis - Purple.” *Cytron Technologies*, www.cytron.io/p-aluminum-2wd-tt-robot-motor-chassis-purple. Accessed 6 Nov. 2020.
78. “HEXBUG Mouse Robotic Cat Toy, Color Varies.” *Chewy*, www.chewy.com/hexbug-mouse-robotic-cat-toy-color/dp/151629. Accessed 6 Nov. 2020.
79. “What Are Standards?” *IRENA*, www.irena.org/inspire/Standards/What-are-Standards. Accessed 6 Nov. 2020.
80. “Library Guides: Standards: Types of Standards.” *University of Washington Libraries*, guides.lib.uw.edu/c.php?g=341616&p=2299355. Accessed 6 Nov. 2020.
81. “Searching for Standards Information via the Internet.” *ANSI*, share.ansi.org/Shared%20Documents/Education%20and%20Training/University%20Outreach%20Program/NSSN%20-%20ASEE%20Regional%20Meeting%20-%2010-04.pdf. Accessed 6 Nov. 2020.
82. “Vision and Mission | IEC.” *IEC*, www.iec.ch/what-we-do/vision-mission?ref=menu. Accessed 6 Nov. 2020.
83. “International Organization for Standardization.” *ISO*, www.iso.org/home.html. Accessed 6 Nov. 2020.
84. “IEEE - The World’s Largest Technical Professional Organization Dedicated to Advancing Technology for the Benefit of Humanity.” *IEEE*, www.ieee.org. Accessed 6 Nov. 2020.
85. “IEEE SA - The IEEE Standards Association - Home.” *IEEE SA - The IEEE Standards Association*, standards.ieee.org. Accessed 6 Nov. 2020.
86. “IPC Standards.” *IPC International, Inc*, www.ipc.org/ipc-standards. Accessed 6 Nov. 2020.
87. “ITU.” *ITU: Committed to Connecting the World*, www.itu.int/en/Pages/default.aspx. Accessed 6 Nov. 2020.
88. Wikipedia contributors. “Prototype.” *Wikipedia*, 4 Dec. 2020, en.wikipedia.org/wiki/Prototype#Basic_prototype_categories. Accessed 6 Nov. 2020.
89. “TESTING AND EVALUATING A PROTOTYPE - WHY?” *ENGINEERING - DESIGN AND TECHNOLOGY*, technologystudent.com/despro_flsh/evalintegr1.html. Accessed 20 Nov. 2020.
90. Sarah. “Step by Step: Testing Your Prototype | Blog.” *Notes from the Field | User Research, User Feedback, UX & Design*, 19 Oct. 2020, qualaroo.com/blog/step-by-step-testing-your-prototype.

91. Vijay. “Web Application Testing Complete Guide (How To Test A Website).” *Software Testing Help*, 1 Dec. 2020, www.softwaretestinghelp.com/web-application-testing.
92. “Raspberry Pi Camera Module - Raspberry Pi Documentation.” *Raspberry Pi*, www.raspberrypi.org/documentation/raspbian/applications/camera.md. Accessed 20 Nov. 2020.
93. Agarwal, Tarun. “Obstacle Avoidance Robotic Vehicle: Working Principle and Its Applications.” *ElProCus - Electronic Projects for Engineering Students*, 1 July 2020, www.elprocus.com/obstacle-avoidance-robotic-vehicle.
94. Dragos-George, Calin. “How To Build an Obstacle Avoiding Robot.” *Into Robotics*, 3 May 2020, www.intorobotics.com/obstacle-avoidance-robot.
95. “RPi-Cam-Web-Interface - ELinux.Org.” *Embedded Linux Wiki*, elinux.org/RPi-Cam-Web-Interface. Accessed 20 Nov. 2020.
96. Popov, Dmitri. “Using the RPi Cam Web Interface.” *Raspberry Pi Geek*, www.raspberry-pi-geek.com/Archive/2016/15/Using-the-RPi-Cam-Web-Interface. Accessed 20 Nov. 2020.
97. Fictiv. “Hardware Testing 101.” *Fictiv*, 25 Sept. 2020, www.fictiv.com/articles/hardware-testing-101.
98. “Hardware Product Testing Strategy – for Complex or Mission-Critical Parts & Systems.” *View Point Systems*, www.viewpointusa.com/TM/wp/hardware-product-testing-strategy-complex-mission-critical-parts. Accessed 20 Nov. 2020.
99. “Product Testing Methods – for Industrial Hardware Products.” *View Point Systems*, www.viewpointusa.com/TM/wp/product-testing-methods-industrial-hardware-products. Accessed 20 Nov. 2020.
100. “How to Test a Voltage Regulator.” *Learning About Electronics*, www.learningaboutelectronics.com/Articles/How-to-test-a-voltage-regulator. Accessed 20 Nov. 2020.
101. “Rechargeable Batteries Guide | NiMH | Li-Ion | NiCd.” *MICROBATTERY*, www.microbattery.com/rechargeable-batteries-guide. Accessed 20 Nov. 2020.
102. “Rechargeable Batteries — Compared and Explained in Detail.” *Michael Bluejay’s Battery Guide*, michaelbluejay.com/batteries/rechargeable.html. Accessed 20 Nov. 2020.
103. Rungta, Krishna. “What Is Software Testing? Definition, Basics & Types.” *Guru99*, 8 Oct. 2020, www.guru99.com/software-testing-introduction-importance.html#4.
104. Rungta, Krishna. “What Is Interface Testing? Types & Example.” *Guru99*, 4 Nov. 2020, www.guru99.com/interface-testing.html.
105. Hanrahan, Robert. “Testing a Power Supply (Part 1).” *EDN*, 26 Mar. 2020, www.edn.com/testing-a-power-supply-part-1.
106. “5 Must-Do Prototyping Tests for IoT Products.” *Indiegogo Education Center*, entrepreneur.indiegogo.com/education/article/5-must-do-prototyping-tests-for-iot-products. Accessed 30 Nov. 2020.
107. “Can You Test a Device’s Bluetooth Strength?” *Phonecheck*, www.phonecheck.com/blog/can-you-test-a-devices-bluetooth-strength. Accessed 30 Nov. 2020.
108. “ESP8266 WiFi Module Testing | ESP-05 | Arduino Due.” *Starting Electronics*, startingelectronics.org/articles/ESP8266-testing. Accessed 30 Nov. 2020.

109. Schweber, Bill. “Wireless Charging, Part 2: Implementation.” *Power Electronic Tips*, WTWH Media LLC, 21 May 2018, www.powerelectronictips.com/wireless-charging-part-2-implementation-faq/.

110. Krejsa, Jiri & Vechet, Stanislav. (2008). Dealing with Sensor Errors in Scan Matching for Simultaneous Localization and Mapping.

111. Andreasson, Henrik. (2008). Local visual feature based localisation and mapping by mobile robots.

112. *Raspberry Pi 4 Information*, PiShop, 2020, www.pishop.us/raspberry-pi-4-information/.

113. “What's New With BLE5? And How Does It Compare To BLE4?” *Vinnter AB*, Vinnter, 8 May 2020, vinnter.se/whats-new-with-ble5-and-how-does-it-compare-to-ble4/.

114. “Wireless Network Diagram Template.” *Lucidchart*, www.lucidchart.com/pages/templates/network-diagram/wireless-network-diagram-template.

115. Feldman, Sarah. “Infographic: The Most Popular Programming Languages.” *Statista Infographics*, 8 Jan. 2019, www.statista.com/chart/16567/popular-programming-languages/.

116. Newth, Jorunn D. “Which Building Materials Can Block Wi-Fi Signals?” *Eye Networks*, 8 Aug. 2020, eyenetworks.no/en/wifi-signal-loss-by-material/.

117. “Engineering Ethics Case Studies Oz Assignments.” *Oz Assignments Australia's Proofreading and Editing Services from Fulltime Experts*, Anaya Learning PTE LTD, www.ozassignments.com/solution/engineering-ethics-case-studies-oz-assignments.

118. “Health Navigation: Sleep.” *Medcor*, Medcor, www.medcor.com/sleep/.

119. Kruger, Bertus. “Raspberry Pi Zero Footprint And Dimensions.” *Electronics Lab*, 27 Nov. 2015, www.electronics-lab.com/raspberry-pi-zero-footprint-and-dimensions/.

120. “Pi Network Monitoring.” *Fun Tech Projects*, 17 Apr. 2020, funprojects.blog/2018/06/24/pi-network-monitoring/.

121. Agrawal, Raunak. (2018). Design and Implementation of Smart Vehicle Theft Detection using Raspberry PI And IOT for Real Time Applications. International Journal for Research in Applied Science and Engineering Technology. 6. 712-715. 10.22214/ijraset.2018.6111.

122. Vieira, Sandra & Pinaya, Walter & Mechelli, Andrea. (2017). Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: Methods and applications. Neuroscience & Biobehavioral Reviews. 74. 10.1016/j.neubiorev.2017.01.002.

123. Santos, Rui. “Complete Guide for Ultrasonic Sensor HC-SR04 with Arduino.” *Random Nerd Tutorials*, 2 Apr. 2019, randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/.

124. Mavridou, Efthimia & Vrochidou, Eleni & Papakostas, George & Pachidis, Theodore & Kaburlasos, Vassilis. (2019). Machine Vision Systems in Precision Agriculture for Crop Farming. Journal of Imaging. 5. 89. 10.3390/jimaging5120089.

125. NXP, “MMA8652FC, 3-Axis, 12-bit, Digital Accelerometer,” MMA8652FC datasheet, Oct. 2013[Revised Oct. 2015].

126. Texas Instruments, “bq5105xB High-Efficiency Qi v1.2-Compliant Wireless Power Receiver and Battery Charger,” bq5105xB datasheet, Jul. 2012 [Revised Jun. 2017].