Mr. Painter-Robot Painting Arm

Senior Design Project

120 Page Documentation

Group 14:

Alan Azargushasb	Electrical Engineering
Alonso Ninalaya	Computer Engineering
Chedlyne Valmyr	Computer Engineering
Trudy Ani-Asamani	Electrical Engineering

Mr. Painter Table of Contents

Table of Tables

1. Executive Summery

Life is meant to be beautiful; aesthetics is divine. What is art but the capturing of a moment, the expression of a thought? Good art transcends time. If it is beautiful then they will come, and hand painted art always in fashion, but what about the cost? An artist needs to eat. Luckily in this new digital world art need not artists. The time has come for the automation of art itself.

Meet Mr. Painter solves this problem of expensive hand drawn/painted art by painting the artwork itself. Mr. Painter is a robot painting arm that paints pictures with precision and ease. The robot arm is made of metal ensuring sturdiness and reliability. Images can be uploaded to Mr. Painter or Mr. Painter can see things through its camera and paint what is in view before it. The camera gives feedback to Mr. Painter so that Mr. Painter can see how he is doing and correct his mistakes along the way.

What makes Mr. Painter different from, let us say, a printer? After all a printer can print any image, whether it be uploaded or pulled live from a camera. The difference is in the stroke. Ink from a printer is flat and uniform. You can obviously tell it came from a printer. Its perfection is its fault. It lacks texture. It is too digital, too perfect, in a word sterile. What is the difference between a Rembrandt and a printed picture of a Rembrandt? The strokes! The paint on the canvas gives it life, gives it meaning. By having Mr. Painter paint on canvas, we reintroduce the texture back to the canvas. The advantages of using a robot arm are listed above, it simulates the human element of painting.

Some disadvantage is the that the robot arm can only lift so much. The further the brush gets away from the base of the arm the weaker it gets. Since the robot uses servos and lots of them, they robot arm makes a cacophonous sound when it moves around, it sound awful. The robot arm is controlled by pulse width modulation. By changing the length of the pulse, we tell the servo how far from the starting position we want the gears to rotate. Through the kinematics the robot arms joints move together in unison to produce motion conductive to painting

2 Project Description

Introduction

Most industries around the world have greatly benefitted from the assimilation of technology into different various businesses. For example, the agricultural industry is changing across the world thanks to technology. Traceability, sustainability and quality of goods have been improved due to engineering. With the help of technology, farmers can now make more use of their lands and introduce better water management techniques to increase their produce and conserve natural resources. The same is true in the health sector, technology has aided in data management issues as well as advanced monitoring devices that help patients and better the healthcare system. The world of business and finance has also had a great relationship with the technological industry. Bank activities can now be monitored by the second and small businesses finally have practical tools to exponential growth and wider market reach. Unfortunately, in the world of Art, there are still some concerns on the impact of technology in this field. While people argue that technology has brought distractions to live performances of art, others say that technology has enabled remote access to art galleries and museums enabling millions of people to experience said art.

There has also been the question of the credibility of machine-generated art. This is due to the fact most of the historic paintings we have come to know and love, have often had a lot to do with the artists that created them. For example, any painting by Michelangelo would easily have great reputation and a huge price tag. Whereas there are currently no globally acknowledged machine generated art pieces. The creation of more machine generated artworks would affect the future of the Art world. This is because museums and art galleries across the world financially thrive on the annual visits and contributions of individuals toward the preservation of various artworks of renown artistes. The question then becomes would people financially support the creation and preservation of machine generated artworks. Thankfully, the robot painting arm has no plans of displacing hardworking artists, or bankrupting museums rather, the robot painting arm serves as an aid to art enthusiasts who may be medically incapacitated to create art.

While there are innumerable articles and Ted-talks on the benefits of art and its impact on individuals, our robot painting arm does not seek to replace the entire activity of painting or creating but rather seeks to recreate the same sensations of comfort, tranquility and relaxation that art brings to its users. Carpal tunnel and arthritis are reasons that multiple artists are not able to continue to pursue their love for art. With the internet flowing with pictures, we don’t get to enjoy the peacefulness of art drawings on a canvas. In addition, the steps for the achievement of drawing human figures is almost impossible for most art enthusiasts. While some are able to naturally conjure the image of human beings in different positions, to the rest of us it would take and art degree and lots of practice. This problem is however fixed with the robot painting arm. Although the opposing foundations of art and science have kept these two worlds apart; science focusing on facts and figures and art dwelling on human expression and creativity, our project seeks to bridge the long-term divide between the two powerful worlds.

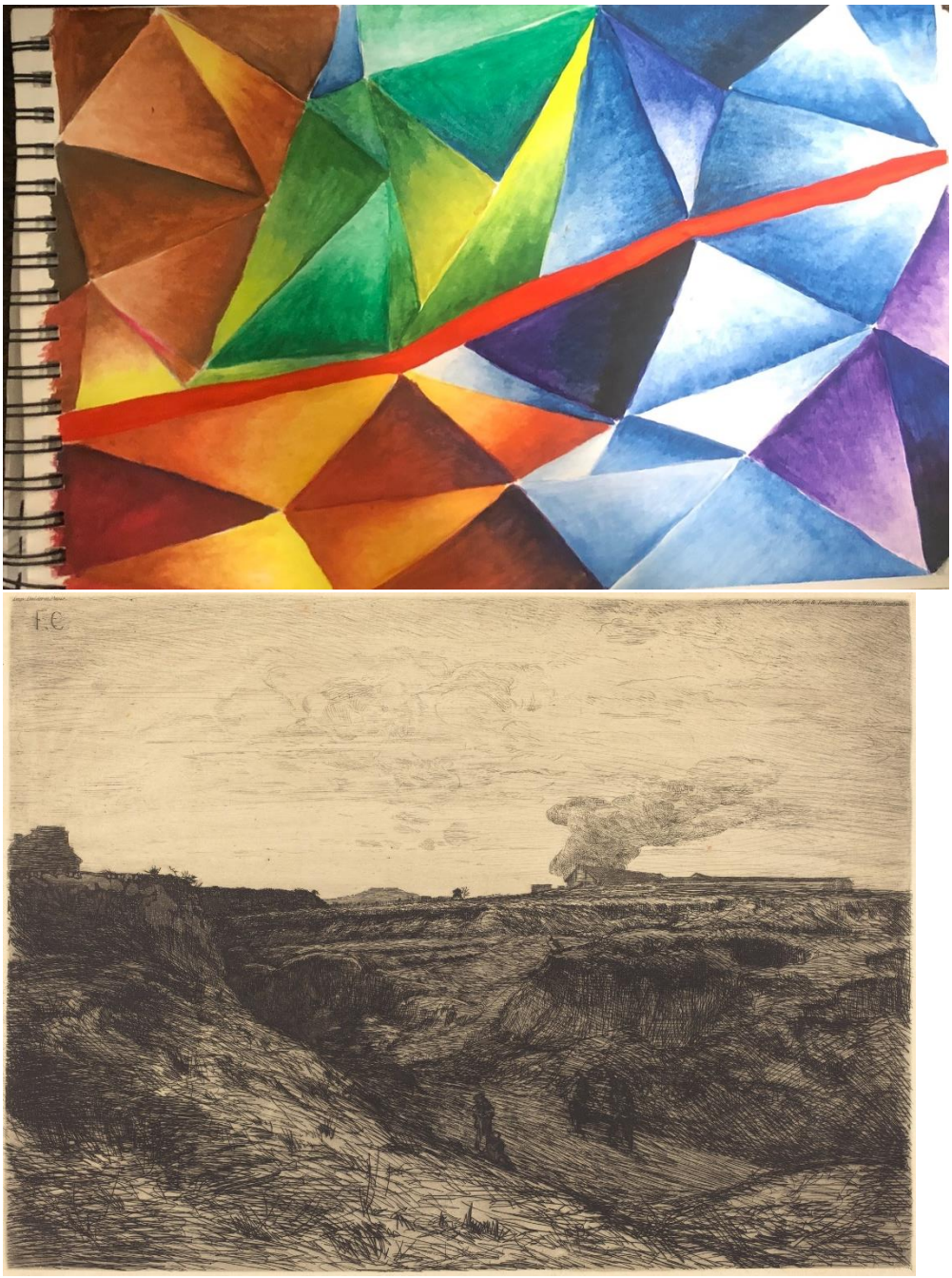


Figure 2.1 François-Nicolas Chifflart
Quarry near Montmartre
1865,in the public domain taken from the National gallery of the arts

Figure 2.2 Landscape by Claude Monet
The Japanese Footbridge
1899, in the public domain taken from the National Gallary of the arts

Designing a robot arm that paints opens the door to solving the problems I mentioned above. And what better place to start then with the painting of landscapes. Art often imitates life. Painting landscapes is an appropriate and sufficient way to demonstrate the power of Mr. Painter and to solve the lack of culture in the modern era. The level of

difficulty with painting landscapes, such as color patterns, precision, and the imagination required presents a clear obstacle to the novice painter, but not so with Mr. Painter, not so indeed. We hope that one day this robot arm will be talented enough to be able to paste celebrity signatures on merchandise, sketch the human face with accuracy and grace, and allow artists damned by the inevitability of old age, to still perform even with failing health (arthritis), and gives us more art to enjoy.

In order to design a robot-arm that paints landscapes we will first purchase a robotic arm from amazon. Buying the robotic device online is easier than having to 3D print the parts which can be very expensive. Also buying the robot arm online allowed us to purchase the motors and screws at the same time making assembling the pieces together less complicated. Next, we researched looking for the best camera that will have sensors knowing when it was close to the canvas, where to paint, and it will have color detection. Which is all required for the robotic arm. Next, we will design the base for the arm so it can be properly placed so that it is sturdy. Once the robot arm is set up, we will then purchase water, paint, a paint brush, and a canvas. There are more parts that are required in order to get a functioning robot arm such as the servos, power supplies, PCB board, etc. Once the device is put together, we can then see how we will be able to program the boards and camera so that it can do the correct movements and successfully paint landscapes on a canvas.

Project definition

The purpose of this project is to design a robotic arm that paints and is able to complete strenuous actions such as rotation, moving vertically and horizontally. Also, it should be able to compare the input and output, how it is doing, and adjust accordingly. Lastly it must be able to detect colors.

2.1 Project motivation

For this project we will demonstrate using what we've learned throughout our time here at the University of Central Florida by designing a robotic arm that paints. The motivation behind this idea came about because we wanted to design a robotic arm that would cook pancakes. This was our idea because we thought it would be cost effective and be faster than us cooking pancakes. Because there were a lot of gray areas such as, flipping the pancake, taking the temperature, and consistency within each batch of pancake mix. We decided to explore other ideas. We then decided on a robotic arm that would cook burgers but finding a robotic arm that would support the heavy lifting without 3D printing was not possible. and then finally we decided on a robotic arm that will paint. We believe once we successfully create our device that paints, we will be able to use the knowledge from this experiment and apply it to building a robotic arm that can cook burgers and pancakes.

We are motivated to implement our idea for this project because it is a learning experience that we can apply in our future careers. With this project it will teach us to work in groups, form ideas, and then proceed to successfully develop our idea into something big. With

this experience it can be very beneficial, and we can build this onto our resumes for future jobs.

2.2 Goals

Setting goals for any project or mission makes the demands and objectives of the project clearer as well as provides actual targets to be met. Having goals is a basis for clearer communication among team members and a rather smooth transition through the team's milestones. The process of setting goals also helps one identify what is important. For example, team members of this group understand and realize that having the color of the robot arm red is not an immediate goal objective. In addition, having a robot arm that can clean up the paint pigment of the brush once it's done painting is not an immediate goal or objective. This is because the primary focus of the project is to have a robot arm that can execute the task of painting effortlessly. The main objective is to have a full working robot painting arm by the end of the Second senior design semester. The milestones in section 9 of this report outline show we plan to achieve this. The goal is no matter the image uploaded to Mr. Painter, the robot painting arm should be able to paint without any issues and in order to reach this main goal, there are a number of sub goals that have been set in order to achieve this.

1. Finding the right robotic arm that is able to lift a paint brush without collapsing, has clamps so that it can grip the paintbrush, and the servos are able to fit.
2. Finding the right camera. This is important because the camera will be the robot arms eyes. Without the camera it won't be able to see what its painting nor it won't be able to distinguish colors, as well as it won't be able to compare what it is drawing with the input. Once the right camera is found we can then program it to do these things.
3. Applying the right power supply. We don't want to short our circuit board because then this will cause delays. We also want to make sure we apply the right amount so that it can function properly.
4. Understanding the mathematics involved with getting the robotic arm to work correctly.
5. Limiting our mistakes by doing in depth research about robot arms.
6. perfecting the code of the robot arm such that each servo aides in executing the task of painting the canvas
7. Having the right type of paint brush attachment that enables the average brush to be used as a tool to execute the task of painting
8. Making the robot arm pencil and marker friendly, enabling users of the robot painting arm change tools if they so desire.

These are some of goals that will guide us to our final goal of getting this robot arm to successfully operate. We understand that there might be some setbacks to achieving these goals, however we are willing to learn from said setbacks and use any failed

experience as a learning opportunity to better our design. Again, setting milestones and having effective communication would enable the achievement of these goals.

2.3 Objective

The purpose of this project is to design a team-based project that will allow us to implement what we've learned throughout our career here at UCF. This project will consist of both electrical and software components. Our ideas will merge into one so that we can present a fully functioning robotic arm that will paint landscape portraits onto a canvas.

Hardware: The hardware will consist of:

- **Robotic arm**
- **Printed circuit board**
- **Base**
- **Servos**
- **Servo control board**
- **Power equipment**
- **Camera**

The hardware will allow the robotic arm to be put together so that eventually it will run efficiently and be able to function correctly. Other parts involved for this robotic arm are small parts like the canvas, paint, and paint brushes. We aren't using a specific brand for this and we are not putting these parts together ourselves, these will be store bought for our robotic arm.

Software: The software will consist of data processing, data collection, input, processing, and output. An example of this is the camera on the robot arm which will compare the inputted image to what is being painted on the canvas. Also programming the servos so that they can all complete the movements required to perform actions.

Control: In order to keep the robotic arm together the control will be the printed circuit board. With the board it will communicate with the other instruments ensuring that an action was completed, or another movement is needed in order to complete a certain action. It will connect the camera to the robotic arm that will be moving with instruction.

Communications: There will not be a wireless communication system device put in place for this robotic arm. With user input the robotic arm will be notified when to start reading and writing. User input will display input that will then be compared using the camera equipment to the output. This is a simpler way to do this without adding any complications or risk to the robotic arm.

Power supply: The power supply serves as the foundation for the entire design. Without a power supply, the robot painting arm will not be functional. In addition, in order to send.

Signals to the micro controller, in order to power the servos and send signals to these servos, the power supply is pivotal. Also, to have a fully functional camera responsible for capturing the input or images to be duplicated, the power supply needs to be present.

2.4 Requirement Specification

• Six Servo Motors - MG996R

Weight: 55 g

Dimension: 40.7 x 19.7 x 42.9 mm approx.

Stall torque: 9.4 kgf.cm (4.8 V), 11 kgf.cm (6 V)

bearing design 55 oC

Operating speed: 0.17 s/60o (4.8 V), 0.14 s/60o (6 V) Operating voltage: 4.8 V a 7.2 V

Running Current 500 mA –

Stall Current 2.5 A (6V)

Dead band width: 5 µs

Stable and shock proof double ball bearing design

Temperature range: 0 oC – 4.8 V a 7.2 V – 900 mA (6V) double ball

• 5V 2A DC Power Supply

Input: 100 - 240 V AC, 50/60Hz

Output Voltage: 12 V DC, 2A

Max Wattage: 24W

▪ Power cord length: 3.9ft/1.2m

Dimensions: 3.86inch/98mm x 1.77inch/45mm x 2.36inch/60mm ▪ Diameter of connector: interior 2.1mm; exterior: 5.5mm

• HC –5 Bluetooth Module (optional) ▪

Hardware features:

-80dBm sensitivity

Up to +4dBm RF transmit power

Low power 1.8V Operation, 1.8 to 3.6V I/O o PIO control

UART interface with a programmable baud rate o Integrated antenna

Edge connector

Software features:

Default Baud rate: 38400

Data bits:8, Stop bit:1, Parity: No parity

9600,19200,38400,57600,115200,230400,460800.

Given a rising pulse in PIO0, device will be disconnected.

Status instruction port PIO1: low-disconnected, high-connected; o PIO10 and PIO11 can be connected to red and blue led separately. When the master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.

Data control: has. Supported baud rate:

Auto-connect to the last device on power as default. Permit pairing device to connect as default. Auto-pairing PINCODE:"0000" as default.

Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection

• Arduino Board (MEGA 2560 REV3)

Arduino will be used for testing the servo motors only

Operating voltage: 5V

Input Voltage(recommended): 7V – 12V

Digital I/O: 54

Analog Input Pins: 16

DC Current per I/O pin: 40mA

DC Current for 3.3V Pin: 50mA

Flash Memory: 256 KB of which 8KB used by bootloader ▪ SRAM: 8KB

Clock Speed: 16 MHz

• Raspberry pi Camera

Open CV compatible

Still resolution: 8 megapixels

Supported OS: Windows 10, 8, 7, Linus, Mac

Weight: 3g

Sensor: Sony IMX219

Video modes: 1080p30, 720p60 and 640 × 480p60/90

Sensor Resolution: 3280 × 2464 pixels

Size: Around 25x24x9mm

C programming API: OpenMAX IL and others available
Sensor Image Area: 3.68 x 2.76mm (4.6 mm diagonal)
Full-frame SLR lens equivalent: 35mm

S/N ratio: 36dB

Dynamic range: 67dB @ 8x gain

Dark current: 16mV/sec @60 C

2.5 House of Quality

Strong positive ↑↓

Positive ↑

Strong negative ↓↓

Negative ↓

No Correlation

Marketing requirements \ Engineering Requirements	Consistency	Efficency	Cost	Install time	Dimensions
Durability	↑↑		↓↓		
Low power	↓↓	↑	↑↑		↓
Easy to use	↑↑			↑↑	
Cost	↓	↓			↓↓

Table A: Marketing requirements vs Engineering requirements

Marketing requirements \ Engineering Requirements	Consistency	Efficency	Cost	Install time	Dimensions
Consistency	X		↓↓		
Efficiency		X	↓↓		↓
Cost	↓↓	↓↓	X		
Install time	↓	↓		X	↓
Dimensions		↓	↓↓	↓	X

Table B: Engineering requirements vs Standard engineering requirements

The House of Quality shows the product development which is inspired by consumers needs verses the resources required to make the product successful and the correlation that this holds. As shown in table 4, we compare marketing requirements with engineering requirements, looking at the positive and negative correlations. For marketing requirements, we look at how the durability of the object, how long will it last, can it hold up without falling apart while being used, is it cheap. These are all important factors when deciding on the durability of the object. We look at the power of the device, does the device require a lot of power to run. Is the device easy to use this is important for user use because if it's too difficult it can have a negative impact? And lastly, costs. Is the device worth the cost, what is the cost to produce. When looking at the correlation of durability, costs, easy to use, low power we kept in mind what we as the consumer would want and how this would affect our likelihood to purchase this device.

For the engineering requirements on table 4 we looked at the efficiency, consistency, cost, install time, and the dimensions. Efficiency for the robot arm is, does the robot arm efficiently run, are there any hiccups, does it run smoothly. For consistency we look at are the movements consistent, does it require adjustments every time it turned on. For cost, we look at is it expensive to produce. For install time we are looking at how long it takes to install for it to turn on. What are the steps taken, how long does it take to adjust the paint brush? And then lastly, we look at the dimensions. We don't want it to be too big, but we also don't want it to be small to the point where it is too weak to lift a brush.

Consistency and durability have a strong positive correlation with each other because if it is durable then it will more likely be consistent while running. Consistency and low power

have a strong negative correlation with each other because we felt because the device uses low power it may not be consistent (in our tests with the first robot arm this turned out to be true). Easy to use and consistency have a strong positive with each other because the easier it is to use the more likely it is to run more consistently. Cost and consistency have a negative correlation because it is more likely to run consistently if it cost more money which is negative.

When comparing the engineering requirement efficiently with the marketing requirements there weren't a lot of correlations except for low power and cost. We thought low power vs efficiency had a positive correlation because we thought if the amount of power being used is not high then that would allow the robot it to run more efficiently (from our tests we learned this is nonsense, various machines are extremely efficient and use a lot of power such as transformers). Cost vs efficiency has a negative correlation because if it's cheap it probably won't be as efficient and if you want it to be more efficient it would require more spending.

The engineering requirement cost also didn't have a lot of correlation with the marketing requirements except for durability and low power. There is a strong negative between cost and durability. If for marketing purposes its more durable on the engineering side, it will be more expensive to produce. As for low power vs cost. The amount of power being used affects the cost. The more power used the more it will cost or produce. For our robot arm this is a positive correlation due to our device not requiring a lot of power, we estimate at this moment, Mr. Painter it would only about 75 watts.

The only strong correlation with install time is easy to use. The simpler it is to put the device together when it is time to use it, the easier it will be for the user to use. This is a strong positive correlation in our design because it is small enough to be moved from place to place, all that is required is placing the brush, water cup, canvas, and paint in a position that is accessible to the robot arm. There weren't other correlations for install time because this has no effect on the cost, how durable it is, and it does not influence the power used by the device.

The engineering requirement dimensions had a negative correlation with power. We saw that if the device is large it will most likely require more power to run. The more parts required the more energy the robot arm will need to run. Dimensions vs cost had a strong negative correlation because the bigger the device the more it will cost. Now looking at table 5 we did a correlation chart comparing the engineering requirements against each other. We didn't look at the correlation if it was for example, consistency vs consistency because such a comparison is nonsensical. For table 5 throughout the whole graph we saw that there was a majority negative correlation or no correlations between consistency, efficiency, install time, cost, and dimensions.

Consistency vs cost had a strong negative correlation because of course it will cost more for it to be more consistent while in use. There was a negative correlation between consistency and install time. We felt if it was consistent then the install time would be longer. For efficiency vs cost there is a strong negative correlation because we felt the

more efficient the device is the more it will cost. This idea is because it would require more expensive parts and more time put towards it for it be more efficient. There is a negative correlation between efficiency and install time, the more efficient it is the more time required to set it up. We also felt the bigger the device the less efficient it will be which is why there is a negative correlation for efficiency vs dimensions.

For cost vs consistency there is a strong negative correlation between them because again, the more it cost the more likely it is to be more consistent. For cost vs the marketing requirements the reasons behind the strong negative were repetitive. For example, the bigger and the more efficient it is the more it will cost. Any adjustments that require it to be faster, more efficient, or simpler results in more spending for better parts. Install time vs dimensions had a negative correlation because the bigger the device the more time it will take to set the device up for use.

3. Research related to Project Definition

This section will examine similar projects that involve a robot arm that paints that will have similar functions and movements. Researching these projects that are similar allowed us to have a better idea and to carefully plan and adjust our design so that it is successful. It also allowed us to factor the time and economic constraints of the project into the design of the robot painting arm. In researching related projects, we focused on projects that were successful in achieving a fully working robot painting arm. We also factored the resources that were used in these similar projects. In trying to be economic since we have a budget of $500, the team members of this group were creative in finding cheaper yet effective alternatives. For example, in the similar projects the base of robot was a metallic base which is comparatively more expensive than a wooden base. Therefore, our economic solution was to carve our own wooden base as seen in figure of the images.

3.1 Existing Similar Projects and Products

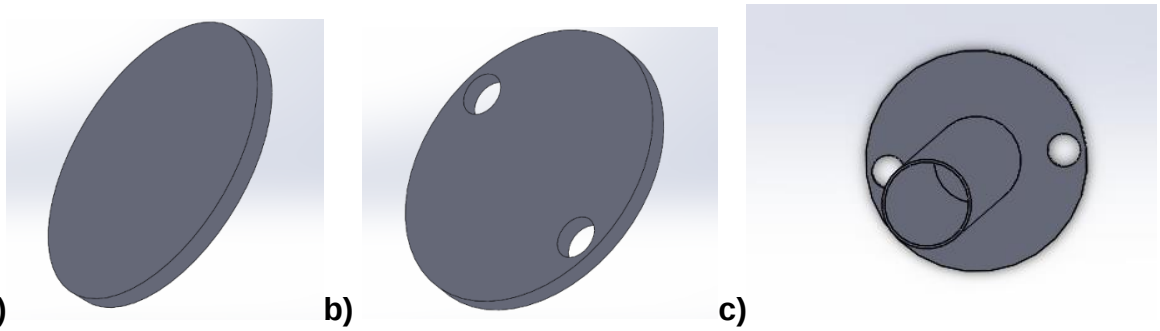
This project is a 3D printed robot which grips the paint brush and paints while the canvas is setup underneath the brush. Although for this project the robot arm paints by moving up and down and side by side using a push button that is pushed by the user. The robot arm performs the same movements with user control. Movements of this robot arm are restricted because the arm is not able to choose the colors that it paints, the arm is not comparing what it is drawing with another image. This is the difference between the arm we will build. For our device it will paint landscapes at an upright position as well as have a camera that will allow the robot arm to better paint images as well as detect different colors to allow a better visual. For this project their idea to grasp the paint brush using a 3D printed part. once the operator puts paint on the brush they will insert the brush onto figure 4 (b) , this robot arm uses a push button that will then grip the brush tighter so that the brush isn't loose allowing the arm to have full control while operating. This robot arm brushes must be changed when there is a color change. The main purpose of this piece is to hold the brush firm enough so that the robot arm can apply pressure on it.

Furthermore, it needs to be screwed in the robot arm. The base design of this piece will be based on the original claw of the robot arm. That way, we are not changing the way the servo behaves with the new piece. This way, whatever rotation movement the other servos due to the entire robot arm, our new piece will behave the same since it's using the same attachment method that the original claw uses.

Our attachment will be designed using SolidWorks. This free software will allow us to create an attachment from scratch. The way we can design our product by setting up the exact dimensions. These dimensions, for our purpose, will come from our brush that already has been decided by the team.

SolidWorks is a solid modelling software that allows any user to design products in 3 dimensions. The technique is generally to sketch 2D profiles then use methods like extruding and lofting to produce the solid shape. It runs on Microsoft Windows. It's a tool for drafting used for making solid model, computer aided drafting which is following the user's command and calculate weight, stress and area of product as per standard record filled in system by the developer. SolidWorks is a solid modeler and utilizes a parametric feature-based approach which was initially developed by PTC (Creo/Pro -Engineer) to create modules and assemblies. Parameters refer to constraints whose values determine the shape or geometry of the model or assembly. Parameters can be either number parameters, such as line lengths or circle diameters, or geometric parameters, such as tangent, parallel, concentric, horizontal or vertical, etc. Design intent is how to the developer of the part wants it to respond to changes and updates. For example, SolidWorks allows the user to specify that the hole is a feature on top surface and will then honor their design intent no matter what height they later assign to the can. Building a model in SolidWorks usually starts with a 2D sketch (although 3D sketches are now available in the newer version). For our purpose, we are using SolidWorks 2019. The sketch consists of geometry such as points, lines, arcs, conics, and splines. Dimensions are added to the sketch to define the size and location of the geometry. Relations are used to define attributes such as tangency, parallelism, perpendicularity, and concentricity. The parametric nature of SolidWorks means that the dimensions and relations drive the geometry and not the other way around. For our project, Solid Work efficient 3D design is an easy to use parametric design modular, meaning you can easily edit the design at any stage in the design process. Real View graphics allow the user to visualize jour design in real time. Also, we can look at each individual part of the design, see accurate mass properties and check for interference, meaning that you won't have to build/manufacture the product before you see any errors, saving time and money and reducing the number of prototypes needed. Compatibility was also a big factor. Due to the popularity of SolidWorks, it is highly likely that a competitor, supplier or customer will be using it, therefore eliminating the need to translate files from one system to another, reducing time and minimizing the chance of errors. It has a short learning curve. Since SolidWorks offers a consistent user interface throughout and drafting procedures that flow logically from start to finish. Furthermore, Solid works has built in tutorials and lots of fantastic pre-sets to make designing speedy and stress-free. For our project, we used the

tutorials from the root website and we also used the resources from www.Lynda.com. This service has a lot of tutorials for students who seek to learn new technologies. This is also free for UCF student. Testing our design. We need to ensure we have the best design before we build it. SolidWorks Simulations validates our design earlier in the process –on-screen-so we can test how it holds up under extreme wind, heat, water, and other conditions. With the answers up front, you can reduce weight, eliminate unneeded materials, and optimize costs, as well as avoid potential liability or safety issues. Design with the environment in mind and measure the sustainability of our product. Fully integrated with the design process, SolidWorks Sustainability tools help us make more informed design decisions in real-time about materials, sourcing, manufacturing, use, and disposal prior to manufacturing.



a) **b)** **c)**

Figure 3.2 (a-c): Paint Brush Attachment

Figure 3.2 shows the output for every step that we took in order to achieve our final attachment. We need a circular basement for our attachment since the attachment from the robot arm is also circular. The Dimensions for figure A are a diameter of 20mm, same as the attachment from the robot arm. Thickness of 1.27 mm, this was optional, but it will add some extra reliability to our part. In Figure B, we use the Sketch option to draw two circles in our base. Each circle will be used to insert our screws. The robot arm attachment is also expecting two screws. The dimensions of these mentioned holes are 3mm diameter. They also need to have 16 mm between the center of each hole to each other. Our robot arm attachment real piece would be expecting 4 screws, but when we took the claw off from the robot, we were able to see that it was able to resist with only using 2 screws. Figure C will show us how a tube coming from the basement. This tube is hollowed. Its depth will vary since we will be having different models. We are using different size of brushes mentioned before. Since we are in the testing stage, we still haven't decided which brush exactly we are going to be using. In total, we have five brushes with different sizes for each one. So, the diameter and the depth will vary from 1 to 5 cm. Our this one shown, we are using a diameter of 7.6 mm and a length of 96.23 mm. We also are going to search for a service that can print the model for us, since due to distance restrictions, most of these services are no longer in the market or are only working with big model requests and since ours is a small model, we are less likely to get approved by a company that does it.

For our design we decided to go with 3D printing. 3D printing by building up objects one layer at a time. This method offers many advantages over traditional manufacturing techniques (for example CNC machining), the most important of which that apply to the industry are covered in this article. 3D Printing is unlikely to replace many traditional manufacturing methods yet there are many applications where a 3D printer can deliver a design quickly, with high accuracy from a functional material. Understanding the advantages of 3D printing allows designers to make better decisions when selecting a manufacturing process and enables them to deliver an optimal product. One of the main advantages of additive manufacture is the speed at which parts can be produced compared to traditional manufacturing methods. Complex designs can be uploaded from a CAD model and printed in a few hours. The advantage of this is the rapid verification and development of design ideas. Additive manufacturing methods generally only use the material needed to build a part. Most processes use raw materials that can be recycled and re-used in more than one builds. As a result, additive manufacturing process produces very little waste. The increase in the number of additive manufacturing machines in the world has also impacted the distance prototypes are shipped. Because tabletop 3D printers have a relatively small learning curve to operate successfully, designs do not need to be sent away to an expert to be manufactured. For this reason, professional 3D printing services are created around the world, even in locations where the cost of land is high. The reduction in shipping requirements has a positive environmental impact. This coupled with the ability to print and produce spare parts on site, results in a much smaller carbon footprint for most parts produced via additive manufacturing.

Comparing this project with our project, the gripping of the paint brushes is a wonderful idea because it allows the brush to be steady allowing more control while the arm itself moves around the canvas. Having to switch the brushes when it's time for color change isn’t efficient enough for us. Yes, it allows the operator to have more control over mistakes. But it is also raising the cost to make if more brushes are required for color change. This robot arm also doesn’t use color detection. The operator controls that. This robot arm does compare what it is drawing with an inputted image. This robot arm WALL-E was tested serval times by the operator for it to memorize the patterns of what it was drawing. After the fifth test, the robot arm WALL-E was able to successfully paint a landscape after adjusting the programming.

This project is also like the robot arm that we are trying to design the Pointillist Painting Robot Arm. This robot arm uses a lot of similar components that we have planned to use for our painting robot. The components used include servos, power supply, push-button, switch, boards, camera, and LED lights.

Comparing this the Pointillist Painting Robot Arm with the robot arm that is being built for this project we wanted an arm that would be small enough to fit on a desk. This would allow the robot arm to be more flexible and mobile. The designer of the Pointillist Painting Robot Arm 3D-printed the parts which is an expensive route to take and is not as simple as buying a robot arm on Amazon. The designer also uses trigonometry math to calculate

desired position for the robot arm based on memory. This part of Aleator777 helped guide us through these equations for calculating the locations when programming our arm.

Understanding the detail and theory of these math equations allows us to dive deeper into our robotic arm. With this math we teach the robot arm how to move, calculating distance between the claw holding the brush and the canvas. The Pointillist Painting Robot Arm uses a camera for image processing. For this robot arm the image input is either hard coded onto the robot arm by the operator or taken from the camera data. Unlike for our project the use of the camera is used only to guide the position of the robot arm and compare what is being drawn to the inputted image. This robot arm only uses one color, so color detection is not necessary for Aleator777 robot arm. While for our robot arm color detection is important in painting landscapes. Pointillist Painting Robot Arm is very similar to our project based on the function of the robot arm itself, how it produces output. We are pushing it a step farther by increasing the level of difficulty our robot arm is expected to produce.

3.2 Relevant Technologies and tools related to Mr. Painter

Raspberry Pi camera are often used in robotic projects because they're small, efficient, and easy to install. The picture above shows one of the many Raspberry Pi camera that are used in projects that require vision. The most updated versions of these camera have a resolution with 8 megapixels as well as sensor capable with 3280 x 2464-pixel static image. Raspberry Pi cameras supports a large range of progressive frames per second making it reliable.

For the robotic arm that paints the Raspberry Pi Camera is probably the best camera to use for this project because this project depends on vision. While the arm compares the inputted picture with what it paints, the camera is used to compare these images ensuring that the robot is accurately painting the image. The camera is also useful for locating and differentiating between a water cup to dip the brush in versus the watercolor paint box. This is important because when there is a color change the robot arm will need vision to know where to position itself. The camera is also used for color detection. Being able to differentiate between colors. If we were painting the sun, then it needs to know the difference between yellow and blue. The camera plays an important role in how the robot arm is able to successfully function.

Servo Motors

Servo motors come in different sizes and are used for the motor skills of the robot arm. Each servo plays a role in how the robot arm functions. Each movement is controlled by the servos. The servo motor receives control signals which in turn it uses to output a position or movement. Servo motors are high in efficiency and power. The bigger the motor the more power required for it to function correctly. Also, the more work required for the servo the more energy and power need for it to perform. Servo motors are used in toy cars, robots, and airplanes. Each servo consists of a control circuit, potentiometer,

motor, output spline, drive gears, and servo case. These parts allow the motor control so that it can run successfully.

The servos are controlled by pulse width modulation through a wire. The pulses are based on maximum, minimum, and repetition. The minimum pulse Width as 1 ms is at a 0 degree, the neutral pulse with a pulse width of 1.5ms is at a 90 degree, and the max pulse with a pulse width of 2ms is at 180 degrees. The degrees affect the position the servo s will cause the arm to turn, life, and rotate.

The servo motors are important for our robot arm to function because each joint in the arm is needed to lift, move side to side, extend, and etc. with the servos it creates a more controlled environment for the robot arm to function. Without the servos there wouldn't be any movement from the robot.

Pulse Modulation

For our project we are using PWM. It's a way to control analog devices with a digital output. Another way to put it is that you can output a modulating signal from a digital device such as an MCU to drive an analog device. It's one of the primary means by which MCUs drive analog devices like variable-speed motors, dimmable lights, actuators, and speakers. PWM is not true analog output, however, PWM fakes an analog-like result by appliying power in pulses, or short

For this robot arm a lot of the movements are repetitive. The robot arm is either swiping up and down on the canvas or dipping up and down when choosing paint color. There are many ways to control this speed, but the simplest way is using pulse width modulation.

Pulse width modulation is used on servo motors to control the speed and movements. For this project we study how the control of the pulse width modulation will affect our robot arm, how we can get the mathematics down so that the arm can perform in an accurate way.

Pulse width modulation control that DC motors use. There are different ways to control this speed, but the simplest way is using pulse width modulation. Pulse width modulation is used on servo motors to control the speed and movements. For this project we study how the control of the pulse width modulation will affect our robot arm, how we can get the mathematics down so that the arm can perform in an accurate way.

3.3 Strategic Components and Part Selections

Robot arm

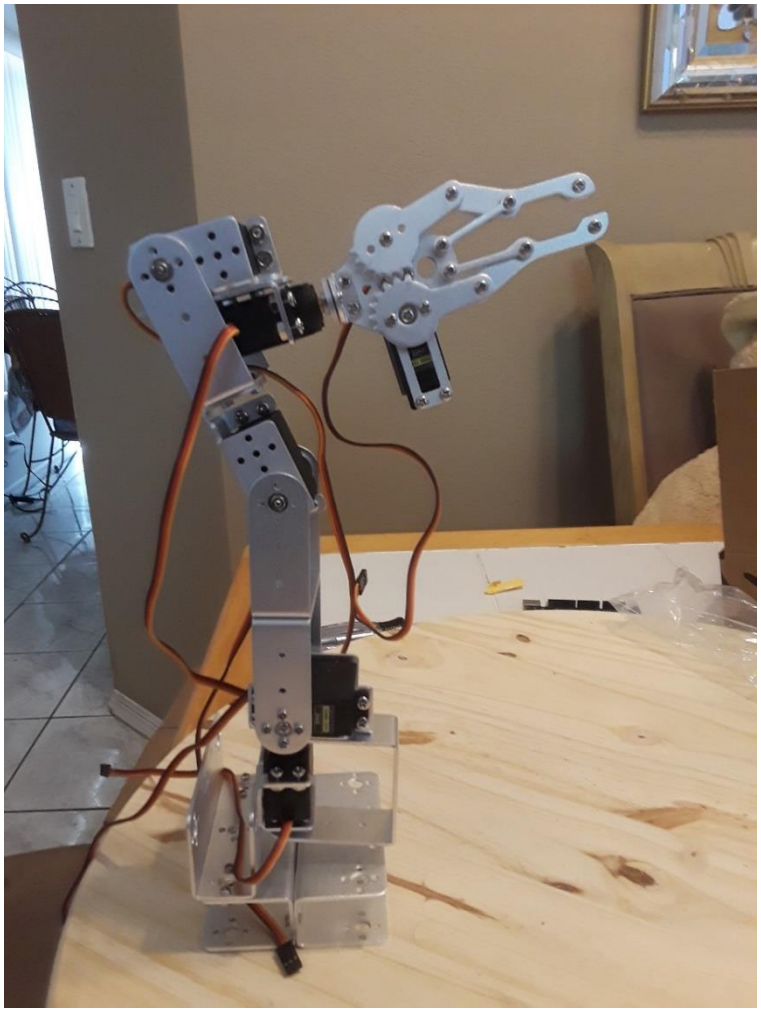


Figure 3.7 Assembled robot arm and robot arm Parts

For our robot arm we decided to purchase it through Amazon because it was the less expensive route to take as well as being the most efficient route to take. The robot arm we purchased is shown in figure 15. Buying the robot arm allowed us to test out different robotic arms to see which would better perform the movements we needed for painting. This robot arm comes a box where we have to assemble the pieces putting it together.

The benefit of this robot arm is that it has

- A radius of gyration of 355 mm
- Rotation angle of 180 degrees
- A height of 460mm and width of 100mm
- Has a clamp to hold the paint brush

These are the reasons we decided to purchase this robot arm for our project. We are also able to adjust the claw/clamp of the robot arm by buying an arm claw kit that can tighten or loosen the grip of the robot arm.

Servo Motors

Figure 3.8: Servo Motors

The servo motors are required for the robot arms movements. The servos shown in figure 16 were recommended on amazon to pair with the robot arm shown in figure 15. The servo motors will be programmed to run using C++ on Arduino.

Camera

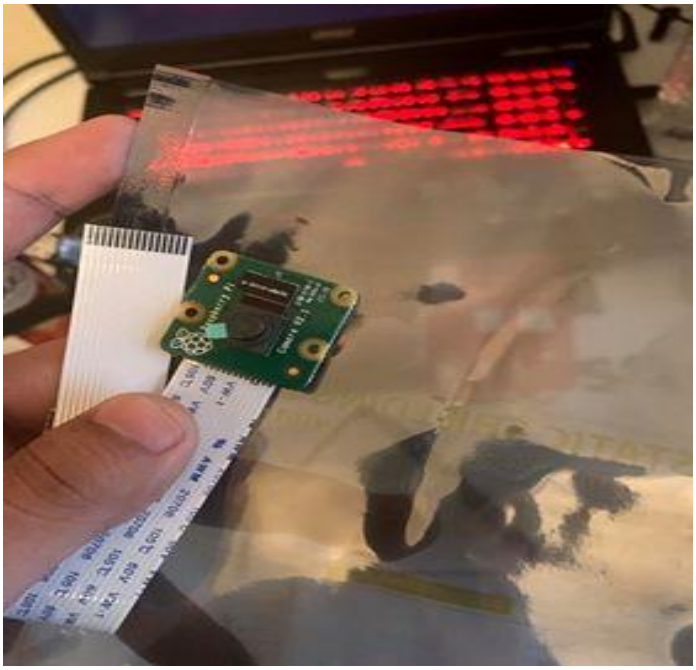


Figure 3.9: Raspberry Pi Camera

We decided to go with this type of Raspberry Pi Camera shown in figure 17 for our robot arm device. This camera is a second-generation module with fixed focus lens. The purpose of this camera is to compare the image being painted with the input image, color detection, and locating the paint.

Paint brush

Figure 3.10: Paint brushes and watercolor.

We decided to purchase our paint brushes through amazon which is shown above in figure 18. They had a wide variety of options. The paint brush will be held by the robot arms clamp. The purpose of the brush is solely to be used to paint the inputted image. We also purchased paint, we chose watercolor paint because it is cheap, as seen above in figure 3.10.

The Watercolor paint we use will has the that are basic such as red, orange, yellow, green, blue, black, and white. These are colors are easier to identify using a color detection camera. The purpose of the water cup is so that the tip of the brush can be dipped into the cup for color change. This is much easier than changing the brush each time it's time to change colors or having the operator clean the brush off. With watercolor paint it is easy to clean off using water.

The base

Figure 3.11 Wooden Base

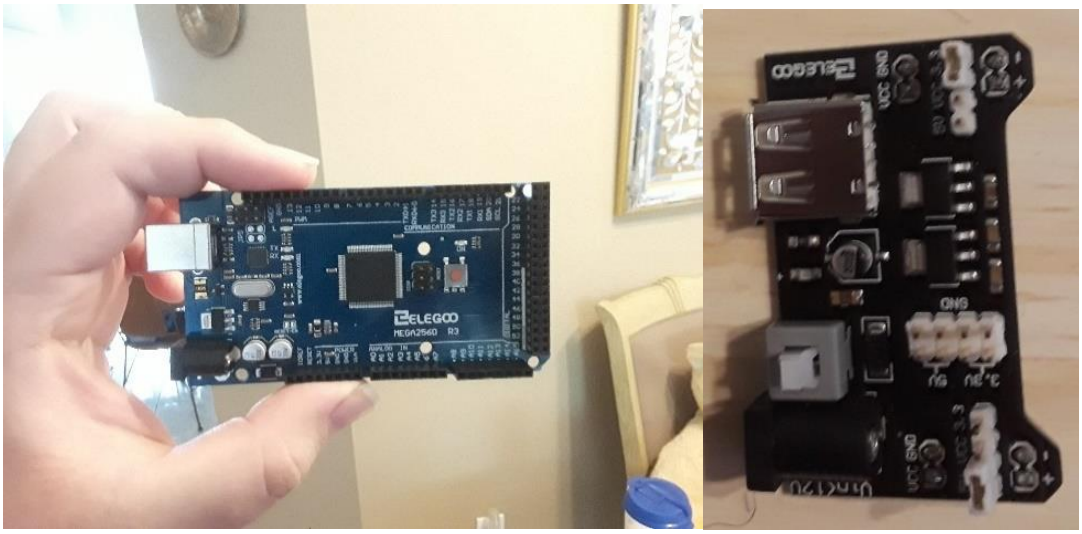
For our robot base we got ourselves a round wooden plate. Judging from the height of the base it should be deep enough to drill screws into it to secure the robot arm. If not we can always just get another one and have the screws go through to wooden plates though that is not desirable, in regard to wasting money. Since I don't often work with wood (last time I did was in high school) this seemed to be the best solution. The base will allow the robot arm to be held together also making it more flexible to move around from one place to another.

Power supply

Figures 3.12 Power Supply 1 left and Power Supply 2 Right

The power adapter shown in figure 21 will allow us to plug our robotic arm in because it will need power to run. The power supply on the left outputs a max of 5 volts at 15 amps This gives us 75 watts of power. The power supply on the right output a max of 9 volts at 1 amp which gives us 9 watts of power.

Microcontroller/PCB



Figures 3.13 Arduino Mega 2560 To the right and a generic PCB designed to handle high power on the left

We will use a microcontroller to control the robot arm. The microcontroller is where the data from the camera and the algorithms that control the robot arm will be computed. Once the data is crunched the pulse width signals will be sent out and the robot arm is move accordingly.

Multimeter

Figure 3.14 Multimeter

The multimeter as seen in the figure above is important because it will allow us to do electrical analysis on our circuits and components. Thanks to the multimeter we can check whether our components are outputting the correct voltage. We can check if the resisters are the correct value. We can check if the servos are getting the correct amount of current. We can check if there is a short circuit. The multimeter will allow us to do efficiency calculations based on input and output power. Without the multimeter we wouldn't be able to do any practical real-world electrical analysis at all.

Canvas

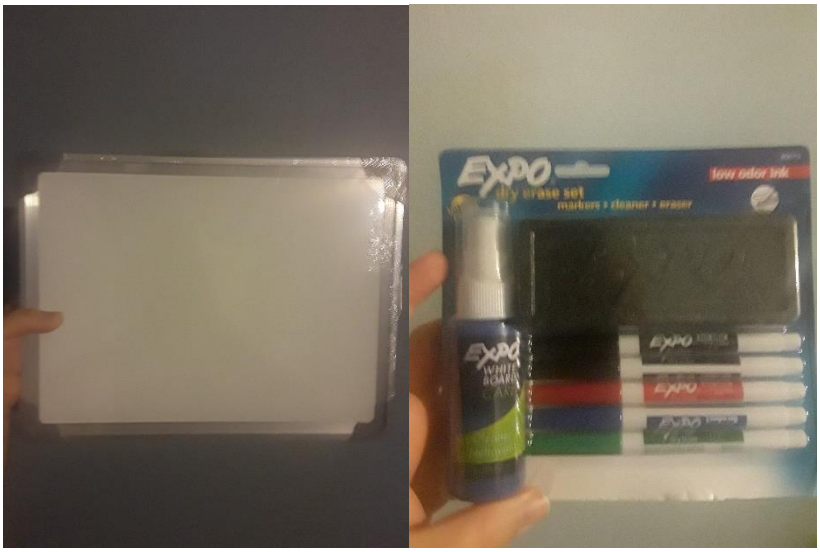


Figure 3.15: Canvas

These canvases shown in the picture above are small enough to use for our robot arm. The inputted image will be painted onto canvases like these which is shown in figure 3.15. finding the right canvas is essential because the robotic device can only reach up to a certain length. Getting a regular sixed canvas will affect how an image is painted onto the canvas. We decided to purchase canvases that would fit onto a regular sized student desk/table with the robot arm nearby. This is better because we don't want a device that includes parts that make it hard to be transportable.Right now to save money we are using a white bourd and markers as seen in figure 3.15

Paint Brush Attachment for Robot arm

The paint brush attachment is a crucial part of the robot arm design. This is because, the attachment is responsible for the accurate execution of painting the canvas. Regardless of the paintbrush or medium used by the operator, the brush holder must help facilitate the execution of the artwork. The mechanical direction of the robot is dependent on the construction of the frame of the robot arm. With no servo motors and no power supply, there will be zero movement on the robot. However, with no paintbrush holder, the principal functionality of the robot arm is lost, thereby reaffirming the importance paintbrush holder. The software direction of the design is controlled by the coding of the servos, as well as incorporating information obtained from the camera. In order to execute the selected artwork of the operator, the software aspect of the design needs to be precise and accurate. Again, the execution of precise and accurate software instructions would be meaningless in the absence the physical paint brush attachment.

Different prototypes of the paint brush attachment are investigated in order to find the best fit for this design. Two main criteria that are essential for the nature of the paint brush attachment are: firstly, the chosen prototype must be able to support our chosen type of brush as well as the standard paint brushes on the market. Secondly, the paint brush attachment must be sturdy and robust to facilitate the flawless execution of the artwork. The paint brush holder helps with reliability, artistic quality, and the general robustness of the design. The paint brush ensures the accurate and precise executions of the paintings.

The dimension of the common brush ranges from 10mm to 100mm, therefore an average length of 50mm was the basis for the paintbrush attachment. The average paint brush weighs less than 40 grams. The width of a paint brush can average about .275 inches. Therefore, the paint brush attachment must support the length, the width and the weight of the average brush, in order to have a proper execution of the painting task. The part of the robot arm responsible for painting is located between a servo and a claw. The original set up of the robot arm comes with a claw at the end as seen in figure. However, the claw at the end of the robot arm will make the painting task difficult if not impossible. Therefore, as a result of this, members of the group began to research and design a prototype of a paint brush attachment that will support the execution of the painting task. In order for the newly designed paint brush attachment fit the robot arm, it needs to be built around the plane in the image below, as this is the foundation of the paint brush attachment. Images of a prototype for the paint brush attachment is seen in figure 3.2.

Figure 3.16 , foundation for paint brush attachment, a generic servo horn

4. Related Standards and Realistic Design Constraints

4.1 Standards

Standards were first created in 1126 AD, as a testament to the constant individualistic ideas and solutions created by human beings. Standards serves as a basis mutual understanding and aid in the facilitation of an idea or a project. In different industries and fields there are standards that exist. For example, in the world of medicine there are health standards or codes that every medical practitioner must adhere to. In the food industry, there are certain sanitary codes of conducts food servers must adhere to. Also, in the world of engineering, there are different standards that engineers and technicians must adhere to. These standards might also differ from country to country. For example, the United States of America is the only country that has its electrical frequency being 60Hz. In Africa and the United Kingdom, this is different as they have an electrical frequency of 50Hz.

Since every member of the team belongs to the college of Engineering it is important that we follow the standards available in the Electrical Engineering industry as well as the Computer Engineering industry. Standards in the electrical engineering industry include the IEEE 802, and this will be discussed in full detail below. In the computer engineering industry, some of the standards include the C++ standard. Again, details on the C++ standards are discussed in much detail below. Another standard discussed is a standard by the Federal Communications Commission. The FCC was founded in 1972 is responsible for the regulation of interstate and international communication either by

radio, by television, wire, satellite and cable in all 50 states of the United States of America. This is an independent US government agency overseen by the Congress. This commission is the primary authority for communications law, regulation and technological innovation. Any details on the provided standards by the commission are discussed in much deeper detail below. Also, members of the team are staying up to date on any information provided by the commission. Here are the standards that are in use with Mr. Painter. Standards are important. The purpose of these standards is to make sure Mr. Painter has a certain level of quality that can be accurately and consistently measured. That the robot is safe to use and doesn't harm us or the environment and that it is used in an ethical manner.

4.1.1 The Federal Communications Commision

Our initial power supply meets the regulations by the federal communications commission. Our power supply has been verified by the Federal Communications Commision to

> "1) designing the digital circuitry in a manner that minimizes radio noise emissions; 2) enclosing the circuitry in a well-grounded case that prevents radio noise from escaping; and, 3) including a well-filtered power supply that keeps the radio noise from leaking onto the electrical power lines." – FCC

Our power Supply has been certified by Intertek, to meet all of OSHA's, the Occupational Safety and Health Administrations standards. The power supply has been tested in the Nationally Recognized Testing Laboratory (NRTL) and was found with the safety regulations OSHA provides. Not only that but Intertek has also certified our power supply also meets the standards for health, safety, and the environment for products in Canada. The power supply features basic insulation.

4.1.2 Python Coding Standards

Our project uses the python coding standards PEP 008 in its code which comprises of (all standards here are taken straight from the python website

Use 4 spaces per indentation level.

- Continuation lines should align wrapped elements either vertically using Python's implicit line joining inside parentheses, brackets and braces, or using a hanging indent [7]. When using a hanging indent, the following should be considered; there should be no arguments on the first line and further indentation should be used to clearly distinguish itself as a continuation line...
- The closing brace/bracket/parenthesis on multiline constructs may either line up under the first non-whitespace character of the last line of list

- Spaces are the preferred indentation method.
- Tabs should be used solely to remain consistent with code that is already indented with tabs
- Imports should usually be on separate lines:
- imports are always put at the top of the file, just after any module comments and docstrings, and before module globals and constants.
- Imports are always put at the top of the file, just after any module comments and docstrings, and before module globals and constants.
    - Standard library imports.
    - Related third party imports.
    - Local application/library specific imports.
- Avoid extraneous whitespace
- Avoid trailing whitespace anywhere
- Always surround these binary operators with a single space on either side: assignment (=), augmented assignment (+=, -= etc.), comparisons (==, <, >, !=, <>, <=, >=, in, not in, is, is not), Booleans (and, or, not).
- Comments should be complete sentences. The first word should be capitalized, unless it is an identifier that begins with a lower-case letter (never alter the case of identifiers!).
- You should use two spaces after a sentence-ending period in multi-sentence comments, except after the final sentence.
- Use inline comments sparingly.
- Never use the characters 'l' (lowercase letter el), 'O' (uppercase letter oh), or 'I' (uppercase letter eye) as single character variable names."

These are just some of the standards of PEP 008 which our code will abide by.

4.1.4 USB

In out project we are using USB 2.0 as a standard. USB stands for universal serial bus. USB are used to connect all sorts of electrical equipment. Not only does USB transmit information but also power. The limit for USB is up to 100 watts. This is good for us since our power supply outputs a max of 9 volts by 1 amp, which is equal to 9 watts. USBs are designed to be durable, and to be able to be hot swappable.

4.1.5 C++ standard

The Arduino microcontroller uses C++ to control and modulate the servos. We have chosen to write our code according to the C++ standards of Geotechnical Software Services listed below (all said standards are taken straight from their website).

- Readability is the highest standard, readability trumps all other standards, any standard listed below can be trumped in the name of readability.
- Names representing types must be in mixed case starting with upper case.
- Variable names must be in mixed case starting with lower case.
- Named constants (including enumeration values) must be all uppercase using underscore to separate words.
- Names representing methods or functions must be verbs and written in mixed case starting with lower case.
- Names representing namespaces should be all lowercase.
- Names representing template types should be a single uppercase letter.
- Abbreviations and acronyms must not be uppercase when used as name
- Global variables should always be referred to using the:: operator.
- Private class variables should have underscore suffix.
- Generic variables should have the same name as their type.
- All names should be written in English.
- Variables with a large scope should have long names, variables with a small scope can have short names
- The name of the object is implicit and should be avoided in a method name.
- The terms get/set must be used where an attribute is accessed directly.
- The term compute can be used in methods where something is computed.
- The term find can be used in methods where something is looked up.
- The term initialize can be used where an object or a concept is established.
- Variables representing GUI components should be suffixed by the component type name.
- Plural form should be used on names representing a collection of objects.
- The prefix n should be used for variables representing several objects.
- The suffix No should be used for variables representing an entity number.
- Iterator variables should be called i, j, k etc.
- The prefix is should be used for Boolean variables and methods.
- Complement names must be used for complementing operations
- Abbreviations in names should be avoided.
- Naming pointers specifically should be avoided.
- Negated Boolean variable names must be avoided.
- Enumeration constants can be prefixed by a common type name.
- Exception classes should be suffixed with Exception.
- Functions (methods returning something) should be named after what they return and procedures (void methods) after what they do.
- A class should be declared in a header file and defined in a source file where the name of the files matches the name of the class.
- All definitions should reside in source files.
- File content must be kept within 80 columns.

- Special characters like TAB and page break must be avoided.
- The incompleteness of split lines must be made obvious
- Include statements should be sorted and grouped. Sorted by their hierarchical position in the system with low level files included first. Leave an empty line between groups of include statements.

These are just some of the standards of Geotechnical Software Services which we will hold our code up to.

4.1.6 IEEE 802

Our Raspberry Pi 4 abides by the standards of IEEE 802. IEEE stands for Institute of Electrical and Electronics Engineers. The Institute of Electrical and Electronics Engineers is standards body that sets standards for electronics and electrical engineers.

> "IEEE is a leading developer of industry standards in a broad range of technologies that drive the functionality, capabilities, and interoperability of products and services, transforming how people live, work, and communicate." –The IEEE

IEEE 802 sets the standards for ethernet, lan, wifi and other means of digital communication over the net. Thanks to IEEE 802 our Raspberry Pi can reliably connect to the internet and interface with local networks with ease.

4.1.2 Design impact of relevant standards

The coding standard helps with readability in the code but does not improve efficiency in any noticeable way. The standard of USB allows our microcontroller to talk to the computer, and for us to be able to transfer our code from the computer to the microcontroller. Thanks to USB we at this moment do not need a power supply for our Rasbery Pi seeing since it is powered by the PC. Since the USB can handle up to 100 watts of power, we know it is safe to use to transfer power from the Laptop to the Raspberry Pi, and from the Raspberry Pi to Arduino. Having our power supplies being approved by OSHA give us confidence that our power supply is safe to use under normal use.

4.2 Realistic Design Constraints

There are some constraints placed on the development of Mr. Painter that need to be considered in order to achieve a cost effective and efficient product. In addition, environmental, social and political concerns are addressed as well. If there are any health, safety or ethical concerns, these have been taken into account and are discussed below. The realistic concerns of the project were amplified due to the current global health

concerns. As a result of the COVID-19 situation, the University of Central Florida's premises has been shut down leaving no room for collaboration. Although the zoom application allows for discussion and deliberation, there is a real concern on the construction and execution of the project. A substantial constraint to this project is the element of time. Due to the lack of collaboration, certain elements of the project that was meant to be done together are now individual projects. For example, assembling the robot parts, the servo motors and the power supply was a one-man job, and this was executed by Alan. In addition, the coding or software aspect of the project was also done individually by Alonso. Designing the PCB to ensure that all components of the project are efficiently connected was the responsibility of Chey. Lastly the task of designing the paint brush attachment was done by Alonso and Trudy. The concerns and constraints listed above are discussed in detail below.

4.2.1 Economic and Time constraints

Economic Constraints

In our discussion of ideas for senior design, the initial plan was to create a robot that made pancakes. However, after considering certain factors it became clear that this would not be feasible nor doable in such a short amount of time due to the fact that the cost of the servos would be too expensive. A robot arm has many servos, and the the weight of both the frying pan and the pancake mixture, exponentially increased our budget. We needed to come with a new idea that reduces the weight of "pan" or a light item that our arm could handle.

Mr. Painter is wonderful project that can help art communities achieve so much more in very little time. It is paramount that we have a practical budget. This is because we have no sponsors for this project and have limited time to finish this project. The budget agreed upon is $500, meaning certain specifications were simplified.

In addition, Mr. Painter could have had much more flexibility in terms of number of servos used. In our specifications, there are six servo motors to aid with the movement of the robot arm. For economic reasons and to remain within budget, six servos are used. A pack of four servo motors costs $20.88 and using higher grade servos would mean an increase in the cost which we decided was not worth it.

In our discussions for Mr. Painter, we initially wanted to 3-d print the robot arm in order to include every specific detail we wanted such as changing the number of servo motors or even changing the positions of these servo motors. After some research on places to 3-D print and doing a cost analysis, we realized that we would go above our budget if we decided to 3-D print. There was no way to 3-D print our robot arm without spending at least a $100. Taken that into account, we made the cost-effective decision of buying a

robot arm and assembling it. With this approach, we could take advantage of the cheap robot arm kits available on amazon. The SunFounder Robotic arm edge kit for Arduino R3 was then for $58.99. Once it arrived it was assembled and tested. We unfortunately realized that it was weak, the material quality was substandard, and it easily collapsed on itself.

We then concluded that it would not be able to perform the task of painting. We then purchased the Silver ROT3U 6DOF Aluminum Robot arm with arm mechanical robotic claw. The great thing about this robot arm is that it came with six servo motors, eliminating the need to purchase the servo motors separately. By purchasing this silver robot arm we were able to save on both the servo motors and the robot arm. Initially we would have spent $100 on 3-D printing the arm and $40 on purchasing the servo motors, a total of $140.Whereas we bought the silver robot arm plus the 6 servo motors for $82.99.

Time Constraints

Considering we have approximately three months to design, create and develop a fully working robot painting arm, there is a time constraint that needs to be discussed. Furthermore, our design needs to go through rigorous testing. It may also need to be redesigned in case of any difficulties. Finally, our group will need to show the functionality of our project by presenting before the judges. The milestone section provided in this report captures the schedule we intend to follow in order to meet the three-month deadline, which in my opinion is a lot to learn for our group, since we never had any real word experience.

Another time constraint that needs to be discussed is the delay in shipment for almost all online purchases from China. Unfortunately, due to the Corona virus, a lot of online shipments are currently being delayed by about an extra week or even an entire month. And this would affect the schedule project. In purchasing the SunFounder Robotic arm edge kit for Arduino R3 in the late part of February this year, it took longer than the standard time to arrive. This has been the Silver ROT3U 6DOF Aluminum Robot arm. Due to this delay, we have had to purchase our parts earlier in order to meet the deadline for this project.

Furthermore, the issue of Corona virus has prevented physical collaboration and therefore assembling the parts or even testing the prototype would be affected and might cause a time constraint. Meaning if we initially intended to test the design over a period of two weeks this might reduce to one week due to the issue of social distancing.

The time constraints for this project will have an impact on the specification of our design. Initially, the plan was to have a pancake making robot, however since Mr. Pancake had many factors that needed to be controlled such as checking the temperature of the pan, and managing spills from the pancake batter, Mr. Painter was selected as the final project

idea. Additionally, to the meet time requirement for this project, Mr. Painter which we initially wanted to be capable of painting anything and everything has now been limited to painting landscapes. This was done in order to have a practical workload for the three months available for testing and redesigning. Thou we hope it can do more.

Another element of this project that will serve as a source of time constraint is OpenCV. OpenCV is a computer program that is a library for functions that aid with implementing real time computer vision. Although OpenCV is simply a library, this software is a crucial part of the entire project. This is because the robot arm can only paint what it is instructed to paint. The camera of the robot becomes an input using the OpenCV software. Becoming conversant with the OpenCV software might take a while, depending on the pace of the learner. Therefore, taking into account the three- month time limitation, members of this team are taking a course on OpenCV through Udemy.

One of the vague reasons why we decided to use OpenCV was because we found a way to use python with the servos as well. Meaning, we would need to install python in our microcontroller, so that we can use our servos and camera using one simple programming language and not having to deal with programming on more than one language anymore, which would increase the time of developing the final algorithm since we would have two completely different scopes.

Lastly, the reason for the three-month design period is due to the fact Senior design 2 which is the second half of Senior design 1 is often offered in the subsequent semester, in this case Summer and Fall. Our team had the option of picking Summer or Fall of 2020 to complete the second half of Senior design. We all agreed on the second half of the semester thereby giving us three months' worth of testing and redesigning. Furthermore, considering the times that we are currently in, Social distancing has increased the time constraint for the project. This is because, times or periods meant for physical collaboration or assembling parts of the robot have now somewhat become a one-man or woman task. Since there can be no social gatherings, for the mean time one person is responsible for certain tasks such as assembling robot parts, testing and we all can redesign in case of any difficulties. Time is not on our side.

4.2.2 Environmental, Social, and Political constraints

Environmental

Present day sustainability goals and efforts to make everything green and efficient caused our team to find ways of sustainable ways of building our robot painting arm. Although, this was not discussed extensively, some design decisions were made with sustainability in mind. For example, we saved economically and on energy by selecting the 5A DC power supply. In addition, by arranging our servo motors in parallel, the servo motors share the 5 V making it more efficient. For the canvas in which the robot arm would paint, we have concrete plans on testing in the most efficient way in order to conserve

resources, specifically trees. When testing the watercolors, we will use both sides of the paper and canvas. At the end of the project, once everything has been tested and our presentation before the judges is made, we have plans of keeping the project by reusing the project to demonstrate our skills at engineering and simply to reminisce on our Senior design days. Since we are not throwing away the Mr. Painter none of it will end up in a landfill and damage the environment.

Social

Some social constraints for this project include social distancing. The case of the virus has affected physical collaborations, online purchases and the schedule of our project. Due to social distancing, we are unable to meet up and this is beginning to become a major constraint that might affect the project.

There is also the social constraint of painters and artists losing their jobs if this robot goes mainstream. In the hierarchy of jobs, artists, painters and most creatives are currently underpaid for their services. Consequently, automation might cause a deeper problem in the creative or art community. A solution to this might be a collaboration between creators of the robot painting arm with art communities. In doing so, artists and painters would not be left out and, this will be an effort to technologically progress the art community as well.

Another social constraint is how these machines might affect people's reaction and even appreciation for arts. In the art community, there is an appreciation for the idiosyncrasies that exist with each artist. For example, Basquiat an African American artist was famous for having little texts in his art pieces. This was his signature. Having a robot painting arm might remove the individualistic effect that comes with art pieces. Robot painting arms could influence the future of art by having certain genres of art being proliferated more because they are easier to reproduce by the robot painters.

For example, in film once digital film making took over of analogue film due to it being easier to shoot and store films became de-saturated because digital film mutes the color of what it records. This in turn led to the rise of color grading film where film studios in post-production add color back into the movie to make up for what was lost. This leads to movies looking synthetic. The tools of the artist effect the art, and Mr. Painter is a tool.

Political

The major political constraint that affects the project is with the war against Artificial intelligence and politics. US Senators feel that robots with facial recognition might be racist and sexist. This is because the elements for creating facial recognition algorithm would include racial features and skin pigmentation in order to perform.

4.2.3 Ethical, Health, and Safety constraints 2 pages

Ethical

Some ethical limitations associated with the robot painting arm has to do with the ethics of the art community. It's completely unethical for a fellow artist or any person to copy or duplicate the work of another artist without acknowledging the original artist. Therefore, with our robot, utmost effort would be put into acknowledging and citing any works that might be duplicated.

Health

The US Department of Labor recognizes that there are some hazard concerns and health limitations associated with robots. They elaborate that being unaware of these hazards can pose a fatal threat to users. Although our robot painting arm might be small compared to numerous robots out there, utmost care needs to be taken in case children or toddlers encounter the robot. According to the department of labor, there are several health concerns related to robots. Some of them include heat stress, ventilation concerns, noise, and indoor air quality concerns.

Heat stress according the Occupational Safety and Health Administration (OSHA) is the net heat load that a user is exposed to. There might be a few things that may amass to the heat that a user is exposed to. For example, physical exertion, environmental factors and the clothing worn. Heat stress needs to be taken serious as this may have the compounding effect of causing heat rashes, heat exhaustion, heat stroke, rhabdomyolysis (destruction of a striated muscle cells), heat syncope and heat cramps. The conditions mentioned above are serious conditions that require medical attention. Therefore, the efficiency of our robot will be worked on in order to reduce any heat losses and furthermore, any heat stress.

Ventilation concerns are generated from emissions, exposures and chemical hazards. According to OSHA, ventilation is deficient in confined spaces, facilities failing to provide adequate maintenance of ventilation equipment, facilities operated to maximize conservation, windowless areas and areas with high occupant densities. Fortunately, our robot painting arm does not generate emissions or chemical hazards. Regardless of that, the team members have no control over the venue for our final presentation. It is paramount to ensure proper ventilation during our final demonstration and testing periods leading up to the final demonstration or presentation.

Noise according to OSHA is any sound at a workplace. The movement of the robot arm, while it performs the task of painting would cause some type of noise. Until the robot is completely assembled and tested, we would have no idea of the amount of noise generated by the robot. The effects of excessive noise in the workplace, may have adverse effects on individuals in that workspace. Some of these effects include auditory effects and worker illness or injury reports. Auditory effects such as hearing loss are one of the most common workplace auditory illnesses. Due to the lack of visual symptoms, it is often ignored and not discussed. In addition, injury related to noise develops over a long period of time and might take a long time to notice. Injury reports related to noise are

12% of all injury reports with 18,000 reporting this case in 2010. In addition, fortunately our robot does not produce any gasses and therefore there are no indoor air quality concerns.

Safety

Some safety concerns associated with this robot are electricity related**.** The robot would be painting with a pigment or fluid. As most of us know, electricity and fluids do not mix. Consequently, this is a limitation to our project and measures need to be taken to ensure the safety of users of this robot. Another major safety concern with this project is that any device or machine that uses electricity needs to follow the basic IEEE standards in order to prevent cases of users getting shocked. The electricity component of this project creates a huge safety concern if care is not given.

Another safety concern is that once the robot painting arm is fully functional, numerous tests need to be run to prevent any accidental spills of the watercolor paint. This can be done by verifying the code for the robot, as well as ensuring that any calculations made are precise and accurate.

5.Project Hardware and Software

The core foundation of the robot painting arm is the hardware and software components of the project. The hardware of the project is the physical machinery of the robot painting arm. This includes the servo motors, the power supply, the robot frame among other components. All these components were purchased, tested, assembled and tested again. The software part of the project is the programs run or instructions given to the robot arm. Some of the programs used in the robot painting arm includes OpenCV and Python. With the make-up of our team being Electrical and Computer Engineers it was important that we did not create mechanical complications that we may not be able to handle. Therefore, in the same school of thought, it was decided that the robot structure or frame would be purchased and assembled allowing the electrical and computer engineers on the team to focus on the hardware and software specifications with regards to electrical and computer engineering. Initially, we decided to 3-D print the robotic arm, however after checking the prices on different websites and considering the time constraint associated with the project, it was decided that the robot arm or structure would be purchased. Deciding on the type of robot structure to use began with avid research on different robot frames out in the market. The criteria for selecting the robot frame were accuracy, feasibility, and other core traits in relation to its functionality. Another intrinsic criterion was that it had to be economically practical, considering this project is funded by Alan Azargushasb who is a student. After this research was done, robot frames were bought and tested.

The task of selecting the suitable robot structure came in three different phases or three different prototypes. The initial prototype was purchased for $60.This was an excellent economic decision since it was the cheapest robot arm on amazon. The servos that

moved the arm were the sg90 and a torque of 1kg/cm. The main functionality of the robot painting arm is to be able to pick up the. The first robotic arm however was to weak to carry out this operation. A detailed explanation on the ineffectiveness of the first robot is provided below. In addition, due to the failure of the first prototype, no further purchases were made until a specific prototype was decided upon. For the next two prototypes, the design specifications are listed out in images 5.4 and 5.5. For the second prototype, it was decided that the power supply would be connected to the camera, the raspberry pi 4 and the Arduino 2560. This is then connected to the PCB, the servos, the robot arm and finally the execution of the painting on the canvas is done by the robot painting arm. For the third prototype, the main difference was including a laptop. In this prototype, the camera is connected to the laptop and the same camera is made to face the canvas to be duplicated. The laptop is then connected to the raspberry pi and then this is connected to Arduino Mega 2650 and then this is connected to the power supply and the robot arm at the same time. The power supply is then directly connected to the PCB.

Another crucial part of the hardware is the servo motor. This part is important is because it controls every movement of the robot painting arm thereby executing the task of painting the canvas. The task of painting would involve the brush being dipped in a pigment, preferably watercolor and further this pigment is applied on the canvas following the instructions provided from the laptop as seen in prototype 3 or from the Arduino Mega 2560 as is common with prototype 2.The reason for having six servo motors is to again allow for maximum motion in order to achieve a better execution of the action of painting. Each servo allows for about 0 to 180 degrees of motion thereby assisting in achieving maximum motion. In addition, each servo motor is in charge or represents each direction.

An additionally important hardware component for this project is the camera module. With the camera present, the location of the pencil or brush will always be known in relation to the canvas. Thankfully with 8 megapixels, the camera has color recognition and therefore any image or landscape uploaded as an input will be visually analyzed in order to be duplicated. With only 2V, the camera is fully operational and ready to work. Syncing each servo motor with the camera would allow each bit to be painted on the canvas. In order to verify or validate the quality of the work being produced by the robot arm, the pixels and color of the original work is constantly compared with the artwork produced by the robot arm. The camera of choice has the ability to work as a CCTV security camera, its able to notice motion detection and also time lapse photography. For better processing of data from the camera, OpenCV and python are the best options. Arrays are the way in which images are saved. The way the robot arm works is that data collected from the camera is recognizable due to the OpenCV library.

5.1 Failed Design 1

A lot of technological advances began with realistic goals or objectives, however once these plans were executed on a practical level, engineers came across a mountain of problems on their first prototype. This was common with SpaceX in the startup years of company as well as Amazon which is now a trillion-dollar company. This is to explain that although we as a team had the best proposal or objective for our project, we were bound to come across some problems. This was realized in our initial purchase of the robot frame. We have since come across issues concerning efforts of physical collaboration due to the COVID-19 situation. In this section of the report, we outline the different designs or prototypes for the project as well as outline the success or failures of each design.

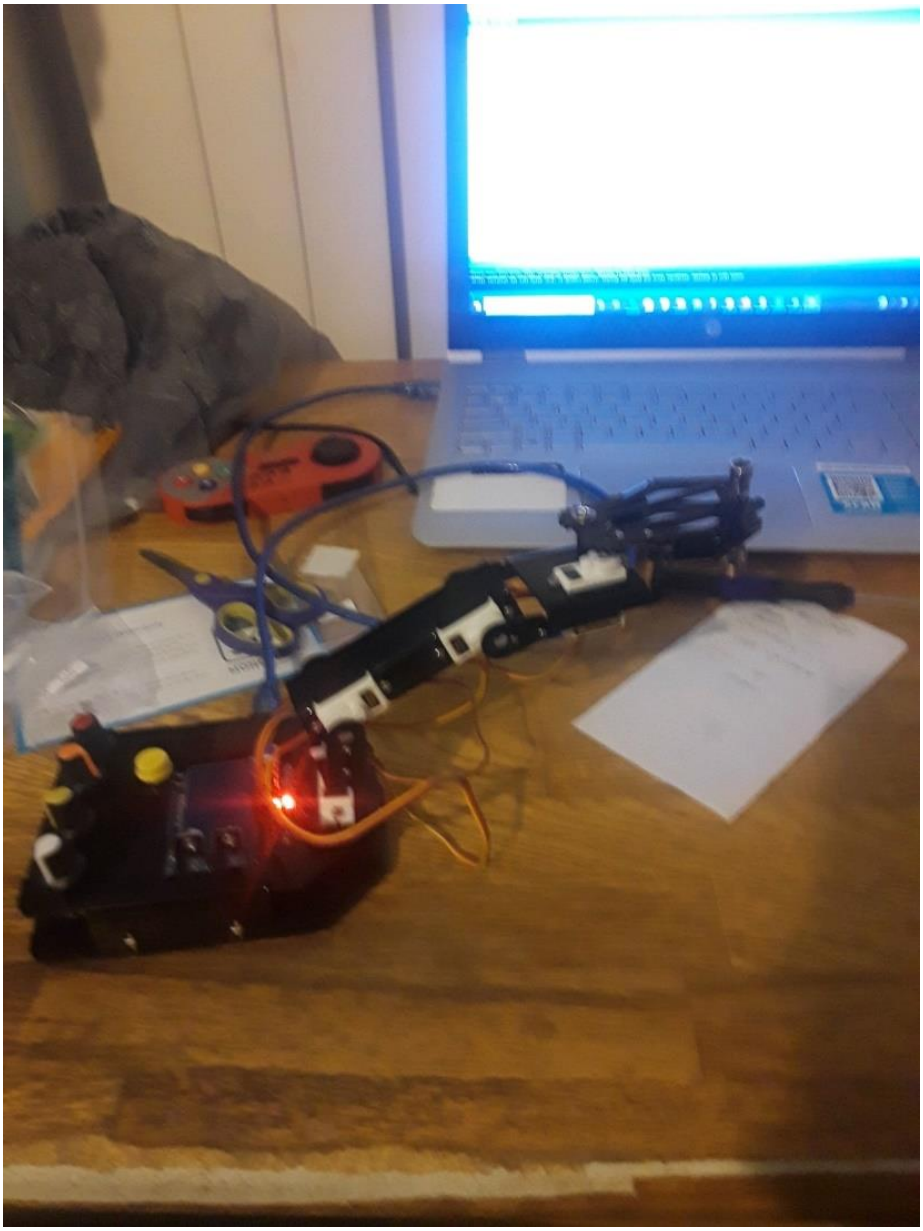


Figure 5.1 Image of the first robot arm Alan assembled.

In the beginning of Mr. Painter, we tried to go get the cheapest robot arm available. The robot arm was only $60 dollars and was indeed the cheapest robot arm on Amazon. The servos that moved the arm was the sg90. It had a torque of 1kg/cm. Since we were not mechanical engineers, we didn't know that that was too weak to do anything with. I thought that 1kg/cm would be enough to pick up a brush, but it was too weak to even pick up my pencil. It could hold it position for with a pencil but could not lift it up. Many times, the servos would act up and do whatever they want, and then collapse in on themselves. I assume this has to do with overloading the servos, but I am not sure.

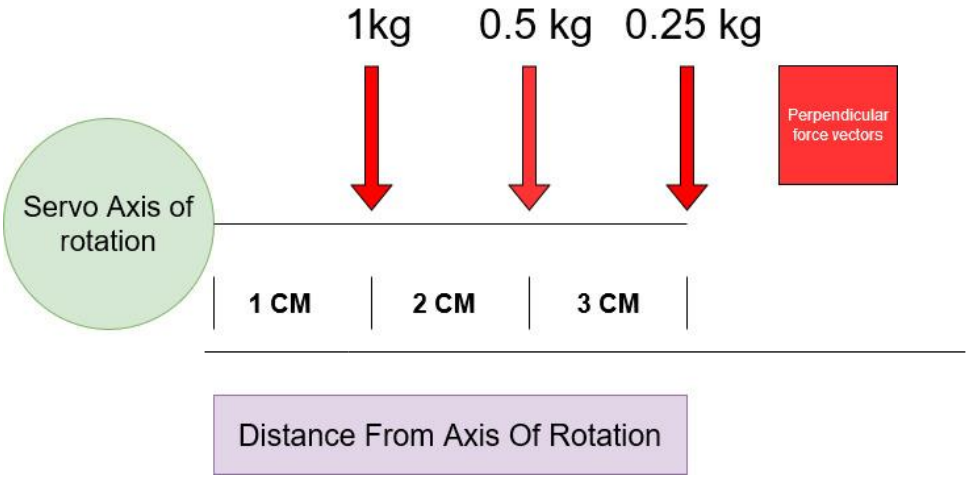


Figure 5.2 Diagram of the mathematics of Torque made by Alan

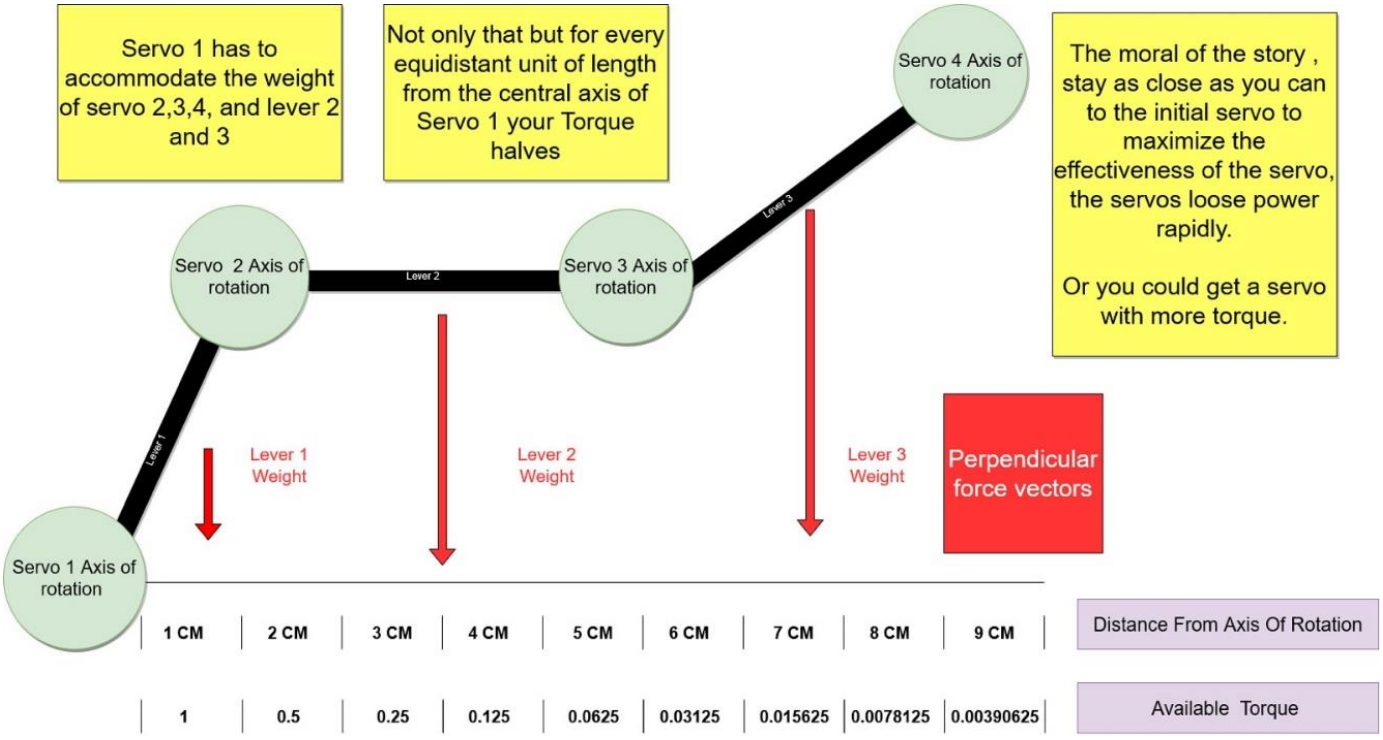


Figure 5.3 Torque Loses diagram

Seeing as how the torque quickly halves per every equidistant unit of length, we are left with a few solutions.

1. Decrease the weight. This would not work since the original robot arm was already made of cheap thin bendable plastic.
2. Limit the range of the arm to minimize torque loss. This would not work since the whole point of the arm is to paint, and in order to paint you need a full range of motion.
3. Increase the servo power. This can work, and so far is our current solution. Our current servos are 9.5 to 11 times more powerful than our original servos.

5.2 Design Details 2nd Draft

This is the second model we have designed so far after our first design failed. The power supply connected to the Raspberry Pi 4 and to the Arduino mega 2560 through the 5-volt output pin powers the servos on the robot arm. The robot arm moves according to the code on the Arduino. The bulk of the calculations however are done with the Raspberry Pi seeing as it is powerful enough to handle high definition images and artificial intelligence. If in the end it turns out we don't need the Arduino we will scrap it and have everything connected to the Raspberry Pi. The pcb on the bottom of the diagram is auxiliary to the Arduino at the moment. We are right now assembled into two teams. Team one is the robot arm and team two is the vision team. The robot arm team is responsible for constructing, assembling, and programing the robot arm. The vision team is in charge of the programing the camera to do object recognition, color recognition, and a controller (which kind we haven't decided, but we are looking into pid at the moment) that tells how the painting is going.

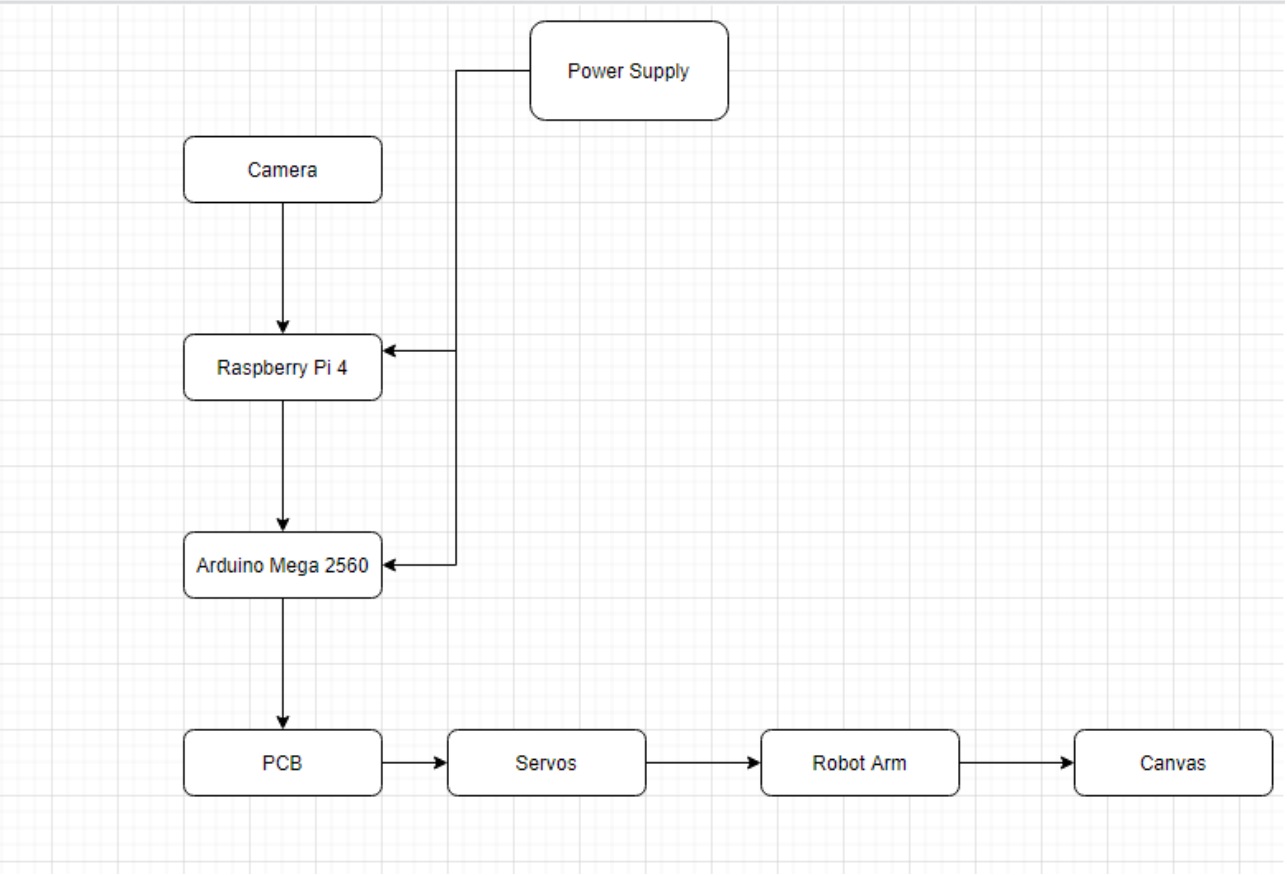


Figure 5.4 Block diagram 2nd design

5.3 Third Design

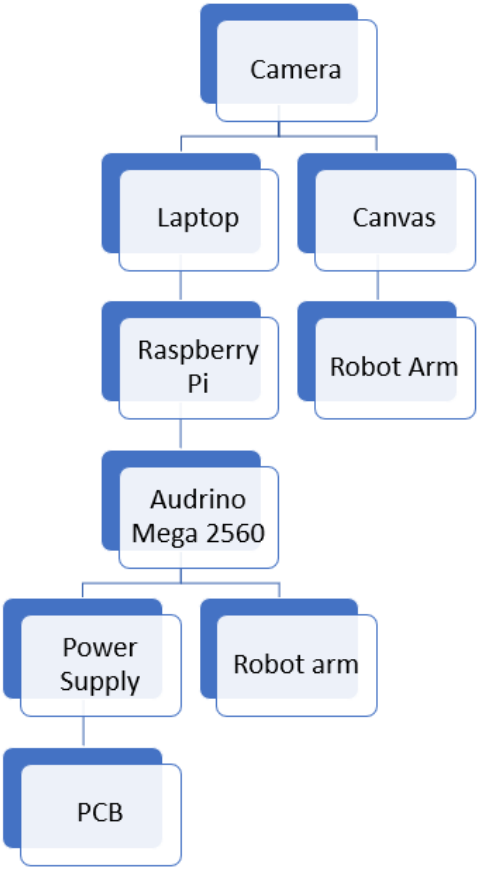


Figure 5.5 Third Iteration of Mr. Painter

After initial testing the servos on the breadboard it became apparent that design 2 could not work because the Arduino only allow a maximum draw current of 20mA. This is insufficient for our needs.

5.4 Requirement Specifications

The robot arm will have multiple requirements to be met to ensure accuracy, feasibility, and other core traits about its functionality. To increase user friendliness the design must be able to achieve a number of tasks within a certain time limit, utilizing communication methods, be able to last for extended periods of time and be ergonomic and easy to use, to list just a few. These factors will determine whether the product can be successful in the marketplace and compete with other robot arms that have existent and dominated the market share. Our requirement specifications can be broken into multiple categories as such, functional, economic, and power consumption. Each component used in our design was carefully chosen based on these three factors. Since the marketplace for robot arms is expensive, our best bet is to try to reduce the economic factor as much as we can without losing the durability factor.

Our design will require a pre-assembled robot arm with set dimensions. These limitations force us to pick our components big/small enough to fit the robot arm for future testing. Its radius of gyration is 355 millimeters and a rotation angle of 180 degrees, meaning that all our servos must have the same rotation angle. The operation voltage for this robot arm is about 4.8 - 7.2 volts, meaning that our components must be within the same range to avoid any possible short cut. The dimensions of the arm after being assembled is 460 millimeters for the height and 98 mm for the wide. Also, since the clamp is the smallest component of the robot arm, it will require the smallest servo motor in order to function properly. The range of the size is about 55 millimeters.

The assembly process of our robot arm will include 6 sets of 25T servo horns that are going to hold our servos. These are made of aluminum and have a pinking aperture of approximately 5.5 millimeters. The servo disc diameter is around 20 millimeters and the mounting hole spacing is about 14 millimeters. Since the parts of the robot are made of metal, the weight will be into high consideration. It will be placed in a table where it would not be able to move when it is performing its action.

- **Six Servo Motors - MG996R:**

Why do we need it?

The use of servo motors is vital for us to control every movement of the arm robot, including the pen. This specific model is mostly used for general movements that don't require to move small pieces. This model has a decent size of 40.7 x 19.7 x 42.9 millimeters approximately. Which means that our robot arm should be wide enough to fit it. Also, with a weight of 55 grams, we need to test how each joint of our robot arm will move. Due to the total weight of our total arm, we need a power supply big enough to provide 6 V for each servo. Current will also be used for the stall torque, 9.4 kgf.cm (4.8 V) to 11 kgf.cm (6V).

Since, our robot arm will be implemented by using more than one direction, we do require one servo per direction. For instance, since our design needs to have a six-way movement, we will need 6 servos for our arm. Each servo can make a rotation from 0 degrees to 180 degrees using an operating speed of 0.17 s/ 60 degrees to 0.14s / 60 degrees using at least 4.8 V and 6V respectively. Each of our servos are stable and shock proof double ball bearing design, which meet our requirements for durability so that we can compete against existent robot arms. Also, temperature is a big factor in our durability requirements. These servos can function properly in a temperature range from 0 degrees Celsius – 55 degrees Celsius

Each servo is controlled by sending signals from a microcontroller. We use the concept of Pulse-width modulation (PWM) to move our servos. This requires the knowledge of duty cycle and frequency that our microcontroller will set and send to our servos. In our design, each of the servo will require a dead band width of 5 microseconds

How does it work?

From a hardware point of view, inside there is a simple set-up: a small DC motor, potentiometer, and a control circuit. The motor is attached by gear to the control wheel. As the motor rotates, the potentiometer's resistance changes, so the control circuit can precisely regulate how much movement there is and in which direction. The desired position is sent via electrical pulses through the signal wire. We also need to take into consideration the voltage needed for each servo. Since all the servos are DC, we only need one power supply and connect all of them in parallel. A voltage of 5 – 10 V would be more than enough to power each servo.

How are we controlling the servos like we desire? One option is we are using raspberry pi to control the servos. This microcontroller brings us the compatibility needed to use the servos we need. We connect the physical pins to the servos that we want to work, then we set them up using Python. This mentioned programming language will bring us a library that will make it easier for use to set, increase or decrease the pulses needed for our servos to work as we desire. However, we still need to calculate the actual distance and slope that our servos need to move as we really want.

Once we figure that out, we can only insert the points that we want our servos to reach, and it will do it without the need from us to set up previously. The math behind this will be explain in another section. The following diagram is done by ours and gives a vague explanation on how to connect the pins from our servos to the pins from our microcontroller involving the power supply pin, ground pin, and the physical pin. This last pin will oversee receiving and sending signals to our specific servo, so that our servo can perform actions as we desire.

- **Camera Module**

Reason for camera module

The use of the camera will be for many reasons: It will guarantee where the pencil or brush is located, so that our robot arm can reach it. Due to its dimensions 1 x 0.9 x 0.3 inches, it will be in a position where it can see the whole process. It will detect how far the paper or canvas will be located so that our robot arm can calculate the distance needed to move the arm along with the pencil to draw the picture. Due to its 8 megapixels, (3280 x 2464 pixels) our camera also has color recognition; we will be using it to draw the right colors of the picture that we want to draw. We also have the option to record the entire process of the drawing since it has a capture video of 1080p30. It is also under our power requirements since it only needs 2V for functioning properly. To sum up, we need the camera to calculate the new distance and movements from our arm every time our robot needs to perform an action. For every signal sent from our robot to our servos. Our servos need to be synchronized with the camera, so that for every bit painted from our picture,

our robot will require a different distance measured. The second functionality of our camera is to measure how much more is left to paint from the original picture. Without this, our robot will be painting the same picture again without having a certain order. The way our robot will work is to paint in a specific order. This means, we are going to break down the entire picture by taking a picture with the camera. The first picture taken will work as the original picture of the picture. After the first picture has been taken, we need to send the signals to the servos. In addition, in order to avoid any repainting unneeded, we will need to take more picture inside a window of x seconds (we are not completely sure of how much exactly we need) to compare it with our original picture. In other words, every picture taken after the first one will have a single purpose. This purpose, for us, is to compare it with the original picture, so that we can paint the desired picture in order. One way to compare each picture to each other is by using pixel and colors. We can use the signals sent to our microcontroller and convert them into pixels so that we can have an actual numerical representation of our picture or new picture. The colors will help us identify which colors are not present in the picture. This will allow us to find a more accurate representation of how much left we need to paint. This will help us to match the picture as close as possible for every move of our arm.

Functionality of Camera Module

Since it has supported applications such as CCTV security camera, motion detection and time lapse photography, we will be connecting it to our raspberry pi microcontroller, from here we will use our laptop to enable the camera. This mentioned microcontroller offers us a module that we can connect to our camera module. Since our microcontroller needs to be connected to our computer using USB to work, this will bring enough power to the camera to work properly. After this, we use the computer to set up our camera with the microcontroller.

The following picture is the exact micro-controller that we are going to use for our project. There are different choices of this same module that are used for certain specific conditions, however, since our project will not require high intense usage or graphics design, we decided to go with the lowest module of 1 GB of RAM. This verson is more suitable for use since we can use it just for testing purposes. For example, we can use it for testing our code that we are going to implement for this project. It will also not be code that will require any PC usage, meaning that it will not be heavy intense for our microcontroller. The way this microcontroller is set up can be complicated, but it will be explained in this section. As mentioned before, we need to use OpenCV and python for our camera module. The reason why we are using them is to have a better control of the camera. We will have access to different methods from this mentioned library that will helps us in the process of converting the data coming from the camera. The microcontroller power supply is from about three to five volts, meaning that we can give it enough power from our laptop/desktop. It comes with 4 USB ports and 1 USB-to-type C port. This last one is mainly used for giving power to the microcontroller. We tend not to use the USB ports for giving it power because we mainly use them for connecting our

keyboard and mouse. We normally use USB ports to transfer data and a fast and more reliable way and not for electricity. After plugging the type C port to a computer, we have two completely different ways to configure the microcontroller. The first and easier option is to plug a mini HDMI port from the microcontroller to a monitor. The second option is to find the IP address from our windows consoles and program it from there. However, both methods will require previous step to boot up this microcontroller. I will proceed to explain these methods.

First, we need to install Raspbian into our microcontroller. The best way to handle this situation is by downloading it into a SD card and then put it into our microcontroller. I like to use SD Card Formatter because if we format the wrong drive, I will lose all data on that drive. It makes it easier for some people who have never installed or formatted SD card. Secondly, we are going to download NOOBS. NOOBS stand out of the Box Software. It is the best place to start for beginners. Rather than installing the Raspbian OS directly, NOOBS is an easy operating system installer that can not only install Raspbian, but also a selection of alternative operating systems. It's really great way to get exposure to the various OS installations available for Raspberry Pi. After installing, navigate to the NOOBS Download Page either the ZIP or Torrent for the full version of NOOBS (not the Lite version). Also, any operating system NOOBS older than 3.1.1 will not be compatible with raspberry Pi 4. After downloading the Zip file, go inside the folder, then copy the contents of the folder to the root of the microSD card. Then, insert the microSD card into the Raspberry Pi 4 and hook up a monitor (his is the hook up monitor way), network cable, keyboard, and mouse. You can also use Wi-Fi to install the operating system, but it may take longer. After hooking it up with a monitor, this one will show the boot screen. You will see some option, but one of them is called Raspbian Full. This is the one we need to install so that we can have all our dependencies. It will take about 15 minutes to install the whole operating system.

Once the installation has finished, we click OK and our Raspberry Pi will reboot. Upon first rebooting up, you will receive a welcome screen and a startup wizard. Note that my IP address will show on this screen. We will need it for later, so it would be nice to write it down somewhere. After that, we are going to take to rebooting steps such as establish the connection to our Wi-Fi, set up the local time and a password. After this, we are ready to start our configuration and remote access. Look over the options. Here we can choose to boot to CLI instead of the Desktop, we can set the system hostname, and we can disable auto-login to the Desktop. For security, it is recommended to uncheck the setting. Now, when the system reboots, it will prompt for username and password to log in.

Then, we go to the Interfaces tab and select Enable next to SSH and VNC. SSH will allow us to remote CLI access to the Raspberry Pi, and VNC will allow us to open a remote desktop using VNC Viewer. Meaning that, after this we can VNC and SSH to the server. To test our connection with SSH. We are going to use Putty from Windows to connect. We need to open Putty and enter in the IP address of our Raspberry Pi (previously showed in the rebooting process). The reason why we do this is because it will be easier

for us to code our project from our desired operating system (Windows or Mac) instead of having to reboot the raspberry Pi itself. After filling the information needed in Putty, we are going to get a security alert, in which we are going to select yes. That's it, we are ready to do any changes to our raspberry Pi from our Windows console. The next step is very optional, and it is used for specific reasons. If we need to enable the GUI of the raspberry operating system in our windows, we need to download VNC Viewer and Run it. After that, we select new connection and enter the credentials needed for our raspberry pi (Ip address and name). Then we hit ok and will require us to enter the credentials for username and password. Once these credentials were validated, we have access now to the GUI of our Raspberry Pi. Even though it is an optional step, it makes it easier to debug or fix any problem happening in the raspberry pi operating system. Doing thinks like enable/disable SSH is much easier to do from the GUI than from the console. Once we are done with these steps, we can upgrade our operating system to the latest and install anything we want from our console. For instance, since we want python to run in our raspberry pi, we only need to run the command needed, which will install all the dependencies needed since our raspberry pi have access to the internet, meaning that we can get or post any data to the internet since we have an endpoint (IP address) where we can go through to get our data online.

Figure 5.6: Our Microcontroller

This is the big reason why we will be implementing the programming language of Python in our whole application. OpenCV makes it easy for us to control the camera as we desire, meaning that we can set up the definition, switch a picture between colors and even recognize what color the original picture has. This will be useful for us since we will need to draw the picture close to perfect. Also, we may need to use it to recognize what parts of the drawing is missing, so that we can finish drawing our picture.

To ensure accuracy, feasibility, and other core traits about its functionality, the camera used should be able to achieve a few tasks within a certain time limit. Task such as color recognition and image clarity. These actions are going to be transformed into data using the camera, which are going to be sent to our microcontroller. Our microcontroller must also implement an algorithm using this data within time window, allowing the user the terminate the tasks within a reasonable time. The following picture represents how our camera can detect blue from and uploaded picture. Our microcontroller will be able to recognize the data gather from the camera thanks to Open CV library. It will take pictures every time our robot finishes a design. Every picture taken from the camera will be taken within a specific period. Our microcontroller will then be able to find the missing section of the original picture. By doing this, we can keep track of our progress and how much time more we need to finish the drawing. We can then display it to the user in a user-friendly way.

The following picture shows how to connect the camera model to our microcontroller by using the camera module built-in in our board. This is the best voltage performance and cost-efficient camera that we can get since it does not require an extra power supply to make our camera work. It also gives us a descent 720 pixels resolution which can be increase all the way to 1080p by changing the settings of the camera from our microcontroller. After inserting the camera like shown in the picture, we need to enable it from our console. We can do it using two methods. The easiest way is to power up our microcontroller and hook it up to a monitor. Once that is done, we can use the monitor and see the GUI ready to enable our camera module. However, this method requires a mini HMI port so that we can see the changes. If we do not have an HDMI port cable, we can easily do it from the console of our windows machine after connecting it thru our server. All we need to do is to connect to our raspberry pi server using a server connection software (in my case Putty) and then connect to the IP address and entering the correct credentials. After that is done, we just need to run a few commands to enable the camera module of our microcontroller. After that is done, we need to reboot our raspberry pi and then we will have established connection to our camera.

Figure 5.7: Our Microcontroller used with camera hooked up

The following picture is a testing product of how our camera will recognize the colors that we specify. After we are done setting up our camera module, we can have access to all the methods that we need to take the pictures from our camera. Once the pictures are being taken, we can use our raspberry pi to display the picture in different window since it has been jpg formatted. After that, we initialized our camera, we can easily compare the colors we had from the pictures taken. This is the main reason why we opted to go with a descent resolution for our camera, because we can recognize the color needed for our picture, so that we can keep track of our drawing with a much better precision. As shown

in the picture, the left side shows the original picture of circles with different colors. This picture has been taken using our camera module and sent it to our microcontroller thru pixels. After that, we can use our console from windows to open the jpg formatted file. From here, we can see what resolution we want. The default resolution is 720p, but we can easily set it up higher to 1080p. This shown is using the default resolution. The right side shows the borders of the object with a green color. In this case we are using green to recognize what objects we need from the original picture. These methods that we use to convert our picture into different colors are from OpenCV which allow us to convert the data from the original picture to useful data that we can use later. OpenCV is a library of programming functions mainly aimed at real-time computer vision. Created by Intel in 1999, it is written in C++, we will be using the Python bindings. There is a full documentation for C++ and Python. It contains many popular algorithms for computer vision, including object detection and tracking algorithms built in. For our project, we will start with the basics and slowly work our way up. To have a better understanding, our team needed to educate themselves to be able to open image files with OpenCV in both a notebook and a python script. Draw a simple geometry on images such as complex drawings of cars or circles and directly interact with an image through callbacks.

The way data is represented in our algorithm will depend on our design. We can represent a picture or each object of a picture on different arrays with values. Normally, we would use number to represent dark or light colors. We are going to represent in a scale from 0 to 1 to represent the darkness of a color. In our case, 1 will represent it is the darkest and 0 will represent it is the lightest. We would have to divide the number we get by 255 to fit within the range. Color images can be represented in red, green and blue (RGB). The reason why we need to know this is because we can represent other colors by mixing up these mentioned principal colors. There are websites that help us to find different colors by adding the right input to RGB. ion will give us the right orange (a bit darker) that we are looking for. So, when we read an image, we have 3 dimensions: height, width and color channels. This means we read an image and we have a shape of, for instance, (720, 1280, 3) which will bring us a 720-pixel height, 1280-pixel width and 3 color channels (one for each color). Keep in mind that our computer will not now if an image is red, blue or green. Now of reading, we have a black and white color. It is up to us to represent those as colors in order to combine them.

Keep in mind that we are using arrays for saving our pictures. After taking the picture, we can know what data type we are dealing with. For instance, this will be a “JpegImageFile”, which means that our imported module cannot deal with it, meaning it cannot read or work with this files format. Numpy is the module that we need to import to our programming environment to deal with arrays in python. We will have to transform our Jpeg file to an array. This will also give us a shape that we can display in our IDE. As mentioned before, the shape given will be given as (height, width, and 3 channels colors). After that, we need to zero out the values of the picture after transforming into an array format. We can

use the concept of RGB to convert the picture colors into the ones we want. For example, red channel ranging from 0 – 255, meaning 0 = no red and 255 = pure red. The picture has certain areas where red is more used versus green. We can also delete an entire channel, for instance red. This will make the picture made of blue and green (RGB) which will give us the actual picture of a purple color. Since we are using OpenCV to read the image, one big problem we got was the order of the values. OpenCV gets the RGB formatted values as (blue, green and red) instead of red then green then blue. Therefore, at the beginning, we didn't see the real colors of our original image. So, what we need to do, is to transform all our data to RGB instead of BGR. It is important to point out that if we want to transform our picture into gray version, then we will have to remove the color channels of our original picture. Keep in mind that, so far, we are using the console to visualize our picture, which can be a bit small for our own eyes or not ideal for others. We can use OpenCV to open the picture into its own windows, which will give us a better visualization of the original picture and any changes that we perform in the future.

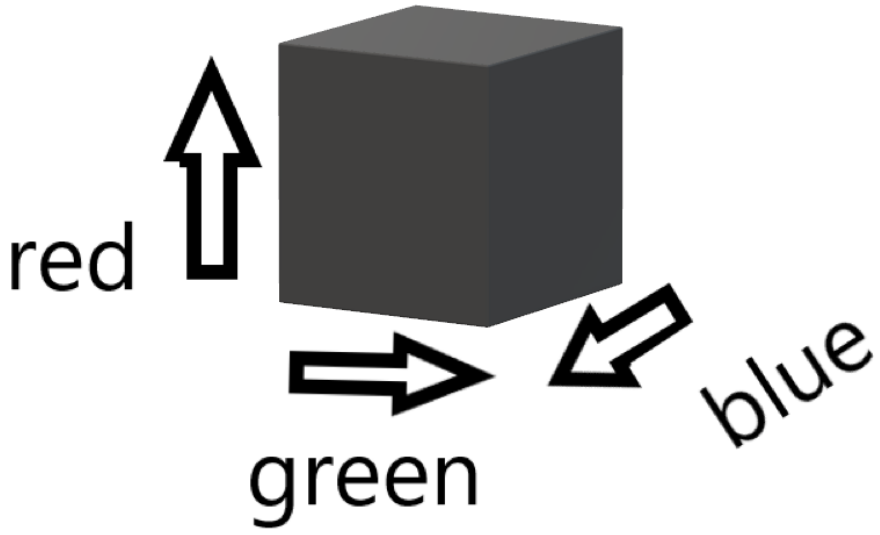


Figure 5.8: Visual representation of colors in RGB model

The picture from above shows a visual representation of colors using the RGB model. Working with RBG color spaces right now is essential since it brings us simplicity. However, RGB coding, colors are modeled as a combination of Red, Green and Blue. In the 1970s HSL (hue, saturation, lightness) and HSV (hue, saturation, value) were developed as alternatives color models. HSL and HSV are more closely aligned with the way human vision perceives color. For our project, RGB model is more than enough to bring us the best accurate color recognition, but it can also be done using these two mentioned models using OpenCV.

OpenCV is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and stat-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in vides, track camera movements, track moving objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database. It has a modular structure, which means that the package includes several shared or static libraries such as core functionality. It is a compact module defining basic data structures, including the dense multi-dimensional array Mat and basic functions used by all other modules. Image processing (imgproc) - an image processing module that includes linear and non-linear image filtering, geometrical image transformations, color space conversions, histograms, and so on. Camera calibration and 3D Reconstruction(calib3d) - basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence algorithms, and elements of 3D reconstruction. 2D Feature Framework (features2d) - salient feature detectors, descriptors, and descriptor matches. Object Detection (objdetect) - detection of objects and instances of the predefined classes. OpenCV deallocates the memory automatically, as well as automatically allocates the memory for output function parameters most of the time. So, if a function has one or more input arrays and some output arrays, the output arrays are automatically allocated or reallocated. The size and type of the output arrays are determined form the size and type of input arrays. If needed, the functions take extra parameters that help to figure out the output array properties. The array frame is automatically allocated since the video frame resolution and the bit-depth is known to the video capturing module. The array edges is automatically allocated by the cvtColor function. It has the same size and the bit-depth as the input array. The number of channels is 1 because the color conversion code cv::COLOR_BGR2GRAY is passed, which means a color to grayscale conversion. Note that frame and edges are allocated only once during the first execution of the loop body since all the next video frames have the same resolution. If you somehow change the video resolution, the arrays are automatically reallocated. The key component of this technology is the Mat::create method. It takes the desired array size and type. If the array already has the specified size and type, the method does nothing. Otherwise, it releases the previously allocated data, if any (this part involves decrementing the reference counter and comparing it with zero), and then allocates a new buffer of the required size. Most functions call the Mat::create method for each output array, and so the automatic output data allocation is implemented. Some notable exceptions from this scheme are cv::mixChannels , cv::RNG::fill, and a few other functions and methods. They are not able to allocate the output array, so you must do this in advance.

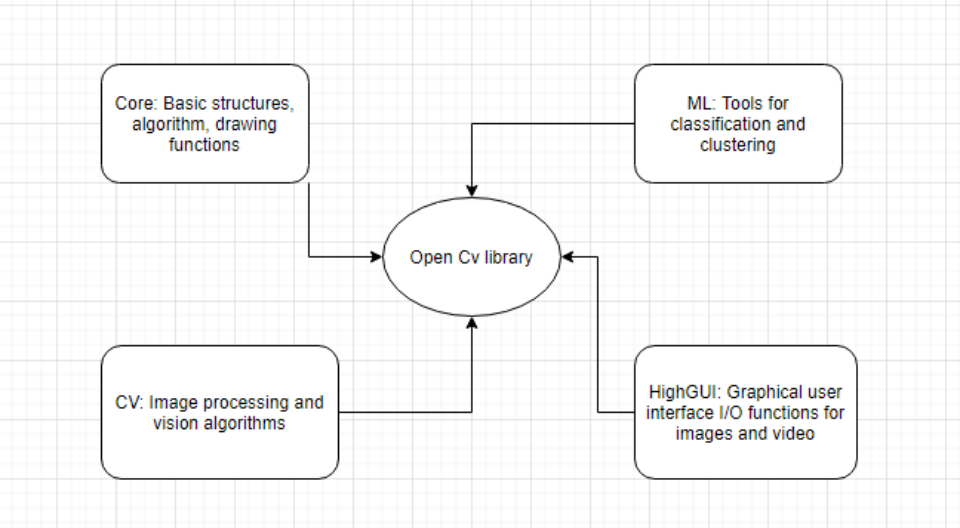


Figure 5.9: Flowchart of OpenCV and advantages

After explaining what OpenCV is and why are we using it for our project, we will proceed to describe how it would work. After setting up the environments of our raspberry pi described before, we need to import the module “PiCamera”. This module will allow us to control our camera. First, we start a preview using the start_preview() function. This turns on the camera and allows it to begin to focus. Then, we wait five seconds to give it enough time to focus by calling the method sleep(integer). This method expects an integer which will dictate how much time it will give to our camera. Then we capture the picture using capture(string). This function expects a string which will be the destination after we take the picture. After that, we run the method stop_preview(). This will allow us to close the live visualization of the picture taken. Finally, whenever we are done, we need to clean the resources of the camera. For this reason, we will be using the close() method. These are pretty much all the methods needed from the raspberry pi in relation with the camera module.

After we took our picture and saved it locally, we need to implement a function that change the format of our data. Imread(pathdestination) method is expecting a string, which will be the path to our picture in our computer and will return the same picture in an array format. Right after that, we need to use imshow(string, img) and waitKey() to show the image taken in its own separate window. This is optional, but it will help us to visualize the picture in a different place other than the console. Once we have our picture, sometimes we don’t get the actual color of the pictures. Every time this happens, we need to use ctvColor(img, CV_BGR2Luv) method. This function converts an input image from one color space to another. In case of a transformation to-from RGB color space, the order of the channels should be specified explicitly (RGB or BGR). Note that the default color format in OpenCV is often referred to as RGB but it is actually BGR (the bytes are reversed). So, the first byte in a standard(24-bit) color image will be an 8-bit Blue component, the second byte will be Green, and the third byte will be Red. The fourth, fifth,

and sixth bytes would then be the second pixel (Blue, then Green, then Red) and so on. The conventional ranges for R, G and B channel values are: 0 to 255 for cv_8u images, 0 to 65535 for cv_16U images and 0 to 1 for CV_32F images. In case of linear transformations, the range does not matter. But in case of a non-linear transformation, an input RGB image should be normalized to the proper value range to get the correct results, for example, for RGB --> L*u*v* transformations. For example, if you have a 32-bit floating-point image directly converted from an 8-bit image without any scaling, then it will have the 0 to 255 value range instead of 0 to 1 assumed by the function. So, before calling cvtColor, you need first to scale the image down. We also may need to resize our pictures so that we can paste a picture or amend a picture with another. We can use the method shape(). This function will return the tuple of the number of rows, columns, and channels (if the image is color). Remember we only need to check if our both images have the same shape before we proceed to any unnecessary process. The following step is only needed if our comparison of these two images return false. We would also need to resize the images since we don't have control of how python handles the images. We can use resize(image, x scale factor, y scale factor). This function is expecting any image and will resize the width to x and the height to y and return the new resized image. The way we are going to blend two pictures will be using addWeighted(src1, alpha, src2, beta, gamma). This function calculates the weighted sum of two arrays. Remember that we changed the format of our image at the beginning of the code. Equation is dst(I) = saturate(src1(I) * alpha + src2(I) * beta + gamma), where I is a multi-dimensional index of array elements. In case of multi-channel arrays, each channel is processed independently. For our case, src 1 will be one of our pictures, alpha will be 0.5, src2 will be our second picture and beta will be 0.5 as well and gamma will be 0. This will return both images blended on top of each other, this way we can see any difference if they are not the same. Another approach is to find out where we need to blend a picture to another picture. For this feature, we are going to need to calculate an area of interest. After we get our dimensions, we can get x and y scale of our picture by using shape(). Since we know the actual size of the picture, we can set a small area by subtracting some number to the Y and X. This area will help us to detect a smaller area from the picture. This is the area where we need to insert our picture or amend it. This new approach will help us build a picture that is almost like the original painting images. So, every time the robot arm does a movement or paint something, we are going to take a picture and amend into a bigger picture so that it looks like the original taken. After this, we have our offset for the area of interest selected of or picture. To calculate the region of interest would be the first image[y_offset:1401, x_offset: 943]. If we save this in a variable and display it, we will be able to see one of the corners (bottom right corner) if the original picture (assuming the size of the original size has sizes 1401 by 943). By doing this, we can set up a new region of interest of our big picture and adding/amending picture to it every time we take a new one from the camera.

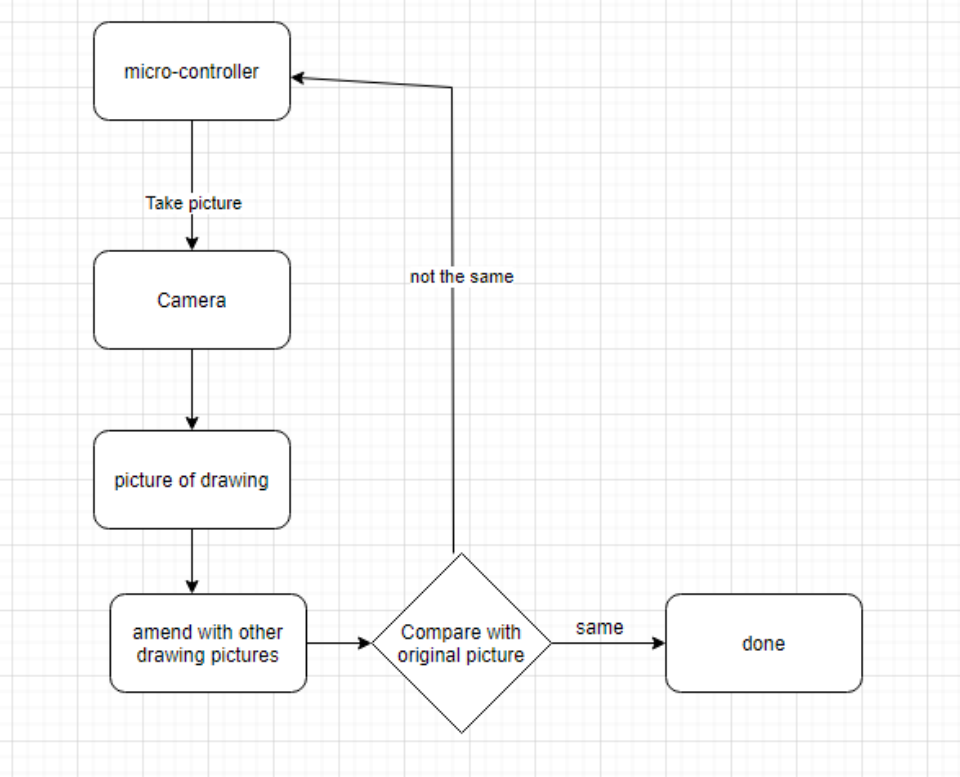


Figure 5.10: Flowchart of the process of pictures comparing

- **Power Supply**

As previously mentioned, since we are cutting a lot of resource on our economic stage, we also need a power supply within our price range. A 6V 5A DC power supply is what we need after carefully reviewing that all our components are also DC. This one has an input frequency of 50/60 Hertz with an output voltage of 12 V DC, 2 A. With a maximum wattage of 24 W, we need to be careful to not give more power than needed. The power supply that we currently have is a 9-volt 1-amp power supply. Using Voltage divider, we lower the power supply to just 5 volts and plug it into the microcontroller.

- **Board Raspberry Pi 4 Model**

Raspberry Pi Foundation is a group of people who decided that kids needed a way to learn to do more with a computer than just scroll through Facebook if they wanted to, and found a way to make it affordable so almost anyone can. At least initially, this was their plan. However, times keeps moving forward and so technology. Now, at version 4, is not a kid's toy. It’s a full-on computer with support for things like 4k video in a tiny package that’s also super-affordable. This has also fostered a huge developer community as well

as long list of after-market accessories that will let you do almost anything with Raspberry Pi 4 without spending a lot of money. Most people with a desktop computer aren't using it for any heavy lifting. After doing some research, the Raspberry Pi 4 is incredible at doing everything at once. It might share some of the same parts with your smartphone, but when it comes to the software, things couldn't be more different. If you use an Android phone, you might now that at it hear, it run atop Linux, but Linux is like a visit to the ice cream parlor and there are so many different flavors. The flavors you can install on a Raspberry Pi are the same flavors you can install on a desktop tower, laptop, or even a mainframe or server. That manes access to desktop-class web browsers, photo and video editing and processing software, and even professional audio production tools are available to install. You can even go a step further and use a Raspberry Pi as a dedicated server for web pages, an FTP host, or a home network file server. All you need to get started the same mouse, keyboard, and a monitor you would need for a more traditional desktop and a place to plug everything in. Advertisements are not a bad thing. Everyone deserves to be paid for the work they have done, and ads are a vehicle that allows it to happen. But, online adds are also a conduit for horrible things like malware and tracking. Do not hate the website that shows you an ad, instead, we should hate the horrible people who try to inject tracking cookies then sell lists with your name on them to companies who buy that sort of things. But when it comes to privacy, none of that matters. They Raspberry Pi makes for a great one that you plug into your network and protect every single device on it from adware and malware. This is free and simple to set up through one of the various applications that allow the Raspberry Pi to block the bad while letting the good flow through. Maybe the coolest thing about Raspberry Pit is how easy it is to control other devices using its standard input and output pins. Accessories like cameras or weather stations already set up to work with the Raspberry Pi, but since switching the input/output pins is so simple, the possibilities are endless. Some examples for the use of Raspberry Pi is to control a small aquarium water and lighting parameters, control the Christmas lights, and even was able to check to see if the mailbox had been opened before the advent of wireless security cameras – which the Raspberry Pi can also control. The raspberry Pi can be act as a web server for running your own websites, a desktop computer, or as a streaming device for applications like Spotify. In addition, the Pi can serve as an internet of things, (IoT) device allowing it to connect to smart devices and can even run Alexa. The most important part of the Pi is the central processing unit (CPU). The CPU is the brains of a computer, and both versions of the Raspberry Pi boast impressive processors. However, the Pi 4's processor is about 50% faster than of the Pi 3B+ an can support two monitors instead of one. These improvements to the Raspberry Pi 4's CPU make it much more versatile than he Pi3, allowing it to perform faster in most situations. Other features allowing the boards facilitate the creation of projects are the 40 GPIO pins, which can be connected to other parts like servo motors, temperature sensors, and distance sensors. In addition, both boards have Bluetooth and Ethernet capabilities allowing for easy connection to other devices like speakers and WIFI networks. The ability for both boards to be programmed with the Python programming language make writing programs for the boards quick. While both boards have similar

features, the Pi 4 is more versatile in its capabilities. In addition to the faster CPU, the Pi 4 has both USB 2.0 and USB 3.0 ports while the Pi3 only supports USB 2.0. The wide array of features allows the Pi to be used in a plethora of technological projects. While the Pi 4 seems like the perfect tool for many small-scale developers, there are a few drawbacks. The relatively recent release of the board means that many software applications commonly used on programming, improving user interfaces, or providing entertainments. One of the applications RetroPie is not compatible with the Raspberry Pi 4. While the Pi 4 has some pre-downloaded software included with the operating system, the scope of what is available for the Pi 3B+ online is far greater. Additionally, since the Pi 4 is so new, people haven't had the chance to fully document the board's capabilities or how some programs work. Without full documentation/tutorials on the Pi 4's functionalities, using the Raspberry Pi 4 could be much trickier than the previous boards. Thoe to this, some individuals might decide it is better to wait for some documentation and applications to support the Pi 4 before obtaining one to experiment with. Having recently purchases and set up a Raspberry Pi 4, I can state the the Pi was easy to set up and is a powerful tool for self-made projects. The operating system was easy to install because of detailed instructions provided by the Raspberry Pi Foundation. Starting out with the Raspberry Pi 4 does not require any previous knowledge of technology or circuitry but having experience with it does help. Online tutorials for Raspberry Pi boards and the Python programming languages are available online, and online forums answered any questions I had while making this project. For my project, my team and I are in the building process, so this will help us with any doubt that we have. To sum up, both Raspberry Pi boards offer similar functionality in many respects, and that is what makes the choice of it the new edition is worth is so challenging to decide. Is it better to experiment with the new board, despite the lack of documentation now, or is it better to use the old board which has less functionality? It is my opinion that clear choice is the Raspberry Pri 4 because of its superior hardware and performance. The lack of online resources for the board will be fixed as more software is updated to support Pi 4 and as more users become familiar with the product.

As previously mentioned, our servos and camera module will receive/send signals from our microcontroller. We chose raspberry pi since it offers physical pins for us and a compatible camera module, so that we can test almost any camera in the market. It is also inside our price range. With dimensions of 3.7 x 3 x 1.1 inches, it can be placed anywhere close to our robot arm. Also, if at some point it needs to be attached to our robot arm, its weight of 2.4 ounces will not be a problem. It supports various interface options, in which PWM channels and SPI are enabled. These two interfaces are needed to control the servo motors and our camera module. It also allows the implementation of Bluetooth, which can be helpful in the future if we need to establish a communication with mobile devices.

To use python on our board, we will need an SD card with Raspbian installed. Once we boot our system, we can install Python, so we can have access to OpenCV library. With an input voltage of 5 volts DC and maximum of 6 volts, it is the best fit within the range of

our robot arm. With a maximum functional temperature of 80 degrees Celsius, so it meets our durability requirements. To reduce thermal output when idling or under light load, the Pi4B reduces the CPU clock speed and voltage. During heavier load the speed and voltage are increased. The internal governor will throttle back both the CPU speed and voltage to make sure the CPU temperature never exceeds 85 degrees Celsius. The Pi4B will operate perfectly well without any extra cooling and is designed for sprint performance – expecting a light use case on average and ramping up the CPU speed when needed. If a user wishes to load the system continually or operate it at a high temperature at full performance, further cooling may be needed.

Its GPIO interface offers us 40 pins for us to control our servos. As well as being able to be used as straightforward software-controlled input and output, GPIO pins can be switched into various other modes backed by dedicated peripheral blocks such as I2C, UART and SPI. The following table will help us in details the default pin pull state and available alternate GPIO functions. Most of these alternate peripheral functions are described in detail in then BCM2711 Peripherals Specifications document which can be downloaded from the hardware documentation section of the website.

Table C of Raspberry Pi 4 Functions

GPIO	Default Pull	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5
0	High	SDA0	SA5	PCLK	SPI3_CE0_N	TXD2	SD6
1	High	SCL0	SA4	DE	SPI3_MISO	RXD2	SCL6
2	High	SDA1	SA3	LCD_VSYNC	SPI3_MOSI	CTS2	SDA3
3	High	SCL1	SA2	LCD_HSYNC	SPI3_SCLK	RTS2	SCL3
4	High	GPCK0	SA1	DPI_D0	SPI4_CE0_N	TXD3	SDA3
5	High	GPCK1	SA0	DPI_D1	SPI4_MISO	RXD3	SCL3
6	High	GPCK2	SO3_N	DPI_D2	SPI4_MOSI	CTS3	SDA4
7	High	SPI0_CE1_N	SWE_N	DPI_D3	SPI4_SCLK	RTS3	SCL4
8	High	SPI0_CE0_N	SD0	DPI_D4	-	TXD4	SDA4
9	Low	SPI0_MISO	SD1	DPI_D5	-	RXD4	SCL4
10	Low	SPI0_MOSI	SD2	DPI_D6	-	CTS4	SDA5
11	Low	SPI0_SCLK	SD3	DPI_D7	-	RTS4	SCL5
12	Low	PWM0	SD4	DPI_D8	SPI5_CE0_N	TXD5	SDA5
13	Low	PWM1	SD5	DPI_D9	SPI5_MISO	RXD5	SCL5
14	Low	TXD0	SD6	DPI_D10	SPI5_MOSI	CTS6	TXD1
15	Low	RXD0	SD7	DPI_D11	SPI5_SCLK	RTS5	RXD1
16	Low	FL0	SD8	DPI_D12	CTS0	SP11_CE2_N	CTS1
17	Low	FL1	SD9	DPI_D13	RTS0	SP11_CE1_N	RTS1
18	Low	PCM_CLK	SD10	DPI_D14	SPI6_CEO_N	SP11_CE0_N	PWM0
19	Low	PCF_FS	SD11	DPI_D15	SPI6_MISO	SP11_MISO	PWM1
20	Low	PCM_FS	SD12	DPI_D16	SPI6_MOSI	SP1_MOSI	GPCLK0
21	Low	PCM_DIN	SD13	DPI_D17	SPI16_SCLK	SP11_SCLK	GPCLK1
22	Low	PCM_DOUT	SD14	DPI_D18	SD1_CLK	ARM_TRST	SDA6
23	Low	SD0_CLK	SD15	DPI_D19	SD1_CMD	ARM_RTCK	SCL6
24	Low	SD0_CND	SD16	DPI_D20	SD1_DAT0	ARM_TDO	SP13_CE1_N
25	Low	SD0_DAT0	SD17	DPI_D21	SD1_DAT1	ARM_TCK	SPI4_CE1_N
26	Low	SD0_DAT1	TE0	DPI_D22	SD1_DAT2	ARM_TDI	SP15_CE1_N
27	Low	SD0_DAT2	TE1	DPI_D23	SD1_DAT3	ARM_TMS	SP16_CE_N

Figure 5.10

The picture from above it probably one of the best things about any Raspberry Pi, including Raspberry Pi 4. We can use it to build all kinds of awesome contraption, from robots to retro gaming consoles and fart detectors. Most of the sensors, motors, lights and other peripherals that make these projects possible connect to the Pi's set of GPIO pins. Every Pi model since the Raspberry Pi B+ has had 40 GPIO pins, though on the Pi Zero and Zero W, you have 40 holes that you can solder pins or wires into. No matter what we are building, in general, we need to know the Raspberry Pi GPIO pinout, the map and explanation of what each pin can do. While some pins provide electricity, others are grounds and still others connect to different kinds of interfaces, all of which will be explained. The GPIO is the most basic, yet accessible aspect of the Raspberry Pi. GPIO pins re digital which means they can have two states, off or on. They can have a direction to receive or send current (input, output respectively) and we can control the state and direction of the pins using programming languages such as Python, JavaScript, node-RED etc. The operating voltage of the GPIO pins is 3.3 voltage with a maximum current draw of 16mA. This means that we can safely power one or two LEDs (Light Emitting Diodes) from a single GPIO pin, via a resistor. But for anything requiring more current, a DC motor for example, we will need to user external components to ensure that we do not damage the GPIO. Controlling the GPIO pin with Python is accomplished by the first importing a library of pre-written code. The most common library is RPI.GPIO and it has been used to create thousands of projects since the early days of the Raspberry. In more recent times a new library called GPIO Zero has been introduced, offering an easier entry for those new to Python and basic electronics. Both libraries come pre-installed with the Raspbian operating system. GPIO pins has multiple names; the first most obvious reference is their physical location on the GPIO. Starting at the top left of the GPIO, and by that we mean the pin nearest to where the micro SD card is inserted, we have physical pin1 which provides 3v3 power. To the right of that pin is physical pin 2 which provides 5b power. The pin numbers then increase as we move down each column, with pin 1 going to pin 3, 5,7, until we reach pin 39. You will quickly see that each pin from 1 to 39 in this column follows an odd number sequence. And for the column starting with pin 2 it will go 4,6,8, until it reaches 40. Following an even number sequence. Physical pin numbering is the most basic way to locate a pin, but many of the tutorials written for the Raspberry Pi follow a different numbering sequence. Broadcom (BCM) pin numbering seems to be chaotic to the average user. With GPIO 14, 22 and 27 following on from each other with little thought to logical numbering. The BCM pin mapping refers to the GPIO pins that have been directly connected to the System on a Chip (SoC) of the Raspberry Pi. We have direct links to the brain of our Pi to connect sensors and components for use in our projects. There are lots of tutorials using this reference and that is because it is the officially supported pin numbering scheme from the Raspberry Pi tutorials using this reference and that is because it is the officially supported pin numbering scheme from the Raspberry Pi Foundation. Certain GPIO pins also have alternate functions that allow them to interface with different kinds of devices that use the I2c, SPI or UART protocols. For example, GPIO3 and GPIO4 are also SDA and SCL I@C pins used to connect devices using the I2C protocol. To use these pins with these protocols we need to enable the

interfaces using the Raspberry Pi Configuration application found in the Raspbian OS, Preference menu. Ground is commonly referred to as GND, gnd or – but they all mean the same thing. GND is where all voltages can be measured from and it also completes and electrical circuit. It is our zero point and by connecting a component, such as a servo motor to a power source and ground the component becomes part of the circuit and current will flow through the servo motor and produce movement. When building circuit, it is always to make your ground connections first before applying any power as it is always wise to make your ground connections first before applying any power as it will prevent any issues with sensitive components. The Raspberry Pi has eight ground connection along the GPIO and each of these ground pins connects to one single ground connection. So, the choice of which ground pin to use is determined by personal preference, or convenience when connecting components. The 5v pins give direct access to the 5v supply coming from your mains adaptor, less power than used by the Raspberry Pi itself. A Pi can be powered directly from these pins, and it can also power other 5v devices. When using pins directly, we need to check our voltage before making a connection because then bypass any safety features, such as the voltage regulators and fuse which are there to protect your Pi. Bypass these with a higher voltage and you could render your Pi inoperable. The 3v pin is there to offer a stable 3.3voltage supply to power components and to test the servo motors. It will be rare that you factor this pin into a build, but it does have a special use. When connecting a servo motor to the GPIO, we first need to make sure that the servo motor is wired up correctly and that it does any relevant movement. By connecting the long leg of the servo motor, the anode to the 3.3v pin via resistor, and the shorter leg, the cathode to any of the GROUND (gnd) pins we can check that our servo motors are moving and is working. This eliminates a hardware fault form the project and enable us to start building our project with confidence.

The Camera and display interface (MIPI CSI connector). These connectors are backwards compatible with legacy Raspberry Pi boards and support all

One of the mains reasons why we chose this board was because of its 64-bit Soc at 1.5 Gigahertz. Also 2.4 GHz and 5.0 GHz IEEE 802, which gives us enough power for such a cheap price. The name of the processor is ARM Cortex-A72 MPCore Processor which level of an asserted signal depends on whether the signal is active-HIGH or active-LOW. In simple terms, the A72 is faster, more efficient and smaller version of the cortex A57 (previous model). Its combined of 40% to 60% on power reduction across multiple workloads in comparison to previous models will help in a better sustain, maximum frequency performance and lower cost designs.

- High for active-HIGH signals
- Low for active-LOW signals

It is a high-performance, low-power processor that implements the ARMv8-A 64-bit instruction architecture. Also, its components such as instruction fetch, which fetches instructions from L1 instruction cache and delivers up to three instruction per cycle to the instruction decode unit. It supports dynamic and static prediction. It has a 48-entryu full

associative L1 instruction Translation Lookaside Buffer (TLB) with native support for 4KB, 64KB, and 1MB page sizes. Other component that really caught our attention was integer execute which includes: two arithmetic logical unit pipelines, integer multiply-accumulate and ALU pipelines, iterative integer divides hardware, branch and instruction condition codes resolution logic.

The advantages of the new microcontroller we decided to go with is that it has an increase mobile SoC performance that reduces power. Compared to other microcontrollers the Cortex-A57 has a 2GHz max, 2.2GHZ max, and 1.3 GHz equivalent performance using only 28nm, It uses less than 50% energy at ISO-process. It also uses 75% less energy at target process. This performance compared to the Cortex A72 which uses 2 GHz, 2.5GHz, and 1.1GHz, awhile Cortex- A15 uses 1.6 GHz. the energy consumed for the same workloads var y on a large scale one using less energy than the other. It makes more sense to use the microcontroller that's going to saves us energy.

6. Project Prototype Construction and Coding

The intent of prototyping is to create a real-life model of the system to be tested upon for functionality. These tests would be conducted on the hardware as well as the software portion of this project. Regarding testing, it is imperative in a system design to be able to validate that all the separate section and subsections are working to their expected standard. If a single section is not working properly, it is also vital that someone can identify the problem. In order to have a cohesive system the integration and testing must be established clearly and be administered thoroughly.

Every component must be tested respectively in order to determine if each part is working properly. We should also have an order of relevance when it comes to testing. Smaller components such as resistors and capacitors or diodes should be tested before anything. These components have a bigger room for failure, meaning if we are able to verify that these small components work perfectly, we would save time when testing our design after being completely assembled.

In our design, hardware testing would be much more relevant than software testing. Our code will be much easier to test once we have our robot functioning as we expected. Our code will only need to send electrical signals, which are going to be done by our hardware components. Finally, when every subsystem is considered then the system will be tested and that ultimately leads to the end of the product. The actual integration of our system is more in the scope of designing rather than quality checking. The challenge of integration is the last step in obtaining a finalized product. Connecting each subsystem brings the questions of which communication protocol to use as well as how power will be distributed. These problems will be described better in more of the following sections.

6.1 Integrated Schematics

Early concepts of integrated circuits date back to 1949 and have since developed at a rapid rate. Integrated circuits are the foundation of all modern electronics. Also, as the world has fully transitioned from the analog electronics to digital electronics, more and more of technological inventions are based off on integrated circuits. Visiting a best buy show room now is just a friendly reminder of how integrated circuits are the foundation to close to 100% of all technological inventions. Some of the famous integrated circuits known in the world of electrical and computer engineering are the microcontrollers and microprocessors. They are the bedrock of all technological projects in the engineering field. Microprocessors and microcontrollers are also the largest integrated circuits on the market. Thankfully, the University of Central Florida has fully equipped students on the utilization of microcontrollers and microprocessors. Therefore, it was easy for our team to decide on including a microcontroller to better facilitate the project. In this project a microcontroller is utilized to achieve the functionality of the robot painting arm. To begin, a power supply of 6V is connected to dc converter. The reason for using a DC converter is to bring up or bring down the voltage supplied to the robot. In this case, since the operation of robot painting arm might require more power thereby an increase in the power supply it is important to have the dc converter. For example, the maximum voltage the microcontroller can sustain is 10V. The dc converter is then connected to microcontroller sending 10mA of current to the microcontroller. The microcontroller serves as the control center or the brain for the entire project. It is responsible for sending instructions to the servo motors. The servo motors are responsible for the movement of the robot painting arm thereby responsible for the execution of the task of painting the canvas. The images from the camera capture the desired input of the user of the robot. For example, if the camera is made to face a landscape painting, this serves as the input for the robot arm. The is analyzed in terms of the dimension, the color and other characteristics. This image is then converted into arrays and this data is used as the foundation for the output on the canvas.

Integration in short is how to connect datelines and power lines across systems when it comes to electrical components by making sure our components meet our safety requirements. In the software sense, the program integration is more of a transfer of information between the actions done by certain electrical components flowing towards our chip, and vice versa.

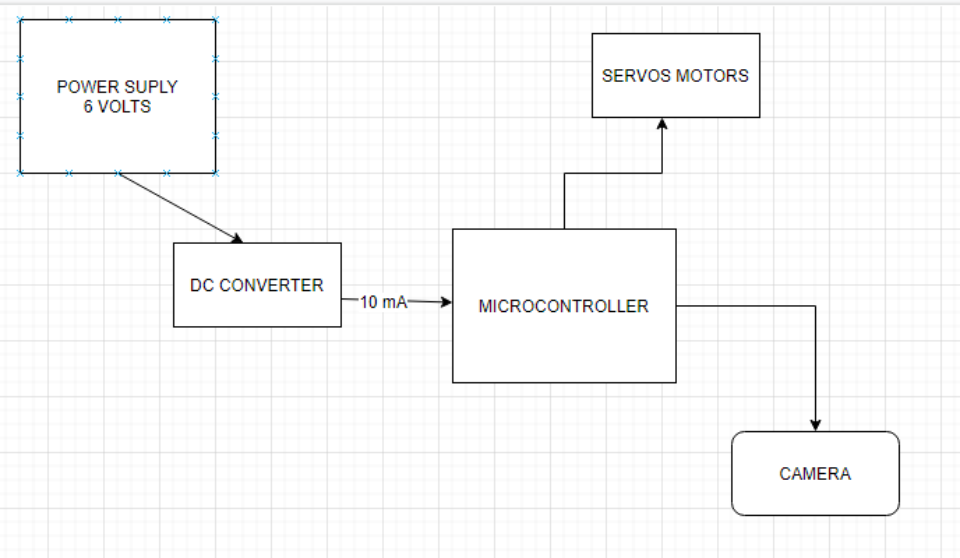


Figure 6.1: Basic Schematic

Figure Theoretical Schematic

We designed a basic diagram that will help us understand the logic flow of each of our components in our entire project. This way, we can have a better understanding of the basic's functionality of our project. This also helped us develop a much more organized representation of the more complex problems that we are going to have in the future with some specifications. Whenever we are going deeper in the development of the project, we still need to respect the basic functionality of this diagram. Every progress that our team makes needs to have a basic understanding of our basic schematic. In our block diagram above, you can see how the data path ultimately ends in our camera module and sensors. The integration can be broken down simply as having two analog front ends that communicate with each other. Each specific system will be analyzed by using one controller. It was possible to communicate with our servo sensor and the camera module by using only one programming language. By doing this way, we were able to reduce our scope into a single microcontroller. Our microcontroller will be using asynchronous system, meaning it is be powerful enough to send signals to each servo motor and at the same time read/sent data from the camera module. Our power supply is capable enough to provide power to all our components every time they require to pull up energy. This basic schematic was also of help when developing the project in real life. It tells us which are the most important components in our project.

6.2.1 Arduino & Pulse Width Modulation

For the moment we are conducting tests on the servos using pulse width modulation from the Arduino Mega 2560 microcontroller.

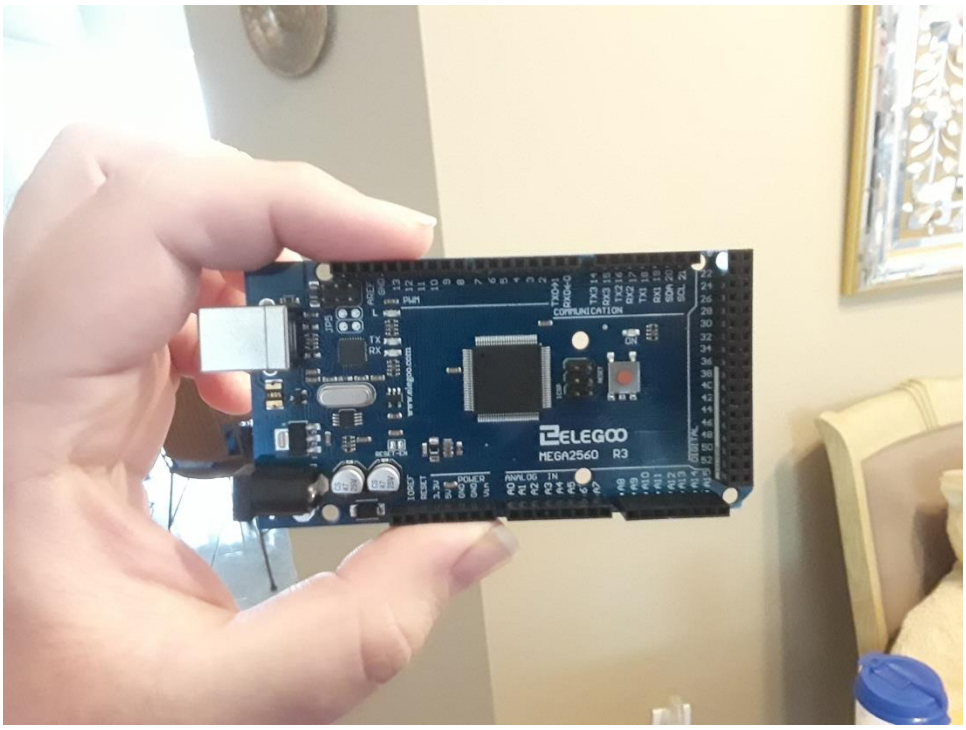


Figure 6.2 Picture of the Arduino mega 2560

It is easier to control the pulse width by changing code in C++ then it is to control the pulse width by manually turning potentiometers to alter the voltage being sent to the servo with a pulse width modulation chip. Since we have 6 servos in our Mr. Painter Robot Arm it is simply impractical to do it manually. A microcontroller is the best option.

In retrospect I should have gotten the Arduino uno since I could them pop out the microcontroller which has by then been programed and solder it onto a PCB board. Whether I do that in the future or keep it is yet to be seen. It would add a lot of complexity to the PCBboard design which is right now being focused delivering power to maximize the torque of the servos

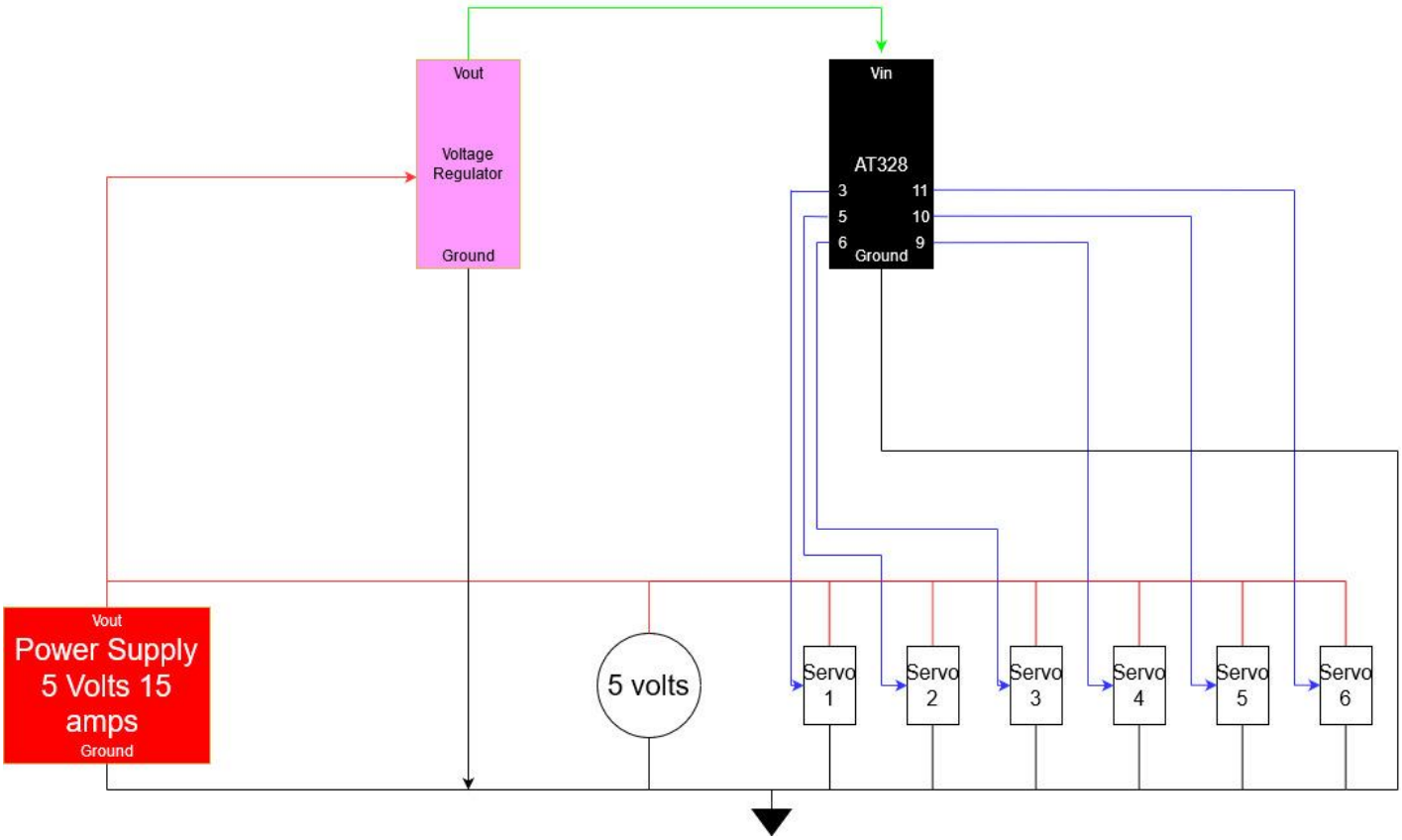


Figure 6.3 Arduino Mega 328P Schematic rough draft

If I were to use a preloaded (from the Arduino) ATMega 328P on the PCB this would be a rough draft of the schematic. Black is the ground node. Red is the voltage node. Blue are the signal pins which are connected to the AT328.

6.2.2Servo Test

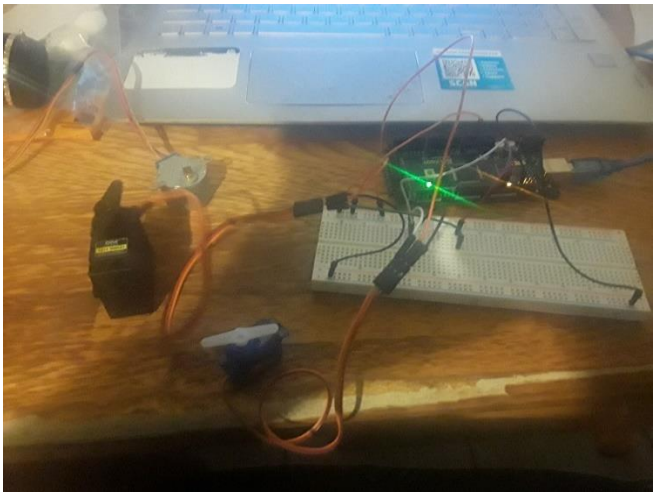


Figure 6.4 Alan testing the servo motors. From left to right, mg996r sg90

Every servo must first be tested to ensure that all of them are working correctly. On Amazon many people complained that some of the servos that came in the set did not work. This has yet to be done though. As of right now my first test was with the sg90 micro servo. This was the original servo in our original robotic arm. Everything worked as intended. The test was done with the Arduino Mega 2560. I then moved to the mg996r servo, simply pulled out the wires from the sg90 and placed them in the corresponding places and it too worked. I then changed the code to allow for 2 servos to be programmed once and the results startled me. I could only get one of them to work. This was my code at the time.

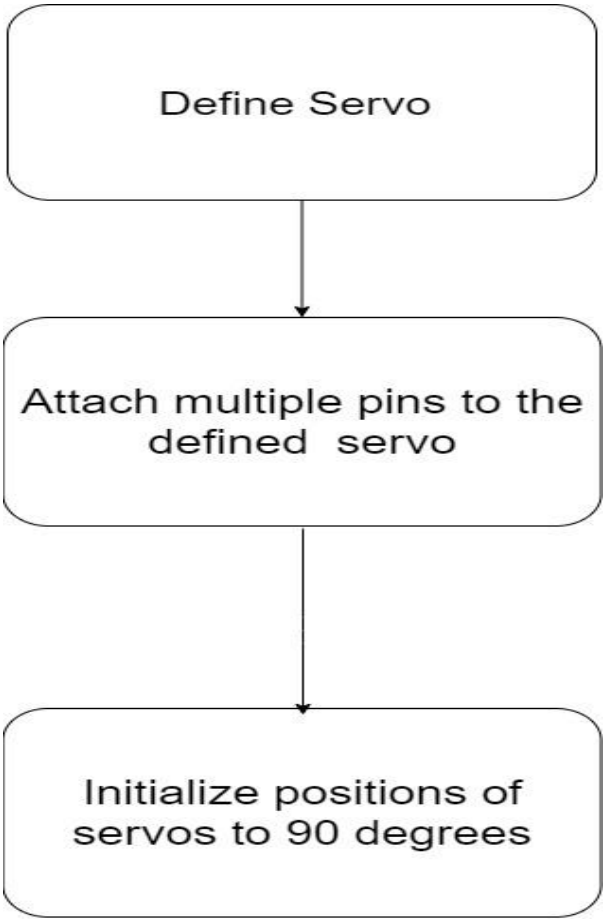


Figure 6.5 Initial code used for working with servos

I checked my wiring and it was fine. I sent the signal wire into either servo and nothing changed. It was only when I unplugged one servo did things begin to work. I didn't have a multimeter so at first I didn't see what was wrong (I still don't, and will soon be purchasing one) , so I began to think about it and I came to the conclusion that I didn't have enough power. I thought to myself how could this be, shouldn't the Arduino be powerful enough to handle 2 servos? Both had 5 volts dc in parallel, but then it occurred to me that volts is not power. Voltage is just voltage. Power is the product of voltage and current.

I looked up online the max draw current for the Arduino Mega and according to Arduino webpage the max draw current is 20mA. 5 volts with 20 mA is equal to 0.1 watts of power. There is no way I can run the robot arm on the Arduino like this. Now I have a power supply module for my breadboard that when I plug in my 9 volts an amp power supply can output 5 volts max at 700mA. Which is equal to 3.5 watts. Now the question is will this be enough to power all 5 servos. Further tests need to be made in order to verify this.

6.2.3 Servo Max Theoretical Schematic, Possible future PCB design

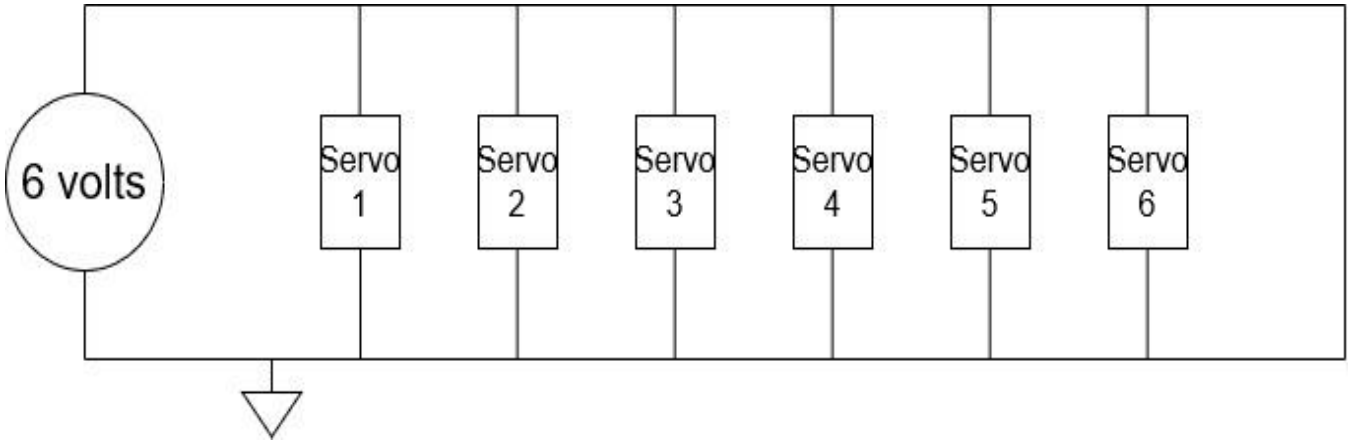


Figure 6.6 Schematic for maximum servo power designed by Alan

According to the datasheet of the mg996 the maximum torque, 11 kg/cm is achieved at 6 volts. The stall torque for the servo motor is 2.5 amps. Now there are 6 servos in the robot arm. In order to get the maximum power in a parallel configuration seen above, we would need a power supply that outputs 6 volts at 15 amps. 6 volts at 15 amps gives us 90 Watts. After looking at power supplies on amazon I was able to fine 5 volts at 15 amps. Even though its max is 5 volts at 15 apps, which equals 75 amps, it was recommended by the manufacturer to go no further then 60 watts for typical use. Economically, I don't think it is worth it to try to to 6 volts, when 5 volts should be enough if this circuit is to be used.

6.2.5 Testing the servos individually and then all at once

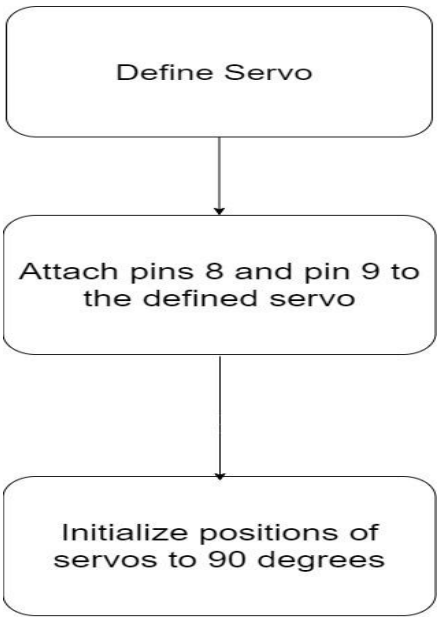


Figure 6.7 First Code Diagram of Wrong Code Flowchart when trying to test the servos.

Later on, I went back to the breadboard and checked out all 6 of the MG996R servos. Everyone worked just fine. I checked my code and realized I made an error. The problem wasn't power as I initially thought. The problem was my code. I didn't know that you could only assign one pin to one servo. My initial code was assigning multiple pins to the same servo which in the end canceled out the former pin assignment. In the beginning I only tried to use two servos at once, this is why I was only using pin 8 and 9. Above is a diagram of my wrong code as seen in figure 6.8.

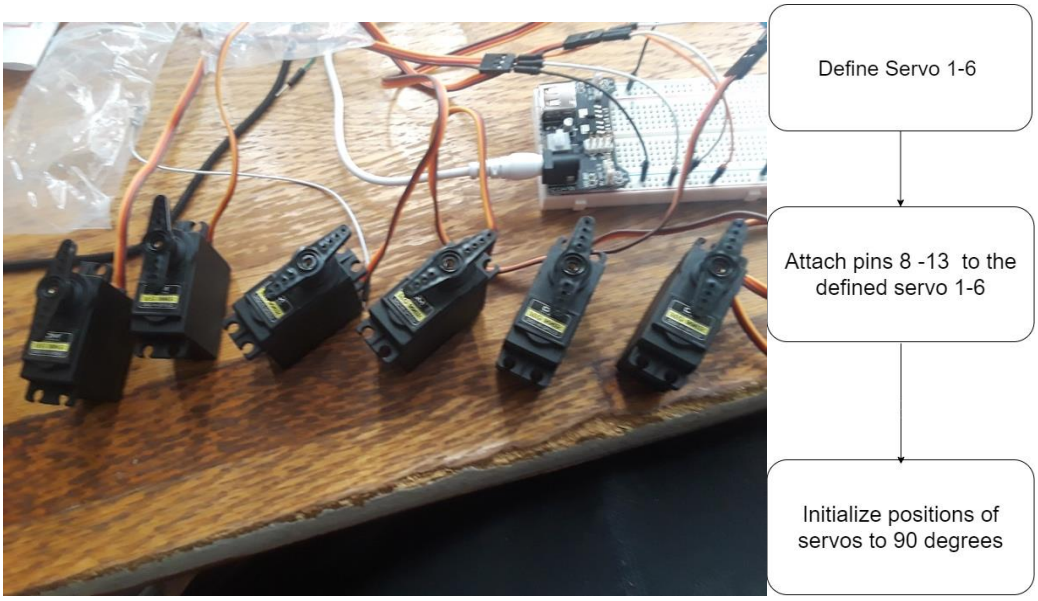


Figure 6.8 all 6 servos working at once and Second Code diagram with Corrected Code diagram

When I defined more than one servo and assigned one pin to one servo I was able to have the Arduino use more than one servo at a time. I tested this by cranking all 6 servos to a random position and then telling the Arduino to reset each position to 90 degrees. The figure above shows all six servos working at once, running on a 9 volt 1 amp power supply.

6.2.6 Constructing the root arm

Before Constructing the robot arm, I initialized all the robot servos in Arduino. This is what the code did as seen in the code flowchart in figure below.

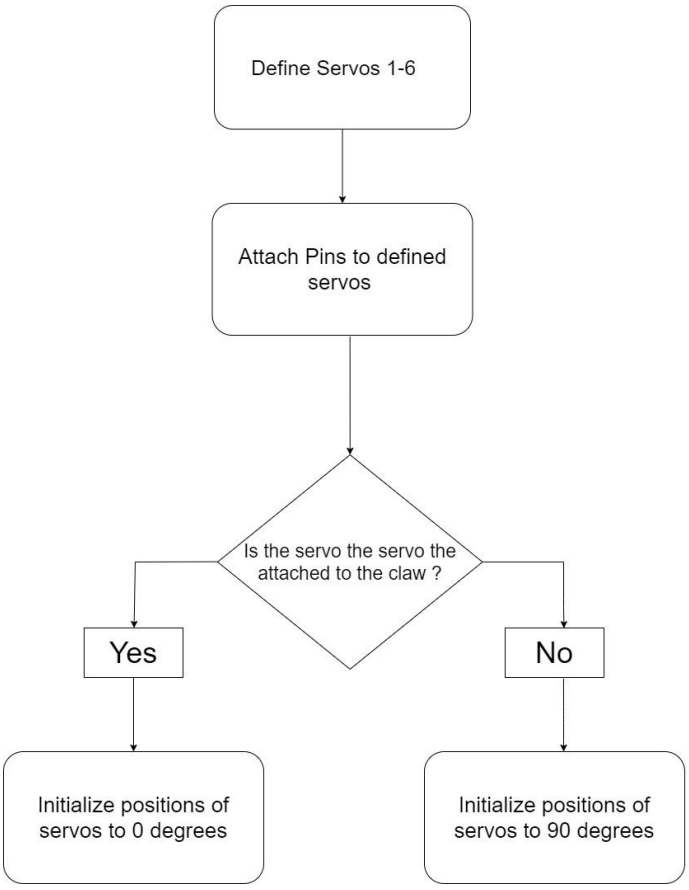


Figure 6.9 Third Code Diagram Servo initialization

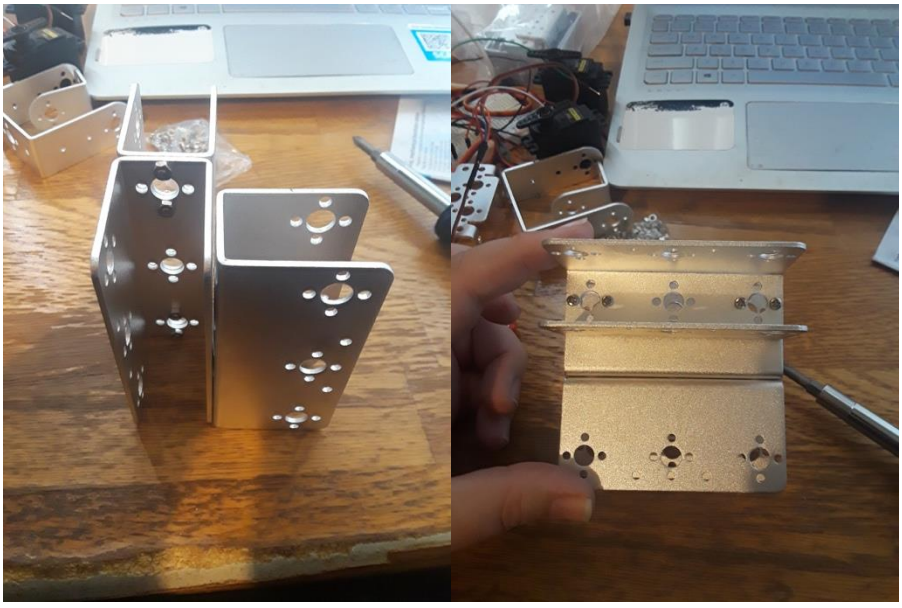
The reason why the servo attached to the claw is initialized to zero and not initialize to 90 degrees is because we don't need upper degrees for the claw. The servos set to 90 degrees are all responsible to providing the robot with degrees of freedom. In order to maximize the range of motion for the robot all those servos were set to the middle position or 90 degrees out of 180.

Figure 6.10

First, we take two long U brackets as seen in figure 6.10 above. This will be the base of the metal base of the robot arm. The metal seen in the picture is aluminum and very lightweight. This helps in general with all the other servos by minimizing the load due to weight. All the servos will be running under load since all the robot parts do have weight to them. However, as a base this is terrible because the base is so light it doesn't stop the robot arm from tipping over. The robot arm tips over because its center of gravity is not in the middle but elsewhere away from the base. Many times, when testing the robot arm, I had to hold the base or grab the robot arm from tipping over. The robot base does however have holes which screws can be used to bolt it to a heavier base. This is yet to be done though.

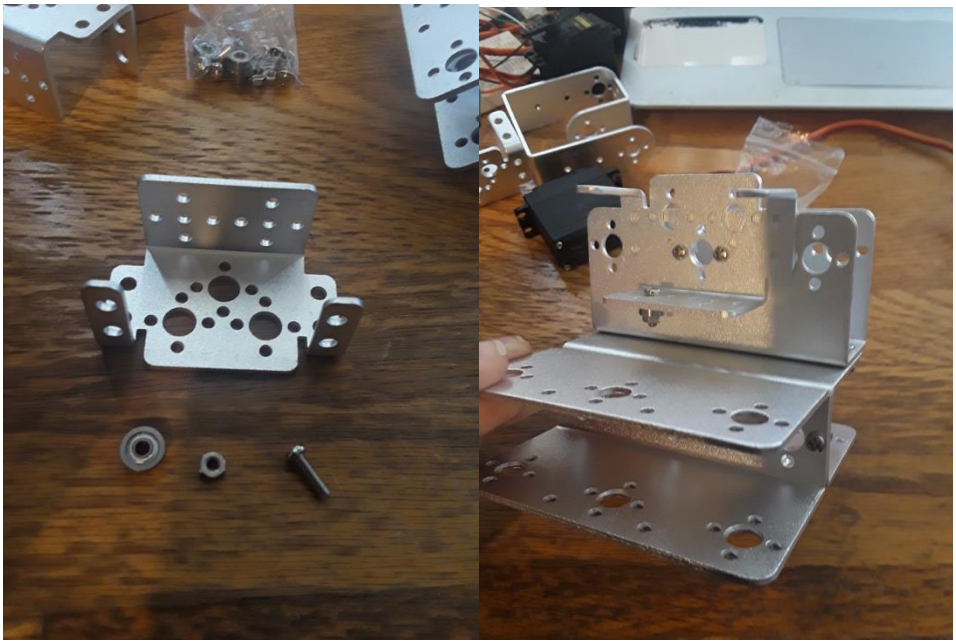
Figures 6.11

Then we fasten them together using 4 screws and 4 bolts as seen in figures 6.11 above. The screws and bolts are M3*8.



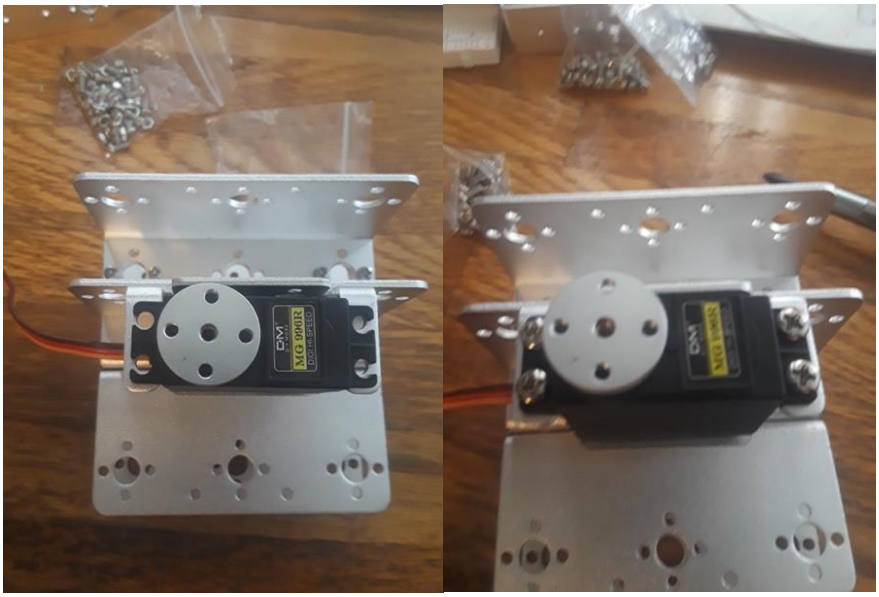
Figures 6.12

Now we take another bracket and fasten it onto the horizontal brackets forming a L shape as seen in figure 6.12 above.



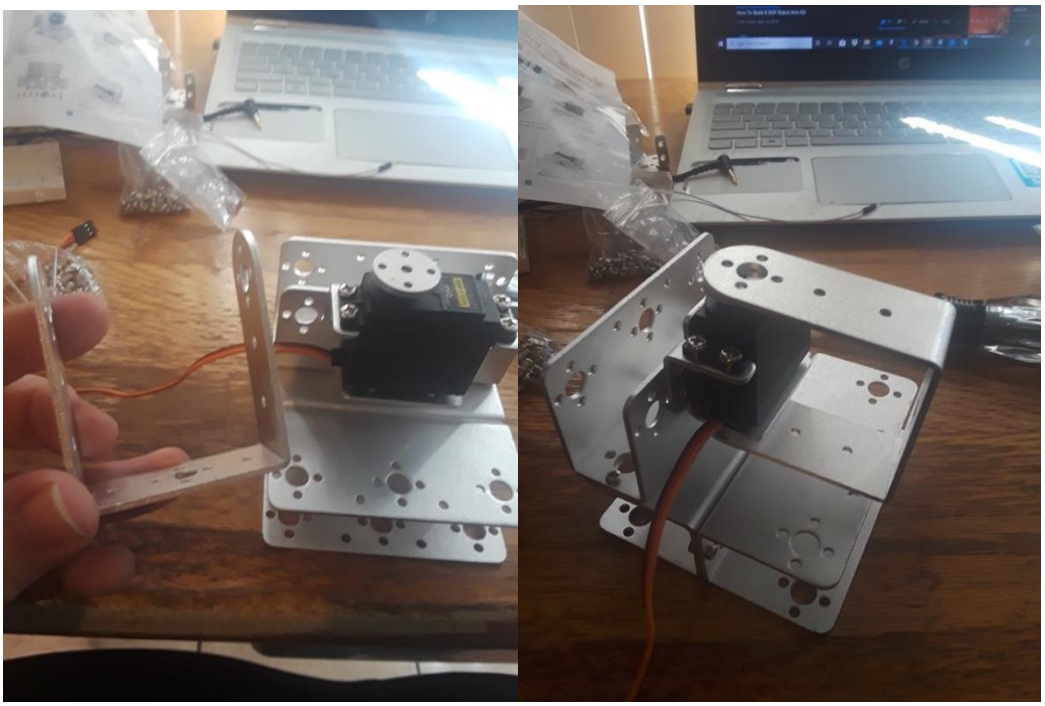
Figures 6.13

Now we take a M3*10 screw and bolt and attach it to the mutli-functional servo bracket on the bottom as seen in figure 6.13.This is important because this grove created by the M3*10 screw allows the bracket attached to the servo motor to rotate. Then we attach the multifunctional servo bracket to the long U type brackets using two M3*8 screws and bolts as seen in figure 6.13 above. The M3*10 screw is seen at the bottom of the servo bracket in figure 6.13 above.



Figures 6.14

Now we take our MG996R servo and attach our aluminum servo horn on top. Then we slide it into the servo bracket and fasten it in with 4 M4*10 screws and M4 nuts as seen in figures 6.14 above.



Figures 6.15

Now we take a U bracket and place it over the servo horn. This servo has just provided us with our first degree of freedom. This servo is now in charge of horizonal rotation and rotates horizontally from 0 to 180 degrees. When I installed this servo, I made sure to test if the servo was correctly installed to insure, we had the full range of motion that was due. If incorrectly installed we would get less than the full range of motion as the u bracket would collide with the l bracket and just was power while being under max load (the reason why it is at max load is because these servos are no ware need powerful enough to overcome and shift the aluminum blocking their path). This when running the robot arm would be awful because all the servos are connected in parallel, and if one of the robot arms is at max load, it will suck up all the current for itself, leaving the other servos underpowered. When testing the servos, I used to different methods with the Arduino, and each one had its own code.

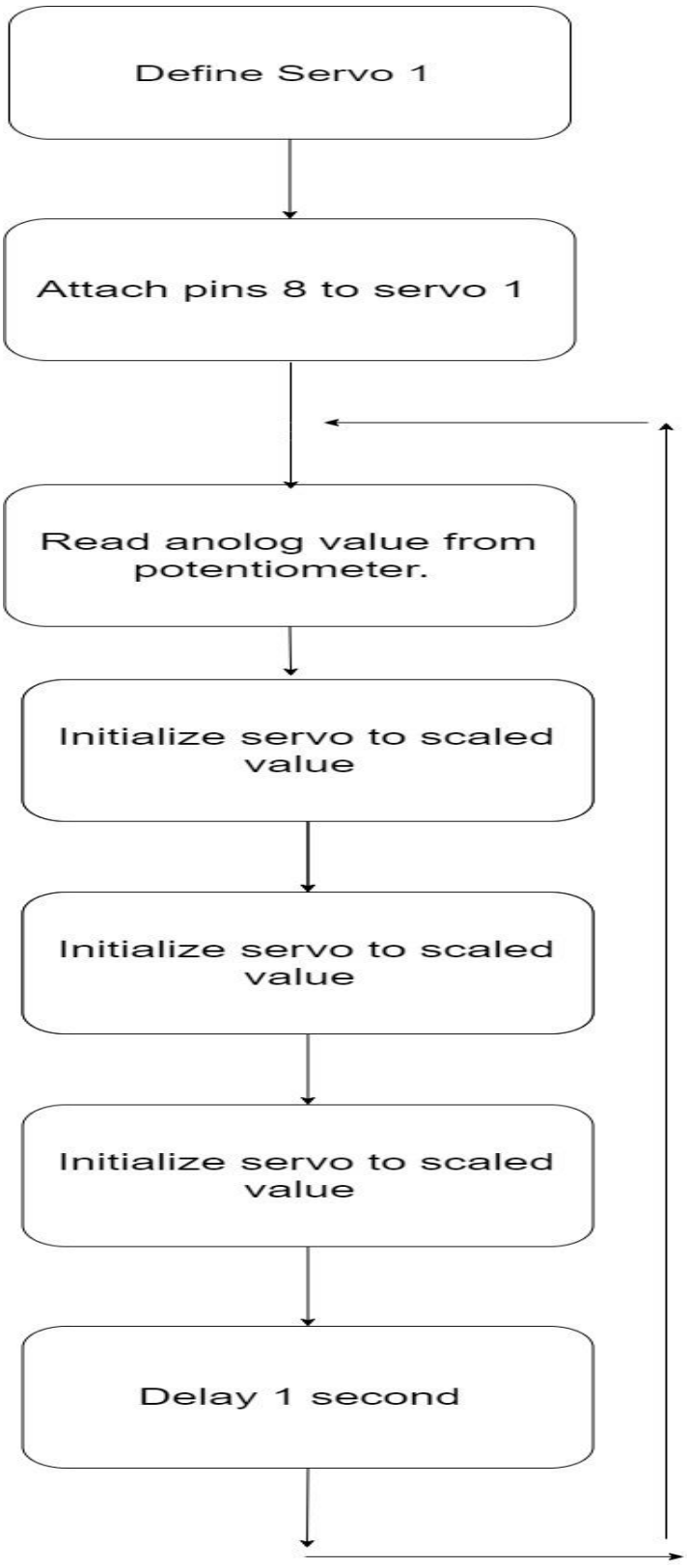


Figure 6.16 Fourth Code Diagram Initial Code Flowchart for testing range of motion of installed servos

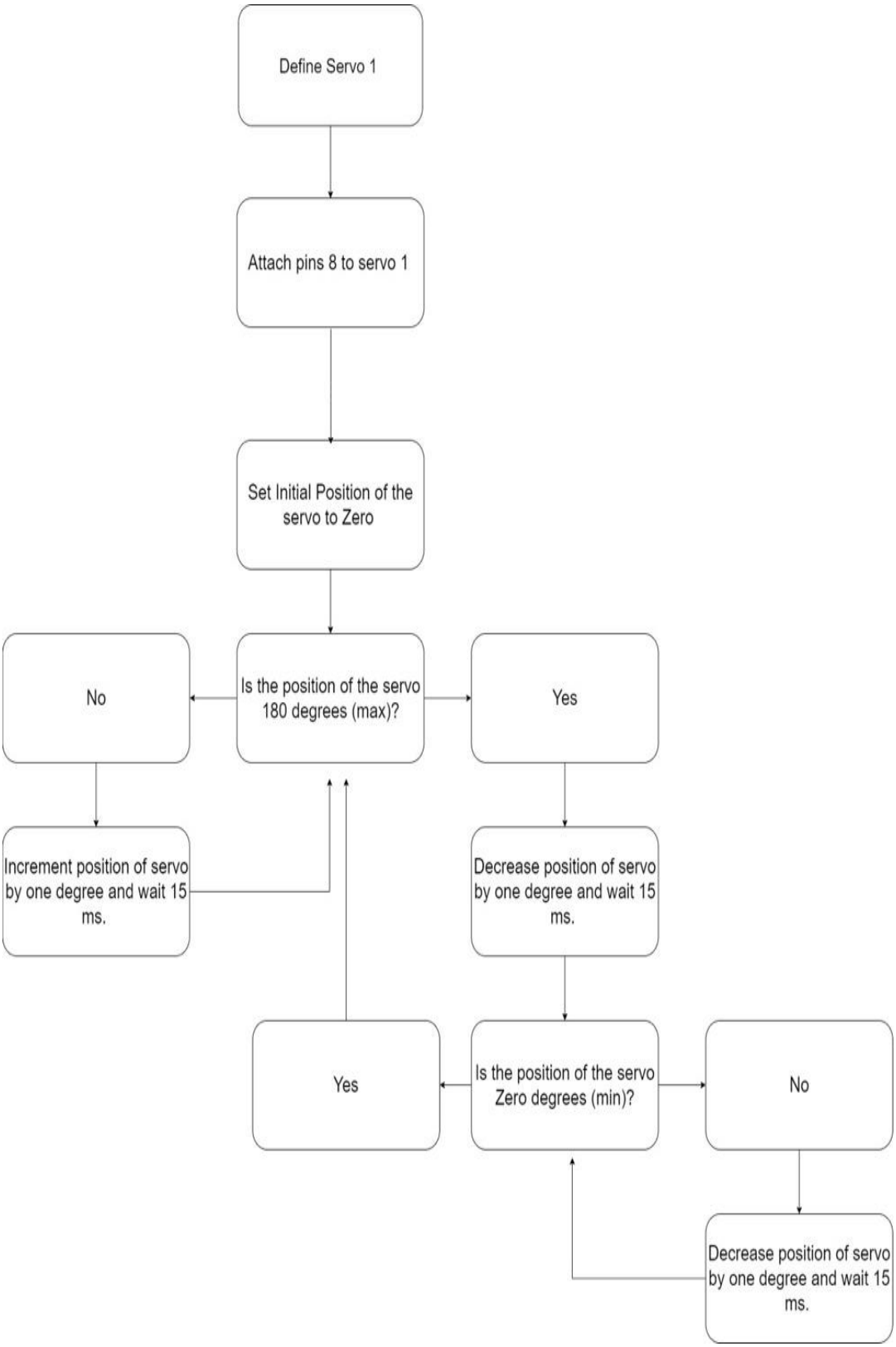
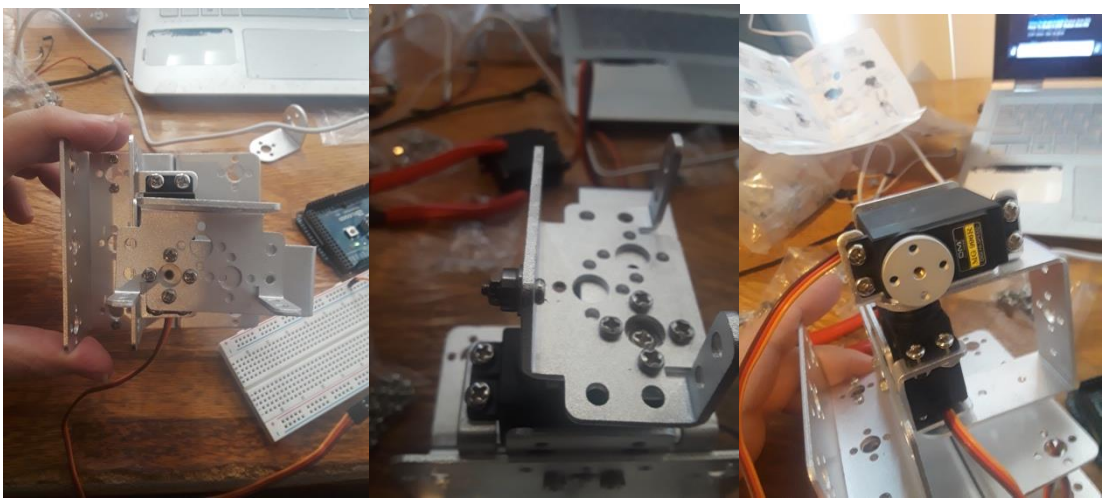


Figure 6.17 Fifth Code Diagram Sweeping code for calibrating the servo motors

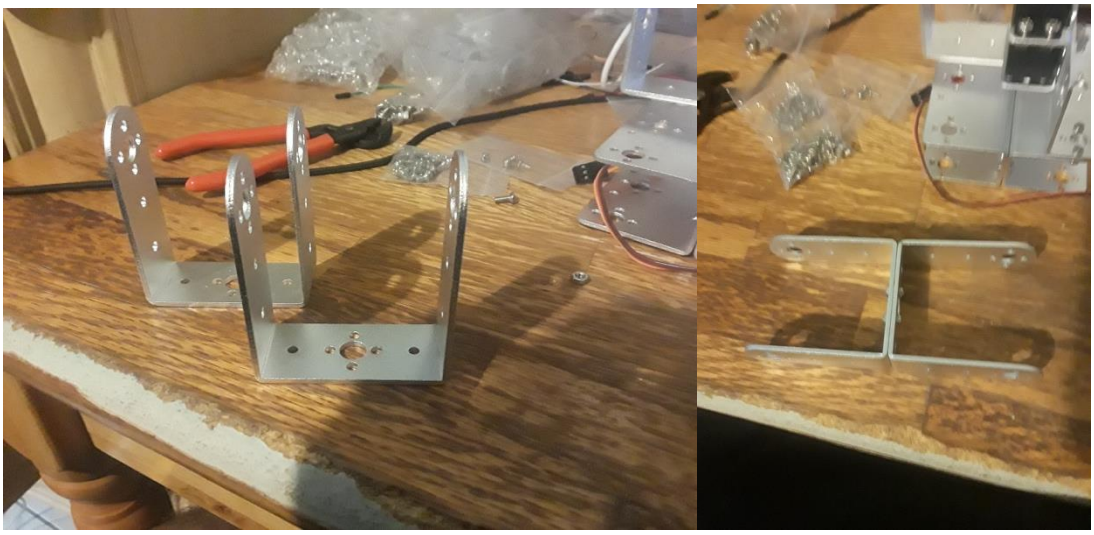


Figures 6.18

Now we take another servo bracket and fasten it to the servo horn with 4 M3*8 screws as seen in figure 6.18 above. Then we repeat screwing in a M3*10 screw and bolt as seen in figure 6.18 above. Then we take a MG996R and put it inside the servo bracket and fasten it in with 4 M4*10 screws as seen in figure 6.18 above. This servo will give us our second degree of freedom and will be in rotating the central lever vertically. This servo is the most important servo because it carries with it the most weight. Thus, it is the most prone to failure. After I installed it, I calibrated it an Arduino, C plus plus code, and a potentiometer. The purpose of the code was to tell the servo to move as I rotated the dial on the potentiometer. The potentiometer is a variable resister. As I turn the crank I increase or decrease the resistance of the resister. The Arduino reads the voltage of the node the resister is connected too as an analog signal. When I decrease the resistance, the voltage increases and when I increase the resistance the voltage decreases. The Arduino takes the voltage and records it as a numerical value. It then converts that numerical value to be a integer between 0-180. It then tells the servo to go to said position, and then loops back to looking to the voltage to see if it has been changed by twisting the potentiometer ever 15 milliseconds. When I used this code as a method of calibration, I got very inconsistent results. The problem wasn't the code but the potentiometer. I had gotten a cheap potentiometer as a part of an electronics bundle and the pins that come out of that potentiometer are very small and frail. The potentiometer would rock and wobble whenever I would turn the crank. This would cause the servo to jerk back and forth when it shouldn't have. Also the pins were to small and would sometimes loose connection to the metal in the breadboard. After I while I decided this wouldn't work until I got better potentiometers.

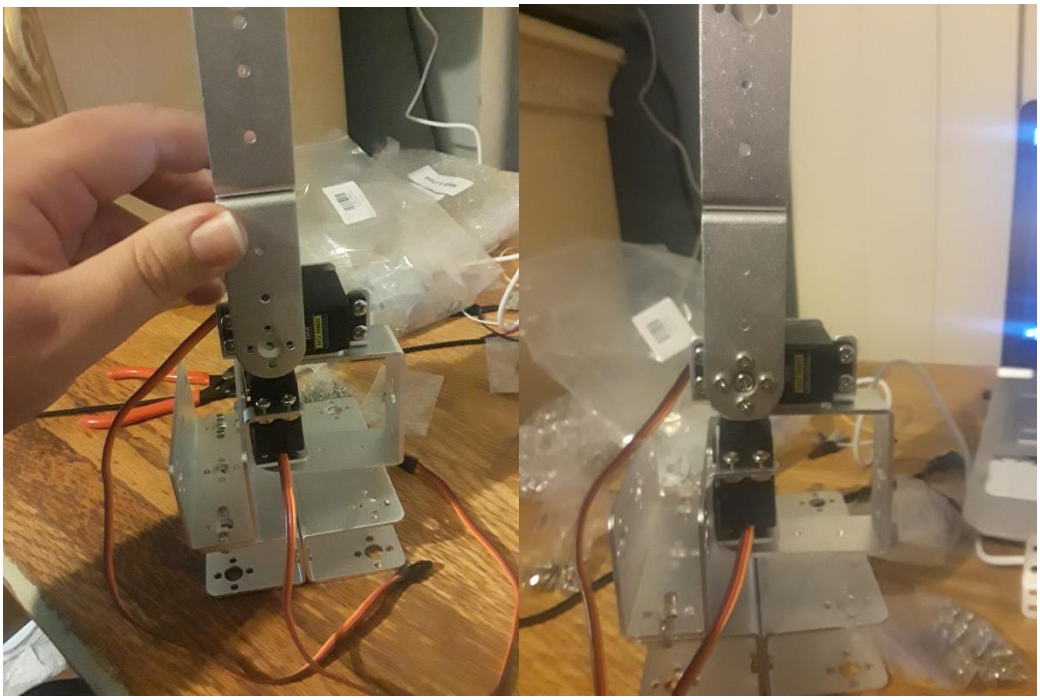
I changed my approach and redid my code. Instead of having the potentiometer be the input that changed the position of the servo I would instead automate the rotation of the servo with my code in C. The servo would go to its initial position, then slowly turn until it

reached its max, then rotate back to zero. The program would loop indefinitely from o to 180 to 0 till the end of time. This was a better approach because it worked 100 percent of the time. Using this method, I could see when my servo was not properly calibrated and adjust accordingly. Why would a servo not be calibrated correctly if the initial position was preset? Well while the servo initial position was never wrong (since I had preprogrammed them to the correct position initially) the servo horn could be installed incorrectly. The servo horn has an even number of holes, however, the gears in the servo do not allow the servo horn to be installed completely flat. The servo horns are always a few degrees from the normal either positive or negative. This causes the servo axis of rotation to be off and sometimes it collides directly with the metal bracket. This is why I had to test and calibrate every servo to see if the servo has a free range of motion or not and hits the aluminum brackets.



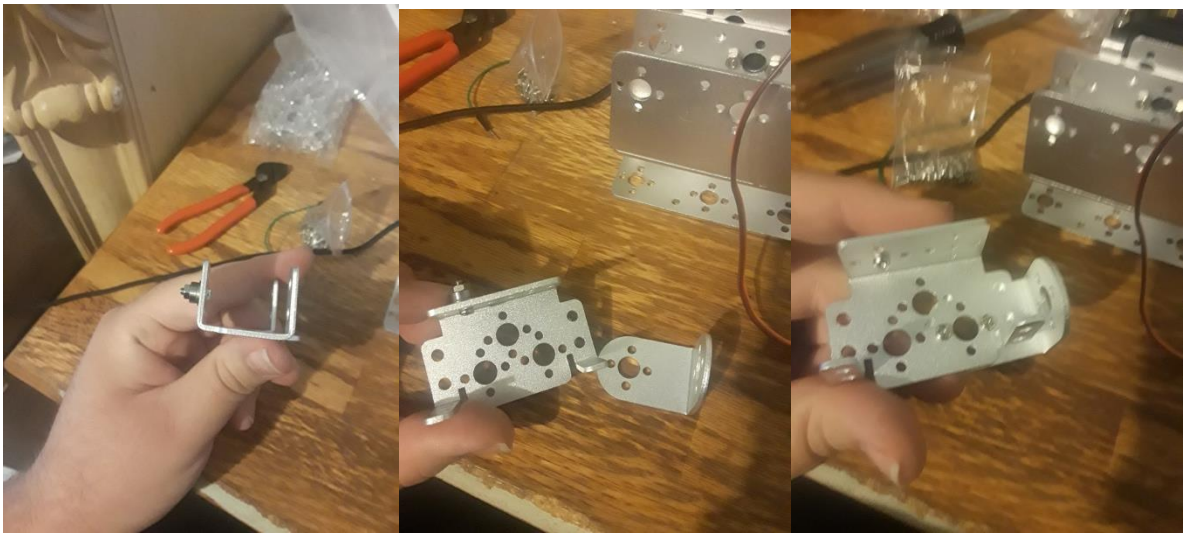
Figures 6.19

Now we take 2 U-brackets as seen in figure 6.19 above and fasten them together with R M3*8 screws as seen in figure 6.19 above. This will be the main lever or arm in the robot arm. The servo attached to this will have to contend with the most weight.



Figures 6.20

Now we take the arm or lever that we assembled as seen in figure 6.20 above and fasten it to the servo horn with 4 M3*8 screws as seen in figure 6.20 above.

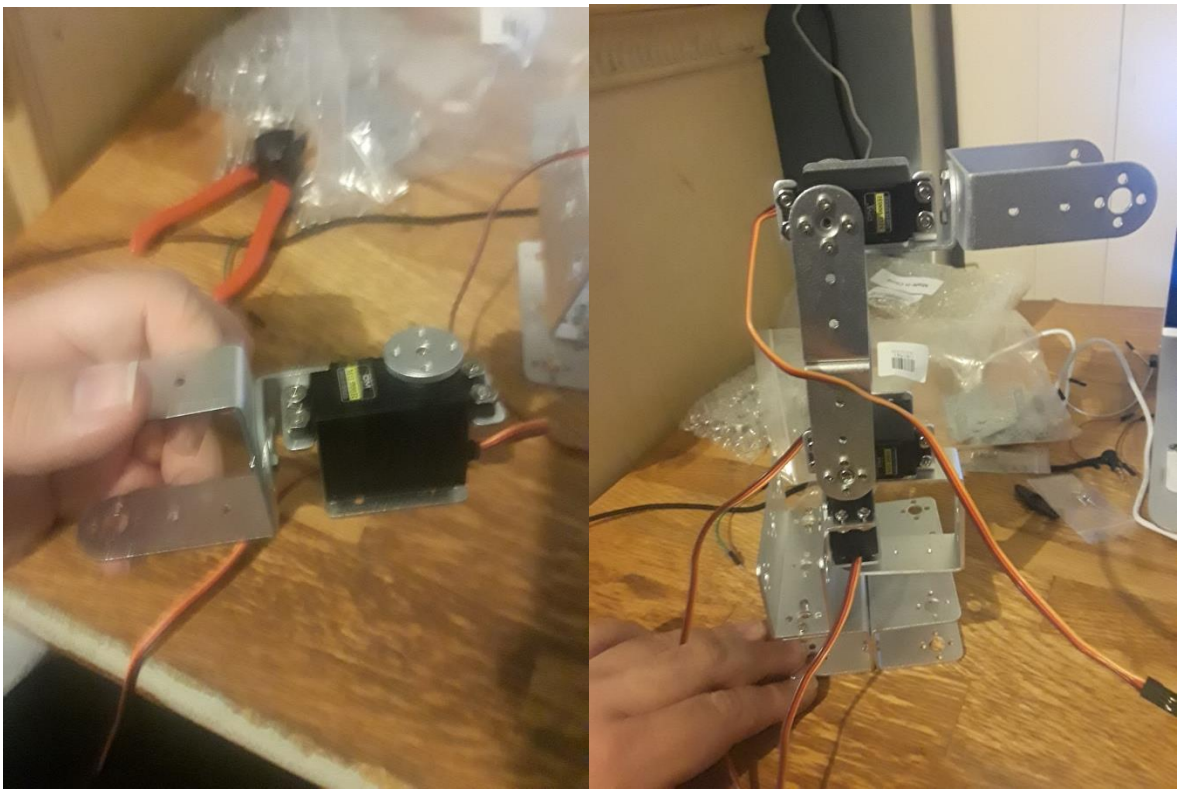


Figures 6.21

Now we take another servo bracket and fasten a M3*10 screw and bolt as seen in figure … above. Then we take the servo bracket and a L bracket and fasten them together using 2 M3*8 screws and bolts as seen in figures 6.21 above.

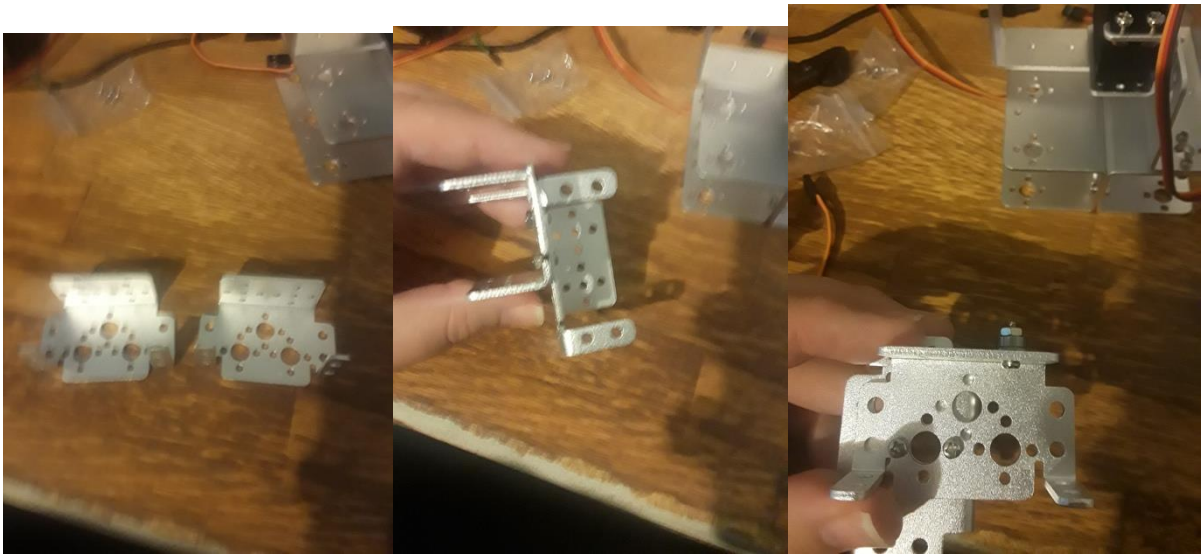
Figures 6.22

Now we take the servo bracket and fasten it to the u bracket with 2 M3*8 screws and bolts as seen in figures 6.22. This is the second vertical arm.



Figures 6.23

Now we attach the M6996R to the servo bracket with 4 M4*10 as seen in figure 6.23 above. Then we attach the secondary arm to the main arm through the servo horn with 4 M3*8 screws as seen in figure 6.23 above. This will be our third degree of freedom. This second servo will allow us contract and expand in vertical space. Thanks to this we can now get closer or father away from and to an object. Technically speaking we could end the robot arm right here. As we know have 3 degrees of movement. With three degrees of movement we can operate fully in 3d space. This would however require us to tweak our design a bit. Instead of the having a claw fastened onto the last servo we could have it fastened unto this one. Or we could somehow attach the paint brush to the servo horn. Why would we do this. Well the other half of the robot is heavy. It may be better just to cut the excessive degrees of freedom and allow all the torque to be focused on manipulating the brush instead of lifting the other aluminum levers.



Figures 6.24

Now we take 2 servo brackets and attach them with 2 M3*8 screws and bolts as seen in figures 6.24 through 6.24

Figures 6.25

Now we take the bolted servo bracket and attach a MG996R with 4 M4*10 screws and bolts as seen in figure 6.25 above. Then we attach the servo bracket with the U bracket and attach a MG996R with 4 M4*10 screws and bolts to the second servo bracket as seen in figure 6.25

Figures 6.26

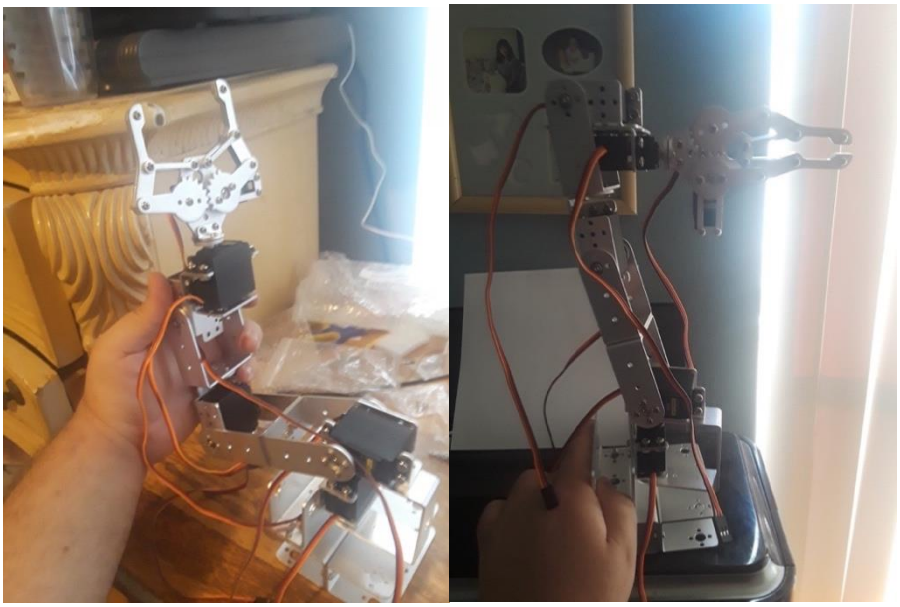
Now comes our final part. The assembly of the robot claw as seen in figure 6.26 above. We start with taking a servo horn and placing it under the robot claw as seen in figure /// above.

Figures 6.27

Now we screw in the servo horn into the robot claw gears with 2 M3*8 screws as seen in figure 6.27 above. Then we flip over robot claw and attach it to the robot claw with 4 M4*10 screws as seen in figure 6.27 above.

Figures 6.28

Now we take the full assembled robot arm as seen in figure 6.28 above and attach it to the servo bracket using 4 M4*10 screws and bolts as seen in figure 6.28 above.



Figures 6.29

Now we are done. The robot arm with 6 degrees of freedom is complete. This is what the complete version looks like in figure 6.29.

6.3 Initial Coding Planning

Without program files or the task of coding, a robot would not be able to execute its functionality. The instructions given to any machine allow for the execution of specific assigned tasks. For example, in the field of medicine there has been a surge in the number of robots that carry out simple and complex surgical procedures. In the situation where there is no coding plan or instruction for the robot, a patient might just lay on an operation table with useless heavy machinery mounted upon them. The utilization of robots in surgical operations is more crucial therefore the coding plan or instructions need to be accurate as the absence of a program may lead to complications or even the death of a patient. Another important industry that requires precise software component is the manufacturing industry. Imagine an industrial plant responsible for the manufacturing of containers for soft drinks. In the absence of the software component responsible for instructing a long chain of heavy machinery to manufacture containers for soft drinks. This might affect a company's ability to meet its demand and in the long run affect large American food chains. Again, imagine McDonalds did not have your favorite beverage due to a manufacturing plant's inability to meet its demand. In addition, it is equally dreadful if the software component of the manufacturing plant was imprecise. Hundreds and thousands of containers will be flawed causing the company money. Once more I engage the reader to imagine your can of coke crushed on its side.

The relevance of an initial coding plan allowed our team to fully decide on the functionality of the robot arm as well as how to achieve that. The main functionality of the robot arm

would be to paint any image inputted through the camera on the provided canvas. The execution of this task would require a paint brush placed in the paint brush attachment and the servo motors programmed to move in order to the output. Our team decided to program in Python, allowing the inclusion of different libraries that exist. Also, the final design of the algorithm encompasses the entire operation of the robot arm. Meaning from the moment signals are sent to the camera to the moment where the paint brush touches the canvas, the final algorithm is responsible for these different tasks.

We divided our source code into two different sections with different functionality and background knowledge, even though we are using the same programming language. We decided to give priority to our servos. It took us a fair decent amount of time to fully understand how the servos work using python. The main concepts of it are duty cycle, so that we can initiate the rotation needed for certain amount of time on our servos. Also, the equation of distance will come handy because we will need to calculate the distance between a certain point from our drawing to another one. It is necessary for us to solve these equations accurately because in order for the robotic arm to paint an image accuracy is important. For example, if we are attempting to draw a painting that involves the color contrast between the sky and the buildings, with the use of these formula the device will know the difference between top and bottom of the canvas. This will allow the painting to be more accurate.

$$\text{2D: } distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$\text{3D: } distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Figure 6.30 The distance formula for 2d, and 3d space.

This distance needs to be coded in a way that it will receive only two points and our algorithm should be able to send the appropriate distance to our servos in order for the arm to move accordingly between two points. The third concept is pure math to find the slope equation. By doing this, it will help us rotate our specific servo by only just giving it an input to our algorithm. What's good about this section is that this section does not involve reading or fetching any data from the servo motors. The microcontroller just sends signals and the servos will do the rest. We just need to do the math as accurate as possible.

For the second section involving the camera module, the process will be larger and more expensive since it will involve more functionality. Before anything, our algorithm needs to read and send data to our camera module. We try to gather accurate from the picture being drawn almost every 10 seconds. The reason is because we need to keep track of

almost every time our robot arm completes some part of the drawing. Our algorithm should be able to save the data locally every time we read from the camera module, which is sending signals to our microcontroller every time we desire. Remember that we as the developers, can control when we need to activate the camera to get some information that we will need to build our algorithm. Furthermore, our microcontroller will wait for the next time we read data. By doing this way, our algorithm must wait for the signal every time or at least every certain period, maybe every 10 seconds. After that, our algorithm should be built in a way that saves the new incoming data dynamically. Meaning, we do not need to set up anything. Our algorithm will work by itself.

The final design of our algorithm will be that it should do everything by itself. Meaning, from the moment it sends a signal to the camera to get the picture in pixels, all the way to the last pixels of the picture and finally finishing sending the last signals to our servos to finish the painting. Every time we gather data, our algorithm will do a comparison between these two last times. However, so far, we haven't done any coding after this step, but the way it will work will be converting the data read by the camera to pixels. We found that comparing pixels, of the first input from the camera, to pixels, of the second input from the camera will be the most convenient way. This is because we will need to convert our pixels missing of the picture, meaning what we have left of the real picture, to our servos. Obviously, our servos do not accept any data other than pulse width modulation. Meaning, sending pixels from our microcontroller will not work.

One idea that we have is to find the distance we need from our robot arm to the paper and find the slope equation, along with the distance equation. The reason is because we want an algorithm that is just expecting any kind of inputs, in this case it will be pixels or maybe we may need to convert it to percentage, and send it to our function, so that it returns the distance needed for our servos to move the robot arm. In order to achieve this, our set up, meaning our robot arm and the paper, will be located and placed in one static place. It will not be moving at any time. By doing it this way, we can calculate the slope equation for the angles and the distance equation for the robot arm only once. Meaning, we would only need to code the function needed only once and recycle for every time we get new data from our servos and send it to our servos.

6.4 Camera software testing

For certain robot technologies on the market, the camera software is a prime characteristic of the robot. From robots that serve as vacuum cleaners to robots that simply carry heavy material, the camera located on the robot serves as the eyes of the robot. Without the camera, precise calculation of its location in proximity to a nearby object would be unknown. With cameras used in robots that assist in carrying out surgical operations, the scale of serious is on another level. In the absence of precise calculations, a patient's life might be at risk. This is also because in the field or medicine, accuracy and precision are the cornerstones of most surgical operations carried out. The camera used

in this project serves as the input for the ongoing data operation and therefore the data generated from the camera is essential in achieving the desired output on the canvas. The functionality therefore of the camera must be of good quality. Before testing the camera software, properties such as the camera resolution, the color recognition and the brightness of the camera were tested to ensure they were in best condition. At this point in the project, it would be impossible to fully test the software capabilities of the camera. This is because not all parts of the robot painting arm are together. Therefore, testing the camera in correspondence to the servo motors is almost impossible.

Before starting to develop the algorithm, we need to test the camera so that it meets our requirements for our design. Since we are dealing with drawing that include color, our camera should see and recognize color in a simple way. To make is simple for us, we will need to use python to change the setting of the camera. We can modify the setting of the camera such as the resolution by going all the way up to 1080p (it is 720p by default). Color recognition is also an important requirement for our design. We can set up the brightness of our camera can recognize the colors being drawn in any environment. We will have to test it on a very illuminated room, where the sun may be pointing to the paper where the robot arm will do the painting. Another testing environment will be dark rooms and changing the brightness of the camera to the maximum. By doing it this way, we can ensure that our camera can work anywhere and can also be adaptable under any circumstance. The first testing environment involved connecting the camera to a different power supply.

As mentioned in previous sections, our camera is power up by plugging it into a laptop or desktop. This gives us enough power (5 volts) to power up the camera. If we try to give a different power supply to the microcontroller, we wanted to make sure our camera still working. We tried plugging into a small power supply that will gives us up to 20 volts. After playing around with the values of the power supply, we determined that the maximum voltage the microcontroller can sustain is 10 volts. It was starting to suffer overheating; however, the camera was still working just fine. It had some delay to do operations such as converting the incoming data into useful and manageable data, but that was because our processor was getting too much energy. Our next test was to see how waterproof or water-resistance our camera really is. Since our project will be in the outside, it will be in contact with some wet environments or accidents may happen. It is always important to check the limits of our hardware. I way to test this case scenario is to apply raindrops into our camera. The first case was to apply the raindrops followed by performing an action and wait how blurry the picture looks like or if the camera does send any signal. We waited 120 seconds after dropping the raindrops and the camera was still giving us the signals we need, but with a blurry definition, which was expected. The next case scenario is to create a loop where our camera takes pictures indefinitely and videos. Since our camera is in the middle of a process, we can see the precise moment a raindrop appears in the camera and check if we get any lost signal from the camera. Fortunately, we didn’t get any lost signal from our camera module and we were able to perform our actions.

However, this test was using drops, we are not going to test big quantities of water since it may damage our hardware.

The third test was about packet lost from our camera to our microcontroller in a set period. This test requires to keep getting data from the camera module and check in real-time that data. We are printing in a new window every time we get a picture. It should never lose any package data coming from the camera module. We should also keep track of the times we are taking the picture in that set of time. For instance, if we are giving it 10 seconds, we can easily measure 10 pictures at the end of those 10 seconds. At the end of, we should have ten new windows with ten different pictures. We are not going to test microseconds since our project does not require to have a precision measured in microseconds. After doing the changes to our algorithm, we were able to perform the ten pictures in order without any packet loss. Keep in mind that even though, we did only 10 seconds, the number of pictures taken were too few. After that, we tested sixty pictures in one whole minute. This will bring us enough information to rely on the precision of our camera at a certain time. After that is done, we were able to see the sixty pictures after the minute was over. This tell us that our camera was reliable enough for use to use it in our module. Another test was to measure how well it recognizes the change of color. Our algorithm can read the color in an array format; however, we rely 100% on our camera to tell us what color it after the same color is has been changed. For instance, if we draw a blue circle and read the picture, our algorithm will tell us that the picture is mainly blue. Furthermore, if we apply red color to our blue circle, then we should be able to still recognize the main color of the circle, in this case blue. After testing the color, we were able to see mainly the blue color even after reading the data.

As mentioned in previous sections, our camera will need to capture the color of a hand drawing picture. This will make it more difficult for use since we are trying to reduce the cost of every component as much as possible. Meaning that a more expensive camera, normally means better capture of the colors. However, by using our camera module v2, I think it will be more than enough for this mentioned camera to capture the color as neat and fresh as we want. Our test will require me Alonso, to draw by hand pictures using paint, pencils, etc. Then, I am going to place our camera in a static position where it will be looking at whatever I do with my hand. After that, our camera will need to send pictures every certain period, as explained in the previous section.

Another point of access is that the whole team will need to have access to the code every time from anywhere. We decided to post our code publicly on GitHub so that anyone of us can have access to it. We don’t want to make it privately since this code needs to be used with hardware, therefore whoever see this code will not represent a serious thread for us in terms of hacking. Furthermore, this new repository will have only one admin access. In this case, Alonso will be admin. Since he is responsible of the entire camera software side, he will oversee post the code every time he decides to. Also, if anyone of the group feels like there is a better way of doing a certain part of the code, this person can leave a comment or request a change of the base code. The user who wants to make

a change to the base code, will not be able to do it until the admin (in this case Alonso) accepts the request first. That way, we can always have someone in charge of the code and who is responsible for the development of this software only. By organizing this way, we can set up roles of distribution. This means, we can assign a specific development of the project to one person. However, this does not mean that this person will only be doing this part. Everyone will do a little bit of everything at the end; however, it is much better for the team to have each of the members focused entirely in one part of the project. After creating this repository on GitHub and adding the new base code from a computer. The next step is to add the members of this repository as a way that they are only able to make any pull requests. Pull requests are simple requests from people who have access to the base code and want to make changes. This way, the team can see what changes each member of the team were done. GitHub let us see a lot of information every time a new pull request has been requested until it was merged into the base code. For instance, it gives us the lines of code that were changed by the requester and it doesn't get merged into the base code unless the admin accepts it. Also, it tells us if there is anything that is conflicting with the base code from the request and it doesn't let the request to merge the request until these issues are addressed. This tool will bring us a much better organizing way of thinking from a developer point of view. It will also give us more understanding and awareness on how the project's code is going. It also tells us what is exactly missing from the final product, so that anyone of the group can jump on it and fix it as soon as possible. It will also tell us how many days it has been since the last merge was made. This will be helpful for us since we have three months for development and testing. It will tell us how fast the code is being worked and how much it is getting done.

6.5 Camera Flowchart

The camera flowchart captures the step by step operations of the camera. It also captures the interactions between the camera and microprocessor as well as some of the if loop conditions to be created in order to have an effective camera software unit. There is also the element of rechecking the validity of the input of the robot arm. For example, in the operation of the camera, the first picture is taken, and this first image is stored locally to be compared with any future images captured. If the images captured after the first image matches the original image captured, then there is no need to recapture the image. The original image therefore becomes the input for the robot painting arm and this image becomes the basis for the painting on the canvas.

The following picture represents the flowchart of just the camera module in a step by step process. It explains from the moment the micro controller sends the action to take a picture to the camera. It needs to get the data received from the camera to perform further actions that are going to be explained. After receiving the first picture, we need to save it locally since it will be used for comparison every time, we take another picture. We are

going to check if the objects match completely the original picture. If our new picture does match the original picture, this means that we don't need to draw the new picture. In terms of servos, this means we don't need to send new signals to our servos to draw the new picture. After that we keep taking pictures until it fails. However, if the new picture does not match the original picture, we need to specify if we have any specific object in the new picture that match or do not match. If there are no objects that match the objects from the original picture, then we are good to proceed to draw the objects that are missing in the original picture. After that, we send the correct signals to our servos to draw the specific objects. However, if there are some objects matching some objects of the original picture, then we only need to specify the existing objects in the drawing. That way, we are avoiding drawing the existing objects in the drawing. In terms of servos, we will not send signals to our servos for these specific matching objects found.

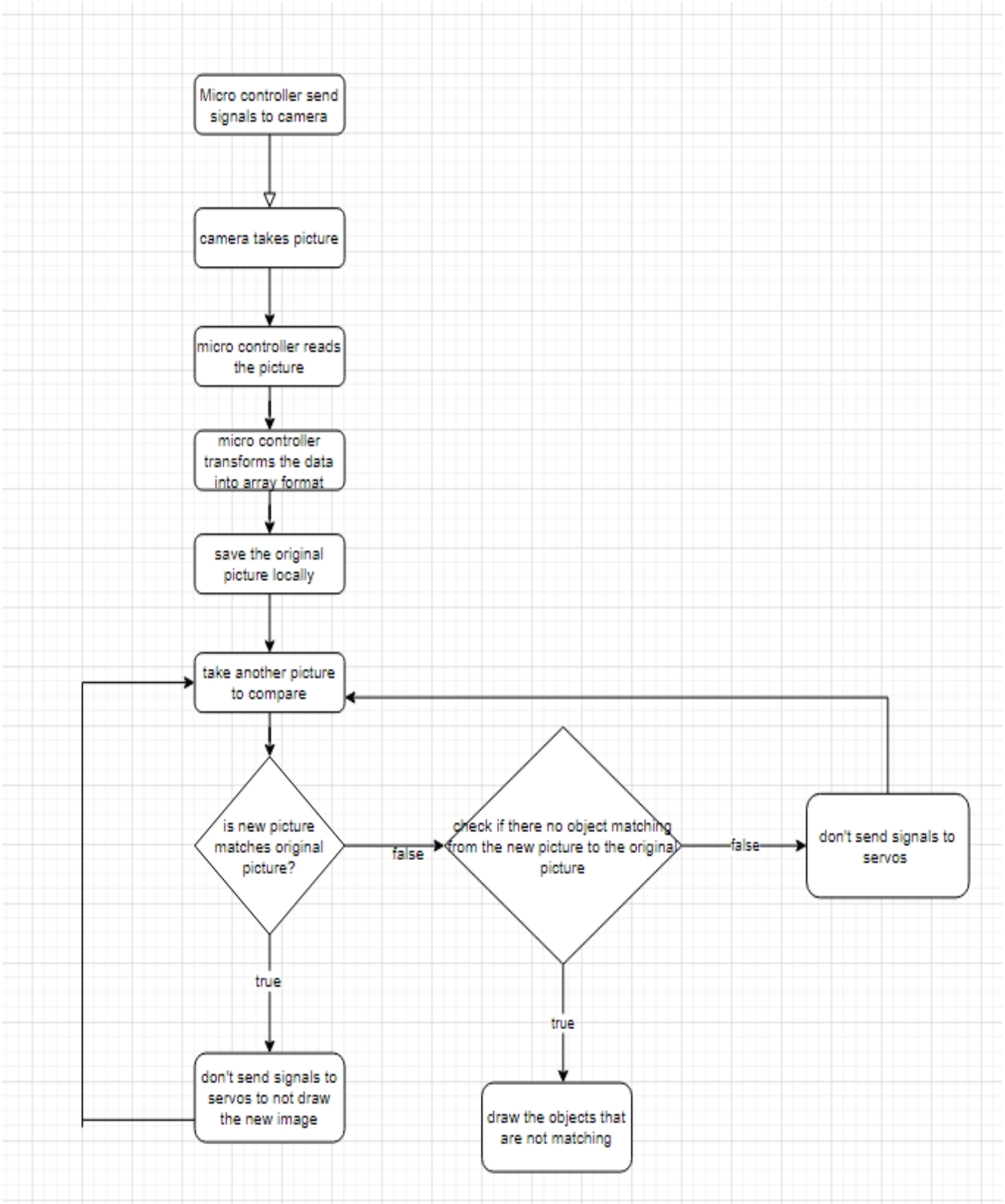


Figure 6.31: Flowchart for camera use

7 Generic Flowchart of Robot Painting Arm

Every single component of the robot painting arm is placed or represented on the Printed Circuit board. However, despite all components of this design can be found on the printed circuit board, the order of operation that enables the final task of painting is also important. The operation of the robot arm is broken down in two parts. These parts are the hardware and the software which principal fragments of the operation of the robot painting arm. The hardware part of the project is responsible for the physical representation of the project, and this includes elements such as the power supply, the servo motors, the printed circuit board, the micro controller, as well as the camera. Whereas the software part of the project has to do with the commands written to control and execute the final action of painting the canvas. The software section of the project involved creating a code or algorithm that enables the camera to capture an image as an input. This same algorithm must include a line of instruction where the servo motors are instructed to initiate and execute the task of painting the desired image.

The entire design begins with the robot connected to a power source. This power source should be enough to power the servo motors. This is because the servo motors are responsible for all of the mechanical function of the robot painting arm. It is therefore paramount that these components are powered. The power supply is then connected to DC converter. As is known as the convention, the power supply that runs through our homes are in AC, however the robot painting arm is powered with DC. Therefore, the DC converter converts the AC power to DC power. Once the robot painting arm is powered, the micro controller is activated and then sends signals to the Raspberry pi camera. The Raspberry pi camera then takes a picture of the desired image. The micro controller then reads the image. This read image or input is then converted into an array format. The original image is then stored locally, and then the Raspberry pi camera takes another picture. This image is then compared with the original image, if the new image is the same as the original image, signals are not sent to the servos to draw a new picture. However, the Raspberry takes another picture of the original image. If the new image is not the same as the original image, the algorithm of camera module would check for object matching from the new image to the original image. Once there is no object matching from the original image and the new image, the servo motors and paint brush attachment is then instructed to paint the original image. The figure below captures the flowchart that represents the operation of the robot painting arm. It showcases the different steps and junctions of the operation of the robot painting arm.

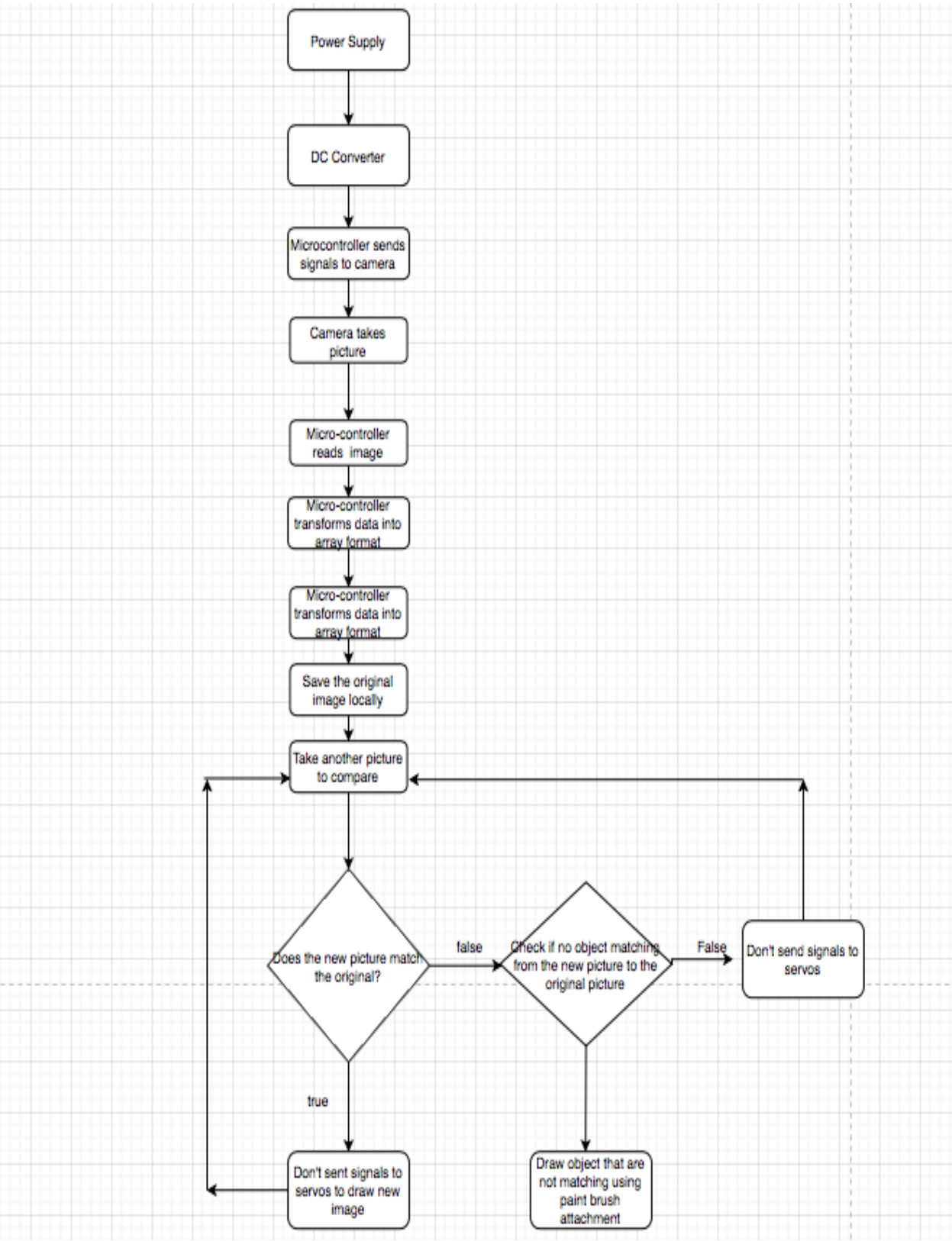


Figure 7.1: Captures the generic flowchart for the entire robot arm.

8. Printed Circuit Board

The purpose of the PCB in our project will be to mechanically and electrically support components that are needed to run our robot-arm so that it is able to successfully paint landscape drawings. The circuit board will be designed using the EAGLE software. The circuit board will consist of these important components:

- **Battery**: This will provide power for the circuit
- **Resistors**: Will limit the current. Making sure that too much current does not destroy the circuit.
- **Capacitors**: will hold or release electrical charge
- **Diodes**: allows electricity to flow in a certain direction
- **Switch**: this will either block current or allow it to flow. If the switch is open current will flow, if not current will be blocked.
- **Transistor**: a switch controlled by the electrical signals
- **LED**: the LED will be used to provide a visual feedback
- **Integrated circuit:** This will be programmed to perform a specific function

After researching the purpose of a PCB, I found that these were the important components that come into play when designing a PCB board. These components on a circuit board all perform an individual task that allows the overall function of the PCB. My group members and I struggled to understand the complexity and what the bored can be used for in the design. After conducting some research, we saw that the PCB board will play its role in the design of the robot arm, allowing it to function properly. We also learned that some possible circuit issues would be open circuit or short circuit issues. The open circuit would be cause if there is loose connection nor a broken wire on the board that will cause the board to not conduct electricity properly. While a short circuit on the board will occur more power than necessary travels through the circuit it will damage the power being supplied. This can ultimately destroy the whole design causing us to start over if we short our circuit on the PCB board especially when connecting it to the robot arm. Having to really understand these issues allowed us to fully expand our research. Making sure we understood the vital steps we will need to take in order to successfully design a PCB board that will function properly.

Our project is now past the protype stage, so we've decided to start designing our PCB board. The software we used to design our first PCB board was Eagle. Eagle is a PCB software tool that generates schematics, make board layouts, and get PCB's manufactured in factories. Output industry Gerber Files parts lists can export the industry files to a board house to get manufactured. Libraries that we are familiar with are Adafruit which we downloaded onto our laptop devices last semester in Junior Design. The libraries define what parts you can use in your schematic and your board with components. In order to find specific parts that fit our needs I used the google search

engine to look up specific parts like XBEE, Atemega, etc. for the schematic. For our first printed circuit board it consisted of a total of 7 major parts which will be listed below.

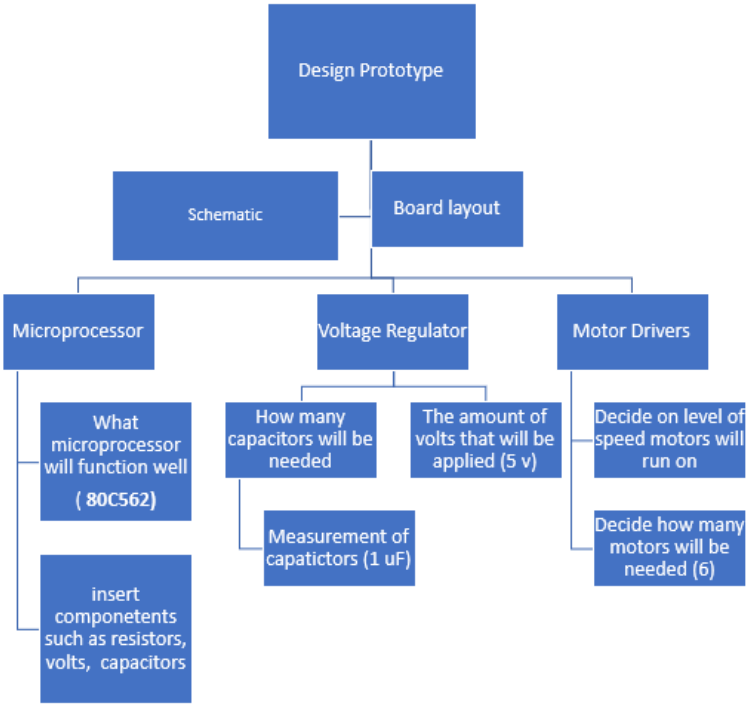


Table 8.1: Hierarchy table, designing the PCB

Due to our strict budget we have decided to first start off with an example circuit board that we will use to practice design on Eagle. To start our circuit board design, we first decided on what we wanted the boards functions to be. Of course, we needed a processor that we would hook our servo motors on in order for them to run when the robot arm is in use. So, we decided to start the ATMEGA328-2GP Single-chip 8-bit Microchip which is shown below on **figure 1**. This microcontroller is familiar with the inputs of the servo motors we are using for the robot arm. It is also able to handle a certain amount of power for the input and outputs which is important for our robotic device because there is a lot of movement required and we are connecting parts that will be connected to the arm itself. Being able to lift objects will also require high/low power.

Schematic Components

This subsection will discuss the major parts we have decided to include in our printed circuit design. These are our screenshotted schematic images from the Eagle software.

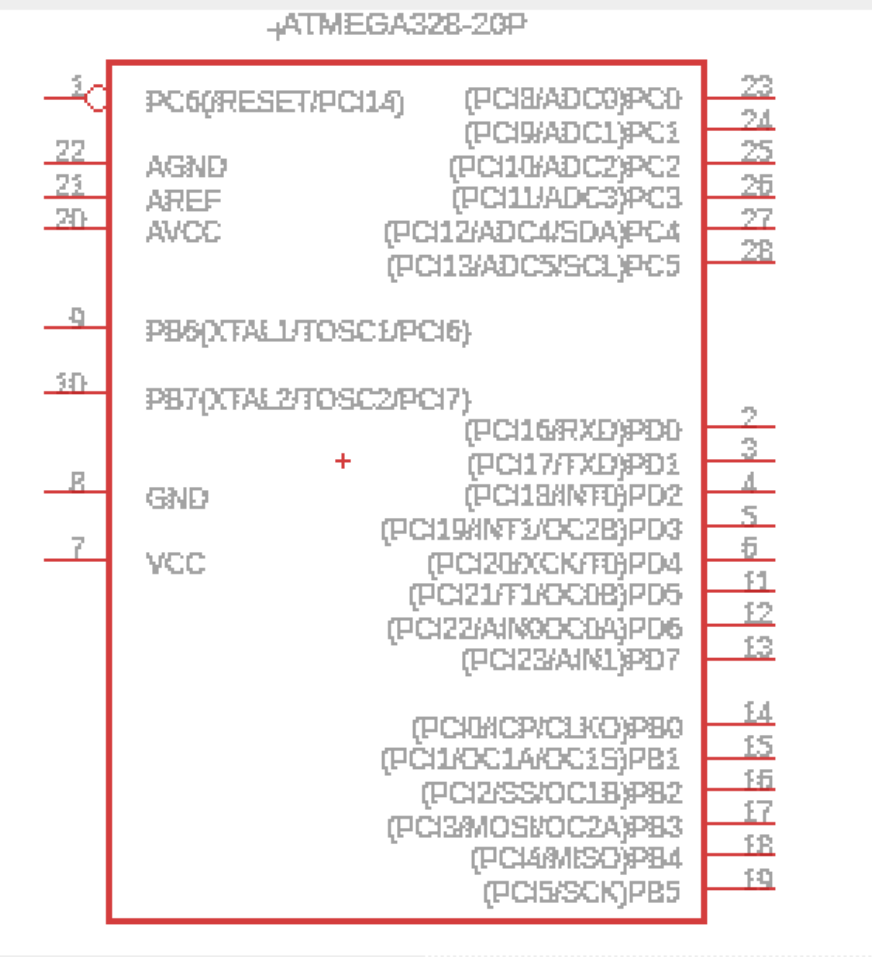


Figure 8.1 ATMEGA328-2GP Microchip

To start our circuit board design, we first decided on what we wanted the boards functions to be. Of course, we needed a processor that we would hook our servo motors on in order for them to run when the robot arm is in use. So, we decided to start with the ATMEGA328-2GP Single-chip 8-bit Microchip which is shown above on **figure 8.1**. This microchip is familiar with the inputs of the servo motors we are using for the robot arm. It is also able to handle a certain amount of power for the input and outputs which is important for our robotic device because there is a lot of movement required and we are connecting parts that will be connected to the arm itself. Being able to lift objects will also require high/low power.

Next, we looked at what type of power supply we wanted to supply our device to run on. This was tricky because of course we don't want to spend a large amount of our budget on power voltage if it's going to be high or too low. We wanted a component that had a good input voltage range. We decided to go with a L200 version 2 voltage regulator, shown below on **figure 8. 2.** This has an input voltage range from 0V to 60V which include an

input over voltage protection. a current rating of 2 amp, and an output voltage as low as 2.85 volts, and a ground of 0 Volts. The difference between the input and output voltage is released as heat. L200 is an efficient voltage regulator we decided to start with this specific one because of the range that it has. We have taken note that the heat released can affect the whole board causing malfunctions such as short circuits, waste of energy, etc. controlling the input voltage is how we plan to steer away from these problems. We can avoid heatsink by using the formal Heat generated to calculate how much energy is being wasted. This formula is *Heat generated = (input voltage – 5) x output current.* This will give us a visual on how to decrease or increase our input values. For example, if our input voltage is five and our output voltage is five, energy won't be wasted, and we can calculate it. the L200 also has short circuit, output transistor, and thermal overload protection. Making it a safe option as our voltage regulator.

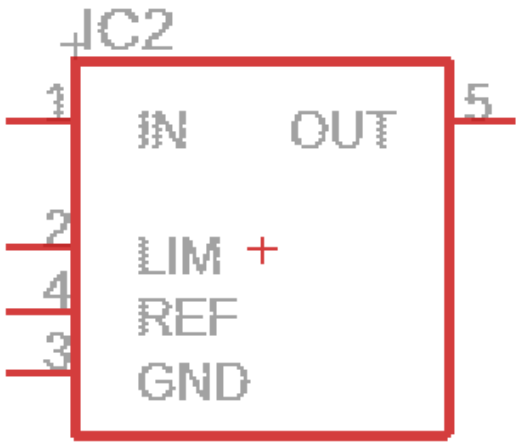


Figure 7.2: L200 Voltage regulator

As of now we decided to connect to the voltage regulator a switch button that will power on or off. This will help regulate energy by turning it off if it is not in use. We also used capacitors on our voltage regulators to filter residual AC noise. This will allow a clean DC signal allowing the circuit to work efficiently. Reducing the noise from the alternating current is only necessary if the noise on the line is too much. The capacitors allow the maximization of voltage regulation. This is included in our PCB as of now because we want to be able to have a control environment where we know if energy is being wasted or not.

Figure 8.3: DC power jack

Another component of our PCB is the DC power jack (shown above in figure 8.3), this is essential in providing the whole board power when plugged into an outlet wall. We will be inserting a power supply cord into the circuit. Of course, for this version of the PCB as mention above we included a switch which is our back up to turn the device on and off. We know that with a power jack we are going to be able to use this component to turn the device on and off.

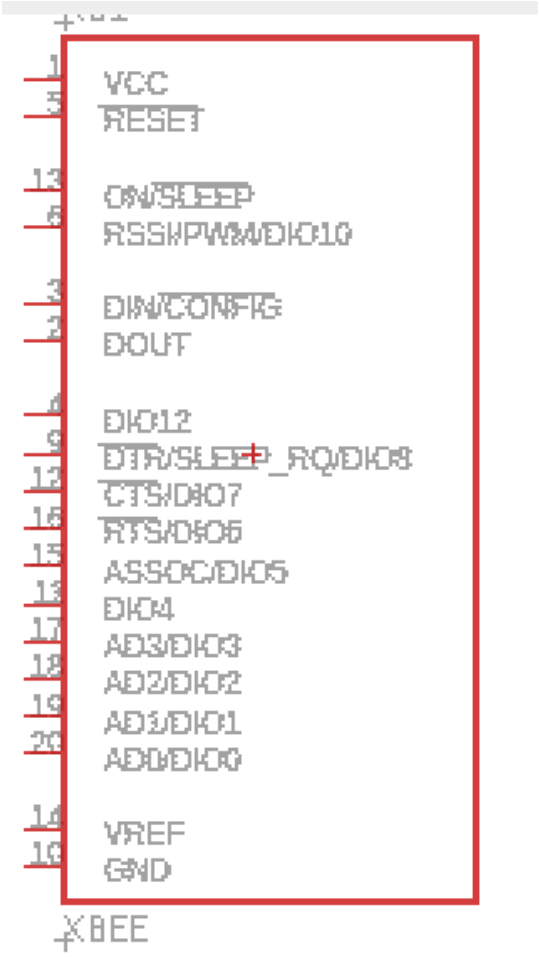


Figure 8.4: XBEE-Pro

For the Raspberry Pi Camera and the servo motors to communicate the XBEE-pro is needed for that action.As shown in **figure 8.4** there are a lot of I/O ports. The XBEE-pro allows wireless communication ranging from 0-300ft with low power consumption. this will be programmed using Audrino's IDE and connected by the baud. The XBEE is connected to a serial connector in order to take input, the type of serial connector used is shown below in **figure 8.5.** The XBEE also has its own voltage regulator to control voltage and current flow so that it run properly. Doing this allows the motor drivers and the Raspberry Pi Camera to run together.

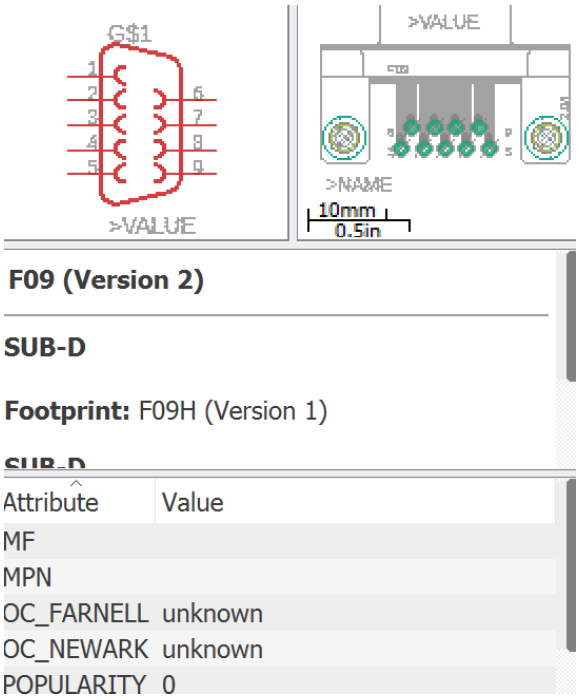


Figure 8.5: Serial Connector

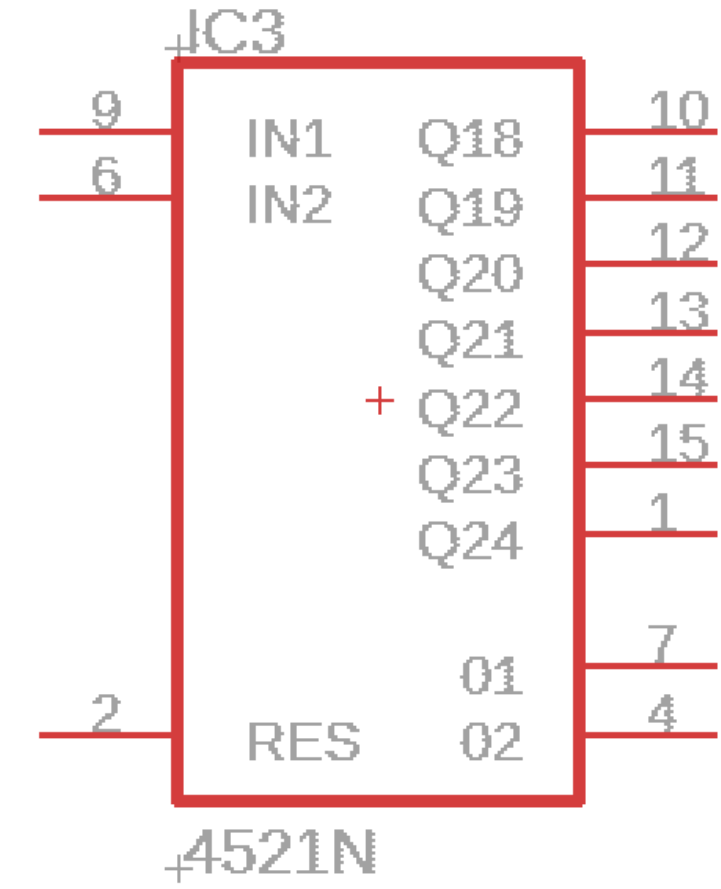


Figure 8.6: SLA7060M Motor Driver

Finally, we used the motor driver SLA7060M which is shown in **figure 3** below. We chose this motor driver because it could operate on high current and high voltage currents. This is useful for when we are ready to test our robotic arm, we can see if it is necessary for use to use a higher power supply. The SLA7060M motor driver allows us to also test low powers when using our motors. The motor driver's role will be to control the power supplied to the servos when performing actions such as lifting the brush, moving up and down the canvas, moving down when dipping into paint. These actions can be strenuous on the robot arm so with the right amount of power the robot arm can function properly with the SLA7060M motor drivers.

These were the major components that were chose to be placed onto our PCB for now. We will discuss more on how the communication between the Raspberry camera and the robot arm will work together to perform in further pages. This communication will affect the arrangements of the circuit board because that will require an input/output pin of its own if the decision is to place them on the same circuit board or connect them using a USB port on the circuit board.

Once we have decided on these components, we will then begin wiring them together placing capacitors, resistors, and grounds where necessary. As stated above these are necessary in limiting current, holding, or releasing electrical energy. Overall, their role is controlling the electrical components. We chose values based on what was recommended in the datasheets for the components above. We are working with trial and error because right now we do not have access to tools needed to build a at home circuit board on a breadboard because of the closure of the school due to the pandemic. We did research based on what other projects used and it compared to what the datasheets recommended. We are trying to design a simple circuit board due to these circumstances.

Below I will include screenshot images of the PCB schematic with all the components connected this will be **figures 8.6-8.11**.

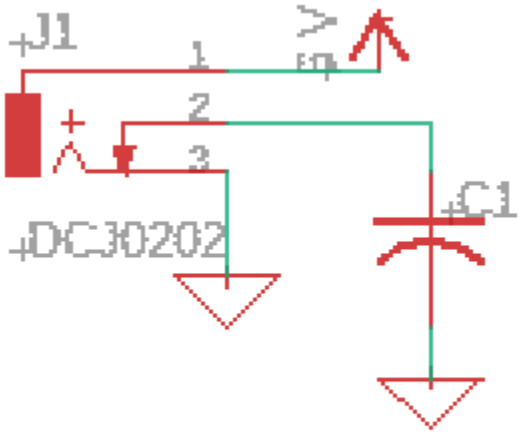


Figure 8.7: DC power jack, 5V input and .22uF capacitor

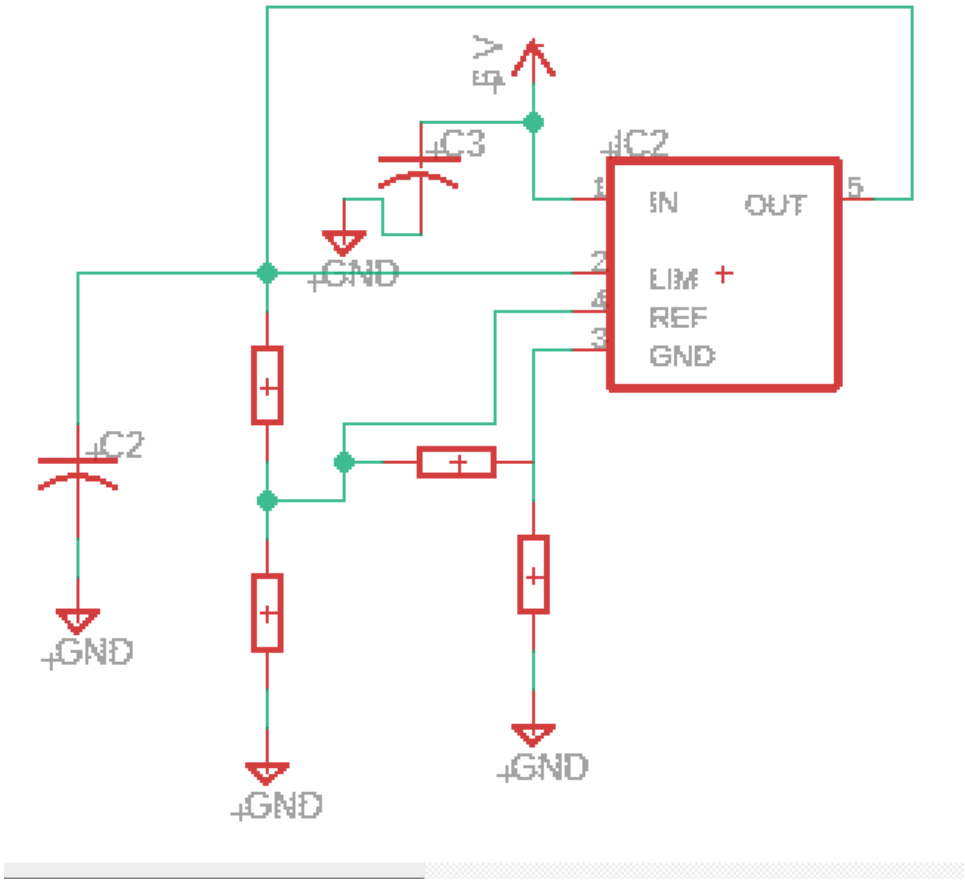


Figure 8.8: L200 Voltage regulator (Components: 5 V input, 4 resistors, 2 capacitors)

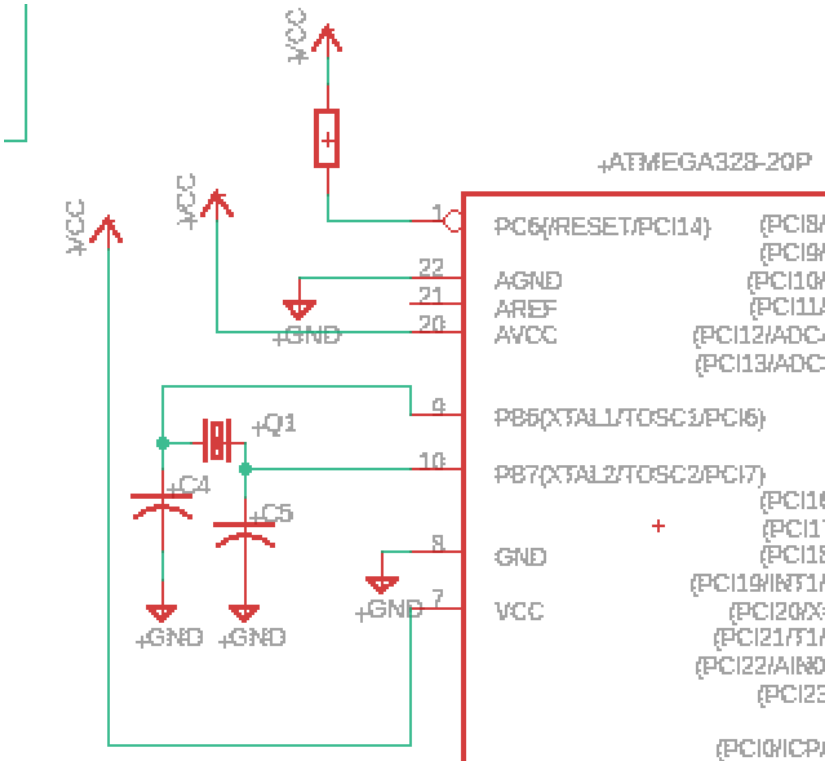


Figure 8.9: Atmega328 schematic side A

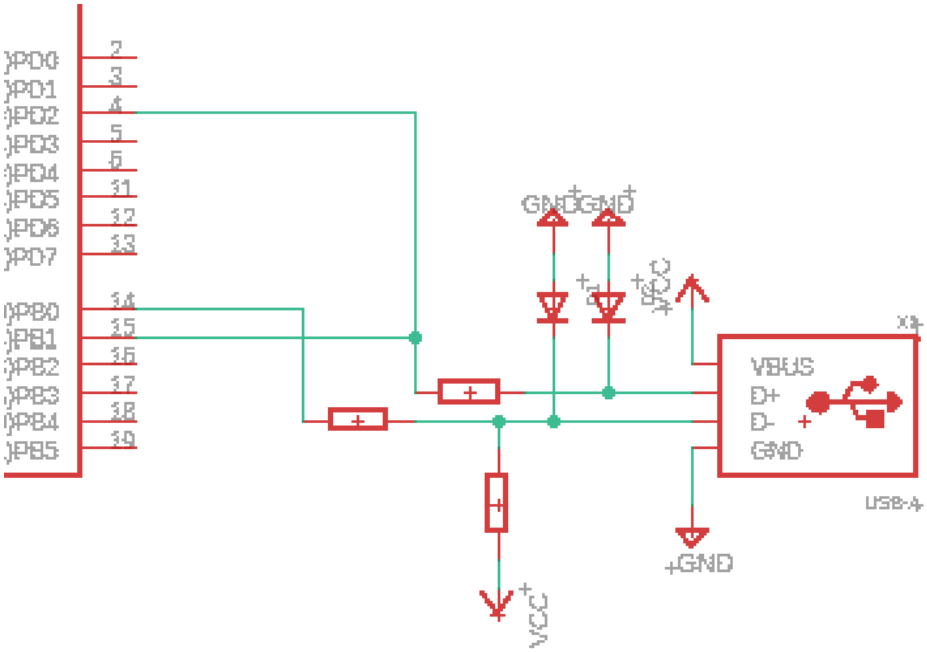


Figure 8.10: Atmega328 schematic circuit side B

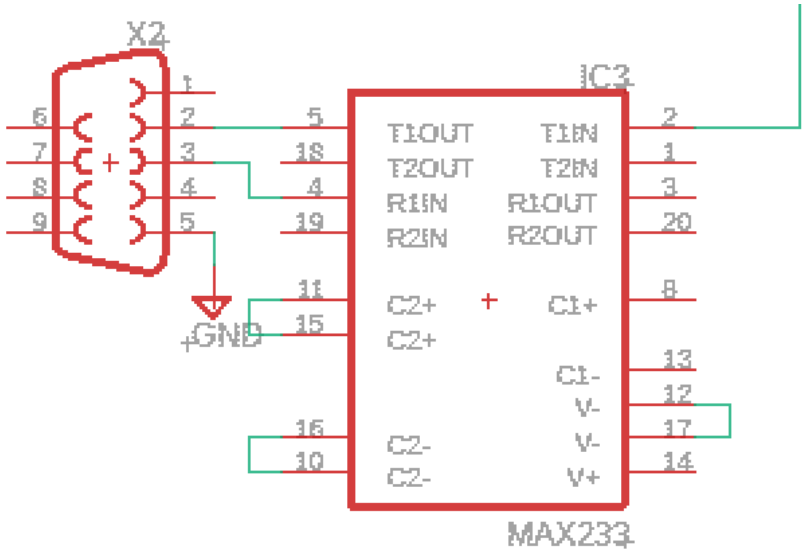


Figure 8.11: Serial connector and Max233 voltage regulator, connected to XBEE

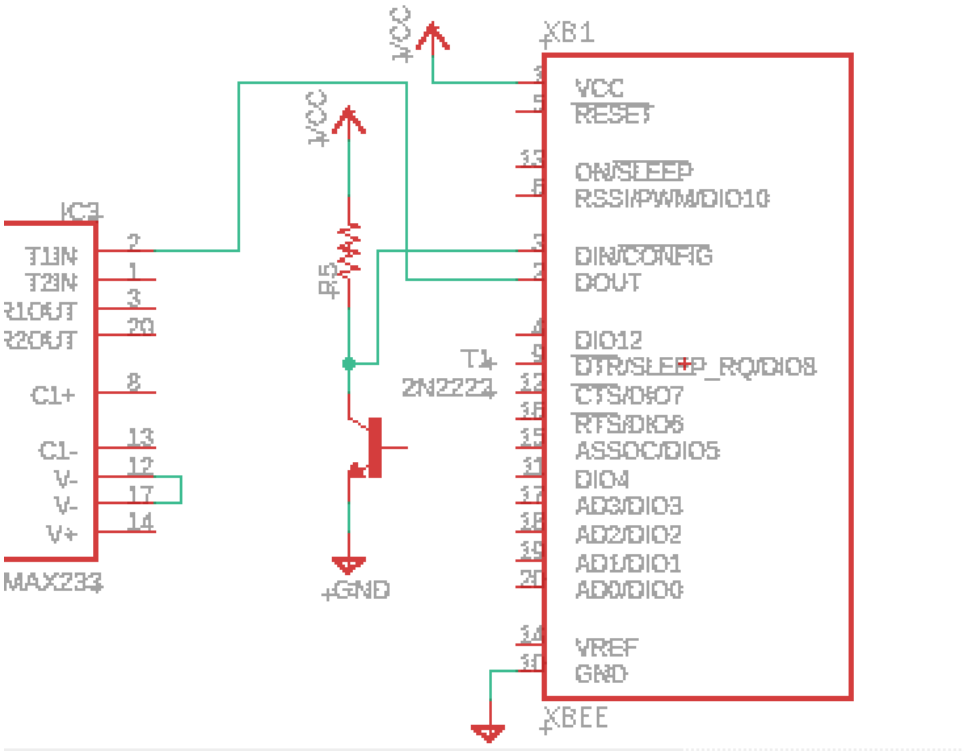


Figure 8.12: XBEE-Pro connection with voltage regulator

Figure 8.13 shown below our previous schematic circuit board which included components such as 7805 voltage regulator, SLA7060M motor drivers, and a Philips 80C562/83C562 Single-chip 8-

bit microcontroller. After doing some research we saw how we can develop a proficient and professional PCB that does not require a lot of power while being able to run efficiently. Also, after doing more research we saw that it would be better to get parts that could be programmed using the Arduino IDE because these parts would have to be compatible with the Raspberry Pi Camera and the Raspberry Pi board. The communication between boards is what will allow the robot arm to properly function and conduct its movements successfully. We have created room in our budget for this trial and error phase for our robotic arm device. With enough research and test runs we expect within the second try to have a more accurate circuit bored designed. There is not a lot of wiring involved between the components our goal for this was to keep it as simple as possible being that it is our tester circuit board for our device. The difference between this schematic and our new one is that our newer schematic is well designed, more accurate, and readable.

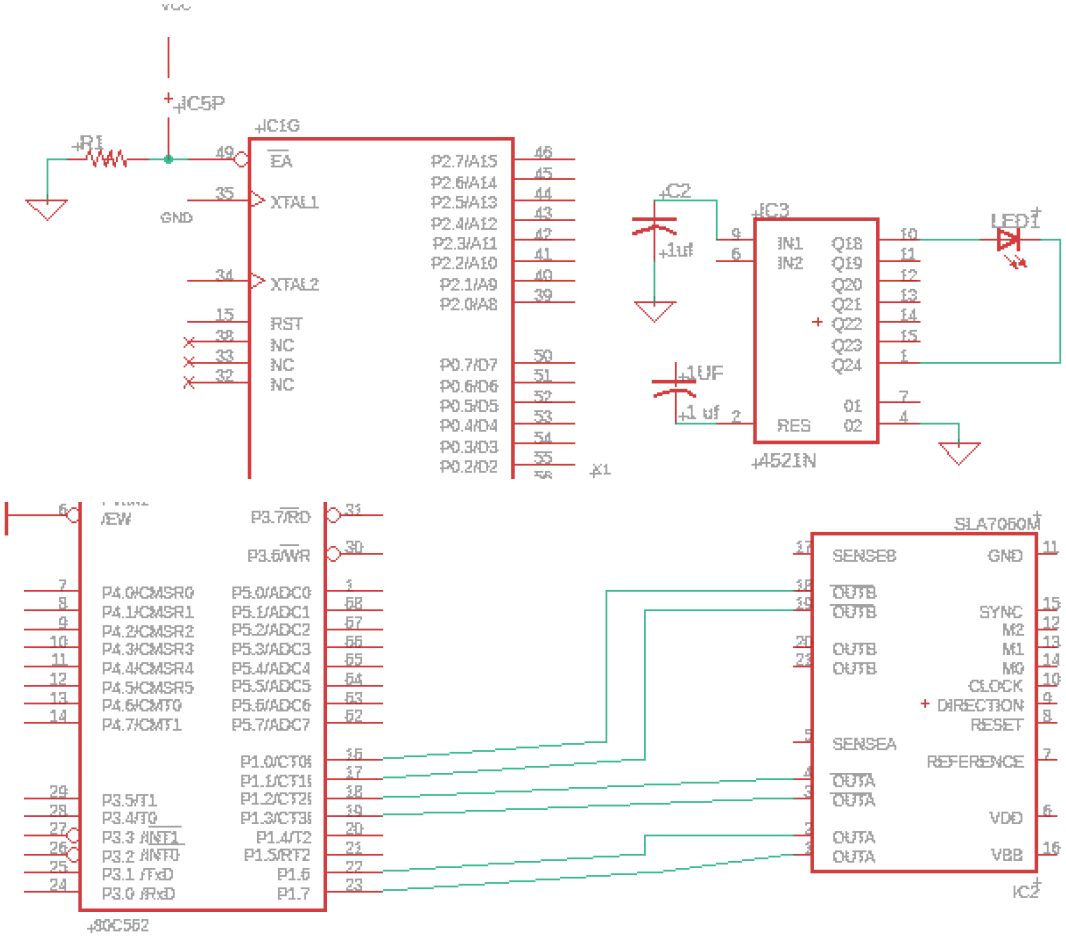


Figure 8.13: Older PCB schematic preview

The next part in designing a circuit board is the board itself, rewiring and placing the items the way you would want to see it on the physical board itself. To start, **8.14(a) and 8.14(b)** show the process that is taken from schematic to board layout to placing the components into the yellow box. The yellow box is our “board”. This part of designing the circuit board was difficult for us

because it took the most time. Rewiring the circuit making sure they don't touch and making sure the components are well connected is important too. Also fitting the circuit into a small frame, trying to get the smallest size possible to not waste space on the circuit board. Once this step is complete, we will then purchase the circuit we designed. Due to the pandemic it might take some time before the circuit board arrives for us to test it without device. Because of that we will start working on the next circuit board just in case we need to make some changes on the board or change the components that are currently being used. In order to meet the deadline, we are trying to be as productive as possible to accomplish our goal of having all of our devices here and ready to be installed by mid-May 2020.

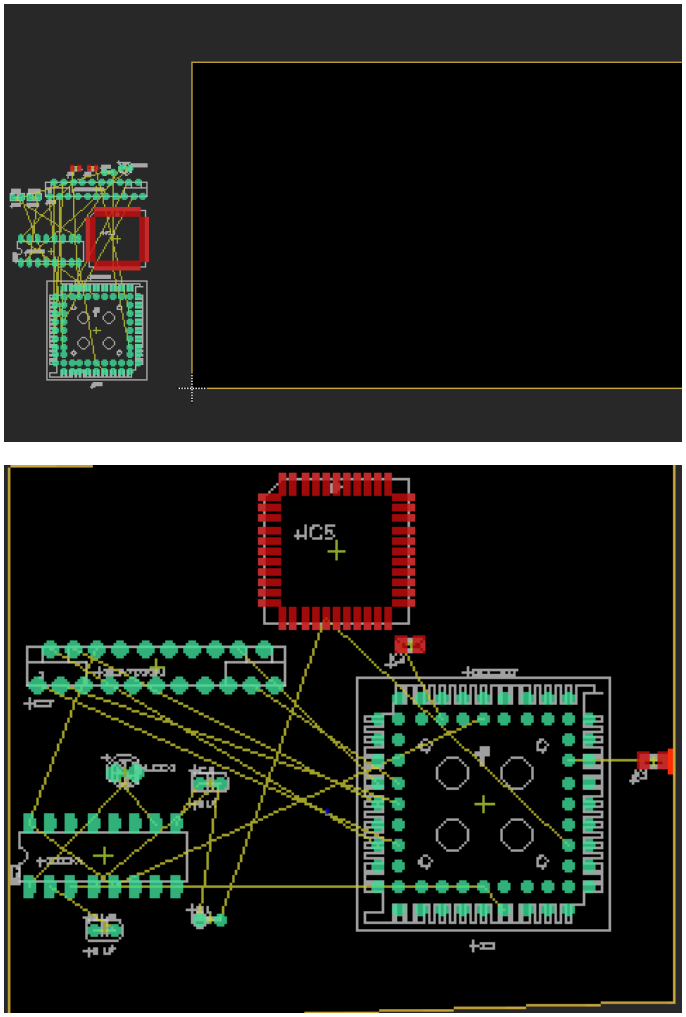


Figure 8.14a/Figure 8.14b: Circuit board preview

9.1 Milestone Discussion

Being able to stay on schedule during the design and development stages of an engineering project are paramount to the success of the project. To successful develop Mr. Painter, the different elements of the projects or the different responsibilities in the project need to be shared and in order to achieve the goals and milestones. The main tasks or responsibilities available for the project include the following: assembling the robot frame, installing the servo motors, connecting the power supply, connecting the microprocessors, installing the canvas, wiring the entire robot and, finally testing and redesigning. This project also has the task of programming since we need to program OpenCV and tell the robot what to paint.

Unfortunately, due to issues beyond our control, the University has moved online completely. This has greatly affected the growth and rate of progress for the entire project. Our goals and milestones must be readjusted in order to achieve the goals at the end of the semester. Social distancing has become a real threat to the success of the project. Tasks such as assembling the robot and programming the robot were going to be done as a team. However, due to the present conditions these collaborative efforts have now become a one-person job. For example, for the assembling the frame of the robot, Alonso Ninalaya has the task of receiving the purchase of the robot frame from Amazon and is responsible for the task of assembling the project. Again, unfortunately this needs to be done with the camera installation, the OpenCV programming and the installation of the power supply.

Another key part of the project is the project documentation. This is because every decision or discussion made needs to be documented as a record of the progression of the project. Consequently, the semester began with each student submitting a project idea. Here all members of the team came up with several ideas for the group project. We all then settled on one idea individually. After that, we then analyzed each idea and picked the best idea based of the time required to complete the project and how demanding the project would be. We then settled on Mr. Pancake; a robot capable of making pancakes from the batter of the pancake mixture.

We spent weeks analyzing the different components of this project. We settled on purchasing the robotic arm rather than 3-D printing it. This comes after a cost analysis of both options. After this, we considered other parts of the design as well such as whether Mr. Pancake was able to make other breakfast meals such as waffles. However, after meeting with our Professor, we realized that there were so many parts of the project we had not thought about. For example, we had no plans of verifying the temperature of the pan when frying the pancake, and we had no specific plans to handle any event of batter spills in the project. We also realized that in checking the temperature of the pan, we would need to include a temperature sensor. Another realization that was made was that the weight of the pan may not be able to be handled by the servo motors. Therefore, because of these reasons, our project idea began to focus on a robot painting arm, in which we decided to call Mr. Painter.

For the situation that we are in we are still making progress rapidly. The robot arm should be constructed by the end of the week. This week Alonso just got the raspberry pi camera and should soon get the raspberry pi 4 microcontroller so now he can start tinkering with robot vision. We are considering even adding a third team, beside robot vision, and robot arm. The reason for the third team is that with social distancing it's much more difficult for the other members to contribute. Alan has the arm, and Alonso has the raspberry Pi and camera, Trudy and Chedlyne and Trudy have nothing to do and that is just not fair to them. We all must meet page quotas and without direct experience, what will they have to write about.

Right now, we are thinking about having the third team specialize in making 3d models in Solidworks to print for the robot arm. This is because while a robot gripper is nice and all, the nature of its design doesn't ensure that the paint brush or the pencil will be held in the same place and the same time or that it will be held securely. Most of the robot arms we saw previously, when we were looking up robot painters, had a specialized part that held the artistic instrument. This allowed for a level of precision for the painting robots since the tip of the brush was always in the same place. Since the position of the tip of the brush always in the same place, its location can be factored into an algorithm through the use of kinematic equations. I looked for these pencil/ paintbrush grippers on Amazon and could not find any. If we are not able to get one it seems we will have to design and print it on our own.

After a few weeks of discussions, the responsibilities and work allocations for this project were clear. Considering the time constraint attached to this project as well as the economic constraints, we were able to come up with a much better strategy to aid us in realizing our goals for the project. Alan was responsible for the construction and assembling of the robot. In a timeframe of about a week, he was able to completely assemble a full working robot. He was responsible for checking each servo motor and whether or not it worked effectively. He was responsible for connecting a power supply and ensuring that the wiring of the robot prevented any shorts or overloads. Prior to assembling the parts of the robot, a lot of research and discussions was placed into determining the type of robot to be purchased. After having purchased the SunFounder Robotic Arm Edge, Alan realized that this robot arm was too weak and flimsy. Since this was an early trial in our schedule for constructing the robot painting arm, this setback did negatively impact the time constraint for the project. The figure below captures the previous robotic arm edge that was used in place of the current robotic arm. From the images below, one can tell that SunFounder robotic arm was frail, fragile or even feeble.

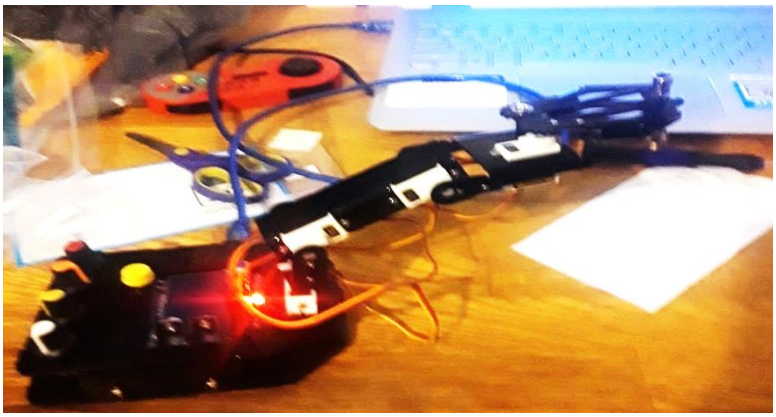


Figure 9.1, captures the SunFounder Robotic Arm Edge

The figures in 6.2.6 capture the step by step process of achieving the robot painting arm. Anything hardware related was the responsibility of Alan, however if there was the need for assistance or support, any team member was willing and able to support. The task of assembling the robot took hours of screwing, numerous minutes of crosschecking and rechecking the wiring of the robot and days of testing the functionality of each servo motor. The flowchart in section 6 captures the step by step operation of the robot arm as well as provide enough documentation on the inner workings of the robot arm. Since the physical robot arm is vital part of the design, it was important that Alan stayed on the track. Alan ordered for the parts of the robot on the 17th of March 2020. He then received it on the 20th of March 2020.

For the software portion of the project, it was the responsibility of Alonso. The software portion of the project involves the camera module and setting up any program that came along with the camera module. In addition, the planning for coding was distributed in terms of the functionality. Top priority was giving to the coding instructions for the servos. This was top priority because the servos controlled the generic movement of the robot. With the absence of the code for the servos, there will be no movement seen in the robot. The movement of the robot is a crucial part of the functionality of the design. In order to write the code for the servos, research on movement and the distance formula was done in order to incorporate these factors into the code. The accuracy of incorporating the distance into the code is crucial as the task of painting a picture (specifically a landscape) requires that the position of the brush on the canvas determines the artwork produced. Therefore, in order to achieve an image that is a replica of the input of the robot arm, the distance formula used in the code needs to be precise and accurate.

To ensure that certain factors such as the color is correctly captured in the output of the design, certain conditions of the camera needed to checked and rechecked. For example, the resolution of the camera is best at 1080 p (although it has a default resolution of 720p). Another condition to be met is how water resistant or waterproof this camera is. As is known, water is the biggest enemy of electronics or electrical technology, and since the robot painting arm would make use of water-based pigments, it is crucial that the camera be waterproof and water resistant. If the camera is not waterproof and is therefore

affected by liquid or fluid, it will affect image captured by the camera thereby in the long run affect the execution of the painting by the robot arm. In addition, the operational conditions of the camera were tested as well. For example, at one-point Alonso noticed that the camera was overheating, however this was temporary and did not affect the results produced by the camera.

Table D: Initial Table of Milestones

Number	Task	Start	End	Status	Responsible
Senior Design 1					
1	Ideas	01/13/2020	01/17/2020	Completed	Group 14
2	Project Selection & Role Assignment	01/20/2020	01/24/2020	Completed	Group 14
Project Report					
3	Initial document-Divide and Conquer	01/26/2020	01/31/2020	In Progress	Group 14
4	Table of Content	02/03/2020	03/06/2020	In Progress	Group 14
5	First Draft	02/17/2020	03/16/2020	In Progress	Group 14
6	Final Document	01/26/2020	04/26/2020	In progress	Group 14
Research, Documentation, &Design					
8	Schematics	03/16/2020	03/27/2020	Researching	Alan &Trudy
9	Solidworks 3D software (Research)	03/30/2020	04/03/2020	Researching	Group 14
10	3D Printing	04/06/2020	04/17/2020	Researching	Group 14
11	Microcontroller	04/20/2020	04/24/2020	Researching	Chey & Alonso
12	Power Supply	04/27/2020	05/01/2020	Researching	Alan &Trudy
13	Servo Motors	05/04/2020	05/15/2020	Researching	Alan &Trudy
14	PCB Layout	05/18/2020	05/22/2020	Researching	Alan &Trudy
15	Camera (Optional)	05/25/2020	05/29/2020	Researching	Chey & Alonso
16	Install time of product	05/25/2020	05/29/2020	Researching	Chey & Alonso
17	Packaging	05/25/2020	05/29/2020	Researching	Group 14
18	**Order & Test Parts**	05/25/2020	05/29/2020	Researching	Group 14
Senior Design 2					
19	**Build Prototype**	06/01/2020	06/05/2020	Researching	Group 14
20	**Finalize & Redesign**	**TBA**	**TBA**	Researching	Group 14
21	**Finalize Prototype**	**TBA**	**TBA**	Researching	Group 14
22	**Peer Presentation**	**TBA**	**TBA**	Researching	Group 14
23	**Final Report**	**TBA**	**TBA**	Researching	Group 14
24	**Final Presentation**	**TBA**	**TBA**	Researching	Group 14

Table E: Current Table of Milestones

Number	Task	Start	End	Status	Responsibility
Senior Design					
1	Ideas	01/13/2020	01/17/2020	Completed	Group 14
2	Production Selection and Role Assignment	01/20/2020	01/24/2020	Completed	Group 14
Project Report					
3	Initial Document-Divide and Conquer	01/26/2020	01/31/2020	Completed	Group 14
4	Table of Content	02/03/2020	03/06/2020	Completed	Group 14
5	First Draft	02/17/2020	03/16/2020	Completed	Group 14
6	60-page document	01/26/2020	03/27/2020	Completed	Group 14
7	Final Document	01/26/2020	04/26/2020	In progress	Group 14
Research Documentation and Design					
8	PCB Layout	03/16/2020	04/26/2020	In Progress	Chey
9	Assembling robot arm	03/16/2020	04/01/2020	Completed	Alan
10	Paint brush attachment	03/16/2020	04/26/2020	In Progress	Alonso& Trudy
11	Camera	03/16/2020	04/01/2020	Completed	Alonso
12	Coding of the robot arm	03/16/2020	06/05/2020	In progress	Group 14
13	Power Supply	03/17/2020	04/01/2020	Completed	Alan
14	Servo motors	03/18/2020	04/01/2020	Completed	Alan
15	Microcontroller	03/19/2020	06/05/2020	In progress	Alan & Alonso
16	Testing Parts	04/16/2020	06/05/2020	In progress	Group 14
17	Packaging	04/26/2020	06/05/2020	In progress	Group 14
Senior Design 2					
18	**Build Prototype**	03/16/2020	06/05/2020	In Progress	Group 14
19	**Finalize and Redesign**	**TBA**	**TBA**	In Progress	Group 14
20	**Finalize Prototype**	**TBA**	**TBA**	In Progress	Group 14
21	**Peer presentation**	**TBA**	**TBA**	In Progress	Group 14
22	**Final Report**	**TBA**	**TBA**	In Progress	Group 14
23	**Final Presentation**	**TBA**	**TBA**	In Progress	Group 14

9.2 Budget and Inventory Discussion

In the beginning of our project we were not doing a painting robot but a pancake cooking robot. We had made a lot of advances until we got to the robot part of the project. We learned that we could not build a robot arm for the money we had. So, we had to redesign the entire project in line with what we could afford. So far Alan Azargushasb has been the main provider of funds for this project. Everything listed below has been paid for by Alan.

Our initial budget I $500 dollars. This is because that is all Alan has. So far, we have spent $250 dollars. We are on our second robot arm. If this one fails, we will return the robot arm parts and buy a professional for up to $350 bucks.

There is also a secondary factor in play at this moment. The entire economy is grinding to a halt. The DOW has dropped below 20,000. The fed chief announced that we can expect up to 20 percent unemployment. In other words, the value of money, our money, the US dollar is in flux. Who is to say we don't have hyperinflation in the next two months?

At this moment, the federal reserve has lowered the amount of money banks has to have in reserve to zero. Millions are losing their jobs. The congress is considering giving 2000 dollars to everyone. All the major retail chains and restaurants are clamoring for a bailout since they cannot get any customers due to the situation at hand. Las Vegas has shut down, Disney has shut down, everyone is shutting down. What does this mean for us? Well, for the moment, 500 dollars is worth 500 dollars. The best course of action for us to get all our goods now before hyperinflation starts and our money becomes worthless. Nevertheless, let us hope for the best. This is a list of and expenses and their purchase dates at the current time.

The first purchase I made was the Gotham Steel Double Pan on Jan 27 2020.I bought it for $14.99 US dollars. This was back when Mr. Painter was still Mr. Pancake. I received the package on Jan 29 2020. The Gotham Steel Double Pan was said to be nonstick, and ideal for making pancakes, omelets, crepes, hamburgers and other such things. In other words, it was perfect for gooey substances that are cooked on both sides and retain the form of a circle. It was supposed to have a "award winning" coating that made sure nothing sticks, so much so that no oil or butter would be needed. The coating was said to be so strong that it was scratch proof and that it could be worked on with metal utensils safely. When testing the pan this was found out to be a lie. There was nothing spectacular about the coating. It needed butter or oil or any other form of grease to operate as do all the other nonstick pan in existence. The pan was made from aluminum alloy, this was said to be to ensure that distribution of heat was even throughout the pan. I think however that it was just because aluminum is a cheap metal that is commonly used in cooking throughout the world. After all the pan was only $14.99 U.S dollars. It was lightweight at only 1.2 pounds and yet heavy for what we were intending it to do. The pan was small with dimensions of 8 x 6 x 2 inches. It was said to be dishwasher safe; I never use my dishwasher, so I never checked out this feature. It was said to have an ergonomic handles that would be cool when touched and countered for a secure grip. In testing this the rubber

handles were not that cool, and the countered grip was not that ergonomic. It was not painful, but it did not rest comfortably in the hand either. The pan was said to be toxin free, free from perfluorooctanoic acid and other perfluorochemicals. The pan had a bronze sheen. There were two center handles with rubber grips that when closed together made a shell where the batter was to be poured. I tested the flip pan by making a pancake with it. The pancake was huge, and yet burned. The shell itself was not sealed, at the right angle the batter could leak. The materials of the flip pan were of low quality. Also the flip pan was heavy and flimsy at the same time. I had planned on returning it but never got around to it on time, so I missed the return by date. I used it three times and my cooking results were mediocre at best. In the end we decided not to go the cooking robot so I wasted 14.99 U.S dollars. After three uses I threw it away.

The second purchase I made was the SunFounder Robotic Arm Edge Kit for Arduino R3 - an Robot Arm to Learn STEM Education for $58.99 U.S dollars. I bought the SunFounder Robotic Arm on February 24, 2020. The SunFounder Robot Arm was designed to be used by people who had little programing experience, and was able to teach how Arduino interfaced with the servos on the way. This was true, though it wasn't the so much the SundFounder Robotic Arm instructing us to code for the servo motors but instead Arduino has built in example files for the robot arm which the SunFounder Robotic Arm uses. The SunFounder Robotic Arm provided no code, but instead pointed you to the direction of the built in examples of the Arduino IDE. The SunFounder Robotic Arm was said to be interactive and it was. The SunFounder Robotic Arm had 4 potentiometers which were used to pivot and control the SunFounder Robotic. The first potentiometer controlled the base servo in moving the arm left to right. The second potentiometer controlled the vertical leaver around the Z axis. It would rotate from being perpendicular to the Z axis to being aligned around the Z axis. In its resting position it was aligned with the Z axis. Vice versa for the third potentiometer. The third potentiometer controlled the secondary leaver around the Z axis. In its resting state it was perpendicular instead of aligned with the Z axis. The fourth and last potentiometer controlled the claw. In its neutral state claw was open. Twisting the screw on the potentiometer caused the claw to close. The SundFounder Robotic Arm was said to have a good range of movement. The claw was to open 3.54 inches at 260 degrees. It was supposed to have 180 degrees of freedom horizontally from on the x and y axis. As well as 180 degrees of freedom for both the primary and secondary levers. This was to allow the robot arm to move freely around its domain. In practice this didn't happen. The servos were horribly weak, so much so the levers would collapse in on themselves without load. The servos were SG90 micro servo and had a weak output torque at 1.4 kilograms per cm at 4.8V. The claw was said to be wsy to use due to its double layered design with pointed edges and pointed end. These edges were said to aid in gripping objects. This didn't happen, as everything I tried to grab slipped out the claw, and the claw would bend and warp under load. It had the characteristics and integrity of a plastic spork. The claw was just as effective as using two plastic butter knives as chopsticks. Since the thing warped under load whatever grip those edges provided was suspect. The robot arm was said to be easy to build. I wouldn't say that was true. It took me all day to build this thing. They were

gracious enough to provide a screwdriver for the assembly. Many of the pieces did not fit as the instructions said they would and so I had to manhandle the plastic. The instructions were clear enough to build it though. First, I assembled a base. Then I screwed the Arduino Uno into the base. Then I assembled a secondary base over the first base. The secondary base plugged into the pins for Arduino, so I never need to plug in the potentiometers directly myself. I then constructed the robot levers placing their servos in their place and attached the robot claw. After testing the product was deemed by me to be junk. I returned it on March 2, 2020 and got my money back as an Amazon coupon.

After buying the SunFounder Robotic Arm I knew I need something more substantial. I need a robot arm that was stronger. I needed a robot arm that wasn't made of plastic, that was made of metal but at the same time lightweight and cost efficient. I needed servers with more torque, and I needed a new micro controller to control them, and basic electrical equipment to connect them.

For the micro controller I decided to buy the ELEGOO Mega 2560 Project The Most Complete Ultimate Starter Kit w/Tutorial Compatible with Arduino IDE. I bought it on March 3, 2020. I received it two days later on March 5 2020. I bought it for $59.99 U.S dollars on Amazon. The kit features

- 1 Mega 2560 Controller Board
- 1 USB Cable
- 1 Breadboard
- 65 Jumper Wire
- 1 IC 74HC595 8-Bit Serial to Serial/Parallel IC
- 1 Active Buzzer
- 1 Tilt Ball Switch
- 1 Photo resistor
- 5 Yellow LED
- 5 Blue LED
- 5 Green LED
- 5 Red LED
- 5 White LED
- 2 RGB LED
- 5 Button (small)
- 10 Resistors at 10 ohms
- 10 Resistors at 100 ohms
- 10 Resistors set at 220 ohms
- 10 Resistors set at 220 ohms (330R)
- 10 Resistors set at 1 kilo ohms (1K)
- 10 Resistors set at 2 kilo ohms (2K)
- 10 Resistors set at 5.1 kilo ohms (5K1)
- 10 Resistors set at 10 kilo ohms (10K)

- 10 Resistors set at 100 kilo ohms (100K)
- 10 Resistors set at 1 mega ohms (1M)
- 20Female-to-male DuPont Wire.
- 1 LCD1602 Display module with pin header
- 1 DC Motor and L293D Transistor and Fan Blade
- 1 74HC595 Shift Register
- 1 SG90 Servo Motor
- 1 Power Supply Module
- 1 ULN2003 Stepper Motor Driver Module
- 1 GY-521 Module
- 1 Rc522 RFID Module
- 1 Prototype Expansion board
- 1 Infared Receiver Module
- 1 DHT11 Temperature and Humidity Module
- 1 Rotary Encoder Module
- 1 Ultrasonic Sensor
- 1 DS1307 RTC Module
- 1 HC-SR01 PIR Motion Sensor
- 1 Sound Sensor Module
- 2 Ten Killo Ohm Potentiometers
- 1 Joystick Module
- 1 Water Level Detection Sensor Module
- 1 Digital 7 Segment Display
- 1 Four Digit 7 Segment Display
- 1 L293D Quadruple Half-H Drivers
- 1 Remote
- 1 MAX7219 Serially Interfaced, 8-Digit LED Display Drivers
- 1 Passive Buzzer
- 1 Membrane Switch Module
- 1 9V Battery with DC input Jack
- 1 Five Volt Relay
- 1 Nine Volt One Amps Power Supply Adapter
- 2 Electrolytic Capacitor (100 micro Farads at 50 Volts)
- 2 Electrolytic Capacitor (10 micro Farads at 50 Volts)
- 5 Twenty Two Pico Farad Ceramic Capacitors
- 5 One Hundred and Four Pico Farad Ceramic Capacitors
- 5 S8050 NPN Low Voltage High Current Small Signal Transistor
- 5 PN2222 NPN Bipolar Transistors
- 1 Thermistor
- 1 Diode Rectifier
- 2 Photoresistors (Photocells)

- 1 Carrying Case for the above components
- 1 PDF Manual with Sample Code for various projects using the components listed above.

I bought the kit because it had a lot of things I needed at the time, for example I didn't have breadboard and without a breadboard you cannot do any electrical tests whatsoever. I didn't have resistors, I didn't have capacitors. I did not have a power supply. I did not have a micro controller. I did not have any sensors and if I did I had no sample code on how to use them. I wanted a one and done package that would get me most if not all what I needed electronically, this was a good base to start on.

For the robot arm I decided on the diymore 6Sets MG996R 55g Digital Metal Gear Servo Motor + MG995 MG996R Metal Servo Arm 25T Disc Metal Horns for Robot Arm RC Helicopter Airplane Car Boat Robot on Amazon. I bought this robot arm kit for $82.99 U.S dollars. I bought it on the same day I bought the micro controller kit, on March 3, 2020. I received the package on March 5, 2020. This kits robot arm is made of aluminum . The mounting hole spacing is approximately 14 millimeters. The Servo disc diameter is approximetly 20 millimeters. Its servo brackets dimensions are 40 x 20 x 36 millimeters. This kit came with …

- 1 Aluminum Clamp Claw
- 1 L- type servo bracket
- 3 U-type robot waiste brackets
- 4 Long U-type servo brackets
- 4 Miniature Ball Radial Bearing
- 6 Multi Functional Servo brackets
- 6 MG996 55g Servos
- 6 sets of aulunimum servo horns
- 4 sets of Round head M3*10 screws and M3 nuts
- 20 Sets of Round head M3*8 screws and M3 nuts
- 24 Sets of Fixed head M4*10 screws and M4 nuts
- 30 Sets of Round head M3*6 screws and M3 nuts

When researching this robot kit I watched a video on how to assemble it on YouTube. The guy who was building the kit said that the package was incomplete and that it has parts missing and that I had to also buy another package with it. I followed his advice and bought the Mallofusa Servo Arm Horn Metal Aluminum 25t for Rc Car Helicopter Round Mg945 Mg995 Mg996 kit on Amazon. It cost me $7.49 US dollars. I bought it the same day I bought the robot arm, March 3, 2020. I received it on the same day I received the robot arm kit on March 5. It features

- 5 Metal Servos Mount
- 20 Screws

Since I bought this secondary kit I now have excess parts, excess screws and excess servo mounts, but since I did construct the robot arm I am fine with the wasted parts. It took me a day to build the robot arm. The quality of the robot arm was good. The aluminum was strong enough to provide the rigidity that was lacking in the last robot arm. The aluminum however is prone to getting scratches. I already have a few scratches on my robot arm. I didn't expect this to happen, but it is no big deal. There is no structural damage to the robot arm at this time. I am glad I went this route as the other robot arms were in the $120- $140 US dollars price range, another one I was looking at was being sold for around $350 U.S dollars. The robot arm kit I purchased was cost efficient for my budget.

The next thing I purchased was the Raspberry SC15184 Pi 4 Model B 2019 Quad Core 64 Bit WiFi Bluetooth (2GB) on Amazon. I purchased it for $41.98 US dollars. I purchased it on March 15, 2020 and it was delivered to Alonso's house on Mar 21, 2020. The Raspberry Pi features a Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit System on Chip running at 1. 5 Giga Hertz. Two 4 Giga Hertz and 5 Giga Hertz IEEE 802. 11b/g/n/ac wireless LAN, Bluetooth 5. 0, BLE. Two USB 3.0 ports, and USB 2. 0 Ports. Two micro HDMI ports supporting up to 4Kp60 video resolution. A micro SD card slot for loading operating system and data storage. From what I have heard from Alonso, the Raspberry Pi is working out for him. I bought the Raspberry Pi because the Arduino was too week to handle the image processing needed to get Mr. Painter working. The Arduino is only an 8-bit micro controller, and lacks the storage needed to even store a modern jpeg, even if it was compressed. The Raspberry Pi is intended to the brains of our project.

The next thing I bought was the Raspberry Pi Camera Module V2-8 Megapixel,1080p on Amazon. I bought it for $27.91 US dollars. I bought it on March 17, 2020. It was delivered to Alonso on May 31, 2020. The Raspberry Pi Camera Module features an 8 megapixel image Sensor from Sony, the IMX 219. It can take a picture at the resolution of 3280 x 2464 pixel. The Raspberry Camera can record 1080p video at 30 frames per second. It can also record 720p video at 60 frames per second or 640 p video at 60-90 frames per second. I bought this so Mr. Painter could see the canvas, images, colors, etc. Due to social distancing I can not meet my teammates. We assigned Alonso to be in charge of robot vision, but if we can not meet up how are we going to synchronize the robot vision with the robot arm. Due to the circumstances we might just have to double our investment, and buy a secondary Raspberry Pi and Raspberry Pi Camera Module. Our workflow will go like this, I would ask Alsonso to do something and he would write the code, send it to me, and then I could implement the code since I have the robot arm. Right now, I have spent $27.91 US dollars on the camera plus $41.98 US dollars on the Raspberry Pi. With the doubling of the Raspberry Pi and the Camera the total Cost would come to $139.78 U.S Dollars. Since I have not purchased a second Raspberry Pi or the Camera, I have not included it in the budget, but it is an option, and a likely one at that with the way things are going.

When building the robot arm, I needed some tools to construct the project. When watching the assembly video, I saw the guy had plier and screwdriver. I had neither so I went to home depo to buy some. After looking around I eyeballed which screwdriver and which plier I should get. I bought a plier for $7.97 U.S dollars. I bought the screwdriver for$6.97

U.S dollars. I also purchased a base for the robot arm made out of pine wood for $10.35 U.S dollars. I did this on March 17, 2020. After taxes my total came to $26.94 U.S dollars. The screwdriver worked but it was oversized for the screws I was screwing. The same goes for the plier, it was oversized and too fat, the plier has a snub nose instead of a narrow beak as I saw in the video. When I purchased the plier I didn't think anything of it, but only after the fact do, I realize the error of my ways. The snub nose got in the way many times of holding the metal nuts. I got it too work with a bit of finesse though in the end.

The next thing I purchased was the Kaiweets HT118A Digital Multimeter on Amazon. I bought is for $35.99 U.S dollars. I bought it on March 23, 2020. I received it on Mar 25, 2020. The package came with

- 1 Digital Multimeter
- 1 Test Leads
- 2 AA Battery
- 1 User Manual
- 1 Thermocouple

The multimeter can measure: DC voltage, AC voltage, DC current, AC current, resistance, capacitance, frequency, duty-cycle and temperature and test diodes. It has an LCD screen that can light up for ease of use in areas with low light. The screen goes read when in contact with a live wire. It can handle up to six hundred volts and is encased in silicon to protect from fall damage. I bought this because when I was testing the servos on the breadboard I realized I had no way to measure the voltage or the current. I had no way to tell is resisters or electrical components were dead. I had no way to tell if something short circuited. I didn't want to buy this but because the University of Central Florida is under lockdown I have no lab where I can test things using lab equipment, so I bought my own portable multimeter.

The next thing I bought was the 5V 15A 75W Power Supply on Amazon. I purchased it on March 23, 2020. I received in in the mail on Mar 27, 2020. The power supply can take an input of one hundred to two hundred and forty volts AC at 50/60 hertz and outputs 5 volts DV. It output current is up to 15 amps. It has a working temperature range of 0-40 degrees Celsius with translates to a range of thirty two to one hundred and four degrees Fahrenheit. It meets the safety standards of the Peoples Republic of China, the European Union, and the United States. I perchesed this power supply because I was worried I would not be able to get the most out of my servos with the power supply that came with the microcontroller kit, which was rated at 9 volts 1 amp max. The power supply comes with a 2.1millimeter output plug. It also has an adapter that allows me to plug in wires into the power supply. This allows me to run the power supply directly on my breadboard.

The next thing I purchased was the Expo 80675 EXPO Low-Odor Dry Erase Set on Amazon. I purchased this on April 5, 2020. It was delivered to me on April 8, 2020. The set features markers that have nontoxic ink that is quick drying. The ink also is said to be low odor so as not to disturb the user. The markers are said to be smear proof. The colors are said to be so bright that they are easy to see from a distance and have consistent quality. The markers have a pointed tip which allows for fine detailed lines. The ink is

supposed to erase easily. The markers are made in the United States of America. The set comes with 5 markers, 2 are black, one is red, one is blue, the last is green, an eraser, and a cleaning solution. The reason I got this was because canvas is expensive, and ink is expensive, I wanted a low cost way to experiment with the robot arm. So, I chose to use whiteboard marker on a whiteboard.

I also purchased a RIOFLY Small Dry Erase Board - 16" x 12" Magnetic Double-Sided Desktop Whiteboard Portable Easel with Stand & Holder on Amazon. I purchased it the same day I purchased the markers on April 5, 2020. I received it on the same I received the markers on April 7, 2020. The whiteboard is a double-sided whiteboard with a metal handle and a silver frame. It folds out into a triangle and is portable.

The next thing I purchased was the CAMVATE Universal C-Clamp, Aluminum Support Clamp Desktop Mount Holder Stand with 1/4-Inch- 20 & 3/8-Inch-16 metal female socket on Amazon. I bought it on April 5, 2020. I received it on April 8, 2020. I bought it create a makeshift apertus to attach the markers to the servo horn. I haven't tried it out yet so I don't know if it works or not. The clamp cost me $11.99 U.S dollars. The clamp has a hole in the center with groves. In theory I could take one of my excess screws and plug it into the servo horn. Time will tell.

The next thing I bought was a potentiometer knob Kit on Amazon. I bought it on April 9, 2020 for $6.99 U.S dollars. I received it on April 9, 2020. The potenitometers are made out of carbon. The kit features

- 1 2K ± 20% potentiometer
- 1 5K ± 20% potentiometer
- 1 10K ± 20% potentiometer
- 1 20K ± 20% potentiometer
- 1 50K ± 20% potentiometer
- 1 100K ± 20% potentiometer
- 1 250K ± 20% potentiometer
- 1 500K ± 20% potentiometer
- 1 1M ± 30% potentiometer
- 1 Carrying Case

The reason I got this was because the potentiometers I got with the original microcontroller kit were faulty and didn't work well. Also, resisters by themselves are small things that can get lost easily, the size and heft and variability of these potentiometers means they wont get lost, and they have multiple uses. I plan on using these potentiometers to do further tests on the servos.

The next thing I bought was the EDGELEC 120pcs Breadboard Jumper kit on Amazon. I bought it for $5.79 on April 5, 2020. I received it on April 8, 2020. The kit includes

- 40 pin Female to Female jumper wires
- 40pin Male to Female jumper wires
- 40pin Male to Male jumper wires

In total that is 120 pieces of Dupont wire. The reason I got this was because I realized my wires weren't long enough to accommodate the full range of motion of the robot arm. If I were to move my robot arm too much the wires would pop out of the breadboard, or they would get tangled. This isn't safe.

The next thing I bought was the Xubox Paint Brushes Set. It cost me $7.59 U.S dollars. I bought it on April 5, 2020. I received the paint brush set on April 23, 2020. The brush fibers are made of nylon. The handles are wooden with an aluminum finish. The kit features ten different brushes of varying sizes and shapes ranging from pointed round, angular, filbert, flat and liner / rigger. I purchased this because one of the goals of Mr. Painter is to paint and you can't do that without paint brushes. I also purchased a set of watercolor paint to be used with paint brushes. I purchased it on the same day I purchased the paint brushes, on April 5, 2020. The watercolor paint arrived earlier then the brushes, on April 7, 2020. I bought the watercolor paint for $7.49 U.S dollars. The paint is acid free and the kit comes with 36 different colors. The kit seemed a good value for the money.

The next thing I baught was the SanDisk Ultra 32GB MicroSDHC UHS-I Card with Adapter - 98MB/s U1 A1 - SDSQUAR-032G-GN6MA. I purchased this on April 5, 2020. It cost me $7.99. I received it on April 7, 2020. I got this because Alonso said he needed it for the robot vision side of the project. As of this moment I have spent $385.95 dollars. If I get another raspberry pi and and camera module my expensives would rise to $463.83. I am still within the budget range but we are nearing the end of the funds set for this project.

Table F: budget

Item	Budget	Purchased	Cost
Power Supply 1	$15	Yes	$23.99
Bread Board	$10	Yes	Included in Bundle
Wires/Resistors	$10	Yes	$5.79
Micro Processor 1 Raspberry Pi 4	$150	Yes	$59.99
Micro Processor 2 Arduino Mega 2560		Yes	$41.98
Wooden Base	$15	Yes	10.35
Raspberry Pi Camera Module V2-8 Megapixel,1080p	$25	Yes	$27.91
Robot Arm Kit parts	$100	Yes	$90.48
Sensors	$20	No	
Tools • Screwdriver Set • Clasp to hold nuts in bolts in place while screwing the screws	$15	Yes	$14.94
Multimeter	$20	Yes	$35.99
Others This is stuff to get • PCB • Perhaps an AtMega328P • AC boot jack for the bread board	Whatever is left from the budget	No	N/A
Sum			$385.95

10 Summary

Senior design serves as the last and final class to mark one's engineering journey here at the University of Central Florida. It also provides students with the opportunity to be creative and to apply the knowledge acquired from the institution. For our senior project, the team members of this group decided to select a project that possessed enough technical complexity and allowed each member of the group to apply their majors. With two computer engineers and two electrical engineers, we settled on a project that bridge these two different fields. This project was the Robot painting arm. The robot painting arm design comprised of the right amount of computer and electrical engineering expertise on demand. With the robot painting arm, we will be required to program the movement of the servos and the input of the camera to ensure the success of the design. We would also need to connect the robot arm to the right power supply and make accurate electrical connections between components in order to supply power and transmit information. In deciding the goal of our project, there was a unanimous vote that the robot arm would be tasked to paint landscapes, with the image of the desired landscape inputted through the camera.

After deciding on the goal of the project, the engineering requirements followed. The team created a list of the hardware requirements needed for the design, this list was mainly upon with time and economic constraints in mind. For example, instead of following our initial plan to 3-D print the robot frame, the economic decision made was to purchase a robot frame from amazon and to assemble the parts. Other hardware components such as the camera, the power supply and the servos have since then been purchased. In order to have a deeper understanding of the project and its requirement, the team members conducted a lot of research on similar existing ideas. Doing this allowed us to understand the scope of the project and have a step by step draft on how to achieve our design. One crucial part of the project that had to be designed and 3-D printed was the paint brush attachment. This is a valuable part of the design since without the paint brush attachment, we are uncertain of whether a paint brush or pencil can be used on the robot. Specific brushes have been selected from amazon and the average size of these brushes served as the foundation for the calculation and design of the paint brush attachment.

Despite having all these goals and expectations for our design, it was important that we followed the standards provided for engineers. We discussed the different available standards which included standards from the Federal Communications Commission, python coding standards, USB standards, C++ standards, as well as IEEE standards. The constraints that exist with this project are also discussed, we mentioned the economic, and time constraints presented in this project. Here we mentioned how the current pandemic has affected the timeline of our project and changed the milestones associated with this project. For the economic constraints of this project, we discussed that being students, we were limited economically and have had to make conscious decisions that would prevent overspending. Some other constraints discussed in the report include environmental, social and political constraints. For environmental constraints, we discussed sustainable goals and efforts we plan to achieve by making

economic choices and having a recyclable plan for the robot painting arm once the project was completed. We also discussed the social constraint which centered around the unemployment issue artistes and painters might face, if this painting arm was produced on a larger scale. Ethical, health and safety constraints were also discussed for this project.

Three different prototypes were analyzed, and flowcharts were designed to demonstrate the step by step operations of these prototypes. After purchasing the robot frame in design 1, it was realized that the frame was too weak to support the goal of the project. Design 2 and design 3 are very similar in their operation however, they diverge in ideas since the second design does not involve a laptop however the third design includes a laptop. The camera module which acts as the input for the robot painting arm was thoroughly analyzed and the mechanism associated with the camera's functionality was discussed. The microcontroller which serves as the communication hub between the camera and the servo motors was discussed. The language of choice for this project is the Python and OpenCV is to be used as the language for the camera. A number of servo tests were also conducted to contribute to the full functionality of the robot painting arm. These six servos were tested with the power supply in order to prevent the case of any negative surprises in the future. Section 6 also features the step by step assembling of the robot frame with images to capture every stage.

The flowchart in section 7.5 captures the step by step operation of the robot painting arm. It is seen that the power from the power supply goes into a DC converter, this power then goes into the microcontroller. The microcontroller sends signals to the camera, the camera is then instructed to take pictures of the artwork to be recreated. This image captured serves as the input for the operation of the robot painting arm. The microcontroller then reads the image and this image is transformed into arrays. The original image is saved, and then other pictures are taken of the input and compared. Once the images match up, signals are sent to the servos and the execution of the painting task is made. The printed circuit board (PCB) which serves as mechanical and electrical support for the robot painting arm and this involves resistors, capacitors, a battery, diodes, switches, transistors, and some LEDs. The design for the printed circuit board included a microchip, a voltage regulator, a DC power jack, an XBEE pro, a serial connector, a motor driver and other more additions. To conclude, the milestones and budget of the project was discussed in order to successfully complete the project.

Appendices

Appendix A – References and resources

"PWN: PULSE Width Modulation: What is it and how does it work?: Retrieved from: https://www.analogictips.com/pulse-width-modulation-pwm/

HYPERLINK "http://www.camera-module.com/blog/what-is-the-difference-between-mipi-csi-2-vs-csi-3.html"Ada, L. (n.d.). Adafruit Motor Shield V2. Retrieved from https://learn.adafruit.com/adafruit-motor-shield-v2-for-arduino/powering-motors

"A.I. Artists Exploring Creative Algorithms: My Robots Paintings are a record of my progress." cloudpainter. n.d. Web. 03 April 2020. www.cloudpainter.com

Anthony, S. (2015, April 23). ARM details its new high-end CPU core, Cortex A72. Retrieved from https://arstechnica.com/gadgets/2015/04/arm-details-its-new-high-end-cpu-core-cortex-a72/

Arm Ltd. (n.d.). Cortex-A72. Retrieved from https://developer.arm.com/ip-products/processors/cortex-a/cortex-a72

Art, A. R. (2019, April 16). Step by step acrylic painting landscape/ learn how to paint landscape in acrylic - video dailymotion. Retrieved from https://www.dailymotion.com/video/x75xgbu

Bossmann, J. (2016, October 21). Top 9 ethical issues in artificial intelligence. Retrieved from https://www.weforum.org/agenda/2016/10/top-10-ethical-issues-in-artificial-intelligence/

DATASHEET-Raspberry Pi 4 Model B. (2019, June). Retrieved from https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0_preliminary.pdf

Fagerness, T. (2015, November 2). How to Build a Robot - PCB Design. Retrieved from https://es.scribd.com/document/251806583/Digit-November-2014

Frumusanu, A. (2015, April 23). ARM Reveals Cortex-A72 Architecture Details. Retrieved from https://www.anandtech.com/show/9184/arm-reveals-cortex-a72-architecture-details

Guido Van Rossom, Barry Warsaw, and Nick Coghlan. "PEP 8 – Style Guide for Python Code." python ™. 05 July 2001. Web. 03 April 2020. https://www.python.org/dev/peps/pep-0008/

How Do Servo Motors Work? (2002). Retrieved from https://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work.html

IEEE Standards. (n.d.). Retrieved from https://www.ieee.org/standards/index.html

Kane, K. (2019, October 3). Impressionist Realism in Landscapes. Retrieved from https://www.outdoorpainter.com/impressionist-realism-landscape-painting/

Liljegren, J. (2017, September 15). UNITED STATES DEPARTMENT OF LABOR. Retrieved from https://www.osha.gov/dts/osta/otm/otm_iii/otm_iii_4.html#heat_relatedillness

Marks, R. (2019, June 6). Standards from IEEE 802 Unleash the Wireless Internet. Retrieved from http://www.ieee802.org/16/docs/01/80216c-01_10.pdf

Mehta, I. (2020, March 6). How the IEEE 802 group helped shape the modern internet with Ethernet and Wi-Fi protocols. Retrieved from https://thenextweb.com/tech/2020/03/06/how-the-ieee-802-group-helped-shape-the-modern-internet-with-ethernet-and-wi-fi-protocols/

OpenCV Programming the Raspberry Pi:Tutorial-18 Identify Pixels with Given Color with Python. (2014). Retrieved from https://www.youtube.com/watch?v=rVTab3WgynU&t=223s

PIC18FXX2 Data Sheet - Microchip Technology. (n.d.). Retrieved from http://ww1.microchip.com/downloads/en/DeviceDoc/39564c.pdf

PIC18FXX2 Data Sheet High-Performance, Enhanced Flash Microcontrollers with 10-Bit A/D. (2006). Retrieved from http://ww1.microchip.com/downloads/en/DeviceDoc/39564c.pdf

Pointillist Painting Robot Arm. (2017, October 13). Retrieved from https://www.instructables.com/id/Pointillist-Painting-Robot-Arm/#discuss

Pulse Width Modulation Used for Motor Control. (2018, February 16). Retrieved from https://www.electronics-tutorials.ws/blog/pulse-width-modulation.html

Pulse Width Modulation. (n.d.). Retrieved from https://learn.sparkfun.com/tutorials/pulse-width-modulation/all

Raspberry Pi Camera Board v2.1 (8MP, 1080p). (n.d.). Retrieved from https://uk.pi-supply.com/products/raspberry-pi-camera-board-v2-1-8mp-1080p

Raspberry Pi Camera Module V2-8 Megapixel,1080p. (n.d.). Retrieved from https://www.amazon.com/Raspberry-Pi-Camera-Module-Megapixel/dp/B01ER2SKFS/ref=sr_1_3?crid=19UWXNDDGO9UA&keywords=raspberry pi camera module&qid=1584477575&sprefix=ras,aps,186&sr=8-3

Raspberrypi, J. H. (2018, April 24). raspberrypi/documentation. Retrieved from https://github.com/raspberrypi/documentation/tree/master/hardware/camera

Ravi, Thompson, J., & Thompson, J. (2017, December 25). How to Design PCB using Eagle (Printed Circuit Board Layout). Retrieved from https://www.electronicshub.org/pcb-design-eagle/

Robert P. J. Day. “C++ Programming Style Guidelines.” Geotechnical Software Services. January 2011. Web. https://geosoft.no/development/cppstyle.html

Semiconductors, P. (1992, January 8). DataSheet 80C562/83C562 Single-chip 8-bit microcontroller. Retrieved from http://www.elektronikjk.pl/elementy_czynne/IC/80C562.pdf

Thingiverse.com. (2017, May 10). OWI Robot Painting Project by Doggerino. Retrieved from https://www.thingiverse.com/thing:2305426/files

UNITED STATES DEPARTMENT OF LABOR. (n.d.). Retrieved from https://www.osha.gov/SLTC/robotics/hazardrecognition.html

"What is the difference between MIPI CSI-2 and CSI-3?" Camera-Module.com. 15 April 2019. Web. http://www.camera-module.com/blog/what-is-the-difference-between-mipi-csi-2-vs-csi-3.html

"Should you buy the Raspberry Pi 4? Web: https://sandbox.spcollege.edu/index.php/2019/10/should-you-buy-the-raspberry-pi-4/

"What is the definition of SolidWork". Retrieved from blog:https://www.quora.com/What-is-solidwork

"Modeling Technology of SolidWork" Web: https://en.wikipedia.org/wiki/SolidWorks

"Five reasons to use SolidWorks" Web: https://www.cadtek.com/5-reasons-use-solidworks/

"Power" Web: https://www.solidworks.com/sw/why-SOLIDWORKS/solidworks-power.html

"Open CV about" Web: https://opencv.org/

"Documentation for OpenCV" Web: https://picamera.readthedocs.io/en/release-0.6/quickstart.html

"Documentation" Web: https://docs.opencv.org/2.4/modules/core/doc/operations_on_arrays.html

"Why Raspberry PI" Web: https://www.androidcentral.com/why-you-should-buy-raspberry-pi-4-and-what-you-can-do-it

"How To Build 6 DOF Robot Arm Kit" Web: https://www.youtube.com/watch?v=bkyzbqoIi18

"Raspberry Pi GPIO Pinout: What each pin does on Pi 4, earlier models" Web: https://www.tomshardware.com/reviews/raspberry-pi-gpio-pinout,6122.html

"The advantages of 3D printing" Web: https://www.3dhubs.com/knowledge-base/advantages-3d-printing/

"Artist's Loft 36 Color Fundamental Watercolor Pan Set with Paint Brush – Watercolor Set for Beginners and Professionals" Web: https://www.amazon.com/gp/product/B013S1A4Q2/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1

“SanDisk Ultra 32GB MicroSDHC UHS-I Card with Adapter - 98MB/s U1 A1 - SDSQUAR-032G-GN6MA" Web: https://www.amazon.com/gp/product/B073JWXGNT/ref=ppx_yo_dt_b_asin_title_o01_s00?ie=UTF8&psc=1

“Xubox Paint Brushes Set, 10 Pieces Round Pointed Tip Nylon Hair Artist Acrylic Paintbrushes, Paint Brushes for Acrylic Painting Oil Watercolor Face Nail Body Art Craft, Miniature & Rock Painting, Blue” Web: https://www.amazon.com/gp/product/B06XRYMR9L/ref=ppx_yo_dt_b_asin_title_o02_s00?ie=UTF8&psc=1

“EDGELEC 120pcs Breadboard Jumper Wires 10cm 15cm 20cm 30cm 40cm 50cm 100cm Wire Length Optional Dupont Cable Assorted Kit Male to Female Male to Male Female to Female Multicolored Ribbon Cables” Web: https://www.amazon.com/gp/product/B07GD2BWPY/ref=ppx_yo_dt_b_asin_title_o03_s00?ie=UTF8&psc=1

“MCIGICM 5K 10K 50K 100K 500K Potentiometer knob Linear Potentiometer WH148 3Pin 15mm Shaft with Nut and Washe” Web: https://www.amazon.com/gp/product/B0791GYLBZ/ref=ppx_yo_dt_b_asin_title_o04_s00?ie=UTF8&psc=1

“RIOFLY Small Dry Erase Board - 16" x 12" Magnetic Double Sided Desktop Whiteboard Portable Easel with Stand & Holder for Kids Drawing Teaching Memo Office Home” Web: https://www.amazon.com/gp/product/B082LZY22F/ref=ppx_yo_dt_b_asin_title_o05_s00?ie=UTF8&psc=1

“CAMVATE Universal C-Clamp,Aluminum Support Clamp Desktop Mount Holder Stand with 1/4-Inch- 20 & 3/8-Inch-16 metal female socket” Web: https://www.amazon.com/gp/product/B018RLY6B2/ref=ppx_yo_dt_b_asin_title_o05_s01?ie=UTF8&psc=1

“Expo 80675 EXPO Low-Odor Dry Erase Set, Fine Point, Assorted Colors, 7-Piece with Cleaner” Web: https://www.amazon.com/gp/product/B00006IFIM/ref=ppx_yo_dt_b_asin_title_o05_s01?ie=UTF8&psc=1

“5V 15A 75W Power Supply 100V-240V or 110V - 220V AC to DC Adapter 5V 15 amp Switching Converter Charger 5.5x2.1mm Plug for WS2811 WS2812B WS2813 2801 LED Strip Pixel Lights” Web: https://www.amazon.com/gp/product/B07K9Q4DV1/ref=ppx_yo_dt_b_asin_title_o06_s00?ie=UTF8&psc=1

“KAIWEETS Digital Multimeter TRMS 6000 Counts Ohmmeter Voltmeter Auto-Ranging Fast Accurately Measures Voltage Current Amp Resistance Diodes Continuity Duty-Cycle Capacitance Temperature(LED Jacks)” Web: https://www.amazon.com/gp/product/B07SHLS639/ref=ppx_yo_dt_b_asin_title_o07_s00?ie=UTF8&psc=1

"Raspberry Pi Camera Module V2-8 Megapixel,1080p" Web: https://www.amazon.com/gp/product/B01ER2SKFS/ref=ppx_yo_dt_b_asin_title_o09_s00?ie=UTF8&psc=1

"Raspberry SC15184 Pi 4 Model B 2019 Quad Core 64 Bit WiFi Bluetooth (2GB)" Web: https://www.amazon.com/gp/product/B07TD42S27/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1

"diymore 6Sets MG996R 55g Digital Metal Gear Servo Motor + MG995 MG996R Metal Servo Arm 25T Disc Metal Horns for Robot Arm RC Helicopter Airplane Car Boat Robot" Web: https://www.amazon.com/gp/product/B07MDFGHMC/ref=ppx_yo_dt_b_asin_title_o01_s00?ie=UTF8&psc=1

"Mallofusa Servo Arm Horn Metal Aluminum 25t for Rc Car Helicopter Round Mg945 Mg995 Mg996 Silver/Red/Blue (Silver)" Web: https://www.amazon.com/gp/product/B00NOGMK3M/ref=ppx_yo_dt_b_asin_title_o01_s01?ie=UTF8&psc=1

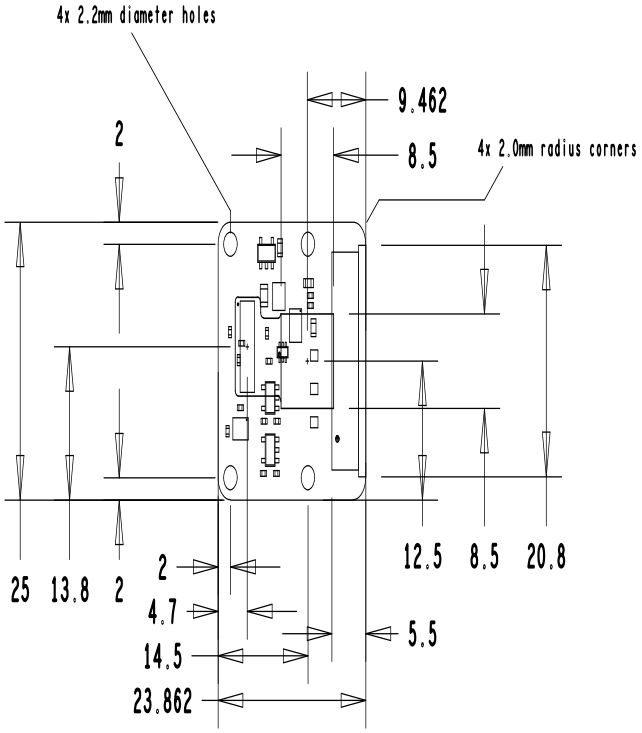
"ELEGOO Mega 2560 Project The Most Complete Ultimate Starter Kit w/Tutorial Compatible with Arduino IDE" Web: https://www.amazon.com/gp/product/B01EWNUUUA/ref=ppx_yo_dt_b_asin_title_o01_s01?ie=UTF8&psc=1

"SunFounder Robotic Arm Edge Kit for Arduino R3 - an Robot Arm to Learn STEM Education(101 Pieces)" Web: https://www.amazon.com/gp/product/B07S6M3KVF/ref=ppx_yo_dt_b_asin_title_o02_s00?ie=UTF8&psc=1

"Gotham Steel Double Pan – Nonstick Copper Easy to Flip Pan with Rubber Grip Handles for Fluffy Pancakes, Perfect Omelets, Frittatas, French Toast and More! Dishwasher Safe" Web: https://www.amazon.com/gp/product/B07BDMHQR7/ref=ppx_yo_dt_b_asin_title_o03_s00?ie=UTF8&psc=1

Appendix B - Datasheets

Data Sheet of the Raspberry Camera



Raspberry Pi
www.raspberrypi.org
© Raspberry Pi 2015

TITLE	RASPBERRY PI CAMERA MODULE V2.1		
DATE	12/11/2015	REF	RPI-CAM-V2_1
DRAWN	Mike Stimson	APVD	James Adams