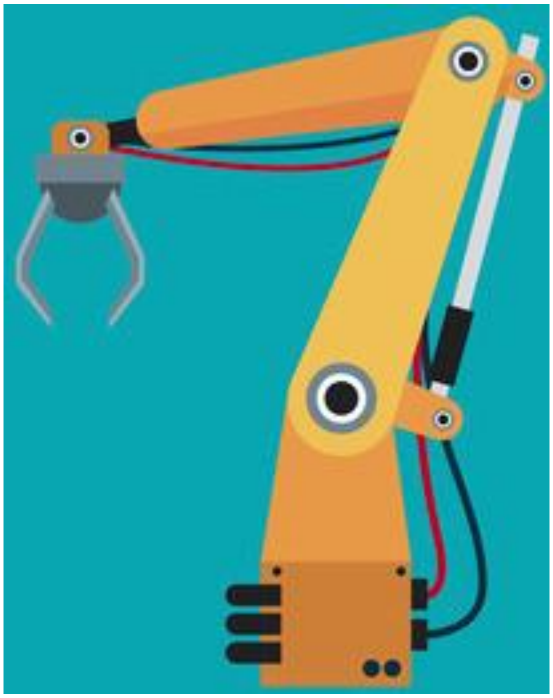
Robotic Flange Assembly



Project & Group Identification Document

Interdisciplinary Project:
Group 18

Sponsor: **Siemens**

Prepared for: **Senior Design I**

Prepared by: **Alana Icenroad (CpE), Antonio Buda (EE), Cassidy Lyons (CpE), Viviana Gonzalez (EE)**

Table of Contents

List of Tables

List of Figures

1. Executive Summary

The following text describes the Robotic Flange Assembly project as stated in the document titled:

“Flanges are often used to attach pipes carrying fluids and gases to various power plant equipment and systems. Flanges are typically round rims welded to the end of pipes with a sequence of holes for threaded fasteners to attach two pipes together. A gasket is usually placed in between two flanged pipe ends to maintain a seal. The assembly of flanges are manually intensive processes requiring careful control so as to maintain a proper seal.

The goal of this project is to develop a scaled prototype robotic flange assembly assistant to demonstrate and better understand the opportunities associated of the system. Benefits of the proposed system include reduced manual effort and improved quality. [1]”

From the project description our interdisciplinary team of two computer engineers, two electrical engineers, three industrial engineers, six mechanical engineers, and three computer science students – a total of 16 students, worked together to gather data in order to define our individual responsibilities. The final product is designed to automate the assembly of flanged pipes with minimal human involvement. The robotic system will be fastened to a platform that is then secured around the pipe by a belt that will provide guidance for stable movement, much like a track. The platform will serve as a carriage for the component that is responsible for driving each bolted screw into place. The process for proper flange assembly requires a strict set of guidelines to adhere to. The most interesting guidelines include requirements for the amount of torque delivered at specific pattern and time intervals. The pattern consists a distinct order that requires thoughtful and repeated movement. The delivery of torque is not all at once for each bolt and the amount delivered depends on time and a percentage of the overall specified torque. To achieve such a system, it was imperative that we recognized how to break down each disciplines role. The mechanical engineers focused on the physical design and the motor(s) that drive the system. The computer and electrical engineers determined the power supply, control units, necessary sensors, and programmable logic that make up the brains of the system. The computer science students will be providing a simulated environment and algorithmic guidance, and finally, the industrial engineers will deliver statistical data on safety and ergonomics that highlight benefits as well as pitfalls in the current and future process that our design depends on.

In this paper we will focus on the electrical and computer engineering roles. We must deviate from the most typical aspects of our expertise in order to gain a deeper knowledge of the mechanics that make up the system that is to be controlled.

2. Project Description

This section discusses a high-level description of the project goals and a preliminary understanding of what is required and how we plan to meet the requirements. We will talk about our motivations, our budget, our milestones, and some prototypes that will define a baseline for our target system. These prototypes may evolve or be combined in any fashion. The Robotic arm flange assembly project consists of five different Engineering disciplines coming together to make a functional robotic arm flange assembler for our sponsor, Siemens. The robotic arm shall be designed to simultaneously fasten bolts and nuts at a given torque. In order to do so, our team of 16 will divide into four tightly coupled teams as we put our knowledge to the test to deliver the desired outcome for the proposed project.

2.1 Project Background

The project, sponsored by Siemens, is interdisciplinary and requires the combined efforts of Mechanical, Electrical, and Computer engineering expertise. The predefined project description shall serve as a baseline as we focus on developing our own unique description of the project. In this early stage as we (the ECE components) integrate ourselves with the Mechanical Engineers (a group of 6), the Computer Science (group of 3) and the Industrial Engineers (group of 3), our primary focus is on discovering the root problem we are tasked to solve. We aim to clearly define our roles and to accurately portray the benefits of the proposed system.

The initial request we interpreted by Siemens based on the project description was to use two hobby shop robotic arms that are prebuilt that will produce a quick turnaround for the prototype. In the beginning the team faced challenges when determining how to fulfill this request when assigning tasks. What role does this leave for the mechanical engineers? Is there a dependency placed on the timeline for ordering a hobby shop robotic arm if we would like to accurately model the mechanical engineer design? If so, does this nullify the quick turnaround? In a meeting with the sponsor, we addressed our concerns and the session resulted in the removal of the previously mentioned suggestion. As the designed progressed the team decided to work in parallel with the mechanical engineers who began targeting a robust final product while the electrical and computer engineers explore a hobby shop design approach. The goal for the ECE team is to optimize control and configure hardware in a customizable way. In addition to discovering limitations and accuracy in terms of expected outputs given a set of inputs, the ECE team will be able to satisfy a vastly different approach that is still able to become integrated in the baseline design of the Mechanical engineer’s design efforts. The ECE approach focus is on a traditional arm design. The mechanical engineers will be designing the base of the system that will carry the main systems component. The base product will be able to accept the main system component and is not described in detail for the majority of the ECE design.

2.1.1 Group Members

Named Member	Role	Degree Major
Antonio Buda	ECE Student Member	Electrical Engineering
Alana Icenroad	ECE Student Member	Computer Engineering
Cassidy Lyons	ECE Student Member	Computer Engineering
Viviana Gonzalez Pascual	ECE Student Member	Electrical Engineering
Rodrigo Duran	ME Student Member	Mechanical Engineering
Fernando Gil	ME Student Member	Mechanical Engineering
Justin Connolly	ME Student Member	Mechanical Engineering
Reed Snowden	ME Student Member	Mechanical Engineering
Juan Meneses	ME Student Member	Mechanical Engineering
Juan Barajas	ME Student Member	Mechanical Engineering
Sopheap Sok	CS Student Member	Computer Science
Deepak Kumar	CS Student Member	Computer Science
Tyler Teixeira	CS Student Member	Computer Science
Kyle Veltre	IE Student Member	Industrial Engineering
Luis Malpica	IE Student Member	Industrial Engineering
Matt Stegall	IE Student Member	Industrial Engineering

2.1.2 Sponsor and Faculty Information

Contributor	Role	Expertise
Gerry Feller	Siemens Sponsor Liaison	Siemens Representative
Eduardo Lopez del Castillo	Project Advisor	NASA Engineer Adjunct Faculty Member
Mark W. Steiner	Initial POC, project coordinator	Professor and Director of Engineering Design Mechanical and Aerospace Engineering

2.1.3 Budget and Financing

Estimated Project budget and financing is $1200, generously provided by Siemens, our sponsor. As a group, we have an ideal overhead budget of approximately $400 for the team members as depicted in Table 1.

Table 1 - Estimated Budget

Description	Quantity	Estimate Cost	Actual Cost
Power Supply	1	$50.00	$50.00
Dual Motor Drive	1	$15.00	$15.00
Microcontroller	1	$25.00	$25.00
PCB	2	$50.00	$100.00
Motion Sensors	5	$5.00	$25.00
Circuit Components	10	$15.00	$150.00
Programmable Robotic Arm	2	$500.00	$1,000.00
Miscellaneous	10	$20.00	$200.00
Total Cost	-	$1,200.00	$1,600.00

2.2 Objectives

The crucial part of our role as ECE students for the robotic flange assembly project is loosely being considered independent of the design portion that the mechanical engineers are responsible for. This is because regardless of any sort of carriage or pulley type system that will be inevitably become part of our overall design, the design must have some sort of appendage and gripper to complete tightening of bolts. More importantly, the device must have an electrical brain programed to execute flange assembly in a specified pattern. While we may only have to slightly adjust any algorithmic data structures for controlling the device after the finalized design, the physical components changing could lead to a catastrophic financial waste. It is our objective to abstractly define the task of robotic flange assembly in the most efficient and correct way. At every opportunity we will choose cost effective alternatives, readily available resources, and existing mechanical design patterns while keeping quality as our number one goal.

2.2.1 Motivation

Motivation to pursue the Robotic Flange Assembly interdisciplinary project stemmed from a need to satisfy a deep curiosity and longtime fascination with robotics. The chance to manipulate a robotic limb is an incredible opportunity, especially for a group of college students. Before we can imagine what it takes to automate a flange assembly, we must carefully study the mechanics of the manual process.

As engineers, we ask ourselves how the process of flange assembly can improve, and we must identify as many shortcomings as possible to ensure we do not carry over any imperfections. During the initial phase of our data gathering, it appears that human error plays a larger role in the pitfalls of flange assembly than does the design aspect.

Flanges, gaskets, and fasteners are three components we have come to read a lot about while exploring nature of our ambitious task at hand. Fasteners can be at the root of failure if they are insufficiently tightened. A too loose fastener will not provide adequate support while a too tight fastener can impose stress on pipes. This is why fasteners must be tightened to a specified torque. Gaskets help to prevent leakage and serve as a seal. Careful installation and proper use should provide a stable design. When engineers select a material that is designed to fill a certain space, a gasket can provide a margin of error to the fasteners being tightened. Flanges must be handled with care to ensure they are not damaged before installation. It is important that proper installation is initially achieved to avoid damage that may occur with less than perfect attempts. While little can be done about flanges becoming warped over time, there does exist ample opportunity for improvement by simply taking precautionary actions.

The most exciting conclusion to draw is the notion that by minimizing the exposure to human error, many failures can be circumvented. The motivation that began as simple intrigue now presents itself as the chance to make a meaningful difference. We are inspired to challenge our intellectual limits and to work alongside other disciplines to achieve our goals.

2.2.2 Prototype A

The first design that came into mind when designing a flange assembly system that could simultaneously fasten two bolts together was one that can wrap around bolts and provide torque. In the figure below, two pieces of this prototype are shown. The very first one is the claw system. This system offers the ability to fasten two bolts simultaneously with the help of two grippers that would wrap around the bolt and torque it to a given value. Both grippers are driven by individual motors that allow them to perform precise tightening. Also, they have sensors in which sense the size of the bolt for proper fitting. The claw is kept together by a hydraulic cylinder at the end which allows the claw to perform proper movement and adapt adequately to the size of the flange without limitations. To have an exact torque on the bolts, a second piece had to be developed. This piece will remain fixed in the flange keeping the bolts from moving indefinitely and allowing the user to reach a given torque.

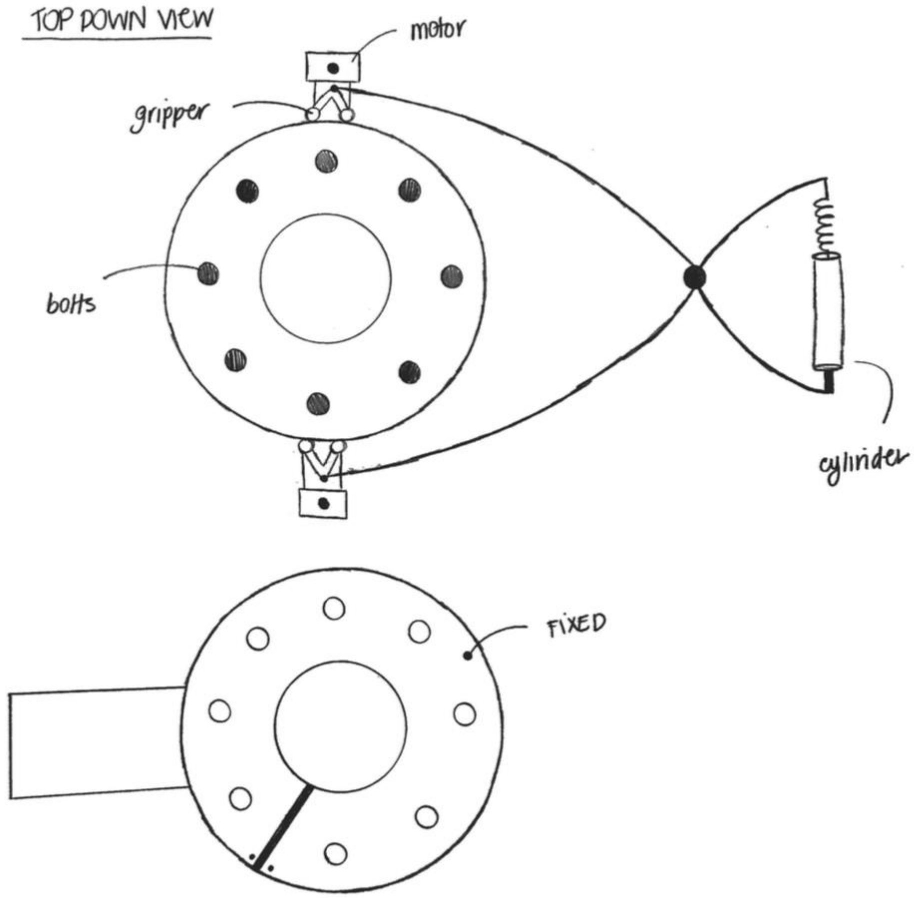


Figure 1 - Prototype A

Figure 1 represents the first prototype considered for this project. The first drawing is the "claw" which will tighten the two bolts simultaneously. The last drawing is a fixed piece that will be attached to the back of the flange limiting the nuts from movingly freely while being torqued.

2.2.3 Prototype B

This is the second prototype being considered for our project. Shown below is a drawing of what can possibly allow simultaneous torqueing of two bolts at the same time. For this design the team came together and thought of developing a carriage that can easily move freely with in the flange itself. The carriage will consist of grippers that will rotate along the edge of the flange allowing the carriage to more from bolt to bolt. In addition, the carriage will have two separate motors that will torque in the bolts simultaneously. For this design the team is applying the same fixed piece, as on the first prototype, to be placed in the back of the flange to ensure that the nuts do not make sudden movements that will jeopardize the accurate torqueing of the system.

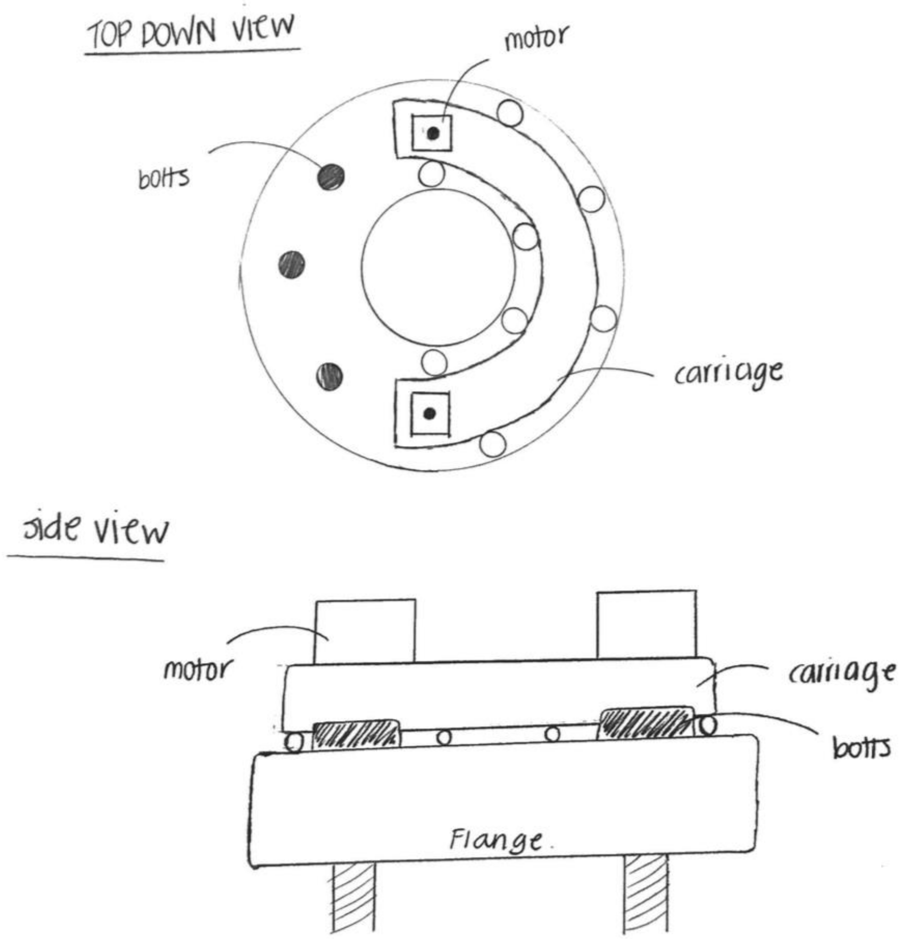


Figure 2 - Prototype B

Figure 2 shows the second design considered for this project. In the top view part of the drawing we can see that the carriage lies within the edges of the flange and can rotate 360 degrees around it allowing it to reach every single bolt. The side view gives us the idea on how the carriage will be able to move around the flange to torque all the necessary bolts.

2.2.4 Prototype C

This is the last prototype being considered for this project. This design is meant to be lightweight and easy to carry allowing the user to take it anywhere without any obstruction. The design is all contained within a box that can be easily taken to the field without any limitations. Inside the box the user will find two retractable arms that can simultaneously work together. One arm will have the ability to torque the bolts while the other makes sure that the nuts at the other side of the flange remain fixed for precise torque.

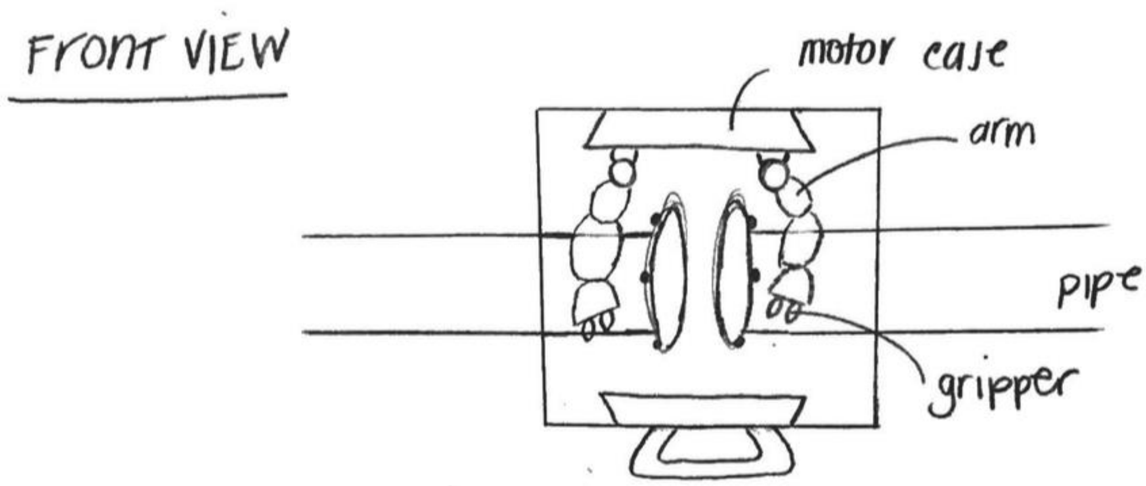


Figure 3 - Prototype C

2.3 Requirement Specifications

The “robot” refers to the final robotic prototype designed to automate the flange assembly with all its components and accessories. A table will provide readability for each set of design requirements. Table 2 represents the overall requirements that all teams must refer to when designing or implementing features. Table 3 and Table 4 are the primary focus for our team of electrical and computer engineers and it is the focus of this document. The computer and electrical engineers will be working together to ensure that all components are accounted for when designing the PCB and where control is concerned. This includes motors, motor drivers, sensors, electrical components and the correct configurations for all as well. It is important to design for efficiency. We do not want to have more components that we need, and we obviously do not want to lack any support. When programming the embedded system, we will aim to use fewer library and system calls in order to take advantage of the fastest interrupt response that the MCU can support. We will also focus on choosing components that allow for a fluid set of configurations.

The computer and electrical engineers will be working together to ensure that all components are accounted for when designing the PCB and where control is concerned. This includes motors, motor drivers, sensors, electrical components and the correct configurations for all as well. It is important to design for efficiency. We do not want to have more components that we need, and we obviously do not want to lack any support.

When programming the embedded system, we will aim to use fewer library and system calls in order to take advantage of the fastest interrupt response that the MCU can support. We will also focus on choosing components that allow for a fluid set of configurations.

Displayed below in Tables 5 and 6, the quality and safety topic requirements are discussed respectively. These requirements ensure that the robotic flange assembly follows standards, performs the desired functionality, and is overall a safe system.
Table 5 and Table 6 describe quality and safety requirements and shall also be referred to be all teams, however, the industrial engineering team will do a great deal of research on topics related to quality and safety. The requirements that are listed are expected to become refined and are subject to change upon customer request, especially if the physical design of the product undergoes optimizations that allow for greater efficiency or changes that alter the scope of the design. The changes shall always represent the worst-case scenario that we expect to be able to achieve and they shall not become more lenient. They shall only change in the case that a failure is realized that requires an alternative approach or a more realistic scope in order to meet customer demands.

Table 2 - Design Topic Requirements

Requirement Statement	
1.	The robot shall comply within the range of at least 2-inch to 10-inch diameter flanges
2.	The robot shall be reproducible financially and mechanically
3.	The robot shall be operated by the efforts of a single worker
4.	The robot shall withstand specified load limits
5.	The robot shall perform in a time that is equal to or less than the efficiency of human assembly
6.	The robot shall be light weight enough to be considered portable
7.	The robot shall follow standard provided by the ASTM and ASME standards **[10]**

Table 3 - Design Electrical Engineering Requirements

Requirement Statement	
1.	The PCB shall support an MCU with multiple servos and sensors
2.	The PCB shall be capable of distributing 6V thought the whole board
3.	The PCB shall perform with expected error margins as defined by schematic representation

Table 4 - Design Computer Engineering Requirements

Requirement Statement	
1.	The MCU shall provide enough I/O PINS to accommodate necessary components
2.	The MCU shall be capable of accepting 6V
3.	The MCU shall be capable of 200 microsecond average response times

The computer and electrical engineers will be working together to ensure that all components are accounted for when designing the PCB and where control is concerned. This includes motors, motor drivers, sensors, electrical components and the correct configurations for all as well. It is important to design for efficiency. We do not want to have more components that we need, and we obviously do not want to lack any support. When programming the embedded system, we will aim to use fewer library and system calls in order to take advantage of the fastest interrupt response that the MCU can support. We will also focus on choosing components that allow for a fluid set of configurations.

Displayed below in Tables 5 and 6, the quality and safety topic requirements are discussed respectively. These requirements ensure that the robotic flange assembly follows standards, performs the desired functionality, and is overall a safe system.

Table 5 - Quality Topic Requirements

Requirement Statement	
1.	The robot shall follow uniform sequence iterations as defined by relevant standards for the flange being operated on according to number of bolts and size
2.	The robot shall measure the required bolt torque via a sensor that is specified to detect load
3.	The robot shall be designed in such a way that portability and installation is equivalent to or greater than current methods of assembly
4.	The robot shall include programming for flange assembly that meet standards specified by the European Sealing Association guidelines (tightening and fitting)

Table 6 - Safety Topic Requirements

Requirement Statement

1.	The robot shall work in collaboration with human effort
2.	The robot shall provide superior efficiency and consistency than work achieved by the human hand
3.	The robot shall offer exceptional installation and operation safety
4.	The robot shall include a failure safe mechanism

2.3.1 Initial Block Diagram

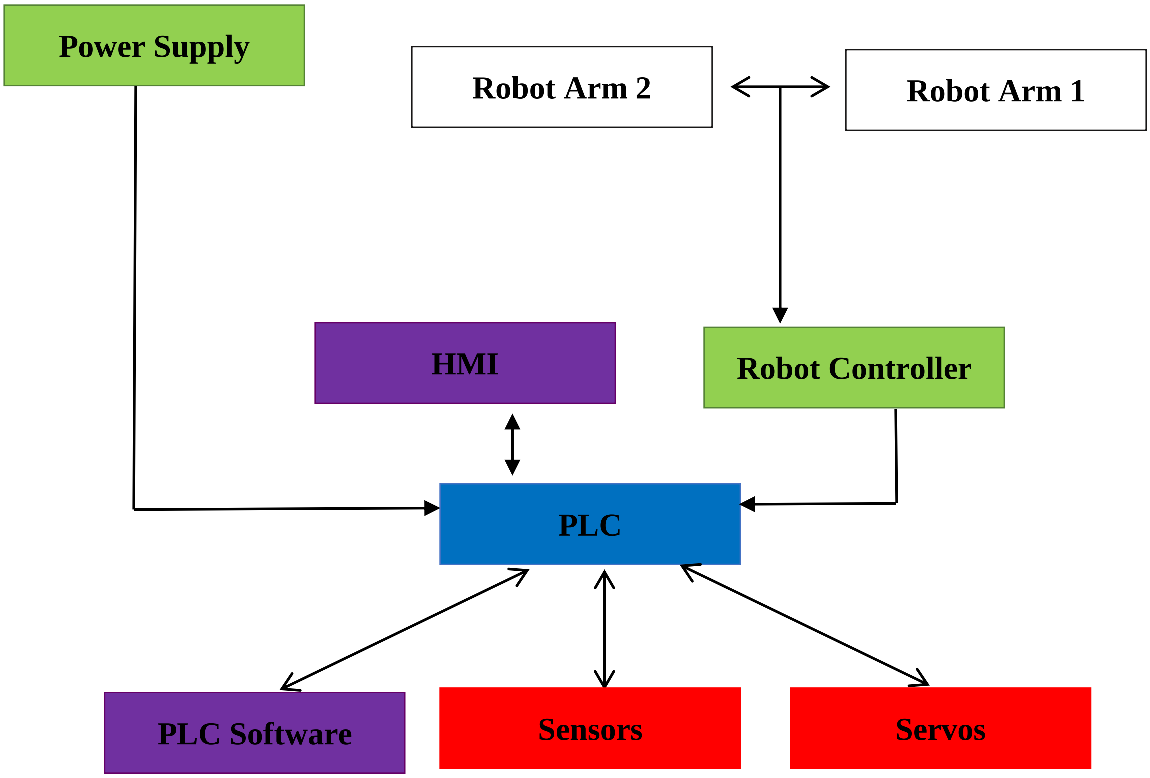


Figure 4 - Initial Block Diagram

Figure 4 represents our current vision for the main components to the Robotic Flange assembly project. The diagram is color coded to illustrate a team member that is responsible. **Table 7** describes the team members responsible for each component. Robotic Arm 1 & 2 will be purchased and for testing purposes only. They do not require the focus of any one individual**.**

Table 7 - Team Member Responsible for each Function

Alana	**Cassidy**	Tony	**Viviana**
Robot Controller Power Supply	PLC Software HMI – Human Machine Interface	Sensors Servos	PLC Hardware - Programmable Logic Controller
✸ NOTE:	Robotic Arm 1 and 2 represent the original requirement as stated by the sponsor's project description. This 'requirement' has since become an unknown at the request of the student design team to alleviate the constraints that such an implementation requirement would cause.		

2.3.2 Project Milestone

Initial milestones as defined by the requirements of Senior Design I is to have a complete design plan for the product system with well-defined roles for each student member, a detailed timeline that serves as a blueprint to be relied upon throughout Senior Design II. In Table 8 we have listed our current milestones and we have left the task list for senior design II dates as to be announced.

Table 8 – Milestones

Senior Design I Task List	**Due Date**
Project Ideas	08/24/18
Project Selection	09/14/18
Assign Member Roles	09/28/18
Initial Divide & Conquer	09/14/18
Divide & Conquer Revision	09/28/18
60-page Submission	11/02/18
100-page Submission	11/16/18
Final Document	12/03/18
Order & Test Parts	01/07/19
Senior Design II Task List	**Due Date**
Build Prototype	TBA
Testing & Redesign	TBA
Finalize Prototype	TBA
Peer Presentation	TBA
Final Report	TBA
Final Presentation	TBA

2.4 House of Quality

For this section, the software “EdrawMax2” [2] was utilized to format and populate it as shown.

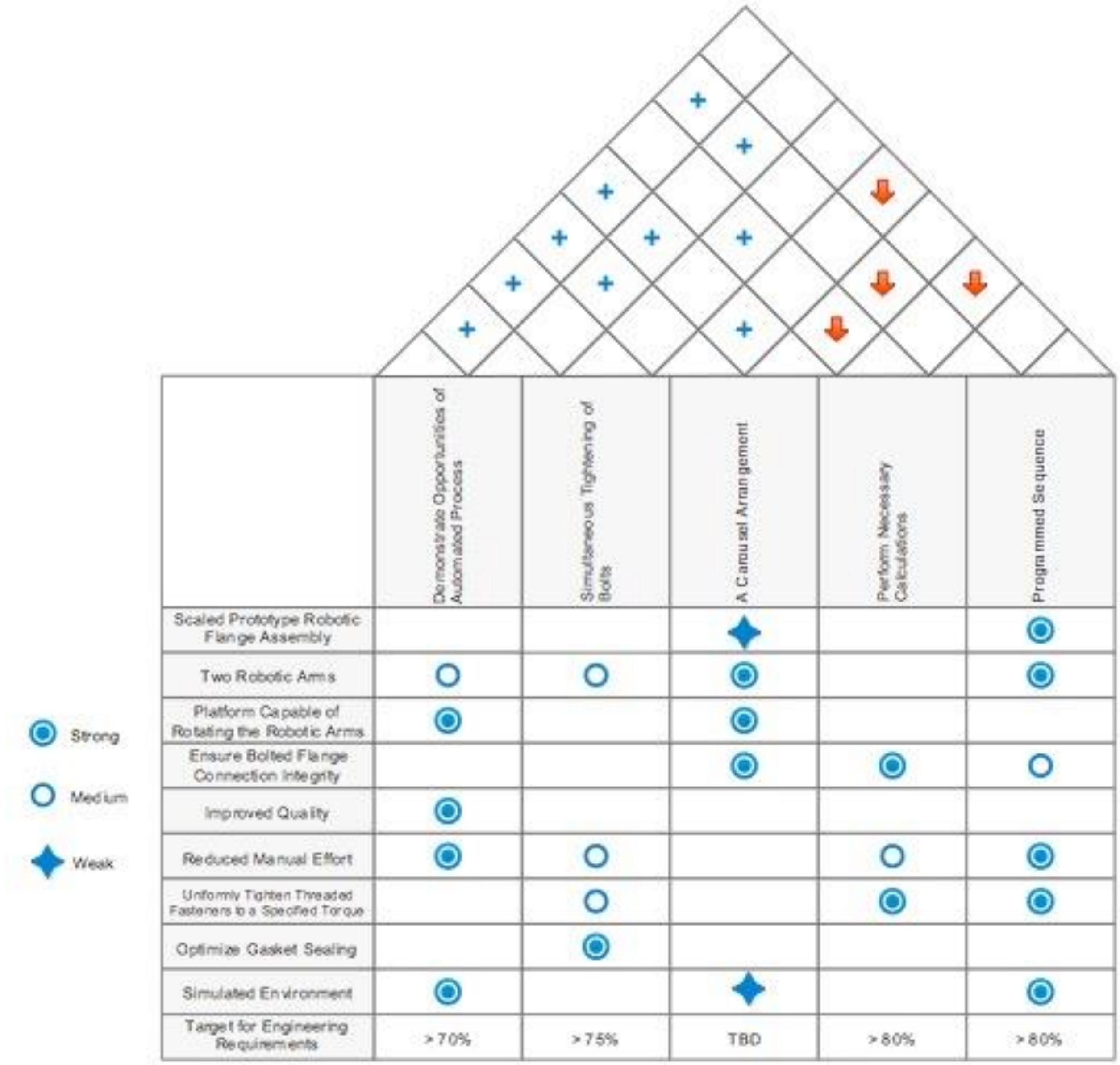


Figure 5 - House of Quality

Figure 5 above represents how strong or weak of a connection each item of importance has with our requirements. The triangular portion at the top represents areas of conflict, and the bottom row establishes an ideal engineering target that we hope to achieve.

3. Research Related to Project

In this section we will be covering many topics that we feel necessary to research in order to gain knowledge of our constraints and to understand our project. Many topics, like the type(s) of pipe and flange patterns that are utilized when assembling a pipe flange in the industry, are discussed in detail because we will be relying on this type of information to program the machine. It is highly important that we understand the process on a deep level so that we can break the method into the smallest instructions possible.

3.1 Pipe and Flange Patterns

For this project our focus is the ability to properly attach two flanges together often used in pipes carrying fluids and gases in sites such as a power plant equipment and systems, water reclamation facilities, oil rigs, and many others. Pipe and flange sizing are a crucial aspect when it comes to any system design. The pipes and flanges that we will study are specific to that of the steam system since Siemens is a company that mainly design and build gas and steam turbines.

Pipeline sizing is dependent regarding the distribution system being used to supply gas or fluid at the correct pressure. Steam piping in boiler systems is often classified as high pressure once exceeding 15 pounds per square inch gauge (PSIG). To sustain high pressures, the pipe and its components must be strong enough to perform its task without any room for failure. When designing piping for boiler systems with high pressure it is necessary to select materials and components that are approved standards by the American Society of Mechanical Engineers (ASME) and by the American Petroleum Institute (API).

There are multiple global piping standards considered. However, the global piping standards are mainly derived by API. Since 1924, the American Petroleum Institute is the largest trade association that establishes and maintains the standards of the oil and natural gas industry. The API categorized pipes by schedule numbers which bear a relation to the pressure rating regarding a specific pipe size. For those pipes that are constantly under pressurized fluids, the wall thickness and pipe strength are very important parameters that need to be addressed. The wall thickness is addressed as the “pipe schedules” or “schedule number”. There are at least eleven pipe schedules that start at a range as low as schedule 5 and as high as schedule 160.

For steam systems the most common nominal size piping is approximately 150mm or smaller and its ideal pipe schedule would be schedule 40 which in most cases is the “standard weight” for other applications. The pipe schedule is dimensionless considering that it interprets the proper pipe size and wall thickness combination to get a uniform relationship between the design pressure and the allowable stress that can be given to a pipe. For any given piping system, the relationship between the pressure and stress that the pipe can withstand is very important. That is why an expression was derived by an English Mathematician named Peter Barlow. The expression relates the internal pressure that a pipe can hold to its dimensions and material strength.

$$P = \frac{2St}{D}$$

P = pressure
S = allowable stress
t = wall thickness
D = outside diameter

The formula (expressed in PSI) relates the internal pressure that a pipe can hold to its dimensions and material strength.

All piping sizes are identified by the nominal pipe size (NPS). The standardized outside diameter (OD) of a pipe remains constant because if there is any variation in the pipe schedule then the only size affected would be the inside diameter (ID). Therefore, as the schedule number increases, the inside diameter (also known as the wall thickening) will increase and the bore diameter is reduced [3]. From ASME and API standards the pipe sizes and wall thicknesses are categorized as Standard, Extra-Strong, and Double Extra-Strong by pipe material used from API Specification 5L. **Table 9** shown below illustrates the nominal piping sizes between 3 different inches that are being considered for this project.

Table 9 - Nominal Pipe Sizes

Nom. Pipe Sizes		**OD**	**OD**	**Schedule Designations ANSI/ASME**	**Wall Thickn. inches**	**Wall Thickn. mm**	**Lbs./Ft**	**Kg/m**
Inches	mm DN	inches	mm					
2”	50	2.375	60.33	5/5S	0.065	1.65	1.604	2.39
2”	50	2.375	60.33	10/10S	0.109	2.77	2.638	3.93
2”	50	2.375	60.33	STD/40/40S	0.154	3.91	3.653	5.44
2”	50	2.375	60.33	XS/80/80S	0.218	5.54	5.022	7.47
2”	50	2.375	60.33	160	0.344	8.74	7.462	1.11
2”	50	2.375	60.33	XX	0.436	11/07	9.029	13.44
4”	100	4.500	114.30	5/5S	0.083	2.11	3.915	5.83
4”	100	4.500	114.30	10/10S	0.120	3.05	5.613	8.35
4”	100	4.500	114.30	STD/40/40S	0.237	6.02	10.790	16.06
4”	100	4.500	114.30	XS/80/80S	0.337	8.56	14.980	22.29
4”	100	4.500	114.30	120	0.438	11.13	19.000	28.28
4”	100	4.500	114.30	160	0.531	13.49	22.510	33.50
4”	100	4.500	114.30	XX	0.374	17.12	27.540	40.99
8”	200	8.625	219.08	10/10S	0.148	3.76	13.600	19.94
8”	200	8.625	219.08	20	0.250	6.35	22.360	33.28
8”	200	8.625	219.08	30	0.277	7.04	24.700	36.76
8”	200	8.625	219.08	STD/40/40S	0.322	8.18	28.550	42.49
8”	200	8.625	219.08	60	0.406	10.31	36.640	53.04
8”	200	8.625	219.08	XS/80/80S	0.500	12.70	43.390	64.58
8”	200	8.625	219.08	100	0.594	15.09	50.950	75.83

8”	200	8.625	219.08	120	0.719	18.26	60.710	90.35
8”	200	8.625	219.08	140	0.812	20.62	67.760	100.84
8”	200	8.625	219.08	XX	0.875	22.23	72.420	107.78
8”	200	8.625	219.08	160	0.906	23.01	74.690	111.16

For any given pipe size and schedule the thickness of the pipe will always remain fixed and found applicable on the ASME standards. The American Society of Mechanical Engineers have related standards specific to the oil and gas industries. Those are ASME B 36.10 Welded and Seamless Wrought Steel Pipe and ASME B 36.19 Stainless Steel Pipe. Common pipes for steam systems are manufactured from carbon steel. Furthermore, for steam at very high temperatures elements, such as chromium and molybdenum, are considered to help improve the strength and resist such high temperatures.

In addition to selecting the correct pipe size and materials that gets the system running, flanges are a primordial part of every pipe. Flanges are commonly used to extend the length of a pipe, connect external equipment (i.e. valves), and to make it easy for future removal due to maintenance procedures. There are many common flanges to consider. Shown in **Figure 6** are ASME B16.5 approved forged flanges that can come in shapes like blank (flat), slip on weld (raised face), weld neck, and socket weld [4]. Most of the flat flanges are made of cast iron and ductile iron. Those that are raised face are commonly found in cast steel and stainless-steel material.

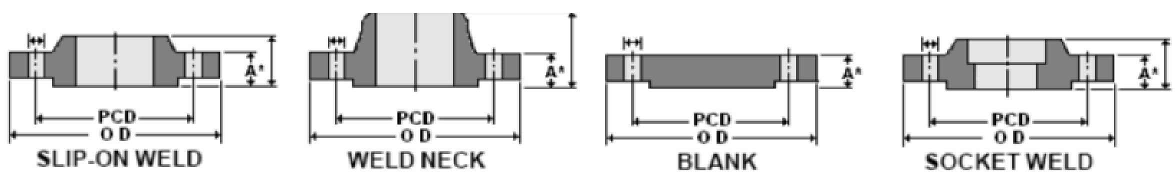


Figure 6 - Flange ASME B16.5 Forged Flanges

The flanges considered for this project are those that support maximum temperature and pressure since we can assume that they will be used for steam piping systems. When considering a flange, it is essential to make sure that proper dimensions and fittings are being chosen. Following dimensions and material standards for carbon steel flanges, commonly used in steam systems, can be narrowed down by the ASME/ANSI B16.5 Pipe Flanges and Flange Fittings and ASTM 105 M Standard Specification for Carbon Steel Forgings for Piping Applications are considered for proper flange set up.

The ASME B16.5 standard targets inputs within a pipe such as pressure and temperature ratings, dimensions, tolerances, marking (flange specifications), and testing. According to the ASME specification flanges have seven pressure class ratings. They are all considered when flanges are foreseeing operational pressure, temperature, and designated environments. Flanges with high class ratings withstand higher pressures compared to those with lower class ratings since the larger the rating the more robust the flange becomes shown in **Figure 7** [5].

Figure 7 - Flange Dimension and Weight Comparison

The difference between the class flange ratings is mainly the outer diameter, number of bolts and the bold circle. **Table 10** shows a few of the flange classes and their specifications in which are included in the ASME B16.5 Forged Flanges standards [6].

Table 10 - Forge Flange Classes and their Specifications

Nominal Pipe Size NPS *(inches)*	**Class 300**			
	Diameter of Flange *(inches)*	**No. of Bolts**	**Diameter of Bolts *(inches)***	**Bolt Circle *(inch es)***
2	6-1/2	8	5/8	5
4	10	8	¾	7-7/8
6	12-1/2	12	¾	10-5/8
8	15	12	7/8	13
	Class 400			
2	6-1/2	8	5/8	5
4	10	8	7/8	7-7/8
6	12-1/2	12	7/8	10-5/8
8	15	12	1	13
	Class 600			
2	6-1/2	8	5/8	5
4	10-3/4	8	7/8	8-1/2
6	14	12	1	11-1/2
8	16-1/2	12	1-1/8	13-3/4
	Class 900			
2	**8-1/2**	**8**	**7/8**	**6-1/2**

4	11-1/2	8	1-1/8	9-1/4
6	15	12	1-1/8	12-1/2
8	**18-1/2**	**12**	**1-3/8**	**15-1/2**

The flange class rating indicates the maximum pressure in PSIG that a flange can withstand when temperatures are very high. While temperatures rise, maximum pressure that a flange can tolerate depends on the type of material it is made of.

Table 11 - Flange Classes and Gauge Pressures

Gage Pressure *(psi)*							
Temperature *(ºF)*	**Flange Class**						
	150	**300**	**400**	**600**	**900**	**1500**	**2500**
	Hydrostatic Test Pressure (PSIG)						
< 100	285	740	985	1480	2220	3705	6170
200	260	680	905	1360	2035	3395	5655
300	230	655	870	1310	1965	3270	5450
400	200	635	845	1265	1900	3170	5280
500	170	605	805	1205	1810	3015	5025
600	140	570	755	1135	1705	2840	4730
650	125	550	730	1100	1650	2745	4575
700	110	530	710	1060	1590	2655	4425
750	95	505	675	1015	1520	2535	4230
800	80	410	550	825	1235	2055	3430
850	65	320	425	640	955	1595	2655

Table 11, shown above, demonstrates the comparison of pressure vs. temperature in a carbon steel flange can endure. When selecting a flange for a new or existing pipeline system it's important to find the material under the ASME B16.5 and the proper flange rating based on pressure temperature [6].

3.1.1 Torque Tightening

Flanges within a pipe system require proper tightening to avoid any possible leaks of fluids or gas. When installing pipes and flanges in site plants it is important to have a planned bolt tightening sequence or torque sequence procedure prior to installation. Torque tightening is the application of a preload to a fastener or bolt while its nut is being turned illustrated in **Figure 8** [7].

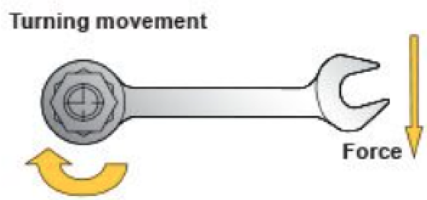


Figure 8 - Torque Tightening Movement

The torque tightening process exerts an axial pre-load tension in the bolt justified in **Figure 9** [7]. The tension load exerted in the flange is equal and opposite to the compressed forced applied on the other assembled components. After suitable tension load is applied to the flange its bolts will behave relative to that of a spring providing the flange with thoroughgoing elasticity properties.

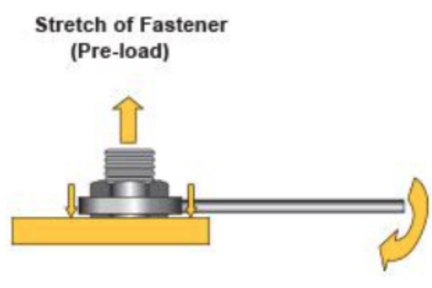


Figure 9 - Pre-load Acting Upon a Bolt

Typically, flanges are torqued with a manual torque wrench. The clamp load provided by the wrench produces a load higher that 75% of the bolts proof load. The flange alignment is highly stressed to ensure that a practical fit has been attained and no residual stress rests in the joints.

3.1.2 Torque Sequencing

The importance of the torque sequence procedure is to avoid point loading and load scattering since the total bolt strength is divided equally among the whole flange. Therefore, it is important to fasten bolts one at a time and at a specific sequence, shown in **Figure 10**, to realize a correct bolt tension [8].

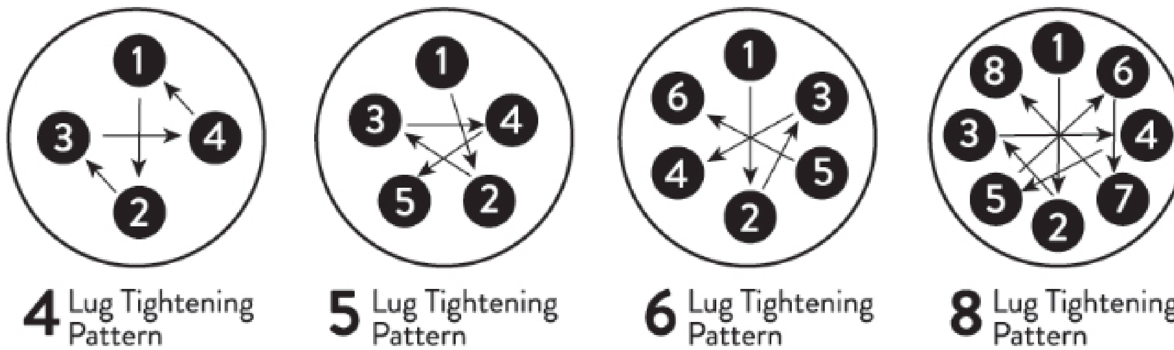


Figure 10 - Torque Sequence Patterns

The figure above shows how to torque bolts and nuts in a "crisscross" sequence. This type sequence practices a minimum of three torque passes with one final pass for conclusive torque. Given that there are many types of flange class ratings in the industry each one of them has a torqueing standard provided by the ASTM and ASME standards [8], shown in **Table 12.**

Table 12 - Torque Table for Flange Class Ratings

Flange Class 300						
Size in inch	**Size in mm**	**No. of Bolts**	**Bolt Diameter**	**Thread Type**	**Bolt Stress lb/in^2**	**Torque lbs-ft**
2	50	4	$^{5}/_{8}$"	UNC	37,000	64
4	100	8	$^{5}/_{8}$"	UNC	45,000	137
6	150	8	$^{3}/_{4}$"	UNC	45,000	137
8	200	8	$^{3}/_{4}$"	UNC	45,000	218
Flange Class 600						
2	50	4	$^{5}/_{8}$"	UNC	50,000	86
4	100	8	$^{5}/_{8}$"	UNC	45,000	218
6	150	8	$^{3}/_{4}$"	UNC	45,000	325
8	200	8	$^{3}/_{4}$"	UNC	50,000	526
Flange Class 900						

2	50	4	$^5/_8$"	UNC	40,000	194
4	100	8	$^5/_8$ "	UNC	45,000	474
6	150	8	$^3/_4$"	UNC	50,000	526
8	200	8	$^3/_4$"	UNC	45,000	894

3.1.3 Torque Sensing Technology

Torque is a measure of force that can cause an object to rotate about an axis. Force is what causes and object to accelerate in linear kinematics, while torque causes an object to acquire angular acceleration [9]. The direction of torque highly depends on the direction of the force action on an axis since torque is a vector quantity. Torque can be classified as either static or dynamic [10]. Static torque does not produce an angular acceleration while dynamic creates angular acceleration. Torque can be calculated by the expression shown below and in relation to **Figure 11**.

$$\tau = F * r \ \sin\theta$$

τ = magnitude of torque vector

r = length of the momentum

θ = angle between the for vector and momentum arm

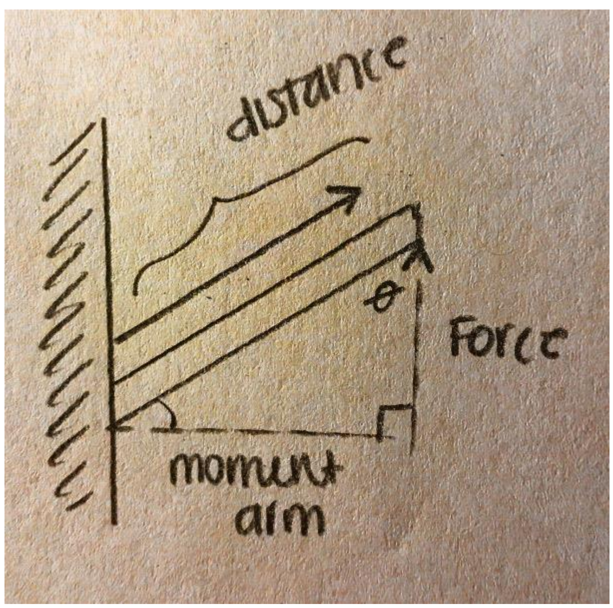


Figure 11 - Torque Analogy

In addition, the direction of torque can be found by using the famous right-hand convention rule. The convention works by curling your hand around the axis of rotation with the fingers pointing in the direction of the force leaving the thumb pointing the direction of the torque vector.

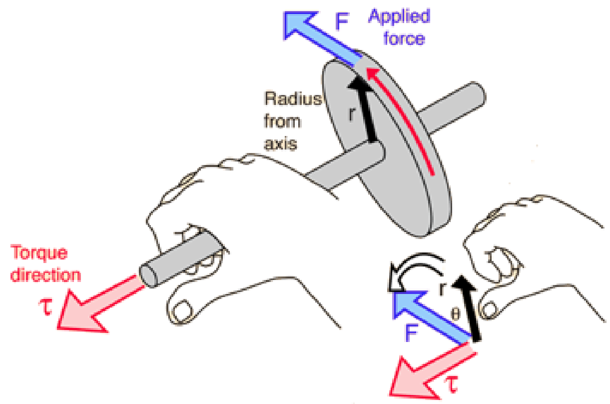


Figure 12 - Right Hand Rule

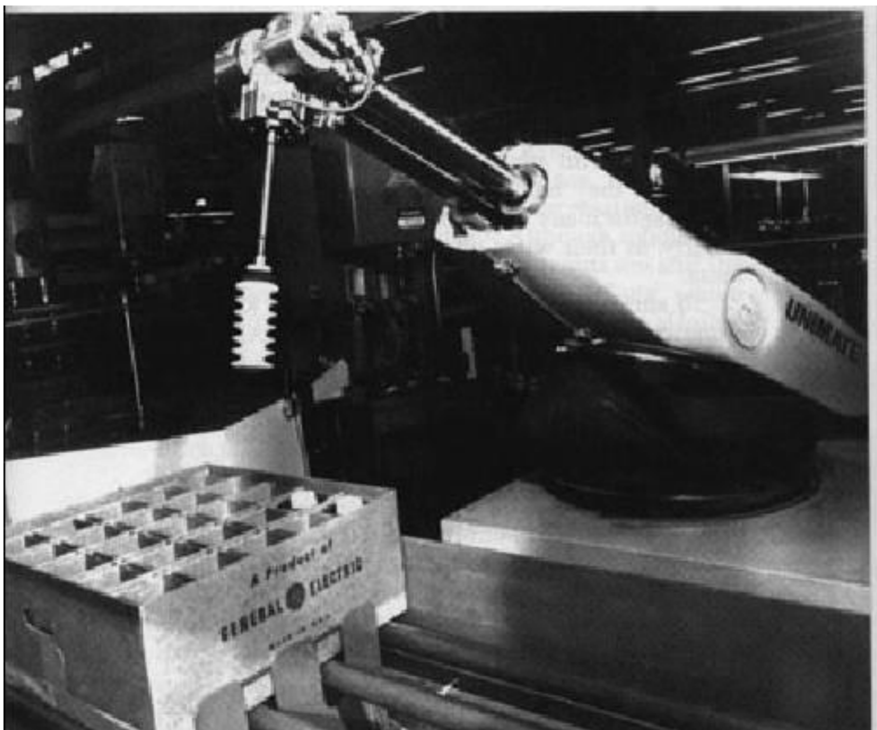
In this project we will be considering the torque sensing technology. The torque sensor is a sensor that converts torsional mechanical inputs into electrical output signals [11]. This technology is used to measure the either the static or dynamic torque on a rotating system. Therefore, there are two different types of torque sensors used to measure multiple type of forces. Dynamic torque sensors measure the rotary or angular force. On the other hand, force through a set distance is measured using static torque sensors. The dynamic torque sensors can either be rotary or non-contacting. Rotary torque sensors are generally used on rotating shaft applications. The rotary sensor mounts on the equipment's shaft and uses an integral slip ring assembly to transfer electrical signal from rotating electronics to stationary electronics. The slip ring uses rotating brushes on the rotating ring in that results in conveying an electrical pathway for incoming excitation and the outgoing signal voltage the system emits. The non-contacting sensors use either magnetic or inductive technology to provide precise measurements at elevated rotational speeds as well as offering long maintenance that is free operation. The static torque sensors offer a lengthier term of reliability since they have non-moving parts. These sensors are suited for industrial applications where angular motion is restricted and in line torque measurements are vital [12].

3.2 Robotic Technology

In this section of the report, research related to robotic technology is discussed. Further details of the discussion include robotic arm advancements, mounting robotic components on to the chassis, collaborative robots, and the possible dangers for humans when working with robots.

3.2.1 Robotic Arm History and Advancements

Over the last few decades, robots and their technologies have been progressing. The development of the robotic arm allowed for many other robotic advancements. Before car and motor companies used robotic arms in assembly lines, these machines had to be created. In the 1950's, George Devol created the first programmable "arm". After a few improvements, it got its final name Unimate. This first industrial robotic arm was created and developed by Devol and Joe Engleberger [13]. The first job of this robotic arm was to work and operate in an assembly line. Then in 1961, Unimate was able to be installed in a New Jersey General Motors.



Armed for duty. A Unimate robot—really, just an arm—picks up and puts down parts in a General Electric factory.

Figure 13 - Unimate Robot in a New Jersey General Motors

Approximately, about a decade after the first industrial robotic arm was built, a new robotic arm was invented. The Rancho Arm was designed and created by researchers at the Rancho Los Amigos hospital in Downey, California [13]. While this arm was not designed for an assembly line, it was created to help handicapped people that were staying at the hospital. It was one of the first robotic arms to move and act like a human arm and be conducted by a computer. Stanford University then later bought the Rancho Arm for creation and testing purposes.

The Rancho Arm established the ground basis for future robotic arms to come. Many robotic arms to this day are created with six joints like that of the Rancho Arm. Several of industrial robotic arms have the six joints, a "shoulder", an "elbow", and a "wrist." The shoulder part of the robotic arm is the part that is mounted to the stationary base structure [13].

Figure 14 - The Rancho Arm

These two mentioned robotic arm designs from decades ago helped shape and establish the norm for industrial robots and robotic arms. Unimate helped shape the foundation of modern robotics and the Rancho Arm helped shape the creation of robotic arms to be like a human arm. For the robotic flange assembly, the goals and functions desired for this project would be to use a robotic arm much like the Rancho Arm. Applications of bolt tightening and attaching flanges together requires the need for a six jointed, human-like robotic arm.

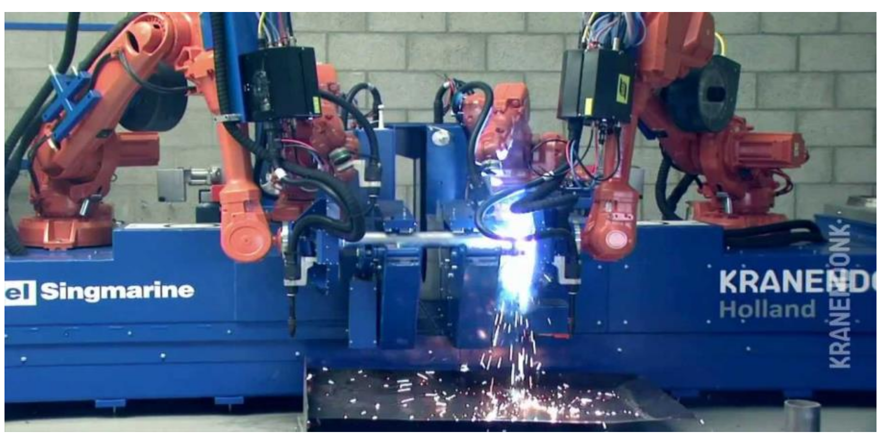


Figure 15 - Robotic Flange Assembly Today

3.2.2 Pre-Built with Customizable Options

"Robotic assembly lowers costs while boosting quality and capacity. Unlike dedicated automation equipment, robots are flexible, off-the-shelf machines that can be reconfigured or deployed as needed. Perhaps of greatest importance, robots are a mature technology, making them a low-risk, high-return investment" [14]. The pre-built assembly robot can be quickly and inexpensively reconfigured if the product design happens to change or even disappear completely. They can be assembled in multiple ways, support various objects on the end of the arm, and be designed for versatility. This means for our robotic flange assembly prototype, we have the option to buying already pre-built robots and customizing them to our project needs and goals. Some places to get a pre-built robotic arm would be from RobotShop, SuperDroid Robots, and SainSmart. This will be extremely beneficial in case we have to change the design or controls of the robotic flange assembly, as well as being able to use an arm for our own testing purposes.

Figure 16 - Pre-built Arduino Braccio Arm from RobotShop

3.2.3 Mounting Solutions

Due to robotic constraints such as size, weight, functionality, and condition, every part on a robot has a different method of mounting and attaching parts. Some of the different parts one would consider when mounting parts of a robot would be motors, sensors, servos, wheels, electronics, and much more. All parts being mounted and attached to the robot would be placed throughout various key locations on the robotic structure. For our chassis, it is necessary to have motors, sensors, servos, electronics, and a printed circuit board (PCB).

One of the main components for the robot is the motor. There are many different types of electric motors such as a direct current (DC) motor, alternating current (AC) motor, servo motor, or stepper motor. For mobile robots, DC motors are the most widely used motors because most robots are powered with direct current coming straight from the batteries. However, for robots that require a large amount of torque or where the motors are connected to wall outlets, an AC motor is used [15]. Motor mounting supplies include motor mounting adapters, motor mounting bases, motor mounting brackets, and motor mounting rings. Motor mounting adapters take care of the differences in mounting hole locations and shaft height without changing the motor axial centerline or the end of shaft extension [16]. Motor mounting bases are designed to help position the motors. Bases include a variety of end plates, shaped brackets, and mounting kits to help mount motors to the bases of the structure. Motor mounting rings help reduce the vibration that motors cause and replace the rings on many motor frames.

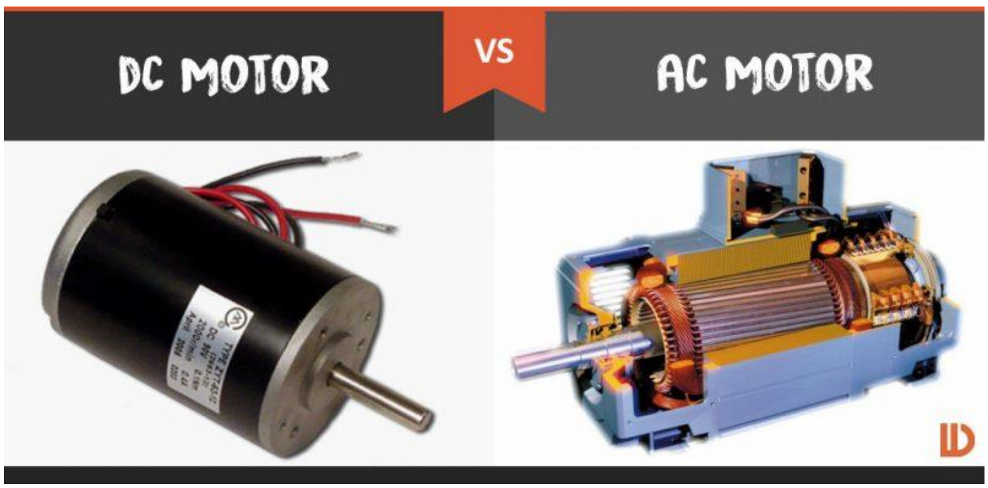


Figure 17 - DC Motor and AC Motor

Sensors are another major key component for a robot to have. Robots use a variety of different sensors to explore and make sense of their environment. A crucial reason why robots use sensors is because it gives the robot the ability to sense objects around itself, the environmental conditions, or its own position and condition. While sensors allow the robot to have human-like senses, sensors are also able to measure physical properties, such as the distance between objects, the frequency of sound, and the presence of light [17]. Some different types of sensors include a contact sensor, pressure sensor, light sensor, sound sensor, and many more. To mount a sensor to a robot, typically brackets, nuts, and bolts are required. However, mounting sensors is a case-by-case basis depending on the robot that is being constructed. It is somewhat difficult to mount sensors because there are very limited places to mount them. They must be protected because any type of damage to them will cause the robot to not function properly. They must also be away from noisy motors because the noise and vibrations can disrupt the sensors from performing their functionality and cause an interference for the robot. It is very dependent on what kind of robot is being built to know where exactly to mount the sensors.

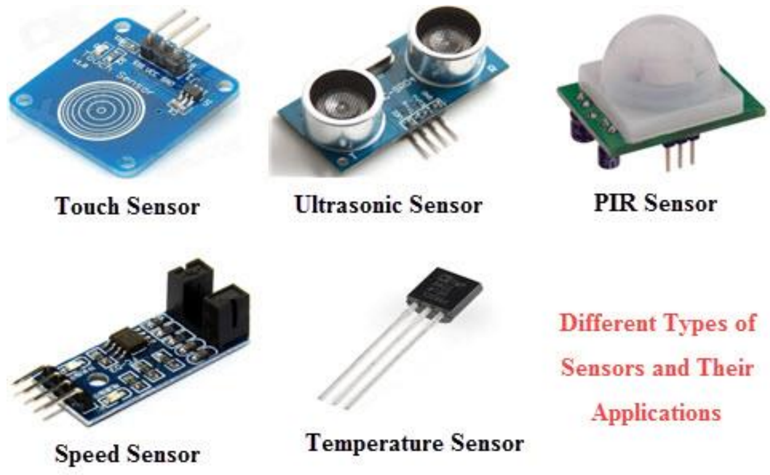


Figure 18 - Different Types of Sensors

Another important component of a robot is the servos. Servos are extremely useful in robotics. They have small motors and built in control circuitry. Even though servos are small, they can be extremely powerful for their size [18]. Depending on the robot will depend on how the mounting is done. It is necessary that servos have a solid position when being mounted. The best way to mount the servo is based on the cost of it and how frequently the servo will need to be replaced. For robotics, servo attachment can be done with either a universal servo mounting bracket or with a customized vertical or metal mount. While universal servo mounting brackets are not too costly, they do require space. When not using a specialized servo bracket, the position of the built-in mounting bracket on standard servo motors makes it very difficult for attachment with a bolt and nut combination. The mounting holes are so close to the servo body that holding a nut while tightening a bolt through the openings is extremely difficult [18]. In addition to brackets and servo mounts, another way to mount servos is to use cable ties. Using cable ties to mount servos is one of the quickest ways to attach a servo motor to the robotic structure. They are cheap and easily available at many stores. Some additional structuring and work may be needed to correctly attach the cables to get all the pieces in the right position.

Figure 19 - Hitec HS-425BB Standard Deluxe BB U Production Servo

There are three different methods when it comes to mounting wheels on robots. These three methods include direct mounting (DM), single bearing (SB), and dual bearing (DB). "Direct mounting is the cheapest and easiest mount but is only used for light duty payloads or with heavy duty motors. The motor shaft takes all of the weight of the robot and any load applied to the robot" [19]. Direct mounting robots support a light duty platform. Although the direct mounting method does not have the same clearance or capacity as the other methods, it performs exceptionally well. It is all direct, with the wheel axles mounted directly to the motor output shafts. For direct mounting, it is necessary to have specific platforms, specific motors and motor mounts, wheels and drive shafts, and hardware which includes nuts, bolts, washers, nylon spacers, cable ties, and cable hold downs.

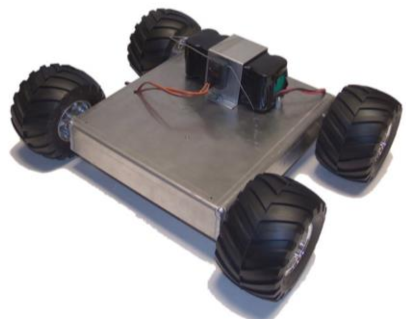


Figure 20 - Direct Mount Wheels

"Single bearing is a much more robust design, where a ball bearing takes most of the load off the motor shaft and puts it on the chassis/motor mount." [19]. Single bearing robots support a medium duty platform. They are supported at each axle by one sealed ball bearing. This allows for most of the load on the motor shaft to be taken off and for the robot to carry more weight. Robots that use single bearing mounted wheels can work on any indoor surface and most outdoor surfaces. For single bearing, it is necessary to have specific platforms, specific motors and motor mounts, wheels and drive shafts, chain coupling, and hardware which includes nuts, bolts, washers, nylon spacers, cable ties, and cable hold downs.

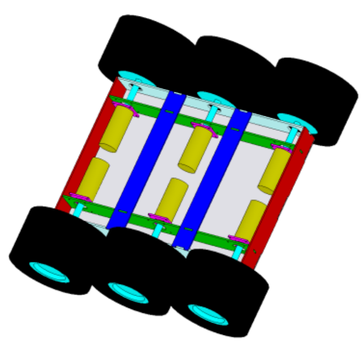


Figure 21 - Single Bearing Wheels

"The most robust system is the dual bearing system. In this, the wheel axle is mounted on two bearings and chain driven. This takes all the load off the motor and puts it on the chassis/motor mount [19]." Dual bearing robots support a heavy-duty platform. It is the sturdiest design of the three and can support heavy weights. For dual bearing, it is necessary to have specific frames, specific motors and motor mounts, wheels and drive shafts, chain coupling, and hardware which includes nuts, bolts, washers, nylon spacers, cable ties, and cable hold downs.

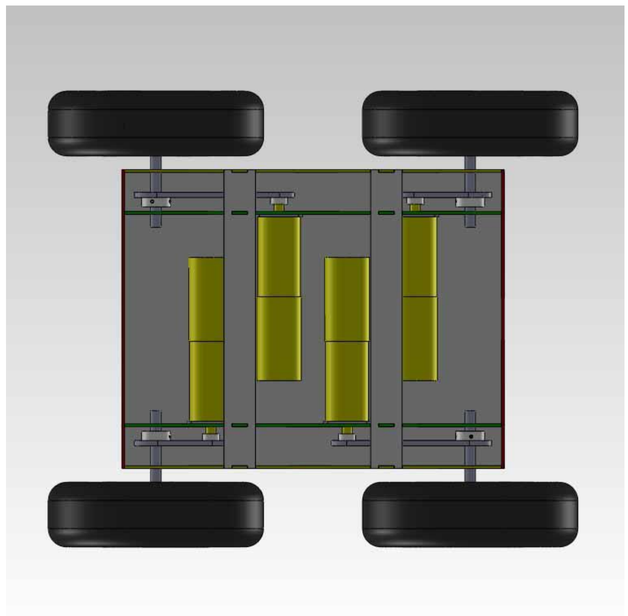


Figure 22 - Dual Bearing Wheels

Specific motors, motor mounts, and size go with each different wheel mounting method. **Table 13** displays the software defined radio (SDR) wheeled robot categories for each type of mounting method. This is the maximum weight that the robot can handle. The weight varies widely depending on the configuration and terrain [19].

Table 13 - SDR Wheeled Robot Categories

Wheel/Axle Mount Method	IG32 (32mm motors)	IG 42 (42mm motors)	IG 52 (52mm motors)
DM (Direct Mount)	10 lbs	-	-
SB (Single Bearing)	75 lbs	100 lbs	150 lbs
DB (Dual Bearing)	-	**200 lbs**	**250 lbs**

Mounting electronics to the robot can be somewhat tricky. Some things to keep in mind when mounting the electronics on the robot are to keep the electronics high up and away from the ground. Wires and cords need to be away from the ground because they can get damaged, tangled up, or throughout time, wear out. Holes should be constructed throughout the robotic structure to create routing holes for the wires. Once getting the wires situated, tape or other adhesives would be placed to keep the wires from dangling or touching each other. Customized PCB boards would also be placed high up and away from the ground. The PCB is what supports and connects the electronic components together with lines and pads and it allows signal and power to be routed between the devices [20].

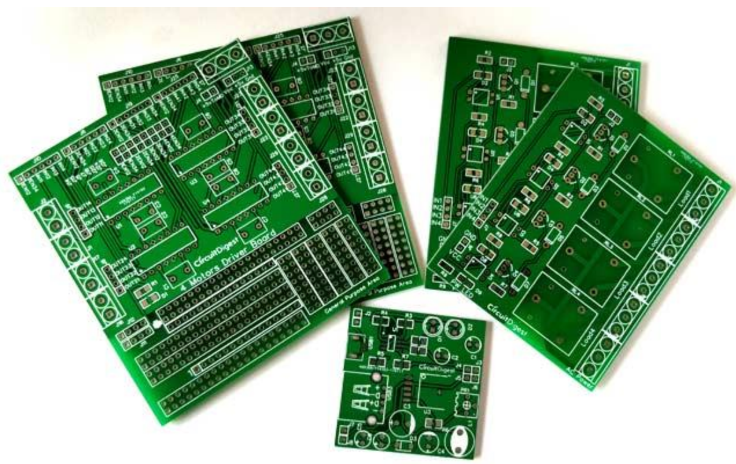


Figure 23 - Custom PCBs

3.2.4 Human-Robot Interactions

The study of interactions between humans and robots is called human-robot interaction (HRI). Some of the different human-robot interactions include human-computer interaction, industrial working machine robots, and artificial intelligence (AI). As technology expands and advances, humans are becoming more familiar with robots. Such human interactions include robotic toys, household appliances, self-driving cars, and industrial workplace robots.

Figure 24 - Flange Assembly

There are many major advantages to using robots. Most robots will usually have powerful, heavy, automated arms that can perform heavy duty tasks such as running an assembly within a workplace. Robots can also help reduce the risk of humans causing self-injuries as well as workplace hazards. Industrial robots would perform tasks that are undesirable or dangerous for human workers [21]. While robots can considerably help humans in many different aspects of life, the final goal of robots is to not replace the job of humans, but to more so have the capability of aiding a human as an assistant or as an extra pair of hands.

Industrial robots are becoming extremely prevalent today. These machines have taken on the jobs that exonerate humans from repetitive and dangerous jobs. Industrial robots have been working in the American workplace with humans for decades. Recently, however, advances in technology have begun to allow for a much greater diversity of robotic systems in the workplace [21]. In addition to the traditional industrial robots, robots can now be professional assistant robots, service robots, collaborative robots, and autonomous robots. These robots can be used in a wide variety of industries, companies, firms, and enterprises. [21].

Collaborative robots, also known as a co-bot, is a robot that is capable of learning multiple tasks so that it can aid human beings. A co-bot, when working side by side with a human, can quickly learn tasks through demonstration and reinforcement learning [22]. Co-bots are not intended to be better than a human worker's capabilities. Instead, they take the form of an arm, providing an extra set of hands. Employers are just now beginning to explore the many possibilities for human-robot collaboration.

Figure 25 - Collaborative Robot (Co-Bot) Working Alongside Humans

The goal for our robotic flange assembly is for the robot to act like a co-bot. It will be a robotic flange assembly assistant that will reduce manual effort and improve quality and preciseness. These extra set of hands will work alongside with humans as a co-bot, demonstrating speed and accuracy in the assembly without tiring or making mistakes. In the case of our robotic flange assembly, the robot will be programmed and "taught" by a human by guiding the robot through desired motions such as how to align the flanges or tighten the bolts.

3.2.5 Robotic Assistant Dangers

Human safety is a primary concern in HRI. As these next-generation robots open and create new possibilities for the future, their increasing interactivity and mobility can perplex the task of ensuring the safety of the humans they work alongside with [19]. Dangerous collisions are likely to occur when humans and robots work alongside together. As more and more robots get brought into industrial environments, newer risks of accidents for workers rise. Some of these dangers include robots malfunctioning, robot insecurity such as getting hacked, lacking sensory capabilities to detect humans nearby, or dangerous tool pass off to humans resulting in an injury.

Because collaborative robots are intended to share workspaces with humans, safety measures are starting to become built into their design. Despite having safety measures built in to these robots though, co-bots are still capable of posing significant threats and risks [21]. The Institute of Electrical and Electronic Engineers (IEEE) Xplore has developed a tool called a "safety map" which is a tool that helps robot developers analyze the safety performance of their robotic designs. "It helps user determine if the robot they're designing is capable of inflicting specific injuries during unexpected collisions [23]."

Figure 26 - IEEE Xplore Digital Library Database

Risk assessments are crucial to safe and successful implementation – and a core requirement of the current safety standards [21]. Risk assessments for co-bots are like risk assessments for traditional industrial robots and machinery. Risk assessments should be operated before, during, and after the set-up of the co-bot. These risk assessments will better ensure that the risks for injuries are lowered. Shown below, **Figure 27** is a diagram of the Risk Assessments and Reduction process [24].

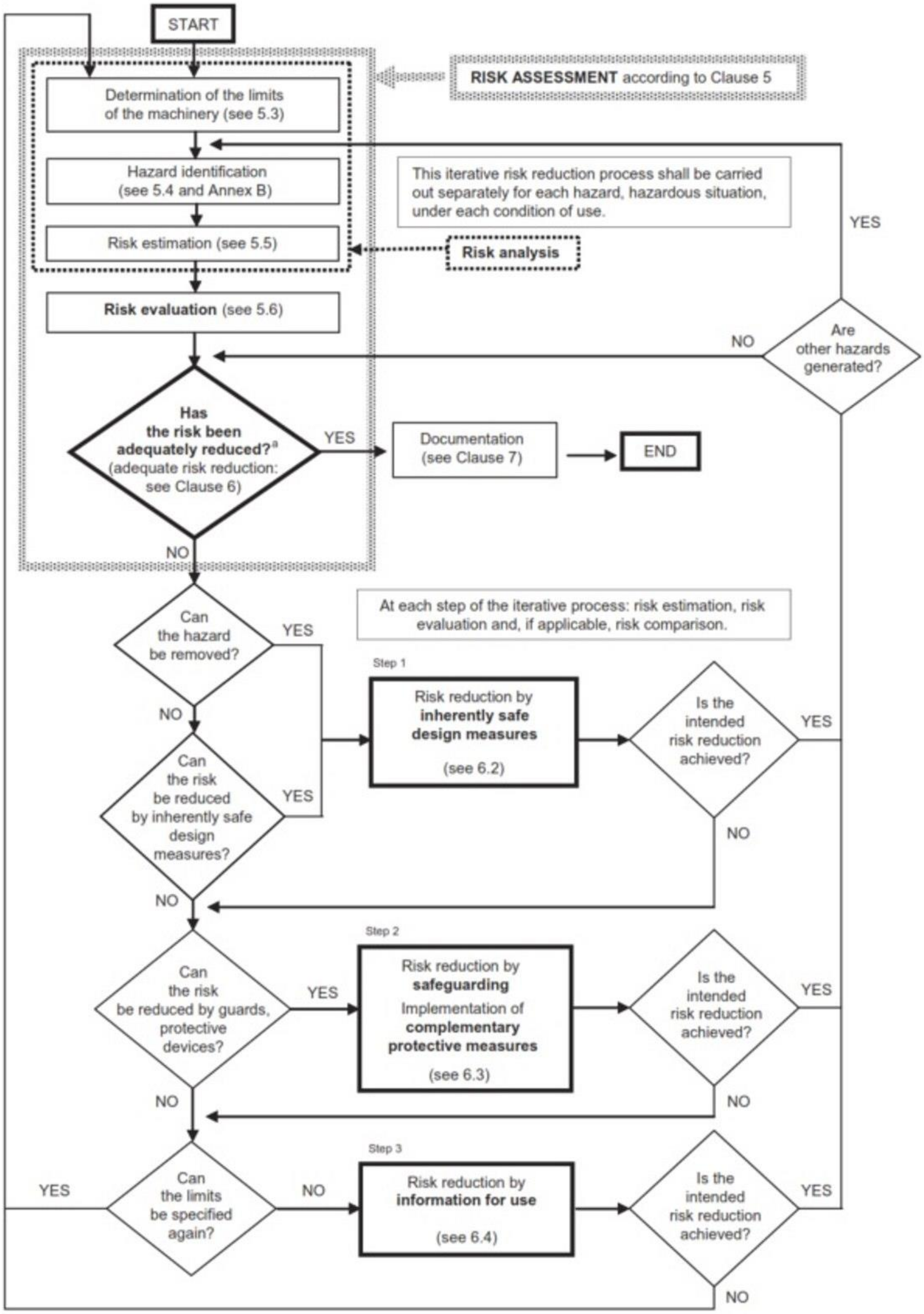


Figure 27 - Risk Assessment for Machinery Flowchart for ISO 12100

To summarize, The Rancho Arm, as displayed in **Figure 14,** will be used as the ground basis for the robotic arm for our robotic flange assembly. Since the goal of the robotic flange assembly arm is to hold two flanged pipes together securely while the bolts are being tightened, it is necessary to have a robotic arm that is like a human arm/hand. Using a robot with fingers (or grippers) will result in a greater flexibility and reliability to grip or grasp the bolts for tightening.

For the robotic arm to be able to move and function, necessary robotic parts are required to be mounted on and throughout the chassis. Some of these tools include motors, sensors, servos, wheels, electronics, and a printed circuit board. Motors are one of the primary tools for robots to move. Motors take electrical energy and convert it into mechanical energy, thus allowing the robot to move, function, and do work. Motors measure speed and torque; the higher the speed the lower the torque and vice versa. Sensors allow the robot to see and examine the type of environment and condition they are in. This will give the robot a pair of "eyes" like that of a human and are based on the functions of human sensory organs. Servos allow the robot to rotate the motor shaft to a specific angle. Wheels can have motors mounted to them and allow the robot to become portable. Lastly, a printed circuit board is extremely important for the robot because it will be specific to the robot as far as what specific components are needed. It is extremely important that our robotic flange assembly arm obtains all these key parts for the robot. More details about what specific parts to use will be discussed in a future section.

The robotic flange assembly arm should act as a collaborative robot and work alongside with humans. It will be a robotic flange assembly assistant that will reduce manual effort and improve quality and preciseness. The human will aid and teach the robot, positioning appropriate parts together and performing certain tasks to assist the robot. The robot will be responsible for aligning the flanges and tightening the bolts of the flanges.

The main goal for the robotic flange assembly is to be as safe as possible while working alongside with humans. When it comes to HRI, human safety is a primary concern. The IEEE Xplore safety map and risk assessments will be conducted to ensure that the robot is constantly being monitored and performing safe interactions with humans. It is extremely important to reduce any risks or dangers in the workplace, as well as prevent humans from doing dangerous and undesirable jobs.

3.3 Programming and Simulation

Programming and simulation will be critical to the implementation of the prototype design. Simulation of the robotic flange assembly will allow for virtual troubleshooting and will provide insight to possible physical constraints. To aid in development, ROS will be used as our operating system and VREP will be used in combination for the actual simulation. ROS is an open source robotic operating system that will serve as our robotic flange assembly controller and VREP is a virtual robotic experimental platform that will simulate the process.

3.3.1 Programming Language

C/C++ allow interaction with low level hardware, allow for real time performance and are very mature programming languages [25]. These are the closest things to a standard programming language for the robotic community [25] and for this reason, we will be using it for much of the software design.

3.3.2 ROS

ROS stands for robotic operating system and provides hardware abstraction, device drivers, visualizers, message passing, package management, and libraries that include common functions to help manage and develop a robotic system [26]. ROS will be the primary environment that our code will be developed in, and we will take advantage of as much code reuse as our system allows to accommodate portability. **Table 14** shows some of the packages provided by ROS that are capable of interfacing with components that are being considered for use in our design.

Table 14 - ROS Package and Compatible Component

ROS Package	Description	Compatible Component
hls_lfcd_lds_driver	Interfaces with LDS-01 to provide 360 Laser Distance Sensor 'HLS-LFCD-LDS' (a.k.a. LDS-01). It is a 2D laser scanner capable of sensing 360 degrees that collects a set of data around the robot to use for SLAM (Simultaneous Localization and Mapping) and Navigation. The LDS-01 is used for TurtleBot3 Burger, Waffle and Waffle Pi models. It	LDS-01 [27]: 

	supports USB interface (USB2LDS) and is easy to install on a PC. It supports UART interface for embedded board. This hls_lfcd_lds_driver package is a driver for LDS-01 [26].	
teraranger	Interfaces with the TeraRanger Duo, which provides calibrated distance readings in millimeters and has a range up to 60m [28]	TeraRanger Duo 

3.3.3 VREP

V-REP will be instrumental in the design process as we will be able to build and customize our design before buying hardware to support the physical prototype. By virtualizing the system, we can visualize customer needs and quickly explore design ideas. The only constraints caused by virtualization is the learning curve that comes with the simulation tools and the time it takes to become familiar with all the customizable options. V-REP will mainly be used to design the prototype and simulate the code that will be developed using C++ and ROS packages. **Figure 28** shows a V-REP template design that can be used to practice simulating movement controlled by an external ROS node.

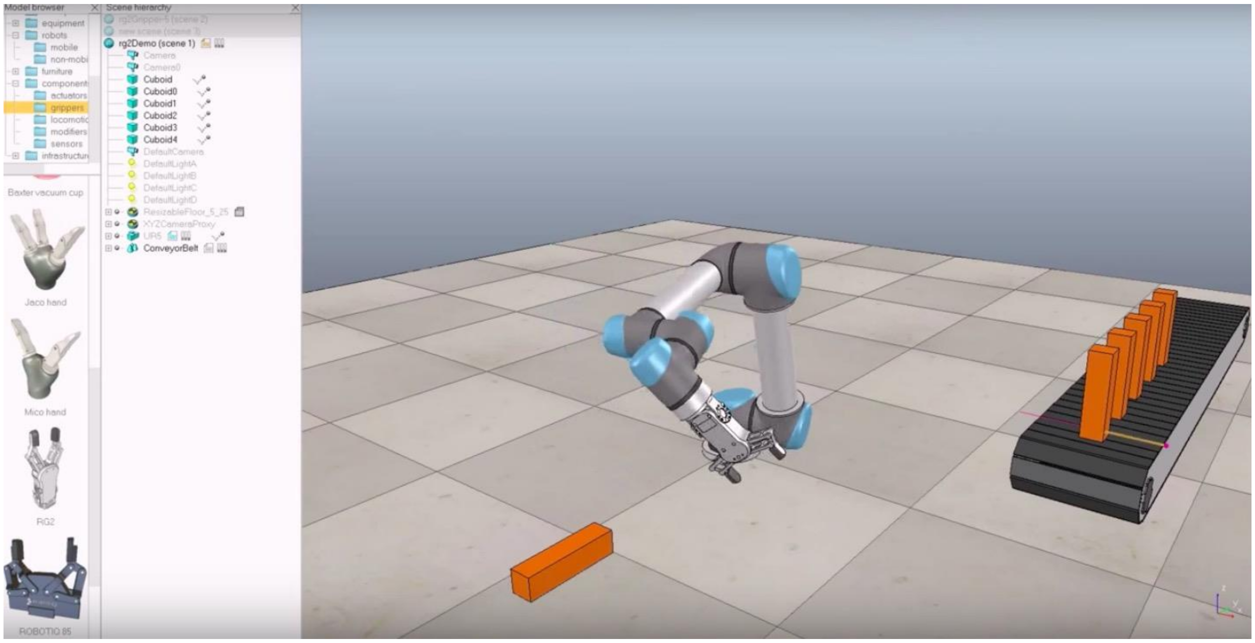


Figure 28 - V-REP Simulation Example

3.3.4 Testing Procedure Discussion

As an interdisciplinary team of 16 students, our biggest challenge has been aligning our milestone deliverables. One compromise we have made is to allow the final design decision to be made late in November. As illustrated in sections 2.2.2 through 2.2.4, our team has come up with several promising designs. Our prototype will largely depend on a mechanical design, but for each design we envision an arm appended in some way. Testing in a simulated environment allows us to test our many different designs without the financial burden, but the virtual design efforts proved to be tedious and involved a steep learning curve for simulating the flange assembly. Being able to test basic inputs and outputs against our expected values will be crucial to final component selection for our chosen design. As we are interested in focusing on the electrical components and programming of these components, we have decided to use an existing robotic arm design to gain baseline knowledge of how the physical design will affect the overall quality.

3.3.5 Testing Equipment

The existing robotic arm design that will be used for testing is a 3D printed arm. The STL files were obtained from an existing design. Since Siemens original suggestion was to use a prebuilt robotic arm, we thought it best to save design time by using an existing model [29]. This arm is not expected to produce the torque values required for steel flanges. Instead, we will obtain relative torque values. Below in Figure 29 [30] is a picture of the arm before assembly.

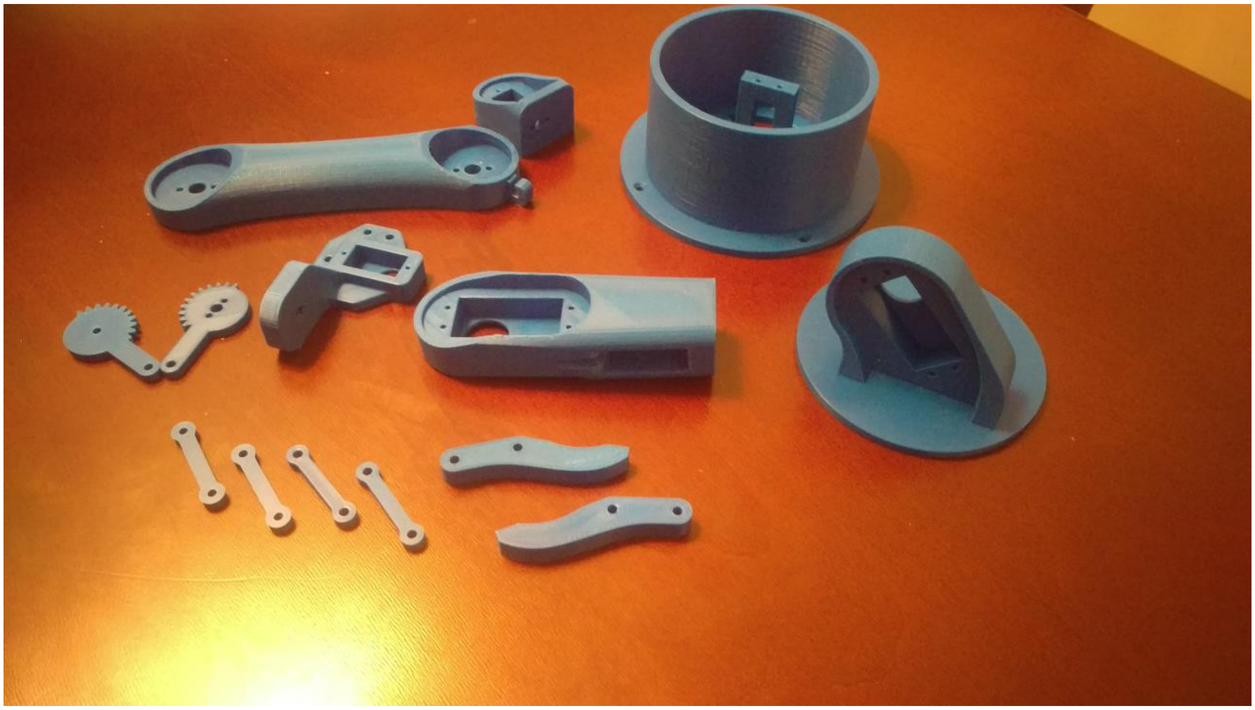


Figure 29 - 3D Printed Robotic Arm Parts

Our first goal, after assembling the 3D printed parts, is to learn how everything should be connected. Additional 3D printed parts we will use for testing are the nuts and bolts. We are currently in possession of the STL files required to print the nuts and bolts shown in Figure 30. Our hope is that if we use 3D printed nuts and bolts for testing, we can have scalable values. Assuming we have correct connections, we can begin our next testing phase. We will place the screw between the gripper of the arm and apply power to the motor that controls the opening and closing mechanism. We will then hold the fastener of the screw in place with some pliers while we use the gripper motor control to close onto the screw and apply a 180-degree angle of rotation. Next, we will release the screw, turn 180 degrees in the opposite direction, and repeat the process. In Table 17 - Stepper VS Servo we compared servo motors to stepper motor and found many similarities. The servo motor is believed to have more control over specifying a torque value. For this reason, we will begin by testing with a servo motor and pay close attention to our output to determine is it meets our expectations. Next, we will repeat the process of learning exactly how to control the key movements, but we will replace each servo motor with a stepper motor. We will do this until we have the most efficient configuration of servo and stepper motors. We will arrange the motors on our 3D printed robot that we decided was the best path for our build material, based on findings listed from. We will also be able to begin to write generic functions that we will be able to apply to the overall robotic flange assembly process.

Figure 30 - 3D Printed Nuts and Bolts

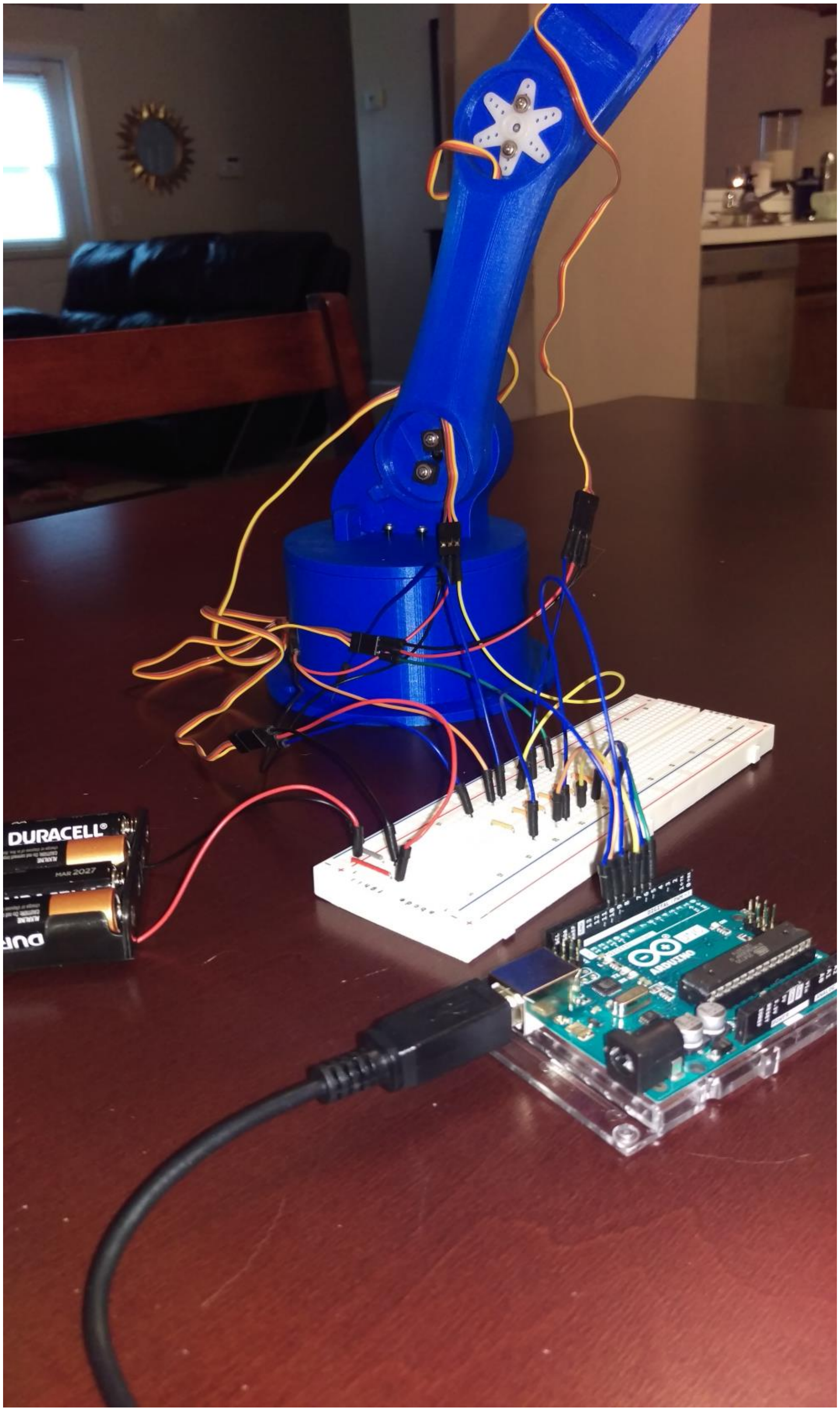


Figure 31 - Tester Arm

Figure 31 - Tester Arm shows how we plan to make our connections on the breadboard. We must use a separate power supply for our motors and for the Arduino Uno, which is what we are currently testing with. The motors are supplied with 6 volts from a battery pack and the Arduino Uno uses a USB 5 voltage power supply that plugs into an electrical outlet.

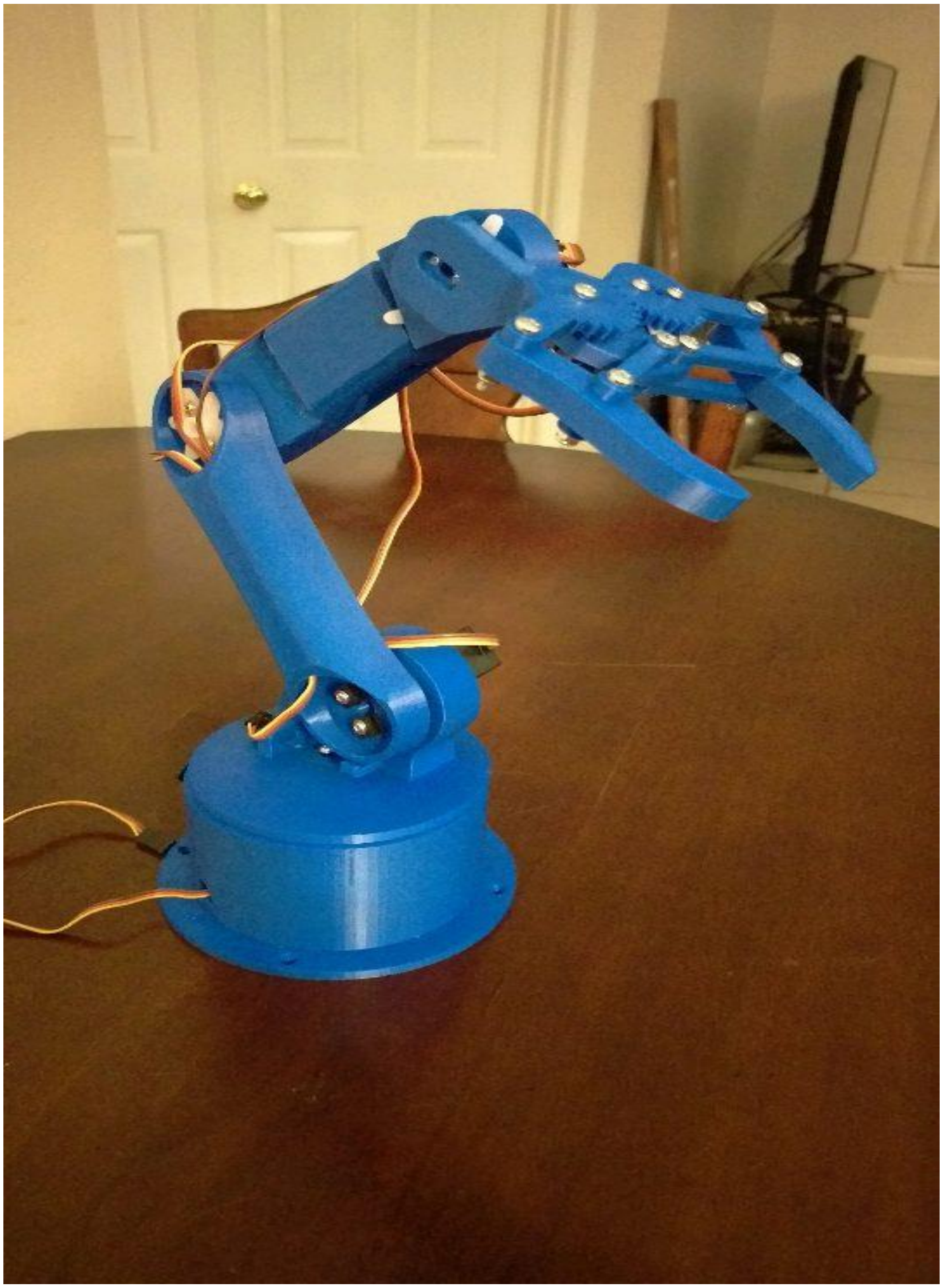


Figure 32 - Tester Arm Full View

Figure 32 is the 3D printed robot arm fully assembled. Up until the gripper was attached the arm appeared to be better than we expected out of plastic parts. However, once the gripper was attached it became unstable. Possible solutions include replacing the metal screws in the gripper with some other light weight screw (possibly 3D printed), adding support material, or rebuilding the arm using other parts like PVC piping or miscellaneous parts found at any home improvement store. The electrical components will still be used in our next iteration and we can still begin testing control with the current tester arm.

3.3.6 Lessons Learned

In addition to being able to test motor control with moving parts, it was anticipated that we would learn about any unexpected pitfalls. Assembling the robot arm was a learning experience because we learned how and where we should be connecting any motors to our physical design. We learned that the design should be heaviest towards the bottom and to use the most lightweight motor possible if any part of the design must support it. The 3D printed robot arm had 5 degrees of freedom. This was great for mobility, but it makes the control more difficult. It also requires more power to support more moving parts. The affect is that the torque that is produced is less than what the motor is capable of. The 3D printed robot arm has more opportunities for failure than we would like to have in our design. It is however a suitable test bench because we can learn a lot about motor control. Being able to test each motor while it is connected to the breadboard has produced predictable responses but having the motors control a physical device is quite a different challenge.

3.4 Power Delivery

This section discusses the power requirements of our future design. It considers the possible components of the system and how they fit into the total power scheme of the design. Many components are used as examples not because the components are specifically what will be used in the design, but because they represent the power requirements of the system well. Some factors taken into consideration are size, cost, performance, and compatibility with other component power supply in the design.

3.4.1 Power Requirements According to Flange Size and Associated Torque

The power supply necessary to tighten each bolt is the flange assembly is related to the amount of torque required to meet the bolt tightening specifications of the flange as well as the angular velocity of the device tightening the bolt. We can see this relationship in the equation

$$P = \tau \cdot \omega$$

With Power (P) in Watts equaling the product of torque (τ) in Nm and angular velocity (ω) in rad/s [31]. Each bolt will have a specified torque based on the flange class and size. More information of these torque requirements can be found in section 3.1.2 of this report. In our prototype we plan on tightening a flange assembly with a diameter between 2 and 8 inches, so our power supply must be enough to attain the highest torque required by this size range. As well, the different classes of flange require different torques. If we assumed the 8-inch diameter of flange and a class 300 flange, according to Table 12 - Torque Table for Flange Class Ratings in section 3.1.2, it would require 218 ft-lbs of torque to meet the tightening specification. Converting this to Nm yields a torque of 295.5683 Nm. If we choose an angular velocity of 2000 RPM like that found in a Dewalt cordless drill (Table 16 - Dewalt 20 V Lithium Ion Battery Specifications), our angular velocity in rad/s would be 209.4395 rad/s [32]. Then, if we plug our specified torque and angular velocity into our power equation, it yields a required power of 61.904 KW. These values may change based on varying flange sizes, power supply methods becoming the determining factor in the torqueing device and changing torque requirements based on prototype decisions. More on torque requirements can be found in section 3.1.1 of this report.

Table 15 - Example Specifications for a Powerdrill

Model	DEWALT® XR® Brushless Compact Hammerdrill
Battery type	Lithium Ion
Battery Size	20V
Max RPM	0-2000RPM
Max Power	460 UWO

3.4.2 Power Delivery Methods to Achieve Each Torque

The delivery of power to the tightening motors to achieve each torque will be determined by how the flange bolts are tightened. Specifically, whether the device will tighten all bolts at once or tighten bolts individually. This decision will determine the power delivery methods. If the bolts are all tightened at once, it would require a larger power supply to be spread out over motors for each built. The power delivery would be shorter and cut down on execution time for the robot but would also increase the total power load for the design and increase costs, weight, and mounting difficulty. Multiple power supplies, perhaps one for each motor, is also a consideration. The latter implementation of tightening all bolts at once could decrease the difficulty level of a mounting solution but, may offer little else in terms of cost and weight benefits.

As explained in **Figure 10 - Torque Sequence Patterns,** a torque sequence procedure is used to avoid point loading and load scattering since the total bolt strength is divided equally among the whole flange. The robot will thus move around the flange in a predetermined tightening sequence as shown in Figure 5 to provide a more equal seal across the flange. During this procedure, a certain power supply will be delivered to the torqueing device to tighten the bolt to a percentage of the max torque for this flange. Because it is an incremental method, power may be supplied in smaller load and will thus require a smaller power supply for the tightening device. This allows the design team to use a smaller, cheaper, device for power supply

3.4.3 Feasibility of Utilizing Tools Based on Cost, Size, or Other Factors

The power supply to the torqueing device as well as the controlling device will have to agree with the budget of the project as well as the size and bulk constraints. One of the main purposes of this project is to decrease human labor intensity by removing the manual struggle of bolting a flange by hand or pneumatic device. If the power supply is so large or heavy that mounting it becomes an issue, then the point of the device is moot. As well, the power supply can't be so bulky that it negatively affects mobility in a tightening sequence.

Most likely our power supply would be no greater than 20V, as this is enough to power a top-quality cordless drill with a rechargeable lithium ion battery [33]. Furthermore, a battery power supply of this size is less than a pound and can be mounted on our assembly in a way that won't cause strain on the user when mounting the assembly and won't get in the way of the carousel's movement around the flange (Figure 27). Most power supplies and batteries of this size can be purchased for under 50 dollars as well [34]. This would fall within our budgeted allotment and would allow for a spare to be purchased if necessary, though it wouldn't likely be needed.

Figure 33 - Dewalt 20V Lithium Ion battery

Another factor in our choice of power supply will be whether it's more cost effective to buy a battery fit specifically to our own purposes or buy a battery designed for something else and modify it to fit our design. For example, if we took the battery pack of a cordless power tool, we could remove some of the outer casing, rewire it, and mount it in a way that is more suitable to our needs. All these factors will come into consideration when we move on to our purchasing stage and narrow down on our desired components.

Table 16 - Dewalt 20 V Lithium Ion Battery Specifications

Characteristic	Value
Capacity	3.0 Ah
Length	5 in.
Height	2.1 in.
Watt Hours	60
Weight	1.1 lbs

3.4.4 Components that Affect Power and Components Affected by Power

The design of our assembly is split between mechanical implements that manually grasp the pipe and flange, and electrical components that power the tightening devices, sensors, and controls. The mechanical implements are manually mounted and as such, don't require any form of electrical power. The electrical components may require differing forms of power for each component depending on the nature of the component and what we choose to purchase and design with.

As discussed previously, the predominant load for our power supply will go to the motor driving our tightening device. If our robot provides torque in incremental doses through a specified pattern, it will cut down significantly on the power requirements for the motor itself. The next component of the tightening appendage the requires power is the device that moves the bolt gripping device up and down in order to move away from the bolt or towards it to meet the bolt while tightening. This motor will be a much more delicate and previous tool than the one used to tighten the bolt itself. While it would need the precision to move down and provide minimal pressure to the bolt head being tightened, it wouldn't need the large amount of torque needed to twist the bolt quickly. As such, it requires a much smaller power supply to achieve its goal. Depending on the type of motor used, it would require a small battery that can most likely either be attached to the carriage the appendage is moving on or be attached to the flange mount in a way that's minimally invasive. A servo motor is probably of similar scale to what would be used and would need to be powered [35].

The final motor present on our device will most likely be that which moves our device around the flange to perform the specific pattern of bolt tightening we assign to the robot. This motor will have to move at a quick speed but nowhere near that of the tightening motor. Most likely it'll be of a slightly larger scale than that of the lateral motor on the appendage, but not much. This power supply will still be small enough not to restrict the device's movement or its speed to a certain point. A simplified diagram of how to supply power to a servo-type motor is given in Figure 30. It's important to note the voltage regulator and capacitor added to provide a stable power load to the servo and not cause damage [35]. These considerations will have to be made in many parts of our design as we aim to provide a stable and reliable device.

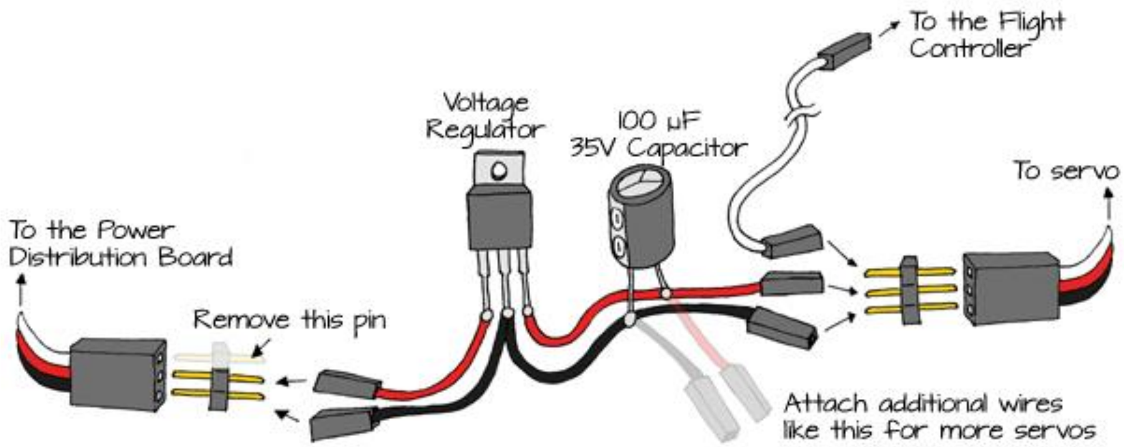


Figure 34 - Simplified Servo Power Supply

Another major aspect of the system to be controlled are the sensors. Depending on what type of sensor we use, the power system could differ significantly at this point from the other devices. Different sensors require different power supplies and as such we'd need to accommodate them in our design. It is well understood by our team that if we have at least one sensor it would most likely be a distance sensor. This sensor would determine how close or far we are from the bolt and when the do can move. A common distance sensor is the Passive Infra-Red sensor (Figure 31) which is a pyroelectric device used to sense motion at a distance through changes in surrounding infrared levels [36]. While this device specifically may not be used in our design, I feel that it gives and accurate idea of what we can expect for power requirements of a small distance sensor.

Figure 35 - Parallax PIR Sensor

According to the datasheet of the Parallax PIR sensor, the power requirements of the components are 3-6 V DC, and 3 mA active load, coming out to 18mW for the maximum power requirements [36]. This is a relatively simple device to power with a very small required load. Many distance sensing devices are of a similar scale to this and wouldn't require much power. In Figure 36 you can through this example circuit with multiple PIR sensors how the power is connected simply though a DC power supply and distributed throughout the circuit [37]. In the top left corner of the breadboard how a voltage regulator and capacitor are connected in series to the power supply for voltage regulation.

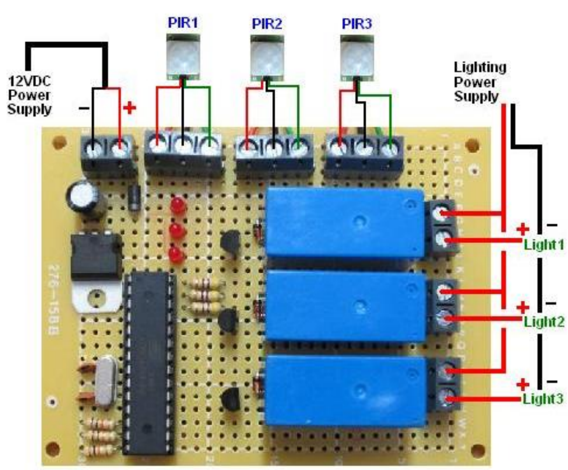


Figure 36 - Example Circuit Involving PIR Sensors

The last major electrical component to be powered will be the microcontroller. This device is the main controlling mechanism of the board and will determine how the device moves and operates. Until we finalize the design for our device, we can only guess at the type of microcontroller used because we don't know what we're controlling or how we're controlling it. For the time being, we'll choose at ATmega328P (Figure 33) microcontroller because this is a popular and basic chip commonly used for robotics and hobby boards, including the Arduino Uno. The ATmega328P has an operating voltage of 1.8V to 5.5V with a current usage of 0.2mA in active mode [38]. Again, not a large power consumption, but a variable one as can be seen in Figure 34. Because of this, the power will need to be well regulated as mentioned with the sensor and servos. In Figure 35 we can see how the microcontroller is implemented on an Arduino board with a DC input in the top left corner of the board. We can see the voltage regulator shown with two capacitors to regulate the DC input for the components on the board.

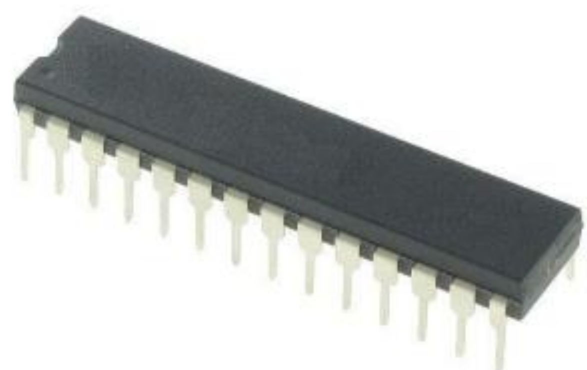


Figure 37 - ATmega328P

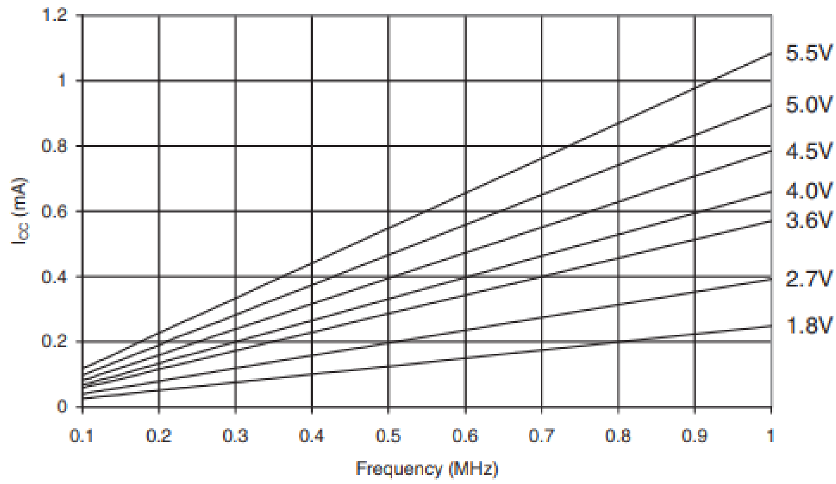


Figure 38 - Active Supply Current versus Frequency

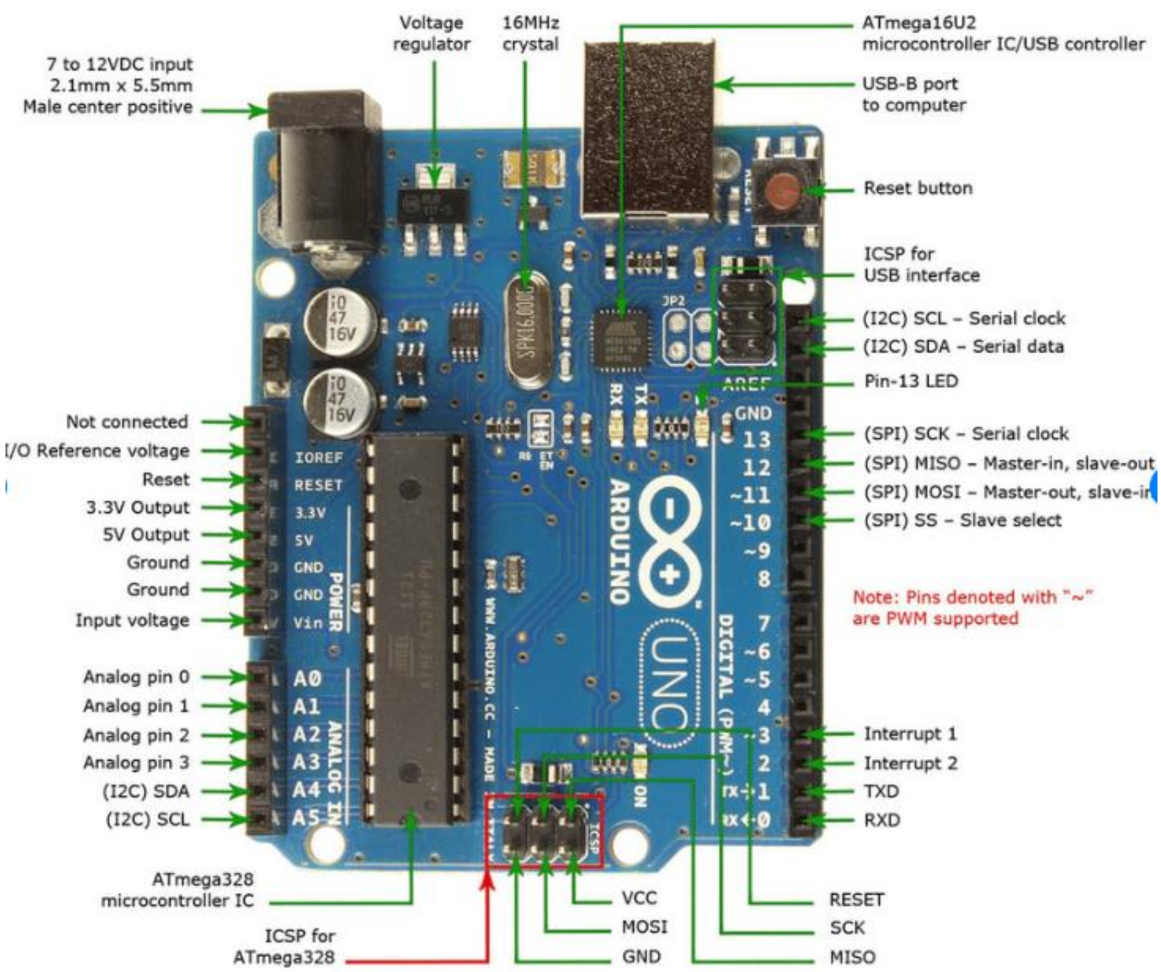


Figure 39 - Arduino Uno with ATmega328P

3.4.5 Compatibility Requirements

Due to differing requirements for the electrical components of the system, we have to design a method to distribute power reliably to all components from a voltage source or varying sources. One issue that may occur is DC to AC conversion if an AC motor is being used. This may be an issue best avoided at all costs because DC to AC converters (or power inverters) are often very bulky, as shown in Figure 36 [39]. A much easier process is AC to DC conversion. An AC/DC converter can either be purchased as a single component or it can be designed using a diode bridge as shown in Figure 37. The diode bridge design doesn't fully transform the AC into DC but rectifies it into a ripple voltage that performs almost like a DC voltage [40]. However, even with this simplified design, it may be better to try to find components that are all compatible in with DC sources.

Figure 40 - 12 V Power Inverter

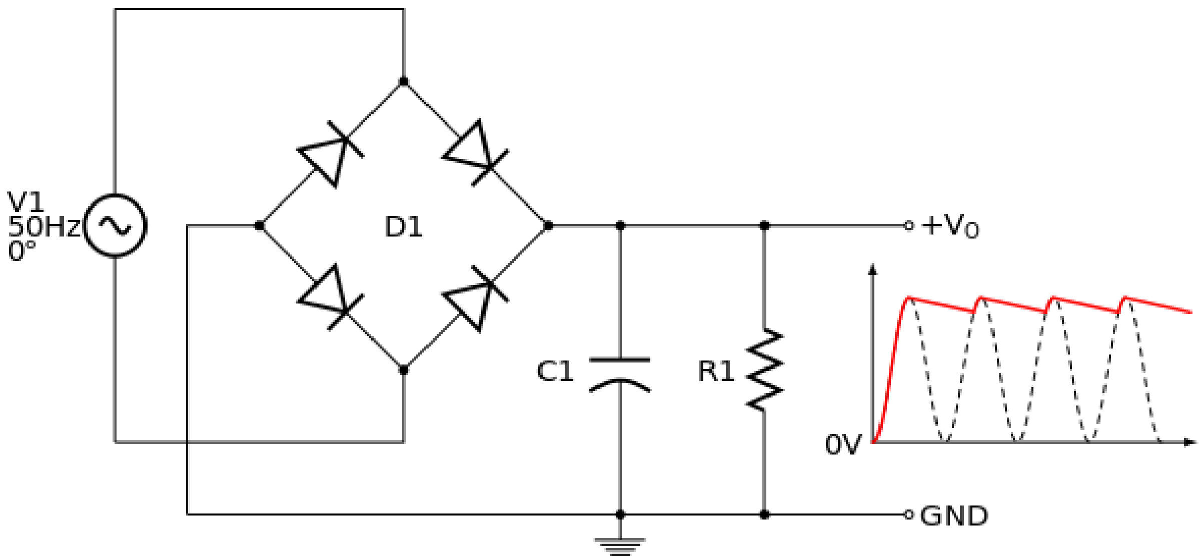


Figure 41 - AC/DC Conversion Schematic and Output

If the components can all be narrowed down to DC voltage sources, then all that is needed is a power distribution board. A power distribution board is a printed board which connects a voltage source to several components of a design (Figure 38). There are three main factors when considering the proper PCB [41]. First, the current rating is an important safety factor which explains the amount of current a board can handle. The different components of our design will have varying current values and will have to be checked for compatibility with the board. Next, the number of connectors needs to be considered. There must be enough connectors for all components in the design but shouldn't be unnecessarily large. Finally, voltage regulation must be considered. Each component will have different operating voltages and must be protected from being damaged by an excessive voltage. The design can either have a voltage regulator soldered to the board, or it can be connected by wire in series directly to the component.

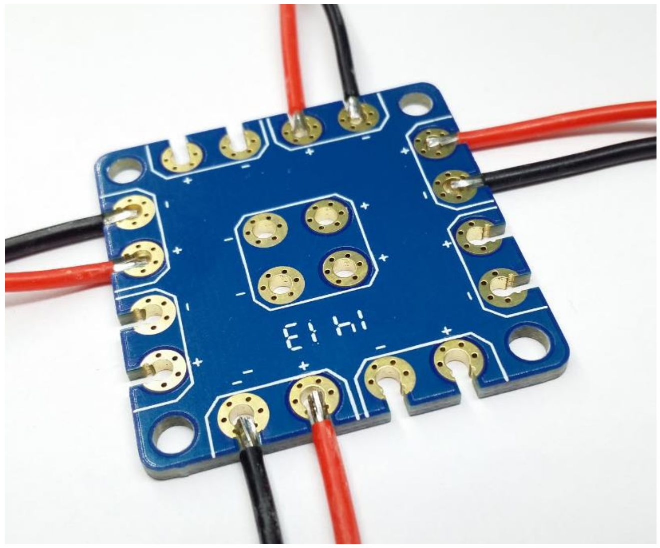


Figure 42 - Power Distribution Board

3.5 Part Selection and Comparisons

In this section we will explain how we determined what components best suit our needs for the system. Some parts will be ideal, and some parts will be a better alternative. We will do our best to demonstrate quality over all else when making our selections.

3.5.1 Motor Options

When selecting a motor, we must consider range of motion, torque capabilities, speed, stability, power, and cost. The two main motor types to consider are stepper motors and servo motors. A table best conveys the comparisons. Below we focus on what will drive our ultimate decision.

Table 17 - Stepper VS Servo

Quality	Stepper Motor	Servo Motor	Advantage
Torque	Operate at full torque	Ability to control torque	Servo
Speed	Generally less than 1200 RPM	Up to 8000 RPM	Servo
Stability	Complete Standstill stability	Stable for smooth operations	Stepper
Power	Requires less voltage for comparable torque	Requires more voltage for comparable torque	Stepper

As Table 17 suggests, servo motors and stepper motors have different advantages. The ability to control torque is extremely important for the flange assembly because each flange requires a specific numerical torque must be met not exceeded. However, some flanges require very high torque and achieving the requirement at a lower and safer voltage will be cost effective, if not necessary. Stepper motors have an additional advantage of being slightly more cost effective, but the advantages that servo motors offer may suite our application needs more effectively. In the next few sections we will learn more about the different types of servo motors and then stepper motors. It is likely that we will use both motors for different aspects of our design.

There are many different types of servo motors for us to consider for our robotic arm prototyped design. There are DC types, AC types, and very small to very large. We will consider small affordable types that are expected to work well with the 3D printed robotic are design that is being used for testing purposes as described in section 3.4.

3.5.2 Standard Servo Motors

A standard Servo Motor is useful for producing high torque values. Standard Servo Motors are specified to operate at 4.8V to 6V, with a recommended voltage of 5VDC [42]. An example of a standard servo motor is model Tower-Pro SG-5010, like the one that is shown in **Figure 43.**

Figure 43 - Standard Servo Motor

3.5.3 Stepper Motor Types

There are three main types of stepper motors, they are: Permanent magnet stepper Hybrid synchronous stepper Variable reluctance stepper. All three types of stepper motors use magnets to generate movement. Permanent magnet steppers have an equal number of gaps and operate based of some amount of repulsion or attraction. Variable reluctance stepper motors have unevenly spaced gaps where the smaller gaps have the least reluctance, and this determines movement. The Hybrid synchronous stepper motor is a combination of the two and is the most powerful. The hybrid stepper can have different magnets activated to determine movement. For our role in the design process we (the ECE members) will first consider a common and affordable stepper motor. We will discuss a common stepper motor in detail in the following section

Figure 44 - Permanent Magnet Stepper Motor

3.5.4 Unipolar Stepper Motor 28-BYJ48

Unipolar Stepper Motors 28-BYJ48 are easy to use, small, and powerful for their size. These stepper motors have permanent magnets in the rotor and attract or repulse to generate movement. This means that control can be clockwise, counterclockwise, or even standstill and maintain a torque value. The motor can make full, half or quarter steps with a full step typically defined as 5.625 degrees. The ability to perform quarter steps allows for precision and high resolution. The Unipolar Stepper Motor 28-BYJ48 has been known to work well for automatic winding machines [43] and for this reason it is being considered for our design. This motor could be used to rotate the wrist of the robotic arm. The constraint for this stepper motor is that is works best at slower speeds. Below in Figure 44 is a photograph of a Unipolar Stepper Motor 28-BYJ48 that will be tested before a decision is made whether it will be the best choice for the wrist control of our robotic arm design.

3.5.5 Drill Motor

As we began researching different motors, we noticed a sharp spike in price as we neared the power and size requirements to produce torque. A powerful torque can be met if the motor is going at a slower speed. But if the motor is not actually large enough to meet the torque, it will burn out trying to reach the torque power. This is all to say that at some point we thought about the power, size and cost of a drill. Once we started talking about taking a drill apart to obtain the motor and gear box, we decided it was a topic to redirect to the mechanical engineers on our team. The part selection is being considered and if selected will be documented in greater detail before the product is finalized.

3.5.6 Test Arm Build Material

To test the inputs and outputs we can produce with our motors and control device, we need to build a robot arm, or purchase an already built robotic arm that is ready to control.

Table 18 - Robot Test Material Selection

Build Material			
Test Arm Solutions	**Advantage**	**Disadvantage**	**Cost**
PVC	Easy to find and customize	Requires many other parts and will be very time consuming to build	Low
Store bought Robot Kit	Comes with all parts and instructions	Quality not quite worth the price point.	High ($80 - $400)
Pre-Assembled Robot Arm	High Quality	Costly to customize, significant financial loss if the design needs to change	Extremely high ($500 - $5,000)
3D Printed Design	Highly customizable, can redesign as much as we need to, extremely affordable	Cannot use a heavy load or expect very high torque production	Free as long as STL files are open source

Based on Table 18 above, we will test our input and output criteria with 3D printed parts we will use for our Robot arm. This is the best option because the disadvantage can be overcome by scaling our torque requirements. In addition to using 3D printed parts for the arm itself, we will 3D print nuts, bolts, and a wrench for the arm to use. The 3D printed parts will be free if we are able to find open source STL files, or if we create our own. This is the biggest advantage due to the many different design possibilities. It is critical that we can accommodate for failure in our design if we want a truly successful product. If we begin testing with a high-quality robot arm whether it is an $80 kit, or a $1,000 prebuilt pretested arm we will be constrained to the preexisting design. If we start with 3D printed parts, we can redesign as often as we need to. If it turns out that the 3D printed arm is not stable enough or is not powerful enough to produce torque at a scalable value, we will at least be able to determine a physical design advantage and we will be able to discover unforeseen downfalls.

3.6 Device for Electronic Integrated Circuits

For this project a compact integrated circuit design system is essential since it typically governs a specific operation using embedded systems. This section will discuss the integrated circuits that are being considered for our design. We will talk about the basics as well as use specific product examples. Our goal is to choose the device that will be versatile enough to meet our needs as we see them now as well as any changes that are likely to be made as our project progresses.

3.6.1 Controlling System

Typically, a controller is used to control some given process. Years ago, controllers were built exclusively from logic components, and were large and heavy boxes. However, as technology keeps on evolving day by day a microcontroller to this date is a simple small circuit board, shown in **Figure 45.** Although simple, there is plenty of room for growth, modification and the ability to customize for our needs exists. A component like this can handle many of our inputs and outputs that we expect and is a strong candidate for reasons like reducing overhead as mentioned below.

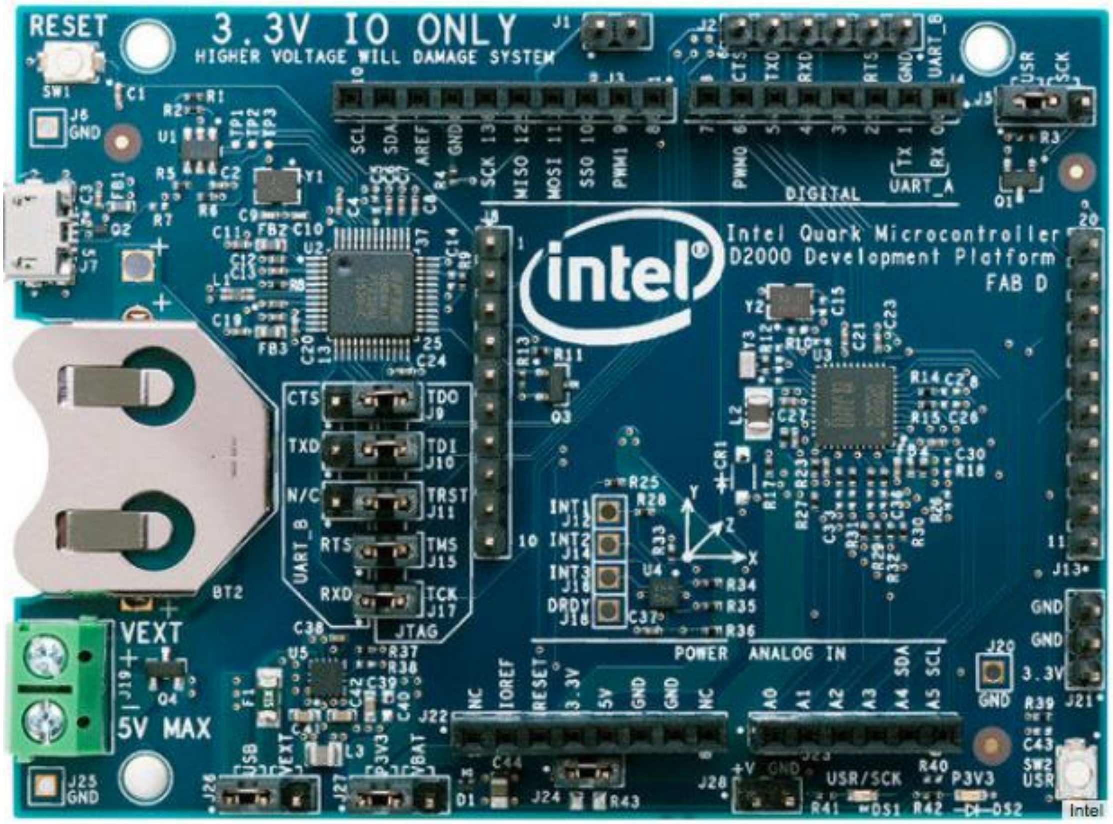


Figure 45 - Intel Microcontroller

A microcontroller is a highly integrated circuit chip, on one chip, that contains all or most of the parts needed for a controller [44]. Most of these controllers include processors (CPU), memory (RAM, EPROM/PROM/ROM), and input/output peripherals (I/O), and timers all on a single chip. These controllers are commonly found in vehicles, robots, medical devices, home appliances, etc. Therefore, they can vary on its built-in features. A typical microcontroller has bit manipulation instructions. The Intel Microcontroller is easy and direct when accessing I/O pins and it is quick and efficient in responding to interrupts. Given the definition and features that the microcontroller offers it is a one-chip solution [45], shown in

Figure 46, in which drastically reduces the part count, design standards, and overall cost.

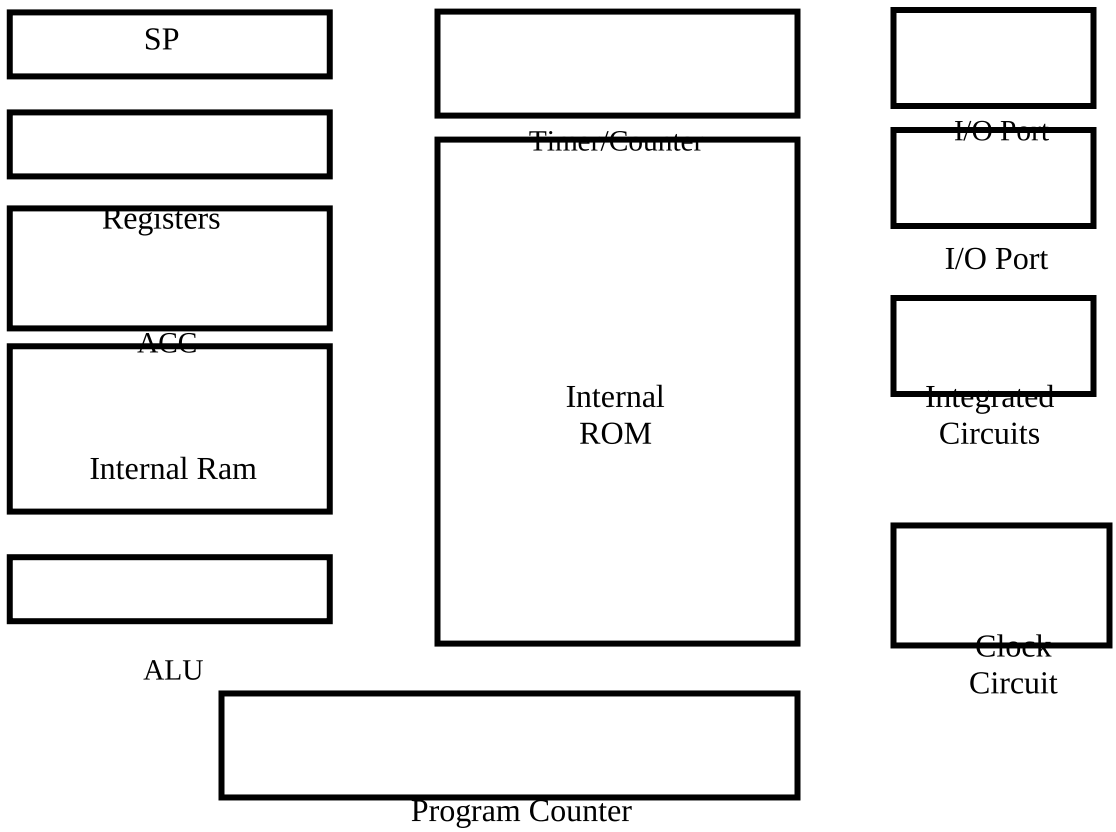


Figure 46 - Block Diagram of a Microcontroller

3.6.2 Microcontrollers Considered for Project

The three microcontrollers that we are considering for this robotic flange projects are Raspberry Pi 3 BCM2847, Texas Instruments' MSP430FR6989 and Arduino's UNO Rev 3 – Atmega328P.

3.6.3 Raspberry Pi 3 - BCM2837

Raspberry Pi offers a variety of boards, but for our project we are considering their BCM2837. This board is dependent of the Broadcom BCM2837 MCU with a 64bit ARMv8 quad core Cortex A53 processor at a speed of 1.2GHz with dual core. In addition, it has 512KB cache memory plus micro SD slot for additional storage, ethernet, wi-fi, and Bluetooth connectivity, and built in ports that offer HDMI, 3.5mm analogue audio-video jack, 4 x USB 2.0, Ethernet, Camera Serial Interface (CSI), and Display Serial Interface (DSI). All these features can be seen in Figure 47. This board acts like a "mini" computer.

Therefore, it might be more than what we want in a board, but we will keep it under consideration since we can be adding more enhancements later in the design [46].

Figure 47 – Raspberry Pi 3 BCM2837

3.6.4 Texas Instruments - MSP430FR6989

The MSP430FR6989 is manufactured by Texas Instruments and it is the best new thing in TI's technology platform. This microcontroller is well known for its use in required laboratories for engineering degree required classes. This device features a 16-bit MCU that can run on a 16MHz clock with 128 KB nonvolatile FRAM. In addition, it operates between 1.8V to 3.6V, has five 16-bit timers, and is supported by both, Mac and Windows, environments. This board is highly recommended for its familiar coding language, ultra-low power achievement, and low cost. Additional features can be easily seen by the given block diagram, illustrated in

Figure 46 [47].

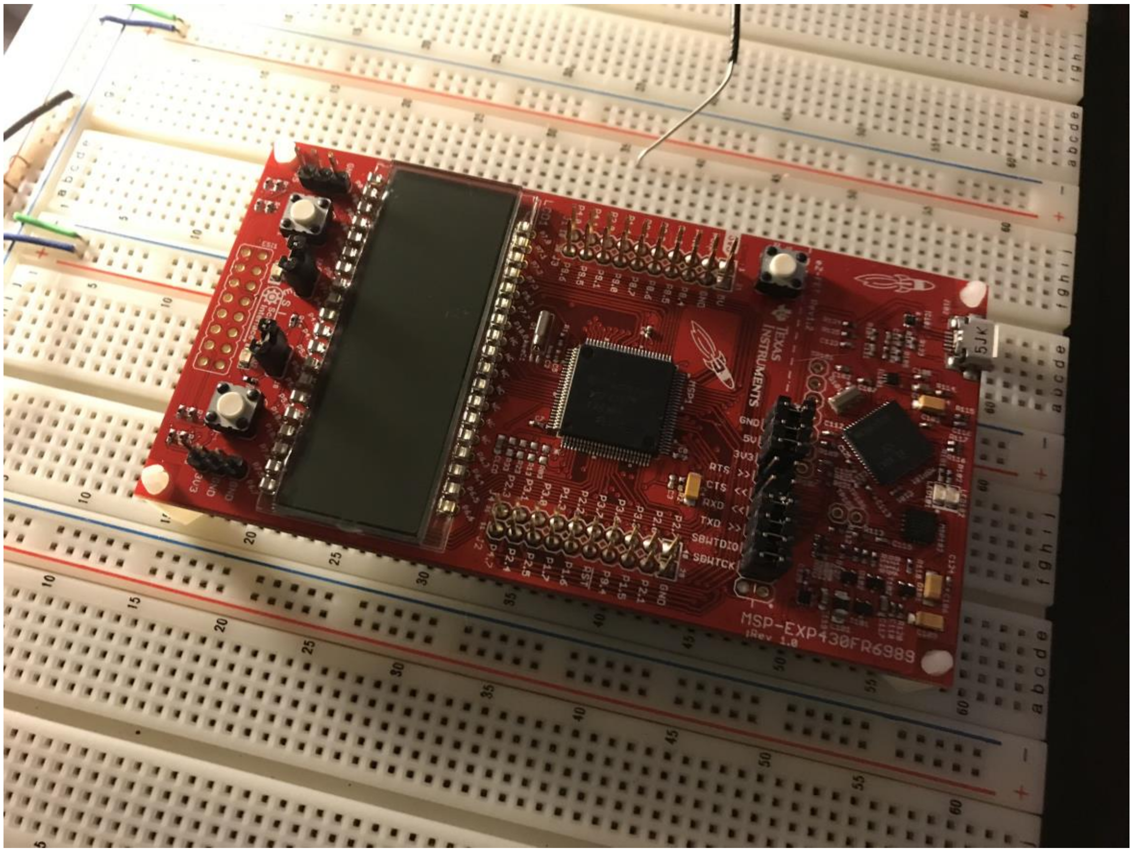


Figure 48 - TI MSP430FR6989

3.6.5 Arduino UNO Rev 3 – Atmega328P

The ATmega328P is part of the Arduino UNO family and it is one of the easiest boards to get started when working with coding electronics. This type of microcontroller is very popular because of it does not rely on heavy coding and it is considered very simple in terms of programming compared to other boards available in the market. This apparatus has an operating voltage of 5V, 32KB of flash memory, and a clock speed of 16MHz. Even though this type of board is not fully integrated in our degree courses many students choose to use it for side projects as it offers great performance and user friendliness. In addition, the system is supported in Mac and PC software. Shown below, on Figure 49, we can see the basic design of the Arduino board [48].

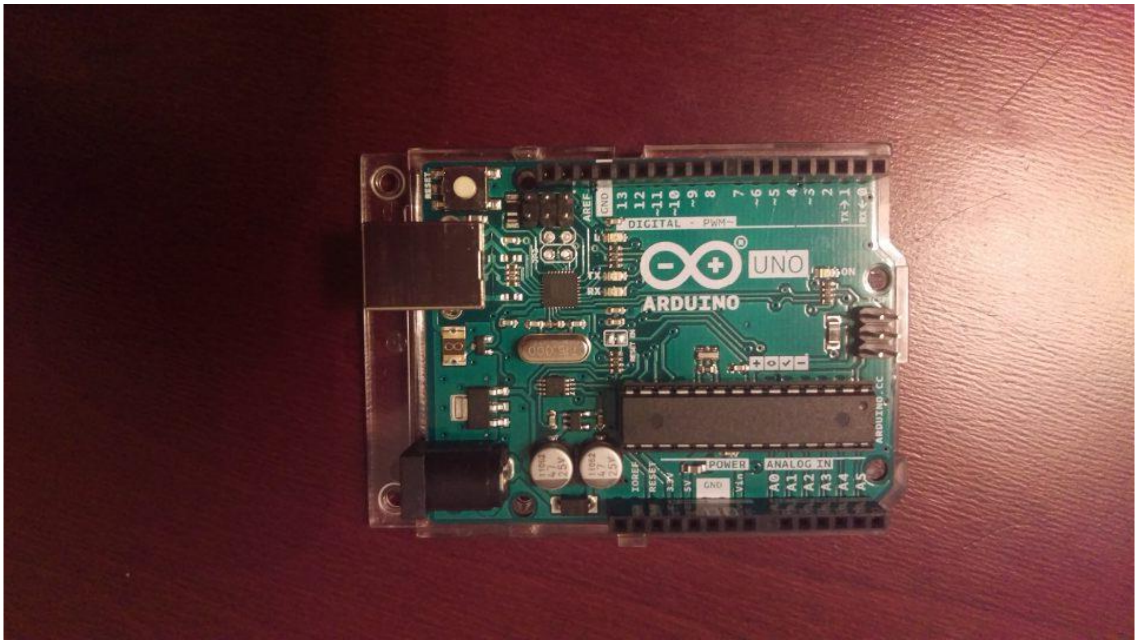


Figure 49 - Arduino ATmega328P

For ease of comparison between the three microcontrollers that we are closely considering for this project we have created **Table 19**, shown below. This table lists the main features that we are observing on each board. All the boards mentioned throughout this section comply with our requirements. However, there are some that offer more than what we need. As we continue further with the design of our project, we will make an ultimate decision in what microcontroller to use.

Table 19 - Microcontroller Comparison

Feature(s):	**MSP430FR6989**	**Atmega328P**	**BCM2837**
Operating Voltage	1.8V to 3.6V	5V	5V
Temperature Range	-40°C to 85°C	-40°C to 85°C	-25°C to 85°C
Maximum Clock Frequency	16-MHz	16-MHz	1.2-GHz
RAM	12KB	12KB SRAM	1GB
Memory	128KB of Nonvolatile Memory	32KB	512 KB L2 Cache
Analog I/O	Both	Input Only	Both
Digital I/O	Both	Both	Both
GPIO Pin Count	83	20	40
Bit Count	16-bit	8-bit	64-bit

Lower Power	Yes	Yes	Yes
Power Consumption	Active Mode: 100 µA/MHz Stand By: 0.4 µA Shutdown: 0.02 µA	Active Mode: 200µA/1MHz Shutdown: 0.1µA	Active Mode: 3500mW
Board Price	$17.99	$22.00	$34.99

3.7 Control Devices

This section provides descriptions and comparisons of three different control technologies and gives our choice of which technology to use. From there, we include a few possible examples to use from that technology and narrow down which model we'll use for our design. Included will be figures representing the hardware and functions of these components as well as tables to give side by side comparisons of the technology and specified components.

3.7.1 FPGA

Field Programmable Gate Arrays (FPGAs) are programmable semiconductor devices designed with an array of logic blocks, gates, and flip-flops [49]. They are usually based around a two-dimensional array with global interconnect corridors between the array cells [50]. Each cell in the FPGA is a logic block containing both logic devices and RAM and can be used to perform a specific function. These devices can be programmed and reprogrammed to fit the desired application by the user. The programming of this device is specified by a hardware description language such as VHDL or Verilog depending on the hardware used [51].

Because of the programmable nature of the FPGA, it has applications across many markets and can made to fit several tasks if the user is capable of programming the task into a specific hardware language. One issue with this flexibility though is that it requires a much larger power drain than certain devices [49]. Another issue with FPGAs is the relative difficulty of programming the device due to the complexity of Hardware Description Language (HDL). However, FPGA is a good device to use if the application requires quick action and multiple inputs and outputs.

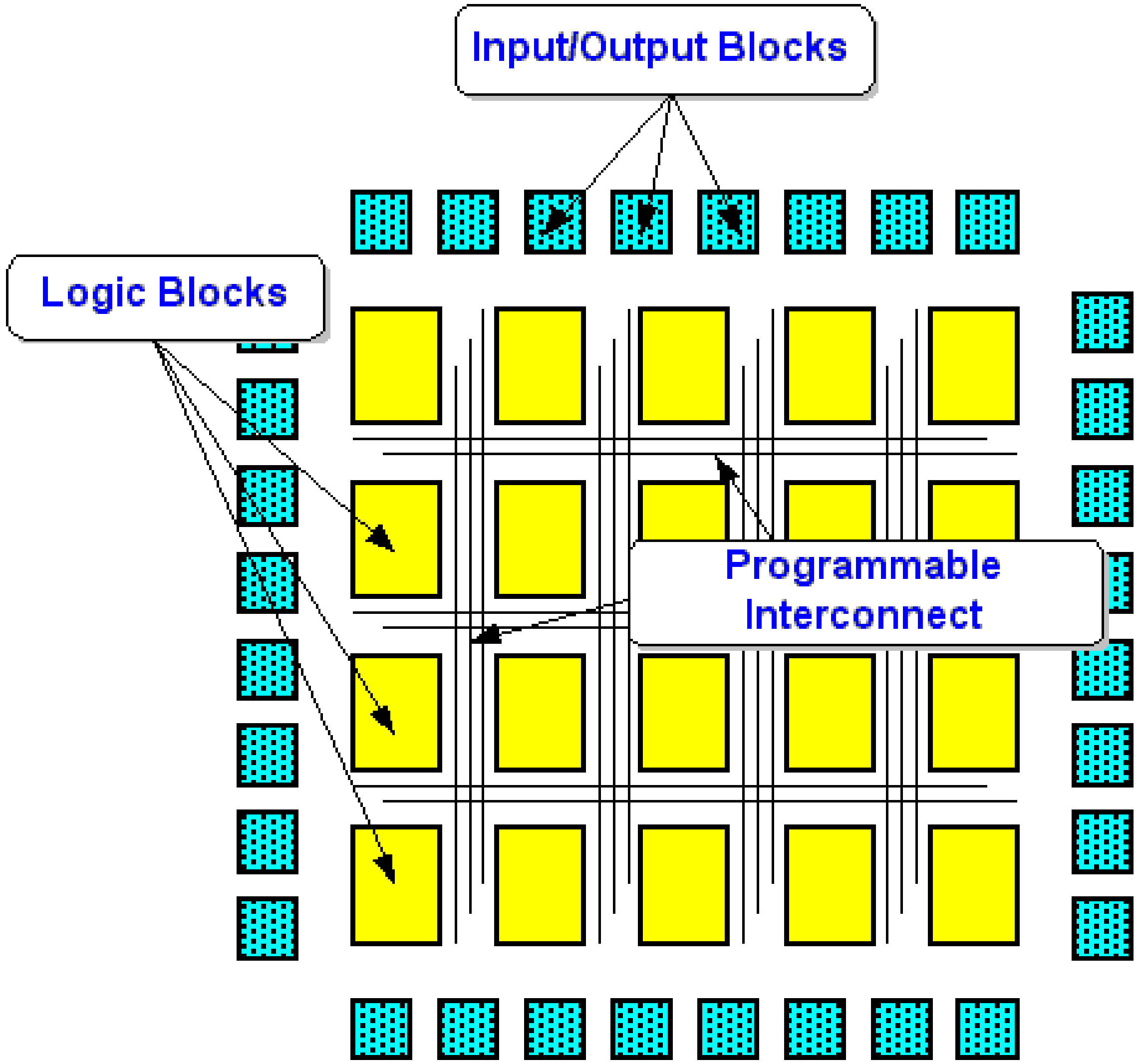


Figure 50 - Typical FPGA Architecture

3.7.2 Microcontroller

A microcontroller is a computer placed on an integrated circuit designed to perform a single task or application [52]. The microcontroller is built around a central processing unit (CPU) which controls the peripheral functions of the microcontroller. The CPU contains the Arithmetic Logic Unit (ALU), basic operating registers, and various logic devices to control resets, interrupts, etc. The microcontroller also contains nonvolatile memory for the program, RAM memory for data, input/output ports, busses, and the clock. Some other peripherals include analog-to-digital converters, digital-to-analog converters, and communication interfaces [50].

Because of the loaded-up nature of a microcontroller, it is often used as an embedded device for specific applications in machines [49]. The single-application aspect of a microcontroller makes it more difficult for use by hobbyists and designers. However, this also makes it a tool with low power requirements. It also decreases overall costs. If you need a device that interfaces with the real world but requires few computations from sensors, microcontrollers are often a good choice [53].

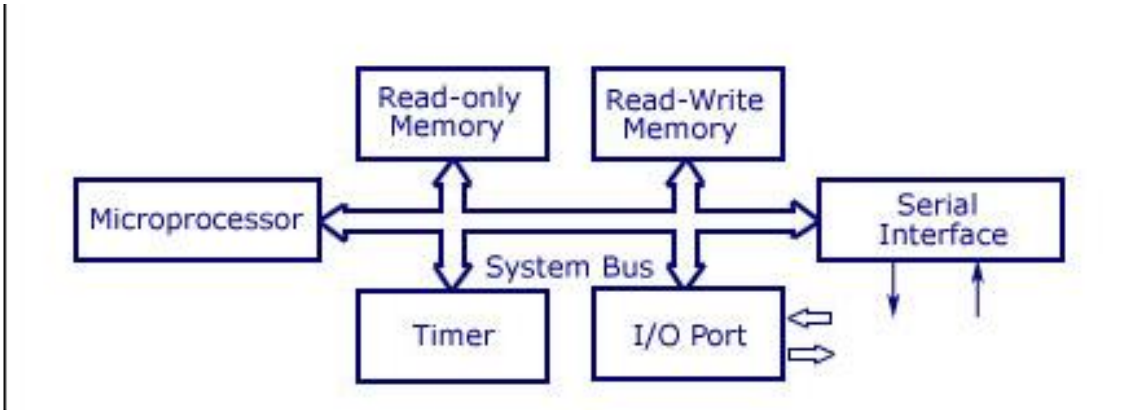


Figure 51 - Typical Microcontroller Anatomy

3.7.3 Single Board Computer

A single-board computer (SBC) is a computer which is a complete computer in which a single circuit board comprises memory, input/output, a microprocessor and all other necessary features [54]. While an SBC doesn't have a lot of raw computing ability or memory, it comes fully equipped with all necessary peripherals found in a desktop and an operating system for programming [55].

The SBC has the flexibility of the FPGA, but with a lot more computing power and a workable operating system. Because of this, the SBC is open to many applications if it can be worked through the operating system. However, more applications can often make input and output integration difficult with the board. As well, peripheral inputs can often fry the board or require separate power supplies depending on the application. If the user application requires heavy computing or a preferred programming language, an SBC is probably the best choice for control.

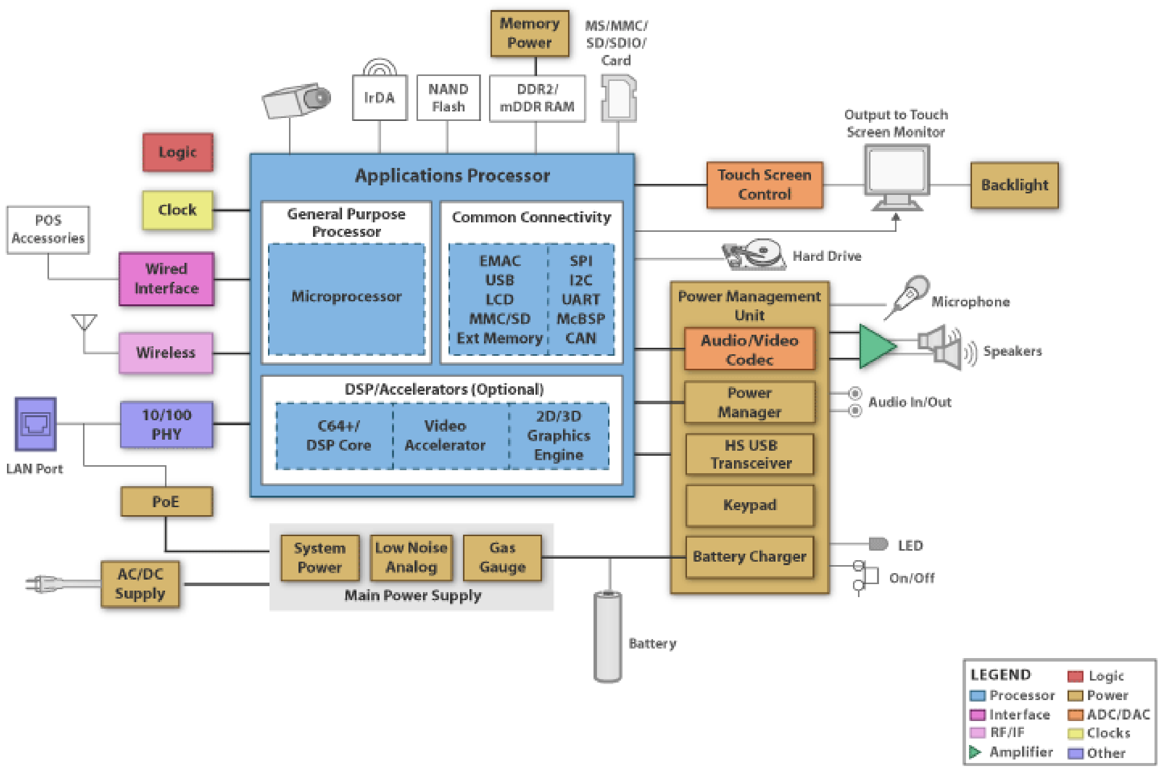


Figure 52 - Block Diagram For an Example of SBD

Table 20 - Control Devices Comparison

	Pros	Cons
SBC	1. Fully equipped with peripherals 2. More computing ability than FPGA 3. Workable OS 4. Flexible	1. Input and output integration more difficult 2. Needs separate power supply for large peripherals
PGA	1. Flexible 2. Fast 3. Can handle multiple parallel I/Os	1. Larger power drain 2. VHDL more difficult to program than other languages
MC	1. Low power requirements 2. Equipped with many peripherals 3. Low cost 4. Easy applicable to a PCB	1. Singularly focused 2. Not for heavy computation 3. Hard to apply to multiple applications

3.7.4 Sensors

It is important for our project to detect the flange's bolt locations and how much torque is being applied to each bolt. Therefore, for out project we plan on adding sensors that have the capability of detecting the bolt's location and applied force torque. These are necessary since we do not want to miss a bolt being torque or have the wrong torque applied that could lead us having a non-symmetrical flange that can potentially cause damage to people and the environment [56].

3.7.5 Types of Sensors

For our design we will be looking at a sensor encoder. An encoder is a sensor of a mechanical motion that produces digital signals in response to something in motion. An encoder device can provide users with information regarding position, velocity, and direction. In order to get an idea of where the sensor is located, **Figure 53** illustrates the block diagrams of an encoders on a dual shaft motor. There are two different types of sensor encoders that the user can choose from. Linear and rotary are the basic types of encoders that can be broken further down into two main types. Before we discuss the difference between linear and rotary encoders, we will evaluate their two basic types.

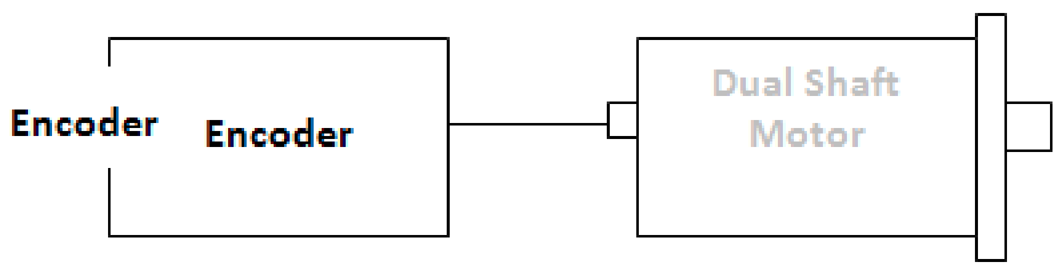


Figure 53 - Encoder on Dual Shaft Motor

3.7.6 Basic Types of Encoders

For the rotary and linear encoders there are two types. Even though they are very similar to one another they differ in their physical properties and their interpretation of movement. The first type is the incremental encoder, also known as the quadrature encoder. This encoder uses sensors with optical and mechanical/magnetic index counting in order to seize angular measurements. These sensors work by utilizing a transparent disk containing opaque sections in which are spaced out with the purpose of measuring movement. A light emitting diode is used to pass through the glass disk that is detected by a photo detector which causes the encoder to generate a train of equal spaced pulses as it rotates, as shown on **Figure 54**. The output of this encoder is measure per revolution in order to keep track of position and measurement of speed.

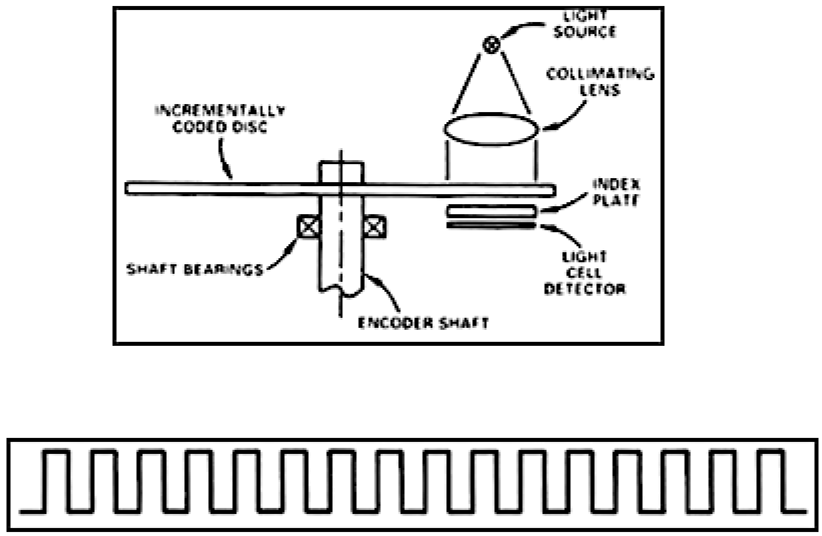


Figure 54 - Pulse Train Produced by Incremental Encoder

The following basic type of an encoder is the absolute encoder. This encoder contains component already found on the incremental encoder. However, instead of using a disk that measures equally spaced lines on a given disc, like in the incremental encoder, this sensor is implemented with a photodetector and an LED light source with a disk containing concentric circular patterns instead. Shown on **Figure 55**, we can see that the absolute encoder utilized mask in between the photodetector and the disk. This causes the output signal to be in digital bits giving a unique position. Therefore, it uses a unique bit configuration.

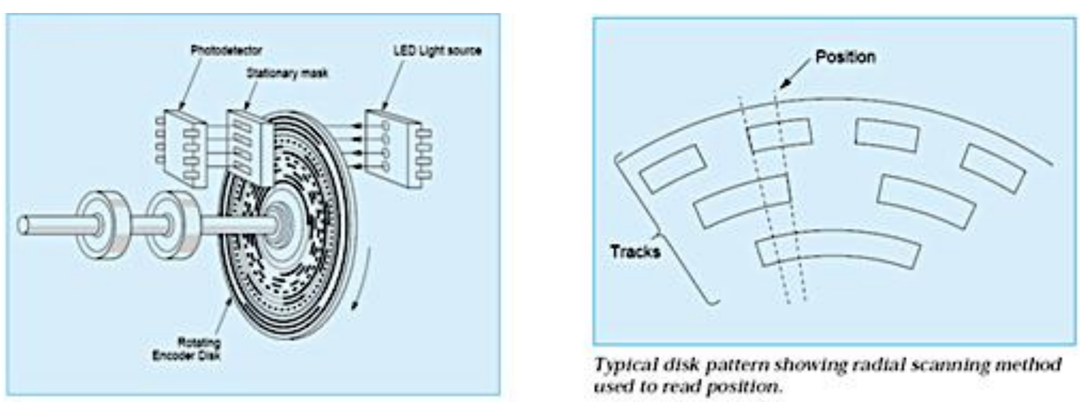


Figure 55 - Components of Absolute Encoder

3.7.7 Linear Encoder

A linear encoder is a sensor, transducer that encodes position. The sensor interprets the reading scale and converts it into position either in analog or digital signal that is later transferred into a digital readout. This is measured by the simple movement that occurs from changes in position and time. This method can be found in optic and magnetic linear encoders, which they can differ in physical properties. These first method of linear encoders that will be discussing is the Optical Linear Encoder. The optical linear encoder works by using a light source and a lens in order to produce a parallel light beam that passes through four different windows, shifted 90 degrees apart, of a scanning reticle. The light then passes through a glass scale, used to transform the detected light beam once the scanning unit moves, and detected by photosensors shown in **Figure 56.**

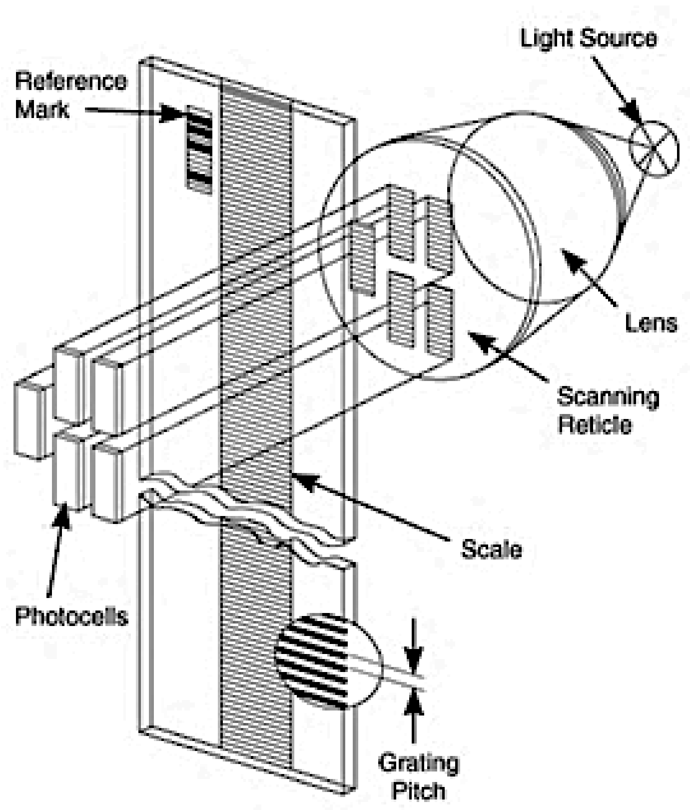


Figure 56 - Linear Encoder Components

The other method for linear encoders is using a magnetic sensor to produce an analog output for two different channels. While the magnetic sensor is utilized to pass along a magnetic scale, once the sensor detects the change in magnetic field it outputs a signal frequency proportional to the measure of speed and the sensor’s displacement**.**

3.7.8 Rotary Encoder

The second type of encoder that we will be analyzing is the rotary sensor. For this project we are looking at the magnetic rotary encoder which consists of two main parts: a sensor and a rotor, shown in **Figure 57.** The rotor in the encoder turns with the help of a shaft and with the presence of altering evenly spaced north and south poles around its circumference [57].

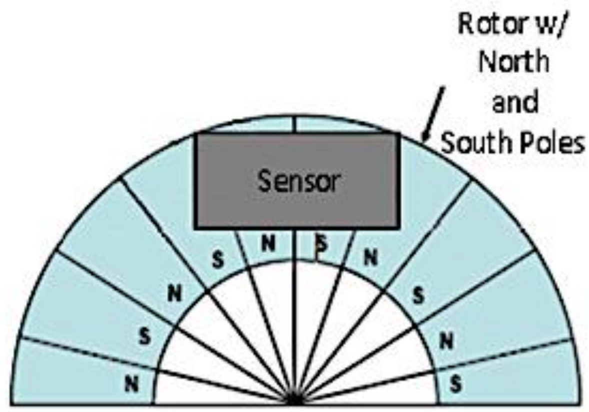


Figure 57 - Rotor and Sensor Using North and South Poles

When discussing the methods of detecting the changes in magnetic field, the rotary encoder considers the Hall Effect and the Magneto Resistive sensors. The Hall Effect, commonly seen in optical encoders, work by the detection of change in voltage by magnetic deflection of electrons. The magneto resistive sensor works by detecting the change in resistance caused by magnetic fields.

Figure 58 - Magnetic Encoder

3.7.9 Communication Encoder

Lastly, we have the communication encoder illustrated in **Figure 59**. This type of sensor contains the same fundamental components that of the incremental encoder. However, this encoder tracks alongside the outer edge of a disk in order to give U/V/W outputs. These encoders work by utilizing a transparent disk included with opaque sections equally spaced to determine movement. In addition, a light emitting diode is used to pass through the glass disk to be distinguished by a photodetector. During this ongoing light emission transmission, the encoder will generate equally spaced pulses at it moves. The output is then measured in pulses per revolution so that one can keep track of position and the determination of speed. The outer part of the encoder disk consists of commutation tracks that communicate with a controller allowing it to know the position of the motor poles so that the proper input can me supplied to the motor. These tracks are used to provide efficiency and proper current to the motor and its rotation.

Figure 59 - Communication Encoder

In conclusion, these sensor encoders have become a vital source for many applications that require feedback when measuring are concerned with speed, direction, and distance. Due to its reliability low cost, and compact size we can find many of these sensors in the automotive, medical, industrial, and military industries.

3.7.10 Part Selection Summary

Our team has ultimately decided to target available resources that possess the quality required to create a baseline for the robotic flange assembly. As previously mentioned, our team (team here meaning all 16 interdisciplinary students) has not finalized our design. This significantly contributed to our part selection. Below in Error! Reference source not found. is a photo of all parts we are currently in possession of. We will be using the electrical components on our final design, but we may be adding additional devices such as sensors mentioned in section 3.6.3.

Figure 60 - Parts in Hand

The crucial part of our role for the robotic flange assembly project is independent of the design portion that the mechanical engineers are responsible for. This is because regardless of any sort of carriage or pulley type system that will be inevitably become part of our overall design, the design must have some sort of appendage and gripper to complete tightening of bolts. While we may only have to adjust any algorithmic data structures for controlling the device after the finalized design, the physical components changing could lead to a catastrophic financial waste. It is our objective to abstractly define the task of robotic flange assembly in the most efficient and correct way.

3.8 Software for Hardware Design

A big part of this project involves electrical connections and components that will be required for the robot to input and output given data accordingly. When designing this project, we will need to use software that will enable us to design and test a complete breadboard model alongside with its components, interconnections, and output signals.

In the marketplace there are multiple computer design software's that allow us to design, test, and implement a created circuit. This can be achieved without the use of a computer and using a breadboard instead. However, for most students it is easier to create a virtual schematic first to later build it on a breadboard with more ease. Therefore, firstly, in this project we will be developing a schematic for the circuit and making sure that all the components and connections are placed and connected properly. Once the schematic is created and outputs the desired outcome, we will assemble a breadboard with such components in order to acquire the same or equivalent results from the computer schematic and look for additional information such as oscilloscope graphs and voltages in real time.

3.8.1 Fritzing

The first software that we are considering for the electrical hardware portion of this project is Fritzing. The software, Fritzing, is an open source hardware initiative that makes electronics accessible as a creative material for anyone [58]. The software tool allows users to better understand electronic circuits, document and share data while working on your project, and prepare your project for future production. Since we will be developing a PCB for our project, I believe this is a worthy software to start developing our schematic since it comes equipped with multiple "viewing" options. The first step to our hardware design will be creating the breadboard that will be later manufactured in a PCB. The software has a "Breadboard" tab that allows us to test our connections between components and the programed microcontroller, shown in **Figure 61**.

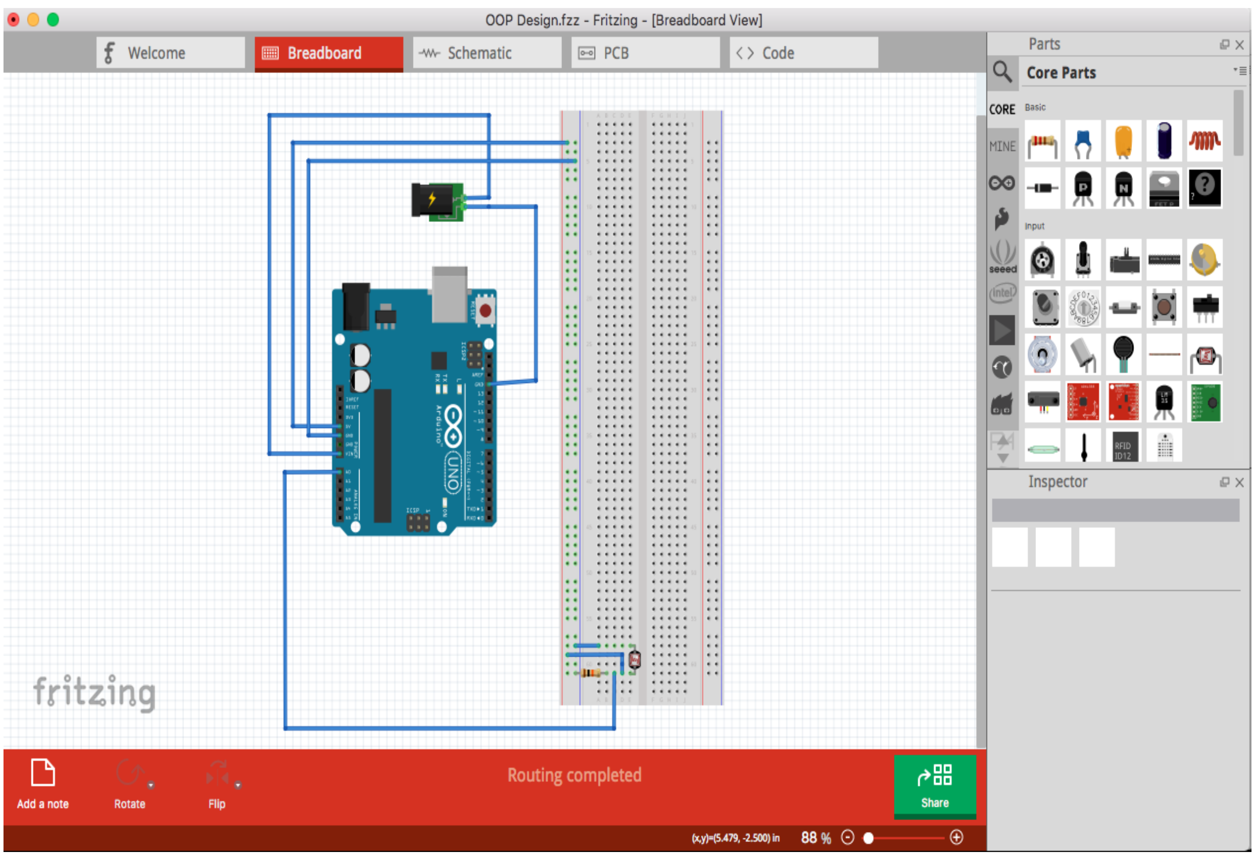


Figure 61 - Fritzing Breadboard View

This window shows a clean picture of the user's wired breadboard. Most of the library parts, those being capacitor, transistors, resistors, LEDs, and others, already come with the software. These standardized library parts can be modified as needed by changing their values to those needed for the project. As for the microcontroller, the software offers brands like Intel, Arduino, and Texas Instruments, among others. The microcontroller can be pre-programmed in the "Code" tab for ease of testing. Once the breadboard's connections have been carefully inspected the user can procced to view the overall schematic of the breadboard, shown in **Figure 62**.

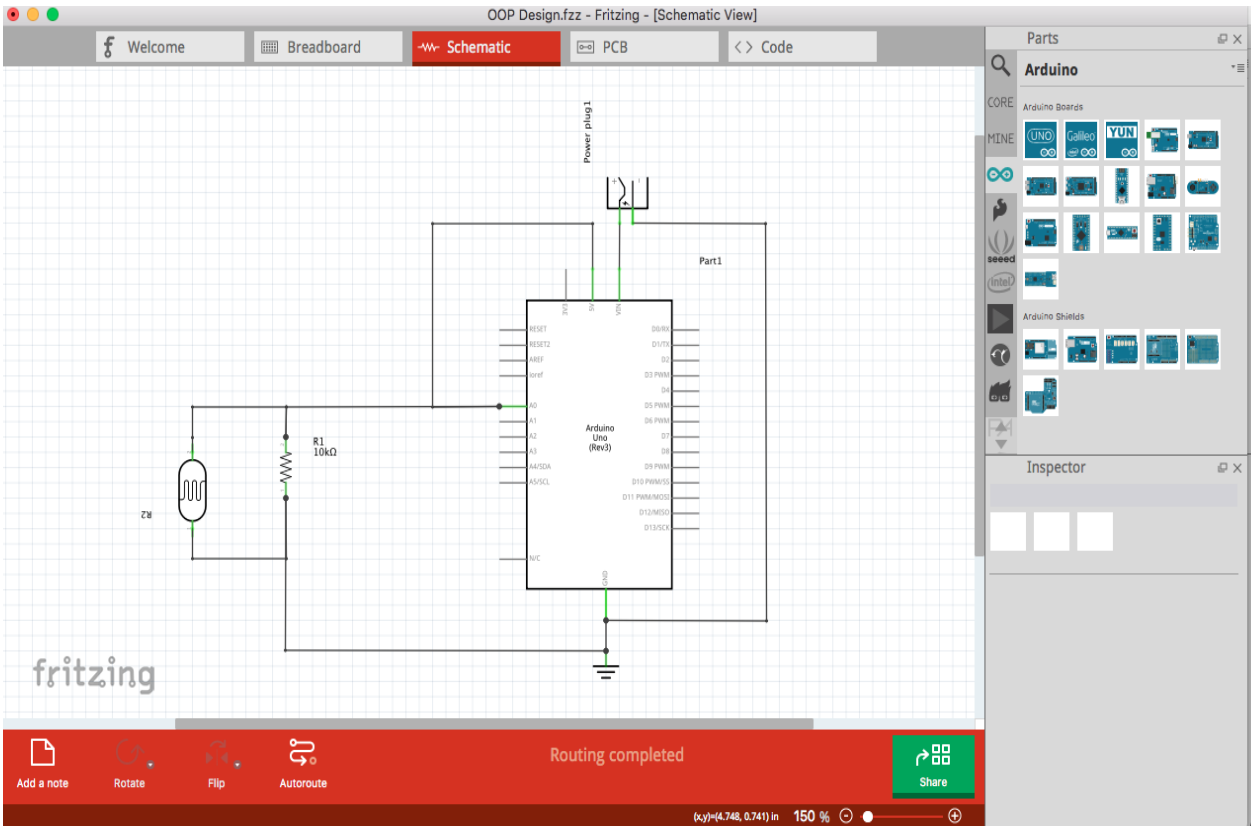


Figure 62 - Fritzing Schematic View

In the "Schematic" tab the user can effortlessly see the diagram in symbolic and simplified form of the breadboard's technical context rather than realistic as shown in the "Breadboard" tab. Lastly, once the breadboard and schematic are complete then comes the PCB design. The PCB will bring the electronic circuit to life in the physical form once sent to be manufactured. The software on the "PCB" tab let us show the design schematic in a PCB design standpoint, down in **Figure 63**.

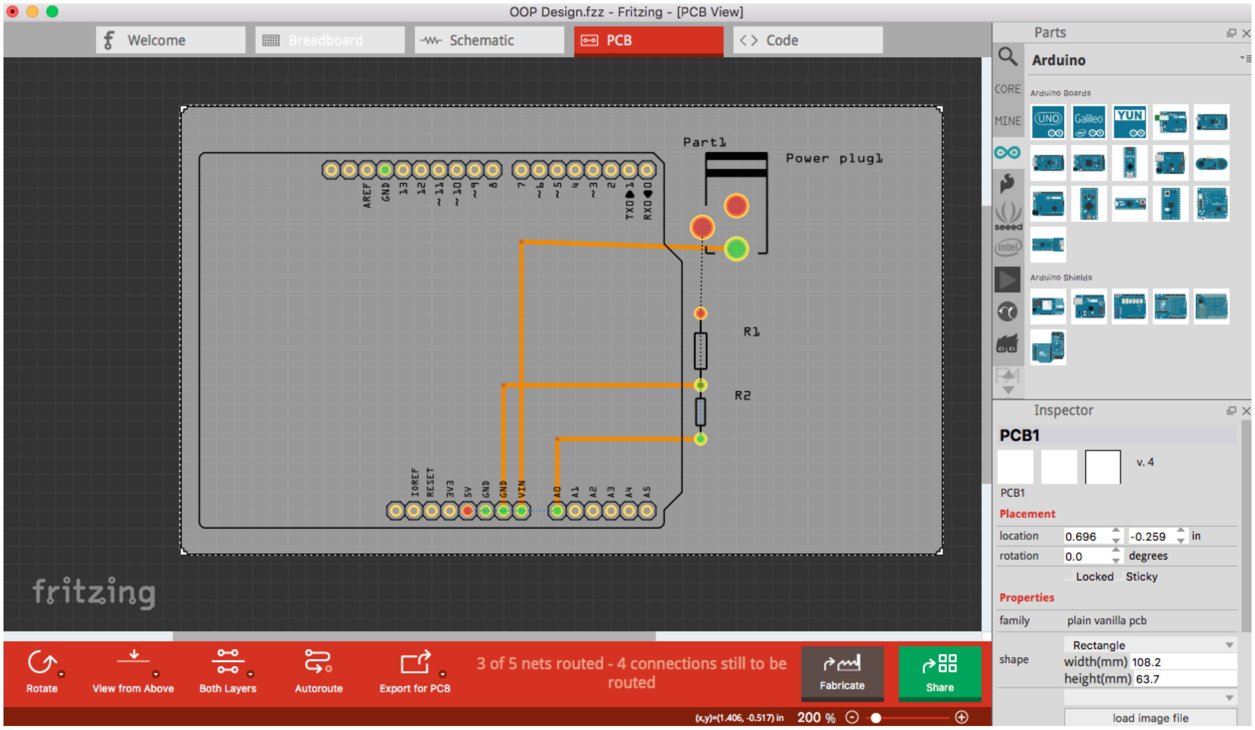


Figure 63 - Fritzing PCB View

Once we have acquired the proper specifications of the mechanical's team design, we will be including the accurate schematics illustrating our design in this section.

3.8.2 Autodesk Eagle

The Autodesk Eagle is our second software consideration for this project. This software is farther more suitable for the business industries, and possibly the one that we will be using for our PCB design when sending it to manufacture. The Autodesk brand has been known for its advance optimized computer software. Autodesk's Eagle software offers users the ability to design a schematic with already included library parts [59]. Once the schematic design is complete you can view the PCB board and make edits to it. This software is not as user friendly compared to Fritzing. This software does not offer the "Breadboard View", unlike Fritzing, for the user to set up their board in the computer. However, Eagle supports more assortments when looking at library parts and it can hold a bigger schematic capacity than Fritzing.

In **Figure 64**, we show the layout of the schematic design in the Eagle software. Since our design can get complex, we are more likely to choose Eagle as our number one schematic layout and the one be sent out to manufacturing. As our design evolves, we will be updating these schematics to include the latest ones that will be accounted for our project.

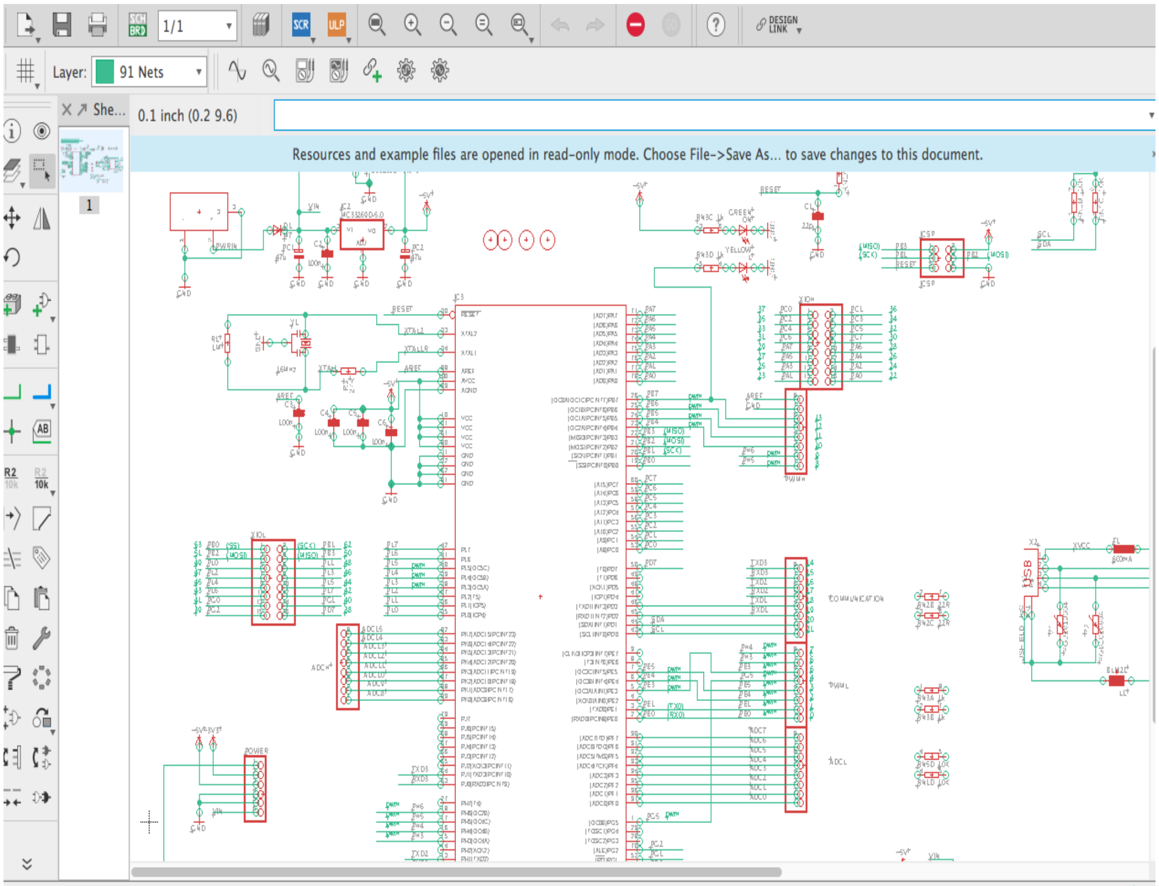


Figure 64 - Eagle Schematic View

3.8.3 EasyEDA

For the last software we will be considering is EasyEDA. This software offers many of the capabilities discussed in the previous ones. However, EasyEDA has the ability of working in the hard drive of your computer or based of the cloud giving its users the ability to work remotely. In addition to the software being based of the cloud, people can collaborate when designing the schematic(s). The collaboration will allow the team to perform better and more efficiently as they make their way through the project. When the team's prototype PCB is ready for manufacture the software has two unique features to make it easy and quick when sending the PCB design to next step, manufacturing. One of the features is giving the user a breakdown of specific branded controllers, amplifiers, motors, sensors, and other components that they can choose from. This information is provided by one of the fastest developing online stores of electronic known as LSCS based out of China. In addition, they provide their approved manufacturer for PCBs, known as JLCPCB. **Figure 65**, shown below, will show the layout of the software based of the web cloud [60].

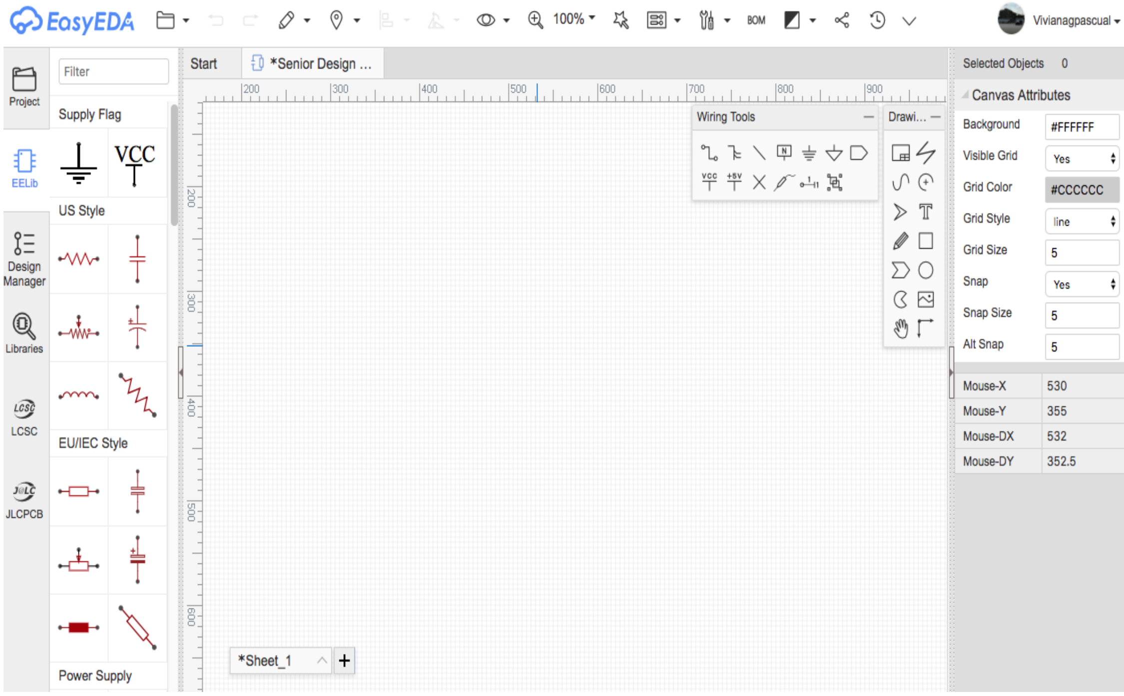


Figure 65 - EasyEDA Schematic View

4. Related Standards and Realistic Design Constraints

This section of the paper discusses related robotic standards and the design constraints that the robotic flange assembly will face throughout the design process.

4.1 Standards

“Engineering standards are documents that specify characteristics and technical details that must be met by the products, systems, and processes that the standards cover [61].” The reason why engineering standards were even developed and created in the first place were to ensure products meet safety requirements, to make sure systems, products, and processes are persistent and repeatable, and to ensure the products minimum overall performance. As technology keeps evolving, standards become extremely important. “Standards promote safety, reliability, productivity, and efficiency in almost every industry that relies on engineering components or equipment [62].” When technologies start becoming more advanced such as working with collaborative robots, it is important that these designs be tested constantly and made sure they are kept up with engineering standards.

Standards are voluntary, not mandatory. Standards provide communication between producers and users. They serve as a common language that defines quality and establishes safety criteria. If procedures are standardized, costs are lowered, training becomes simplified, and interchangeability and interoperability is allowed. Standards and codes are documents that are constantly revised to show changes in development and technical advances such as new designs, new materials, or new applications.

Engineering standards are summaries of the best practices for industrial and manufacturing use. The specifications for the standards include functional, electrical, and mechanical terms and aspects that allow proper usage of available components to build a system. They can also define how tests should be performed and how products should be designed. Engineering standards help with the growth of new technologies, protect public health and safety, enhance the quality of products, and allow for international and global trade.

Standards can be grouped together based on three different categories. The first one is by process standards, the second one is by standard test methodologies, and the third one is by performance standards. Process standards are standards that talk about the overall general system or a basic way of doing things. Standard test methodologies are standards that discuss a specific test being performed and the testing protocol used to evaluate any properties or performance levels of the product. Performance standards are standards that describe the performance qualities and attributes [63].

There are many different types of engineering standards that correspond with different types of engineering. Some of these include the ANSI Standards, ASME Standards, ASTM Standards, and IEEE Standards. As computer and electrical engineers, our primary focus will be the standards that fall under the ANSI Standards, the IEC Standards, the ISO

Standards, and the IEEE Standards. Our overall project will focus on the four previously mentioned standards as well as the ASME Standards.

Figure 66 – Standards

4.1.1 Standards Related to Project

The ANSI has created a new version of the NSSN search engine for national, foreign, regional, and international standards and regulatory documents. "First launched in 1997, the NSSN: A National Resource for Global Standards is a cooperative partnership between ANSI, U.S. private-sector standards organizations, government agencies, and international standards organizations. The site has become the leading provider of technical data and information about developments in the global standardization arena [64]." The redesign of this website allowed users to find standards and related documents easier. ANSI is a private non-profit organization whose goal is to promote, facilitate, and safeguard the integrity of the voluntary standardization and conformity assessment system. With the launch of this new design, the search for standards is simplified.

One of the standards obtained from the ANSI search engine was the IEEE Standard of Ontologies for Robotics and Automation (IEEE P1872). As it relates to robotics and automation, ontology is defined as the branch of metaphysics dealing with the nature of being. It describes the study of things that exist and how they are grouped and related to each other. For robotics, it is important to know how robots exist and how they are grouped and related to each other as well as with humans. "The IEEE Standard of Ontologies for

Robotics and Automation is designed to simplify programming, extend the information-processing and reasoning capabilities of robots, and enable clear robot-to-robot and human-to-robot communication [65].” This standard is mostly related to working on ontology specific to industrial robots, focusing mainly on assembly tasks. Because technology is progressing, and robots are becoming more and more advanced, such as working collaboratively with humans, there is a need for robots to have clear, concise communication.

“The working group's core ontology for robotics and automation, or CORA, is an important step toward achieving this shared understanding. It establishes a formal way of representing knowledge that robots possess to perform tasks in their area of activity such as manufacturing plants or hospitals. This "common ground" enables efficient and reliable exchanges of information and integration of new data [65].” With this structured base of knowledge for the standard, an industrial, manufacturing robot, for example, will know what its required tasks are, how much weight it can lift, if it can collaboratively work with humans, how to detect objects nearby, and other performance-defining features. When new tasks arrive, the robot will know how to analyze the given work and if it’s capable of performing the desired task or not.

This standard is new to IEEE Standards and a starting point for robotics and automation standards. To summarize, this standard is designed to simplify programming, extend the information-processing and reasoning capabilities of robots, and enable clear robot-to-robot and human-to-robot communication. Its main goal is to define what a robot is and how it works with humans as far as the standard goes. It also defines common robotic concepts and provides an organized framework for succeeding efforts to develop for specific classes of robots [65]. These specific classes of robots include industrial robots working on assembly tasks. It will be proposed as a standard for groups working on industrial robots. Displayed in Figure 65 below shows a flowchart that explores the IEEE Ontology for Robotics and Automation for agent interaction. The blue boxes show the concepts from Suggested Upper Merged Ontology (SUMO), while the black boxes show the concepts from point of sale (POS) or CORA.

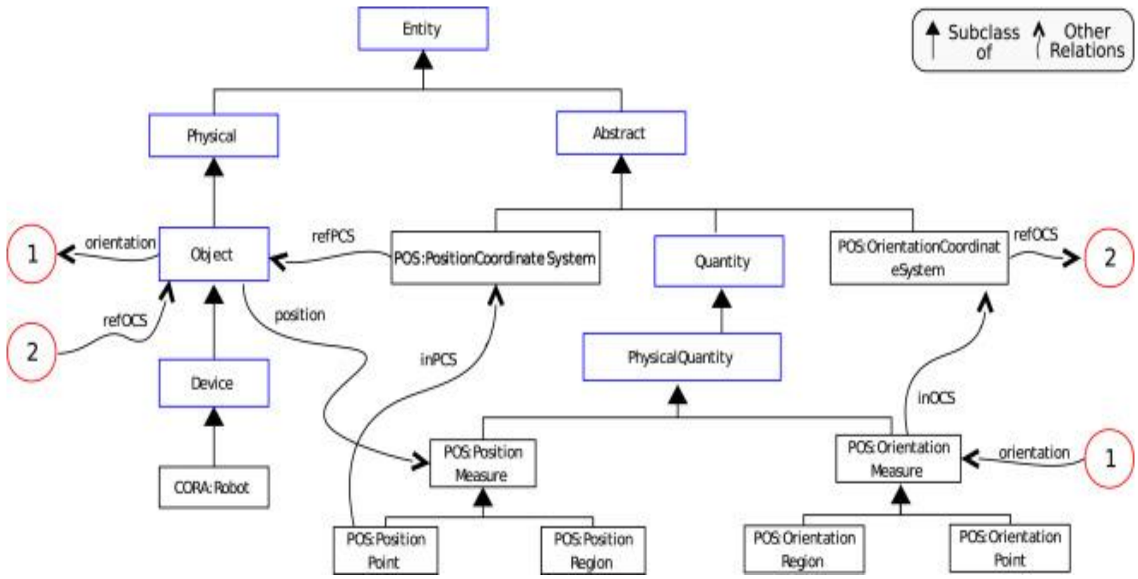


Figure 67 - IEEE Ontology for Robotics for Agent Interaction

RIA stands for the Robotic Industries Association. The RIA TR R15.606-2016 is a specific standard report for robots and robotic devices, primarily focusing on collaborative robots. "Published on December 25, 2016, the RIA TR R15.606-2016, like its ISO predecessor ISO/TS 15066:2016, provides safety guidance for collaborative industrial robot systems where a robot system and people share the same workspace [66]." Effective use of TR 606 assumes that the robot system is compliant with ANSI/RIA R15.06-2012 – industrial robots and robot systems safety requirements. The RIA TR R15.606-2016 standard primarily focuses on how the collaborative industrial robot system must be designed, including risk assessments and hazard identifications. It also talks about the requirements for collaborative robot system applications, including workspace, control system, robot operation, how to transition between collaborative and non-collaborative modes, and collaborative operations. Additionally, it discusses how to establish threshold limit values on the system for the collaborative robot, focusing particularly on power and force limiting applications. It includes verification, validation, and information for use.

Under the International Organization for Standardization (ISO), robots or robotic devices are also covered. ISO standards are constructed in levels. "A-level standards are the highest-level standard. They apply to fundamental safety knowledge, basic design features and general machine aspects. The B-Level standards are more specific to devices that can be found on different types of machines. It is still a general standard, but it goes into specific safety features. C-Level standards are specific safety requirements for a specific kind of machine, a robot for example [67]." ISO 12100 – safety of machinery – defines different basic concepts such as risk assessments and risk reductions for all various types of machinery; this would be classified as an A-level standard. However, the ISO 10218 – robots and robotic devices – are written specifically in terms of robotics, using robotic examples for illustrating safety requirements for industrial robots; this would be classified as a B-level standard. To add, ISO 13482 – personal care robots – allows for close human-robot interaction and contact; this would be classified as a C-level standard. Displayed in Figure 66 below, the standards under the ISO progressively become more and more specialized to a specific kind of machine, which in this case is a robot.

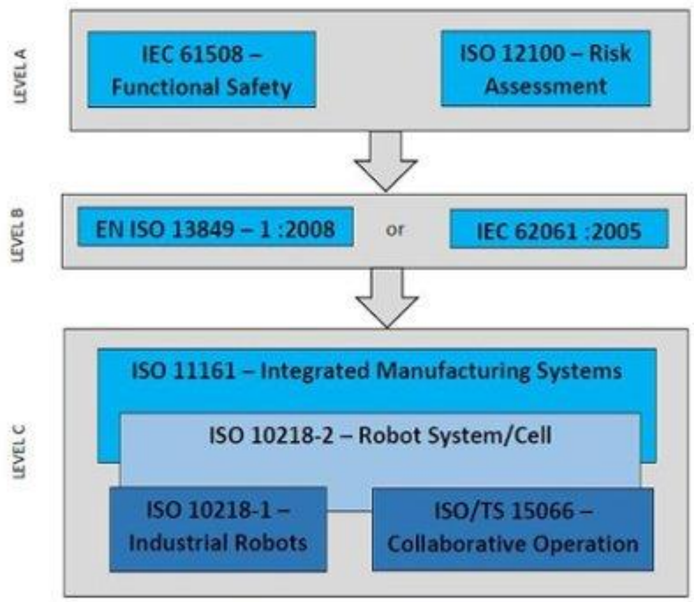


Figure 68 - ISO Standard Levels

Another relevant standard is Electronic Component Standards. "U.S. and international standards for electronic components, capacitors, transducers, surge protectors, LEDs, and resistors find wind application in consumer products, vehicles, medical devices, sensors and controls for industrial use and more [68]." There are nearly 100 standards published under the Electronic Component Standards. The IEC and the IEEE are two standard developing organizations focused on electronics and their design, manufacture, testing, use, safeness, and end-of-life procedures.

Relating to Electronic Component Standards, The Occupational Safety and Health Administration (OSHA) standards primarily focus on the minimization of electrical hazards. OSHA standards focus on the design and the use of electrical equipment and systems. "The standards cover only the exposed or operating elements of an electrical installation such as lighting, equipment, motors, machines, appliances, switches, controls, and enclosures, requiring that they be constructed and installed to minimize workplace electrical dangers [69]." These standards also require that approved testing organizations test and verify any electrical equipment before it is used in the workplace to ensure that it is safe for all workers. This standard is extremely important because electrical shocks and hazards on humans, whether it is direct or indirect contact, can lead to permanent disabilities or death.

Another standard to look at is the Human System Interaction Ergonomics Standards (ISO 9241). These standards for Human System Interaction Ergonomics applies to user interface designers, developers, evaluators, and buyers. These standards also provide design principles and an overall framework for applying those principles to product analysis, design, and evaluation [70]. These standards correspond with human-centered design of software user interfaces and how both software and hardware components of interactive systems can enhance human-system interaction. In addition, it increases usability, issues that are associated with the design of the services, and it augments the equipment people use with a wide range of sensory, physical, and cognitive abilities. Displayed below in Figure 67 is a flowchart that shows ergonomic indicators and parameters, a human-oriented design of collaborative robots.

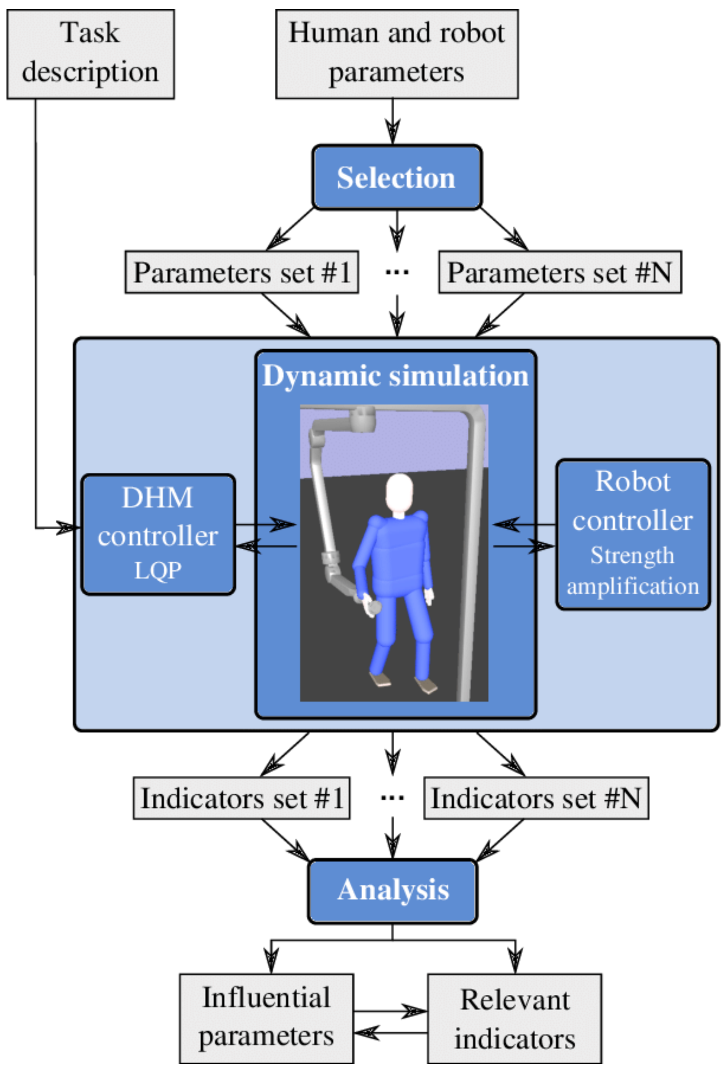


Figure 69 - Ergonomic Indicators for Co-Bot

Software Engineering Standards also play an important role for required standards. "Software, both throughout various industries and as an industry itself, relies on standardization. From the foundation of standardized hardware specifications and interfaces, up through programming languages and interoperability, and using software for the purpose and use case that it was intended for, software development is heavily driven by standardization [71]." Software engineering standards (ISO/IEC TR 19759:2015) approach the process from many different directions, handles documentation, management of the life cycle, assessment, and testing [72]. Standardization is extremely important for software engineering because software engineering is a collaborative effort. Standardization assures that all components are in sync with each other during the process and ensures the quality of the output. Displayed in Figure 69 below are all the software engineering standards categorized into different sections.

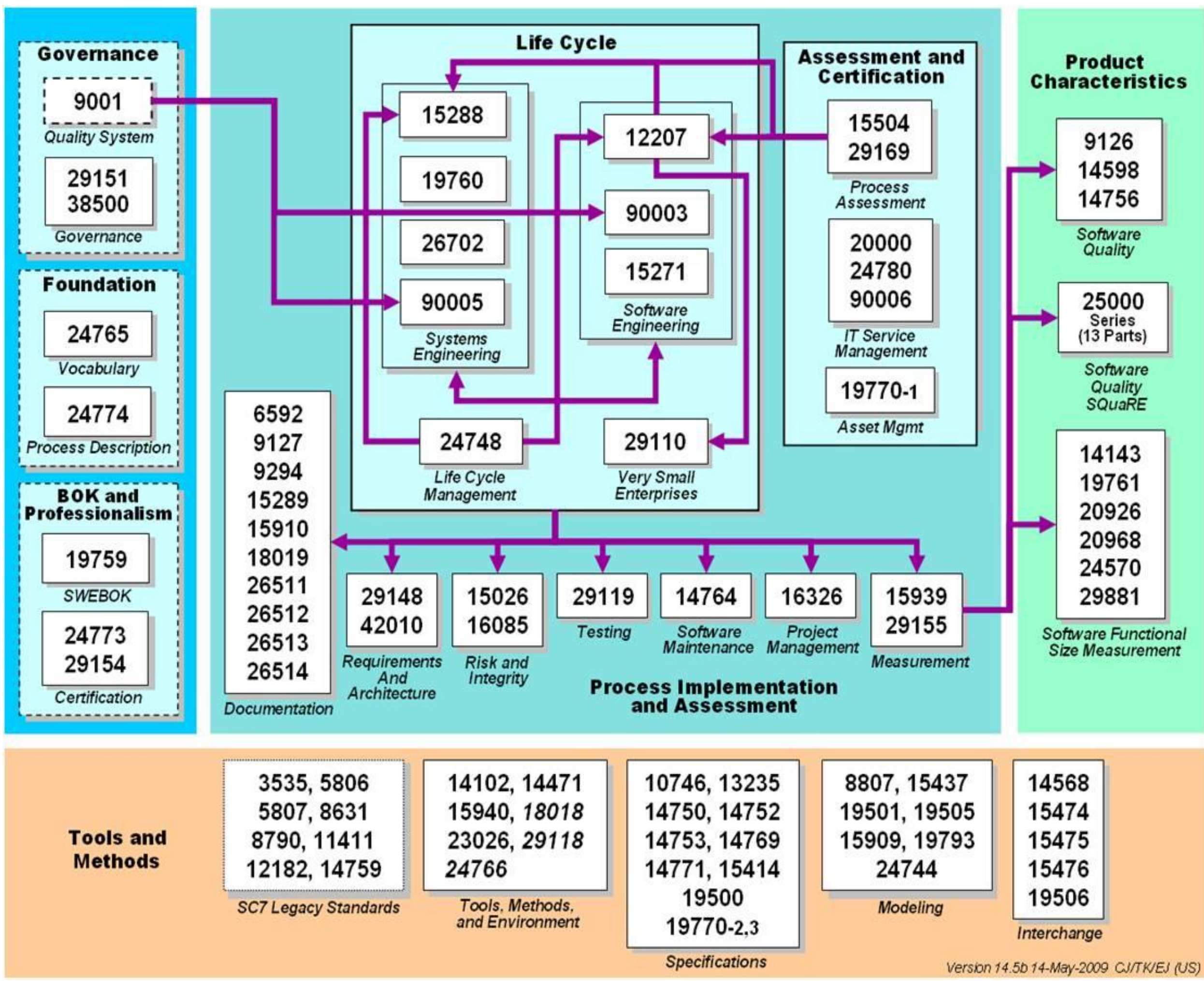


Figure 70 - Software Engineering Standards

4.1.2 Design Impact of Relevant Standards

Design impact is the belief that design can be used to create positive, environmental, and/or economic change. When it comes to standards, standards ensure that engineering designs are created in a way such that the designs create positive, environmental, and/or economic change. The design impact of engineering standards ensures that products meet safety requirements which allows for environmental safety and for humans to be safe in the workplace. It also ensures efficiency, production of standardized products, improve trade, and most importantly, provide safety and quality requirements. Manufacturing to a certain standard implies a certain level of quality to the customer. In many cases, standards provide uniformity, which allows worldwide acceptance and application of a product or material. The goal of design impacts for standards is to facilitate trade, exchange, and technology transfer. Standards help remove technical barriers to trade, leading to new markets and economic growth for the industry [73].

Standards are significantly expanding since the creation of the Internet and the World Trade Organization (WTO). They are increasing impact on society and businesses and they are creating many more stakeholders such as corporate and business leaders. National and international standards affect our industry. Because technology is advancing, and we are moving towards a more global economy, standards are starting to become more and more complex. "Standards are recognized as being essential to helping companies be innovative, reduce costs, improve quality, and maintain competitiveness in an international marketplace [73]." Standards allow processes to run faster, run efficiently, run predictably, and be more cost effective. This is achieved by providing uniformly detailed procedures which help the user produce quality products, allowing users to communicate with one another easily, enhancing product quality and reliability at a reasonable price, improving health, environmental, and safety protection, and allows for a reduction of waste [73]. Standards give huge design impacts, ensuring that all products meet the design requirements as far as reliability, safeness, and quality. Additionally, the design impacts of engineering standards ensure that products developed by other companies are compatible with other products, thus allowing for interchangeability.

For IEEE Standards in the robotics and automation space, “the IEEE strives to ensure that everything involved in the design and development of autonomous and intelligent systems is educated, trained, and empowered to prioritize ethical considerations, so that these technologies are advanced for the benefit of humanity [74].” They shall also promote common measures and definitions in robotics and automation, promote measurability and comparability of robotics and automation technology, and promote integrity, portability, and reusability of robotics and automation technology [74]. When taking into consideration a robotic arm, it can be implemented in many ways.

4.2 Realistic Design Constraints

Design constraints are limitations on a design. These can either be limitations that one can control, or they can be self-imposed limitations to improve a design. There are many different design constraints such as economic and time constraints, environmental constraints, compliance constraints, style constraints, or health and safety constraints. Constraints vary across the board for different projects. For example, a car engine cannot exceed the size the space in which it fits, yet it cannot produce less than the specified power, or if a robotic arm is to be portable, it must be small enough to do the required task while still maintaining portability. Constraints are typically seen as being negative, however, they are conditions that are necessary to happen or would like to have happen with a design.

When it comes to robotics, "a design constraint refers to a limitation on the requirements and/or operation conditions under which a robot is expected to operate. A design constraint can, for example, affect the robot shape, the robot operation features, and the robot functionality. A design constraint can be also related to other aspects such as the manufacturing technology or the available budget for the construction of a robot [75]." For the robotic flange assembly, some of the design constraints would include the reduction of time for the task, size range of flange constraints, motion constraints, price and cost constraints, safety constraints, size constraints, torque constraints, and many more. While there are many constraints to take into consideration, the meeting of these constraints will allow the project to perform to the best of its ability at the highest of quality.

There are many different aspects of a project that must be considered to determine the feasibility of the system; these aspects are constraints. The constraints this project will focus on are economic and time constraints, environmental, social, and political constraints, ethical, health, and safety constraints, and manufacturability and sustainability constraints. It is important that each of these constraints be looked at individually to determine the practicality of attempting to design this robotic flange assembly.

4.2.1 Economic and Time Constraints

Generally, one of the largest limiting factors of any design project is the economic constraint. Economic constraints determine whether a project should be carried out from a financial perspective or not. Primary considerations for economic constraints are the cost of making a product (including fixed and variable costs), the pricing of a product such as if it can compete in the market place, and the cost of ownership for both the producer and consumer.

Economic constraints are a type of external constraint. These constraints involve economic factors that affect a company and are usually out of the company's control. A major economic constraint for this project is the budget. With a given sponsored budget of $1200, finances are going to be tight. One of the reasons why this budget is a concern is because there is a total of sixteen group members for this project, meaning that each person would get $75 evenly to use. Simple robotic arms alone can cost anywhere from $50-$1000. Large industrial robotic arms can cost anywhere from $5000-$25000. Having a budget of $1200 for the entire team places a limit on the versatility, reliability, and sophistication of the desired completed project.

Displayed in Table 21 below shows some of the required general components for the project and their average corresponding prices. These prices will account for having only one robotic arm rather than the ideal two. This table depicts the possible economic constraints of having a $1200 budget.

Table 21 - Component Cost

Robotic Component	Quantity	Average Cost
Basic Robotic Arm	1	$350.00
Controller	1	$100.00
End-Effector (3-finger)	1	$80.00
Sensors	5	$75.00
Servo Motors	2	$30.00
Microcontrollers	1	$30.00
PCB	3	$150.00

As displayed in Table 21 above, the total cost for these general components is $1,015.00 and these are only electrical and computer engineering component requirements. This leaves the sponsors donation amount of $185.00 left. This does not consider the mechanical engineers, the computer science programmers, or the industrial engineers. Evidently, this budget will cause an economic constraint within the robotic flange assembly project.

Time constraints are one of the most overlooked constraints in any design project. Time constraints refer to the limitations on the start and end times of each task in a project's critical path, which is the sequence of tasks that cannot be delayed without delaying the entire project. There are many factors that created time constraints for this project. Some of these factors include the project being only a year long, having to complete a prototype design in time for peer evaluation, having to build and design everything from scratch, having parts ordered and tested on time, and much more. Delays in the critical steps of a time-constrained project are typically unacceptable because they can and will affect the project's completion time. The longer a project is behind schedule, the more resources it will need, thus possibly increasing the cost of the project [76]. To mitigate this constraint as much as possible, a very serious and concrete stance was taken on the scheduling of milestones for the project. Having these milestones allows for any unforeseen events and obstacles to be given the proper amount of time and attention to get handled.

4.2.2 Environmental, Social, and Political Constraints

“Environmental constraints are any limitations on strategy options due to political, external, competition, social requirements and expectations, cultural or economic factors, and technological or legal requirements [77].” Environmental constraints consider the ways that a product can impact the environment, from its manufacturer to its use to its disposal. Environmental constraints can also be defined as the surroundings and conditions that influence the performance of a design. These constraints include variations in temperature and weather. Temperature constraints on the robot will be defined by the components that make up the robot; the robotic flange assembly will have the assumption that primary functions and operations will be done indoors at room temperature. In addition, the device shall not break down when being transported to different locations or to areas that have an opposite climate such as from going cooled down indoors to hot weather outdoors back to indoors.

Features of the environment may also constrain the motions of the hand or object. This is most evident for surfaces which objects rest on, such as tables, counters, and floors. These environmental constraints – when properly used – can aid the grasping functionality of the robotic arm. Since the robotic flange assembly shall be a portable device, these constraints can be preset, such as placing the device on a flat surface to ensure proper grasping functionality.

It is important to understand the basis of human interaction with the environment in order to have the collaborative robotic arm act in a similar manner to what humans do while moving and interacting across their given space. There are many ways the robotic arm interacts with the environment. Some of these interactions include acquiring data from the robot's surroundings through its sensors to provide the necessary input signals to the controller and perform its actions in order to achieve desired tasks. The interaction between the robotic flange assembly and its environment will focus on noncontact tasks. Noncontact tasks allow for unconstrained motion in a free space without any environmental influences on the robot. Some of the noncontact tasks include industrial applications such as pick-and-place, packaging, assembling, or machining. In the case of the robotic flange assembly, some of the noncontact tasks will include holding two flanged pipes together securely and evenly while the bolts are being tightened, tighten two bolts on the flanges simultaneously, and detect and adapt to bolts on the flanges.

"Social constraints are defined as patterns of behavior that provide opportunities for and constraints on implementation of engineering projects. Social constraints can include formal practices such as government regulations or informal norms including cultural preferences [78]." Social constraints are also developing projects that are designed to meet human needs and/or to address social issues. With any design, the impact of social constraints must be considered. Regarding the robotic flange assembly, a positive social impact is desired within Siemens, as they are the sponsors of this proposed project. The scaled prototype design for this project can be used to solve Siemens' current problem with flange assembly and decrease the time it takes to tighten flanged pipes together. Moreover, this project will also give a presentation to all mentioned standards and explain in detail all engineering and design decisions made for the project.

Just as the social constraints of a design are considered, a design's political constraints must also be considered. Social constraints and political constraints are generally like one another; however, they can differ across projects. One needs to understand how engineering and political activities interact, and how to work effectively in this environment. Some key points to examine are how the government acts as a regulator and how the government acts as a customer. The main political constraint of this project is the robotic arms ability to work collaboratively with humans. The characteristics of this robotic flange assembly are desired objectives, needs, and wants for industrial workplaces. Since these characteristics can be expanded on to create large scaled industrial robots, it will be of great political importance. Considering this project is being sponsored by Siemens and created for potential industrial workplace purposes, political impact is augmented. It is also important that all standard regulations be followed in the creation of this device.

4.2.3 Ethical, Safety, and Health Constraints

"Engineering design ethics concerns issues that arise during the design of technological products, processes, systems, and services. This includes issues such as safety, sustainability, user autonomy, and privacy. Ethical concern with respect to technology has often focused on the user phase. Technologies, however, take their shape during the design phase. The engineering design process thus underlies many ethical issues in technology, even when the ethical challenge occurs in operation and use [79]." Engineers need to be made fully aware of any codes of conduct that provide standards of proper behavior in our interactions with others and devices, both inside and outside of the profession. This should not be confused with what we feel is right, what our religious beliefs are, what the law states, or what are the socially accepted norms of behavior.

Ethical constraints for electrical and computer engineers can be identified by using the IEEE code of ethics. If a product, system, or design violates the IEEE code of ethics, it should not be considered an applicable solution to a design problem. It is important that all team members do not violate the IEEE code of ethics. Common ethics violations include safety violations, poor working conditions, and forgery and theft. When it comes to the construction and programming of a robotic arm or any robot, given, some ethics violations include that the robot shall not injure a human and the robot must obey any orders given to it by humans unless it conflicts with the previously mentioned violation. As robots transition into human social environments, a new range of technical, ethical, and legal challenges arise.

For a system or design to be ethically possible it must also be safe. Engineering is about the application of knowledge for the betterment of humanity. Products should be designed such that their everyday use does not cause harm. Rather than health and safety requirements, many industry and governmental regulations and standards typically concentrate on the specification of safety constraints. Safety constraints are defined as another way of specifying safety-related requirements; it is any constraint that specifies a specific safeguard. "Safety constraints typically include things like requiring interlocks and physical barriers around moving parts, safeguards concerning electricity, and the handling of toxic chemicals, and the mandatory placement of warning signs [80]." System safety continues in the workplace by ensuring that components are designed in such a way as to enforce the safety constraints. Accidents occur when a safety constraint is not strictly enforced by the system's components. Safety constraints are extremely important when it comes to engineering designs and products. If a product is not safe, then it's no good.

The robotic flange assembly's main goal is to be safe. Since this robotic arm will be working collaboratively with humans, it is crucial that the arm does not pose any potential threats or injuries to anyone working with the robot. Robot safeguarding is extremely crucial for the robot to have. "The proper selection of an effective robotic safeguarding system should be based upon a hazard analysis of the robot system's use, programming, and maintenance operations. Among the factors to be considered are the tasks a robot will be programmed to perform, start-up and command or programming procedures, environmental conditions, locations and installation requirements, possible human errors, scheduled and unscheduled maintenance, possible robot and system malfunctions, normal mode of operation, and all personnel functions and duties [81]." An effective safeguarding system protects not only the robot operators but also the engineers, programmers, and any other individuals who work on or with the robot that could be exposed to safety and health hazards associated with a robot's operation. Additionally, in correspondence to the robot safeguarding system, the device should have a safety release mechanism in case it was to become attached to any part of the worker.

All robots should meet the minimum design requirements and standards to ensure safe operation by the user. "Every robot should be designed, manufactured, remanufactured, or rebuilt with safe design and manufacturing considerations. Improper design and manufacture can result in hazards to personnel if minimum industry standards are not conformed to on mechanical components, controls, methods of operation, and other required information necessary to insure safe and proper operating procedures. To ensure that robots are designed, manufactured, remanufactured, and rebuilt to ensure safe operation, it is recommended that they comply with Section 4 of the ANSI/RIA R15.06-1992 standard for *Manufacturing, Remanufacture, and Rebuild of Robots* [81]." It is critical that safety and health constraints are mandated throughout all engineering designs. As technology advances, and robots are becoming more and more intelligent, it is imperative that health and safety constraints be placed on these robots as well as all engineering designs.

4.2.4 Manufacturability and Sustainability Constraints

Manufacturability refers to the designing of a product in such a way that it can be manufactured efficiently, reliably, and within acceptable costs. This can include the redesigning of a product to reduce the number of parts it uses, simplify fabrication, or utilize common parts and components. The ability of a system to be produced with as few resources as possible alludes to a system's manufacturability. Design for manufacturability (DFM) describes the process of designing or engineering a product to facilitate the manufacturing process which will reduce its manufacturing costs. DFM allow for potential problems to be fixed in the design phase which is the least expensive place to address them; therefore, manufacturability helps with economic constraints and costs. Some factors that may affect manufacturability include the type of material being used, the form of the material, and secondary processing.

Manufacturability is important for printed circuit boards (PCB). In the PCB design process, DFM leads to a set of design guidelines and steps that try to ensure manufacturability. By doing this, it is likely that production problems may be brought up during the design stage. DFM is constantly evolving. As manufacturing companies evolve and create more and more stages of the processes, the processes tend to become cheaper and more affordable. In the case of the robotic flange assembly, since this design is a machine design, the automatic, repetitive process it does will likely be cheaper than doing so by hand.

Examining the manufacturability of the robotic flange assembly, the robotic arm will possibly be built off already existing technology. It will consist of soldering and wires connecting to sensors. Additionally, customization of the PCB will also be made to make it compatible with our device. With some simple modifications to the chassis, PCB, components, and source code, this will allow the robotic arm that is manufactured to be capable of holding two flanged pipes together securely and evenly while the bolts are being tightened, tighten two bolts on the flanges simultaneously, and detect and adapt to bolts on the flanges.

Sustainability is defined as the ability to be maintained at a certain rate or level. It refers to the ability of an engineering design to perform under normal operating conditions for a given length of time. While choosing the necessary parts for the project as well as the modifications for them, manufacturing lifetime was considered by trying to choose the most generic components possible with multiple providers to ensure a long manufacturing lifetime.

Sustainability is a critical feature. Everyone wants the best products made from the best materials and components. "In the United States, many projects funded with government monies are now required by law to incorporate sustainability principles in their design and function. This is an especially important consideration in the design and construction of large public works such as transportation infrastructure, water and wastewater treatment facilities, and government office buildings [82]." For the robotic flange assembly, the device should be composed of durable and corrosive resistant material; the target material would be steel. Materials like these are extremely robust and ideal for mechanical applications involving strain, stress, and movement. Such materials are easy to obtain, cheap, and sturdy. Additionally, when it comes to motor comparisons, sensor comparisons, software technology comparisons, or other necessary comparisons, it will be determined in a later section what the best materials, components, or programs are necessary to use for this device.

Engineering disciplines are engaged in sustainable design including life cycle analysis, pollution prevention, design for the environment, design for disassembly, and design for recycling. "Training in sustainable practices is valuable for engineering design professionals because it brings these issues to the forefront of design plans and provides engineers and architects with facts that can be used to explain why more expensive options are also be the best options [81]." If this scaled prototype becomes used in an industrial workplace, it is necessary for sustainability constraints to be taken into consideration.

5. Project Software Design Details

The software portion of the robotic flange assembly will be developed in C++ and will utilize ROS, the robotic operating system. The program will be implemented on a microcontroller and will have three main tasks: process input received from sensors that collect data that will define the size and location of the flange, determine if the flanges are aligned based on the input, and produce a pulse width modulation signal to control the pulse going to the motor to define movement.

5.1 Software Functionality

The software will have three broad functions: to determine if the flanges are aligned before beginning assembly, define the path of movement to get to each bolt, and to define the necessary torque/revolutions required to tighten each bolt. To determine if the flanges are aligned, a distance sensor will be used to calculate the size and position of the flange that will then refer to a database that has known values to represent the expected position for a flange according to size. The path to each bolt will refer to the required torque sequence that is discussed in further detail in section 3.1.2, and the actual tightening will be controlled by a pulse width modulation signal that is sent to the motors.

5.1.1 Sensor Input Calculation

One of the assumptions being made for the flange assembly is that the flanges are perfectly aligned. This allows for our product to simply check that this assumption is correct before beginning the process of assembling the flanges. A proximity distance sensor will be used to scan the area for the location of the flange. The current sensor being considered for this task is the LDS-01. The LDS-01 is a 2D laser scanner capable of sensing 360 degrees that collects a set of data around the robot to use for SLAM (Simultaneous Localization and Mapping) [83]. Once the location is determined information will be read from a database that has stored the expected location of flanges in the range of two to eight inches in size. The starting location of the robotic assembler, relative to the flange, will be known. If the flange position does not match any of the expected scenarios, the size of the flange will not be found, and the system will assume that the flanges are not aligned. If the flanges are not aligned the system will sound a warning to alert the user and will shut off. If the location of the flange matches a known set of values, the size of the flange will be returned. The size of the flange will be used to initialize the number of bolts, the toque sequencing pattern that the system must follow, the distance between each bolt according to the order they will be visited (determined by torque sequencing pattern), and the value of torque that each bolt must reach. Once this initialization process is complete the assembly process will begin. The output of the sensor input is a Boolean value. The function returns true if variables have been initialized and returns false if the variables have not been initialized.

5.1.2 Path Execution

The path will begin execution if and only if the flanges have been determined aligned by the previous sensor input calculation function. Our software design will be prepared to execute a path that consists of 4 to 10 bolts. The bolts on a flange are equally spaced allowing us to use algebra to determine the distance between each bolt along the circumference of the flange. Each bolt will be visited a total of three times. The first time each bolt is visited it will be tightened to an initial torque value. The second time each bolt is visited it will be tightened to the intermediate torque value. The final pass will be to bring each bolt to the full torque value. The iterative process is the same process that a human would follow to tighten each bolt in such a way as to avoid misalignment. This function will hold a matrix of key value pairings that will keep track of each bolt (key) and the number of times it has been visited (value). When the bolts have been visited three times each, the process will be finished.

5.1.3 Torque Execution

The required torque will be defined by the characteristics of the flange. These characteristics include the size (flange diameter), number of bolts, and class of the flange. As mentioned in section 4.1.2 above, each bolt will be tightened to a specified torque value at each of three passes. During the first pass each bolt will be tightened to a maximum of 30% of the total torque value. During the second pass each bolt will be tightened to a maximum of 60% of the total torque value. During the final pass the bolts will be tightened to 100% of the torque value. There are many factors to consider when aiming to achieve a specific torque. These factors include the power applied, the class of material, the speed of rotation (RPM), and the length of time spent to achieve the torque. The software design should be flexible enough to produce torque in more than one way. The program will also recognize invalid parameters by referring to a defined set of acceptable input values for various scenarios. Below in Table 22, we explore how to obtain the necessary torque by manipulating different parameters. This allows us to see how much flexibility we have and to determine which factors carry the most weight. We will assume specifications for each 4-inch flange as described in Table 12**.**

It is important to note here that the information in Table 22 is meant to portray the structure for defining an object within the software program. The class for a flange will be abstracted enough to serve as a template for any flange. Also note that the power required to generate the baseline values are not realistic for a portable device. Decreasing the power by a factor of 10 gets the power closer to a reasonable wattage but it is still a concern. To produce the required torque for a manageable power requirement the angular speed must also drastically reduce. The values in the table are purely theoretical and in many cases are not efficient. Another, more probable approach is to scale down the torque enough to satisfy realistic values for both power and RPM values. Defining values that do not seem reasonable, even though they are expected to produce the required torque, may be helpful to define warning modes.

Table 22 - Achieving Required Torque

Torque	Class	RPM (angular speed)	Power (Watts)	Time (seconds)
***Baseline Values**				
137	300	3450	67,000	60
218	600	2904	90,000	60
474	900	1780	120,000	60
Effect of Decreasing Power Only				
137	300	345	6,700	60
218	600	290.4	9,000	60
474	900	178.0	12,000	60
Effect of Decreasing Power and Decreasing RPM				
137	300	51	2,000	30
218	600	51	3,190	30
474	900	22	3,000	30

*** Calculations based on an online Engineering Tool Box** [84].
Here is an example of values our software program can refer to.

5.1.4 Database

The database system will store information for each type and size of flange that will be used by the robotic flange assembler. The information stored will be used to identify what the specified torque must be, how many bolts will be tightened, the distance from one bolt to the next, and the torque sequence pattern. During the first step of the program execution, which is the alignment verification process, the system will gather information about the flange that is in place. The piece of information used to identify all components will be the size of the flange. Although information will exist in the database for several classes, an assumption will be made about the class of the flange for our prototype design. Including information for classes that we do not plan to assemble allow for future growth.
Table 23 is a sample of the type of data that will be fetched when a size for some flange is detected. In the most optimal case a sensor or a set of sensors will detect the flange size. In the worst-case that we aim to achieve, the user will input the size of flange and the relevant date will be fetched. In this worse case a sensor, like an ultrasonic sensor, will at least detect the location of the flange.

Table 23 - Data Storage

8.5 Inch Flange of Class 150	
Number of Bolts	8
Bolt Ordering for Torque Sequence	1,5,3,7,2,6,4,8

Distance Between Bolts (inches)		2.748	
First Pass Torque (ft-lbs)		60	
Second Pass Torque (ft-lbs)		120	
Final Pass Torque (ft-lbs)		200	
Bolt	**X – Location in inches**	**Y – Location in inches**	**Angle in Degrees**
1	3.500	0.000	0.000
2	2.475	2.475	45.000
3	-0.000	3.500	90.000
4	-2.475	2.475	135.000
5	-3.500	-0.000	180.000
6	-2.475	-2.475	225.000
7	0.000	-3.500	270.000
8	2.475	-2.475	315.000

In
Table 23 we assume that the class of the flange is 150. The size of the flange is identified by sensor input and once a flange size is retrieved from the database, the remaining values can be initialized and used to define critical paths in execution. If there does not exist a set of (x, y) locations that match the newly gathered data, it means that the flanges are not aligned. If the flanges are not aligned the system will alert the user and power down, requiring human interaction to ensure flanges are aligned before beginning.

5.1.5 Program Execution Flow Chart

Below in **Figure 71** is an overview of how the program will operate.

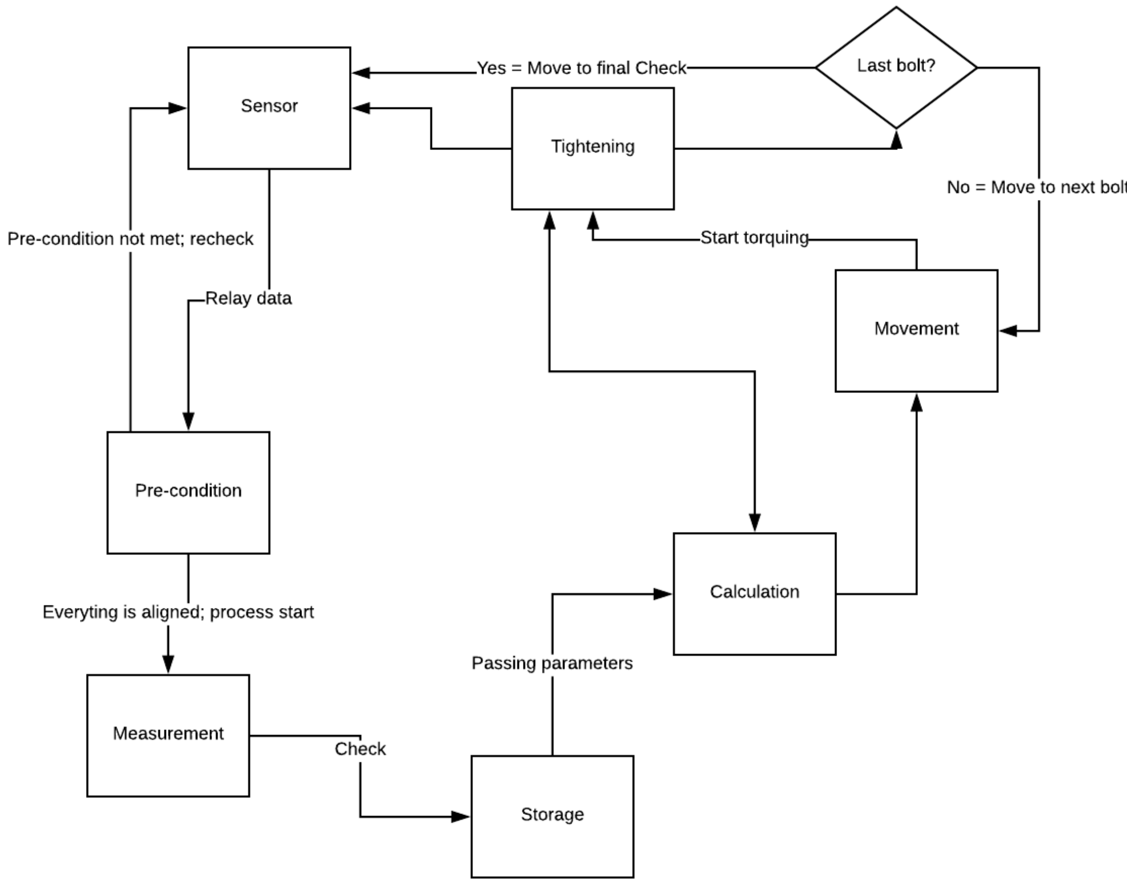


Figure 71 – Program Execution Flow Chart

5.2 Simulation Software

This section provides a comparison of different software tools that can be used to simulate our robot. We've chosen to do a comparison between three open-source simulation packages: Gazebo, ARGoS, and V-REP. Most work on the simulation software will be done by the Computer Science team, and as such, most information on these tools has been gathered from them and their resources. However, it is expected that the Computer Engineering team members will be working alongside the CS team to optimize our design with the results of the simulations.

5.2.1 Gazebo

Gazebo is a 3-D dynamic robot simulator that provides a fair number of features while also having a relatively simple interface. It is available for MacOS, Linux and Windows, but only has a binary package for Linux Debian [85]. This makes it difficult to install and use on MacOS and Windows. It is installed via the command line using third-party package managers on other systems [87]. Only the ODE physics engine is available by default, but it's possible to build Gazebo from source with a different physics engine. Gazebo has a code and scene editor, but lacks mesh manipulation, making modeling more difficult. Objects in the simulation environment can be moved and added, but the world won't reset to its original state after the simulation is completed or reset. Gazebo can also output simulation log files, text files, and video frames as pictures. Gazebo does not offer particle systems [86].

Gazebo possesses a less diverse library of default robots than V-REP, that mostly includes wheeled and flying robots, but more than that of ARGoS. Third-party robot models are available to be uploaded, but their documentation is often not enough for good simulation in the tool. The default models in the Gazebo library are simple and are therefore more appropriate for computationally complex simulations. Model meshes for Gazebo are imported as single objects, and models that contain multiple sub-components have to be assembled in Gazebo from multiple DAE files, each corresponding to one sub-component. These imported meshes cannot be changed and therefore have to be optimized in another third-party 3D modeling software. This leads to increased difficulty in model development [86].

Both Gazebo and ARGoS save their scenes as XML files, which allows for the use of bash script to change and run simulations. Functionality of robots is done by either compiled C++ plugins or as ROS programs, but the lack of scripts makes it difficult to run quick tests. If third party robot models are used, difficulties in recognizing what is occurring in the simulation and what plug-ins are being used can often arise. One major issue with Gazebo programming is that many plug-ins provided with the default robot models can malfunction or are faulty. One positive note about Gazebo is that it contains comprehensive documentation, step-by-step tutorials, and a large user community to aid in user projects [86]. Of all three simulation packages, Gazebo has the worst user interface usability. It is notorious for having the interface freeze when editing the model or running the simulation and often requires a reboot of the program. As well, tool bars are difficult to locate, multiple objects can't be copied and pasted, and a scene can't be overwritten once changes have been made. As well, there are issues with connecting to the library and finding specific models in the library [86].

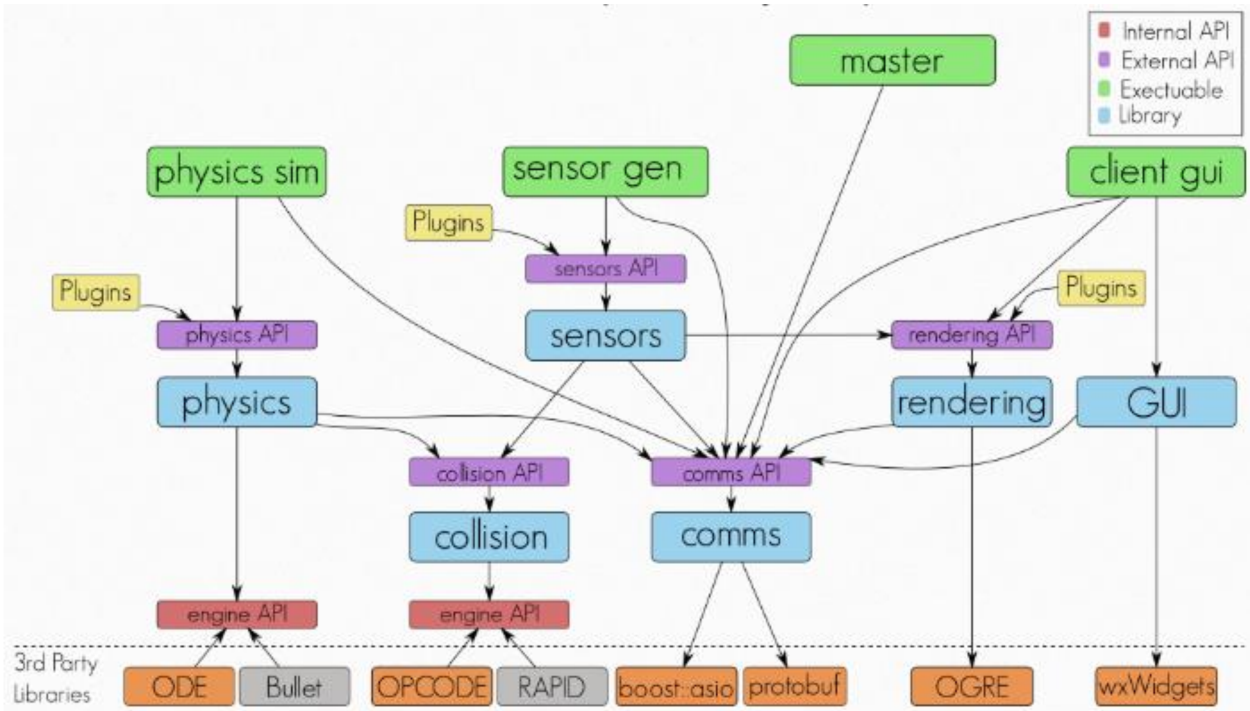


Figure 72: Simplified Gazebo Framework

5.2.2 V-REP

V-REP is a robot simulation tool with an integrated development environment based on a distributed control architecture. In the development environment each object or model can be individually controlled via an embedded script, a plugin, a ROS or BlueZero node, a remote API client, or a custom solution [84]. It's available for MacOS, Linux and Windows and has binary packages available for all platforms. V-REP is available under a commercial license or for free for educational institutions [86].

V-REP comes with several default physics engines, including Bullet 2.78, Bullet 2.83, ODE, Vortex and Newton. It also includes a code and scene editor. Unlike the other two simulation packages, V-REP has meshes that can be manipulated by robots in real time. Scene objects can be fully interacted with in V-REP by the user during simulation. The world returns to its original state when the simulation is reset. Outputs include video, custom data plots and text files. Unlike the other two simulators, V-REP also includes particle systems, creating a more complex and realistic modeling environment [86].

The V-REP library provides a large variety of robots, including bi-pedal, hexapod, wheeled, flying and snake-like robots. It also provides many robot actuators and sensors. This aspect far exceeds the default libraries of the other two simulators. The default models in V-REP are very detailed and therefore appropriate for high-precision simulations. It is possible to simplify the models in V-REP as well. Meshes in V-REP are imported as collections of sub-components, making it possible to manipulate individual parts of an imported model and to change their textures, materials and other properties. Because of this manner of creating meshes, it is possible to simplify, split and combine meshes. This makes it possible to optimize the triangle count of imported models and to manipulate meshes with robot actuators [86].

V-REP saves its scene in a special V-REP format, which forces all scene editing to be done using V-REP interface. For program functionality, there are many options including scripts attached to robots, plug-ins, ROS nodes or separate programs that connect to V-REP via the RemoteAPI. These scripts can be included in robot models and are often used to describe the models and their capabilities. Custom interfaces can be created using V-REP's CustomUI API that is based on QT. Also, Custom UI controllers can be placed on individual robots in the simulation so that individual devices and functions can be interfaced. All scripts and plug-ins provided with the default robot models are simple to implement, good API documentation is provided, and a large library of tutorials and code examples and a large user community are available [86]. V-REP is known for being user friendly. It has few issues with freezing, intuitive functionality, and an organized model library distributed within V-REP. The advantages and disadvantages are listed in **Table 24**.

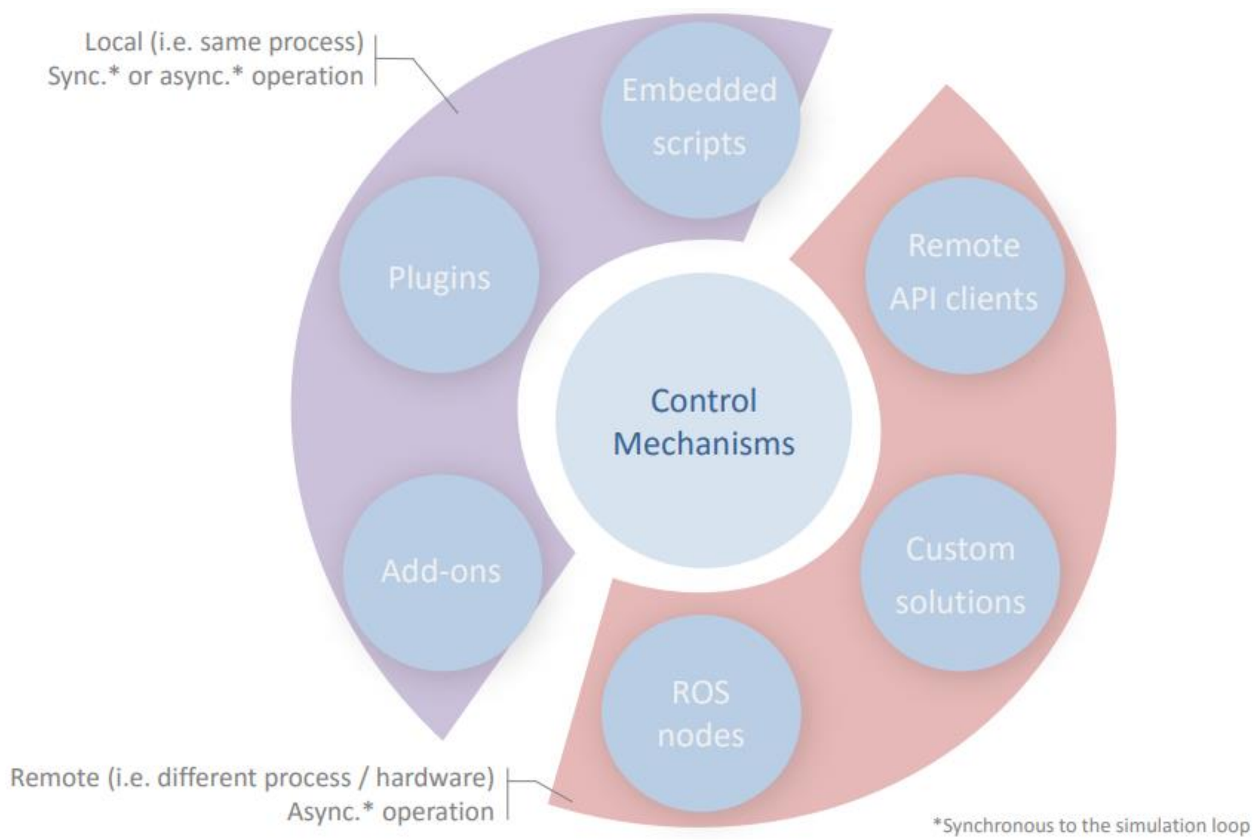


Figure 73: V-REP Interfaces

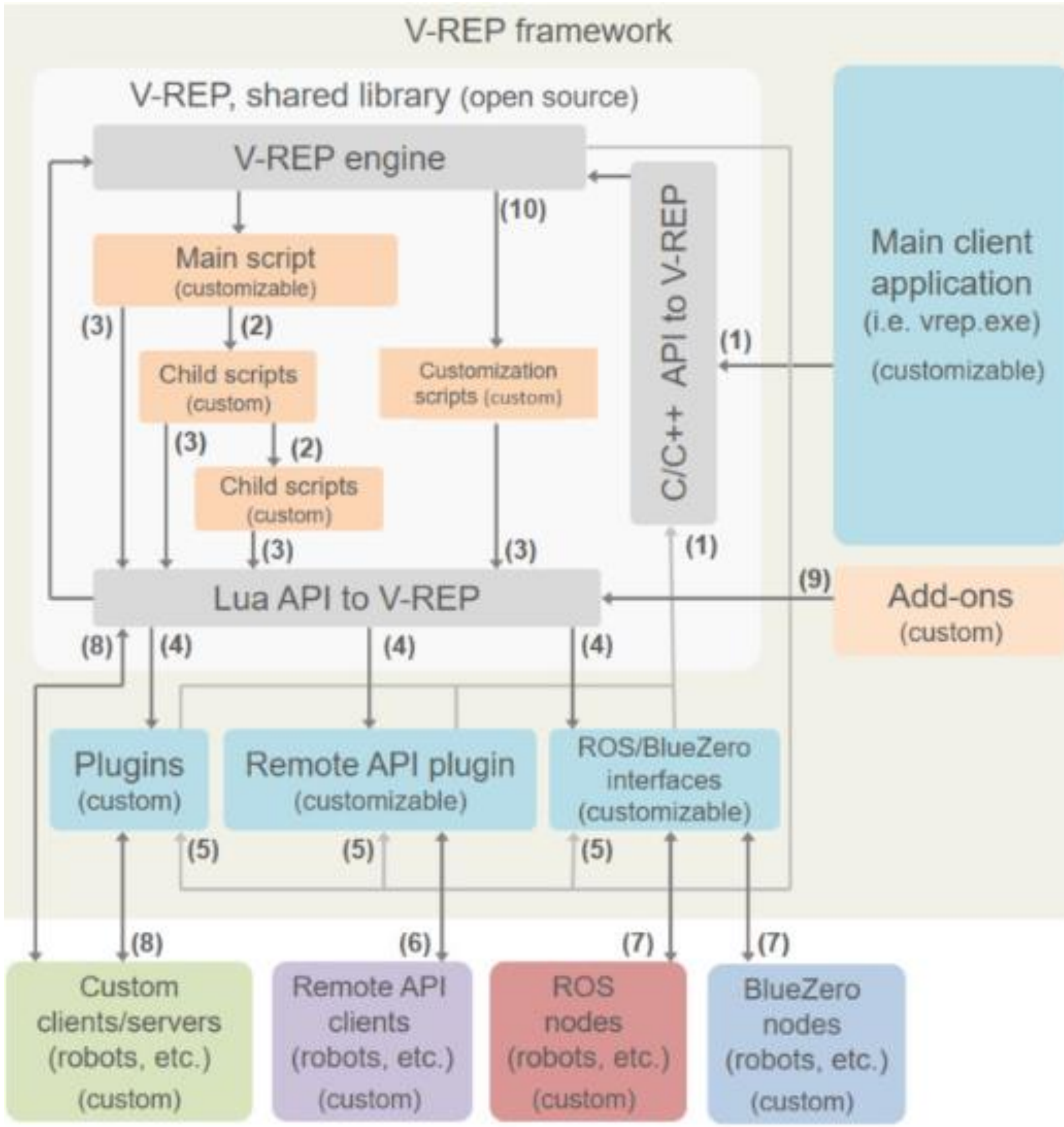


Figure 74: V-REP Framework

Table 24 - Advantages and Disadvantages of V-REP Coding Methods

	Embedded script	Add-on	Plugin	Remote API client	ROS node	BlueZero node
Control entity is external (i.e. can be located on a robot, different machine, etc.)	No	No	No	Yes	Yes	Yes
Difficulty to implement	Easiest	Easiest	Relatively easy	Easy	Relatively easy	Relatively easy
Supported programming language	Lua	Lua	C/C++	C/C++, Python, Java, Matlab, Octave, Lua	Any [1]	C++
Simulator functionality access (available API functions)	500+ functions, extendable	500+ functions, extendable	500+ functions	>100 functions + indirectly all embedded script functions	Indirectly all embedded script functions	Indirectly all embedded script functions
The control entity can control the simulation and simulation objects (models, robots, etc.)	Yes	Yes	Yes	Yes	Yes	Yes
The control entity can start, stop, pause and step a simulation	Start, stop, pause	Start, stop, pause	Start, stop, pause, step	Start, stop, pause, step	Start, stop, pause, step	Start, stop, pause, step
The control entity can customize the simulator	Yes	Yes	Yes	No	No	No
Code execution speed	Relativ. slow [2] (fast with JiT compiler)	Relativ. slow [2] (fast with JiT compiler)	Fast	Depends on programming language	Depends on programming language	Fast
Communication lag	None	None	None	Yes, reduced [3]	Yes, reduced	Yes, reduced
Control entity is fully contained in a scene or model, and is highly portable	Yes	No	No	No	No	No
API mechanism	Regular API	Regular API	Regular API	Remote API	ROS	BlueZero
API can be extended	Yes, with custom Lua functions	Yes, with custom Lua functions	Yes, V-REP is open source	Yes, Remote API is open source	Yes, via embedded scripts	Yes, via embedded scripts
Control entity relies on	V-REP	V-REP	V-REP	Sockets + Remote API plugin	Sockets + ROS framework	Sockets + BlueZero framework
Synchronous operation [4]	Yes, inherent. No delays	Yes, inherent. No delays	Yes, inherent. No delays	Yes. Slower due to comm. Lag	Yes. Slower due to comm. Lag	Yes. Slower due to comm. Lag
Asynchronous operation [4]	Yes, via threaded scripts	No	No (threads available, but API access forbidden)	Yes, default operation mode	Yes, default operation mode	Yes, default operation mode

[1)] Depends on ROS binding

[2)] The execution of API functions is however very fast. Additionally, there is an optional JiT (Just in Time) compiler option that can be activated

[3)] Lag reduced via streaming and data partitioning modes

[4)] *Synchronous* in the sense that each simulation pass runs synchronously with the control entity, i.e. simulation step by step

5.2.3 ARGoS

ARGoS is a multi-physics robot simulator used mostly to simulate large-scale swarms of robots of any kind efficiently [83]. It's available for MacOS and Linux with binary packages available for Linux. On MacOS, ARGoS is installed via the command line using a third-party package manager [86].

ARGoS has a 2D and a 3D custom-built physics engine with very limited capabilities which are available by default. It also has a Lua script editor but no scene editor, no particle systems, and no mesh manipulation. On the other hand, scene objects can be moved by the user during simulation. Outputs include video frames as pictures and text files [86].

ARGoS has a small library of robots, only including the e-puck, eye-bot, Kilobot, marXbot, and Spiri robots. These defaults are simple and are therefore more appropriate for computationally complex simulations. Mesh importing is not available in ARGoS and object representations are coded using OpenGL. Because of these limited features, ARGoS models tend to be very simplistic in representation [86].

In ARGoS, a scene is saved as an XML file. This makes is possible to create a bash script that changes the scene and then runs a simulation. Programming occurs Robots either through Lua scripts or in C++. Some documentation of the robots is provided in ARGoS, but most of how a robot works needs to be deducted from code examples, which is easy enough because the robots are so simplistic. Custom interfaces can be created in C++ by subclassing an ARGoS API class, and these interfaces can be attached to the whole scene or to invididual robots. While there's a good amount of documentation on ARGoS, a small user community is available for information outside of the tool [86].

Because of the simplicity in ARGoS modeling, there are few issues with freezing or slow simulations. As well, the interface is user friendly and the robot models are distributed within ARGoS and it is thus always available regardless of Internet connectivity [86].

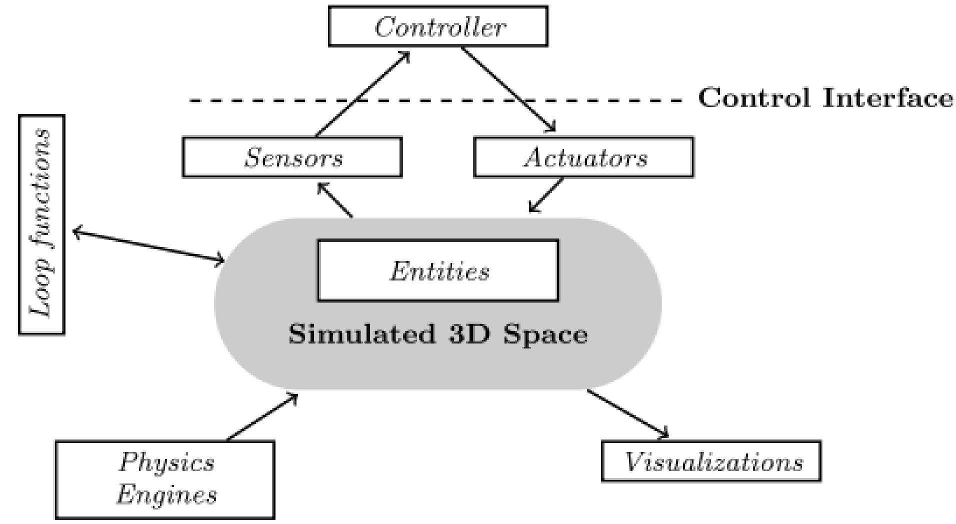


Figure 75: Common Architecture of ARGoS Robot Swarm

```
Initialize
Visualize the simulated 3D space
while experiment is not finished do
   for all robots do
      Update sensor readings          }
      Execute control step            } sense+control
   end for                            }
   for all robots do                  }
      Update robot status             } act
   end for                            }
   for all physics engines do         }
      Update physics                  } physics
   end for                            }
   Visualize the simulated 3D space
end while
Cleanup
```

Figure 76: Pseudo-Code for ARGoS Simulation

Table 25 - Simulator Comparison

	V-REP	Gazebo	ARGoS
Build-in Capabilities	• Bullet 2.78, Bullet 2.83, ODE, Vortex and Newton physics engines • code and a scene editor • Meshes can be manipulated (e.g., cut) by robots in real time • Scene objects can be fully interacted with (e.g., moved or added) by the user during simulation • No particle systems are available	• Only the ODE physics engine is available by default. Possible to build Gazebo from source with a different physics engine. • code and a scene editor • No mesh manipulation • No particle systems are available	• 2D and a 3D custom-built physics engines with very limited capabilities • Lua script editor but no scene editor • No mesh manipulation • No particle systems are available
Robots and models	• Provides a large variety of robots, robot actuators, and sensors. • Very detailed default models • Meshes are imported as collections of sub-components making it possible to manipulate individual parts of an imported model • It is possible to simplify, split and combine meshes	• A less diverse library of default robots • Default models are fairly simple and therefore more appropriate for computationally complex simulations • Meshes are imported as single objects • Imported meshes cannot be changed	• A fairly small library of robots • Default models are fairly simple and therefore more appropriate for computationally complex simulations • Mesh importing is not available
Programming methods	• A scene is saved in a special V-REP format.	• A scene is saved as an XML file	• A scene is saved as an XML file

	• All scene editing has to be done using the V-REP interface • Scripts can be included in robot models • "CustomUI" API, based on QT, is used to create custom interfaces • All scripts and plug-ins provided with the default robot models • Good API documentation, a large library of tutorials and code examples and a large user community are available	• Functionality can be programmed either as compiled C++ plug-ins or via ROS programs • Custom interfaces can be created as plug-ins by using the default QT API • The interfaces can only be attached to the whole scene and not to individual robots. • A fairly comprehensive documentation, step-by-step tutorials and a large user community are available	• Robots can be programmed either through Lua scripts or in C++ • limited documentation on robots • Custom interfaces can be created in C++ by subclassing an ARGoS API class • examples are provided on the ARGoS website • Good documentation, small user community
User Interface	• No freezing issues • functionality is fairly intuitive • Model library is distributed within V-REP	• Freezing issues • UI usability is relatively low • The model library is not distributed within Gazebo	• No freezing issues • functionality is fairly intuitive • The robot models are distributed within ARGoS

Table 25 above is meant to quickly display the areas that are commonly of interest when deciding what simulation software fits your needs as a user or developer.

5.2.4 Simulator Selection

The team has chosen V-REP for its robot simulation tool. V-REP’s ease of installation and superior built-in capabilities allows for use of many physics engines, easy mesh manipulation, superior output options, and inclusion of particle systems. As well, V-REP allows users to add scripts to robot models to describe and enhance their capabilities. V-REP provides the most diverse library of robots, actuators and sensors of the three tools.

6. Final Design

This section represents the final status of our Robotic Flange Assembly Project. We will first give an overview of the design. Then we will discuss the final hardware components that have been selected. We will then explain in greater detail how these components will be integrated into the overall design. We have included a schematic design and a PCB design that is derived from the schematic design. The PCB design is expected to be the first PCB ordered for senior design II. Final component selection in this section takes precedence over any parts claimed as a final selection that is mentioned before chapter 6.

6.1 Official Prototype Overview

Being an interdisciplinary group has brought on many advantages. Since we have the pleasure of working with several fields of study, we have been able to come up with several designs that have been vastly different from each other. From the beginning The Mechanical Engineers have been tasked with the most substantial aspect of the final design. The development of the final concept for the design is documented in detail from the Mechanical Engineering standpoint and is not part of this report. The prototype that showed the most promising outcomes to better facilitate users in the workforce is shown on Figure 77.

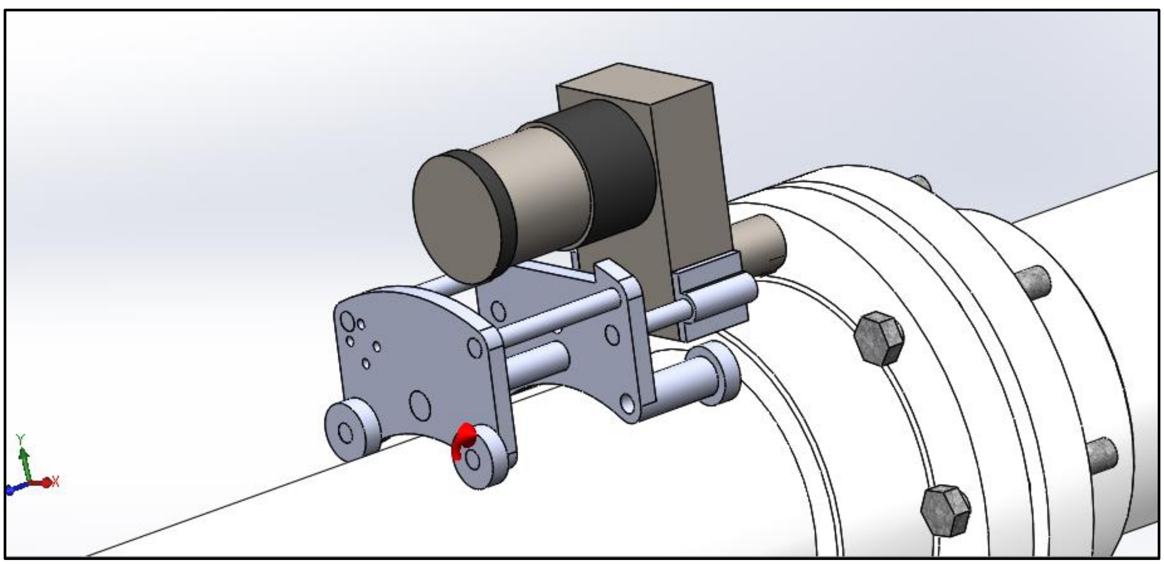


Figure 77 - SolidWorks Prototype

The figure shown above is what the SolidWorks' representation made in collaboration with our Mechanical Engineer colleagues. However, since we remain in the early faces of the designing portion for our project this prototype can be altered in next semester's Senior Design course continuation.

6.1.1 Final Design Details

Not featured in the SolidWorks prototype design is how the mechanism is expected to fasten to the pipe. There has been mention of a carriage in previous sections of this paper, but the details have been hidden. The materials relevant to the carriage component are discussed in more detail by the mechanical engineers and are not included in this paper. However, here we will give a description of the general idea and a bit of the motivation behind it. Early on, it was realized that we will need a carriage that will traverse across the surface of the pipe to be able to access each nut and this carriage shall carry the system. To do this, the team envisioned a system in which a belt is placed along the circumference of the pipe. There will be a small amount of the belt being pulled on a shaft within the carriage. This closed system will be able to hold the carriage so that it is normal to the pipe surface, through the tension forces of the belt, as demonstrated in Figure 78:

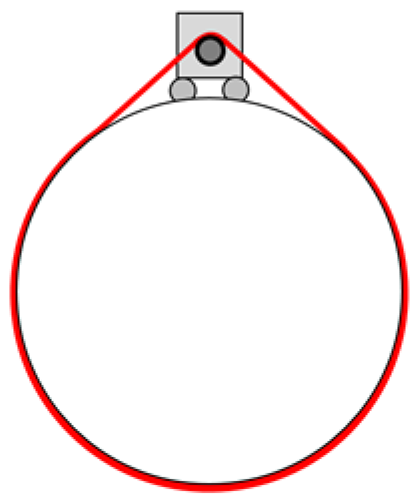


Figure 78 - The Belt Fastener

6.2 Hardware Component Overview

As we undergo on our project's wants and needs, we have made multiple hardware changes from what we have researched and discussed in previous sections throughout this report. Due to sudden design changes we are now considering the Arduino Mega 2560 controller. This change was due to the promising complexity of the project, the requirement for more I/O ports, and storage space. This Arduino features an operating voltage of 5 Volts, a limit input voltage that ranges from 7 to 20 Volts, 54 digital I/O pins, 256KB of flash memory and a clock speed of 16MHz. Besides the main head unit, we will be adding multiple inputs. Those inputs included will be rotary sensors, a wireless Bluetooth sensor, an ultrasonic distance sensor, and a pushbutton. The outputs will be the three motors and an OLED display.

6.2.1 Power Supply

The predominant load for our robot's power supply comes from the large E-30 motor used to tighten the flange bolts. The motor has a peak voltage usage of 24 Volts at 5600 RPM. However, the motor can also operate at a lower voltage when using a lower RPM. For example, at 12 Volts, the motor will operate at 2800 RPM [91]. This allows the power quantity to be raised or lowered depending on the specifications given by the mechanical engineers. For the sake of our initial design we will be using 12 Volts for our larger motors in order to get a baseline functioning model of the prototype. This 12 Volt supply can be created by putting batteries in series that total 12 Volts. By doing this, it would allow us to buy cheaper batteries and make our power supply less heavy and bulky. For our design we will use 8 AA batteries in series to create our power supply. Each AA battery supplies 1.5 V [92].

Figure 79: 12 Volt power supply using 8 AA batteries

6.2.2 Arduino Mega 2560

We've selected the Arduino Mega 2560 as our major control device. This model is a microcontroller board based on the ATmega2560. It's better suited for more complex projects such as robotics because of all the I/O capability and space on the board. The Mega 2560 board has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), and a 16 MHz crystal oscillator [93]. As well, it has an optimum input voltage of 7-12 Volts and can be powered by USB connection or battery connected power jack.

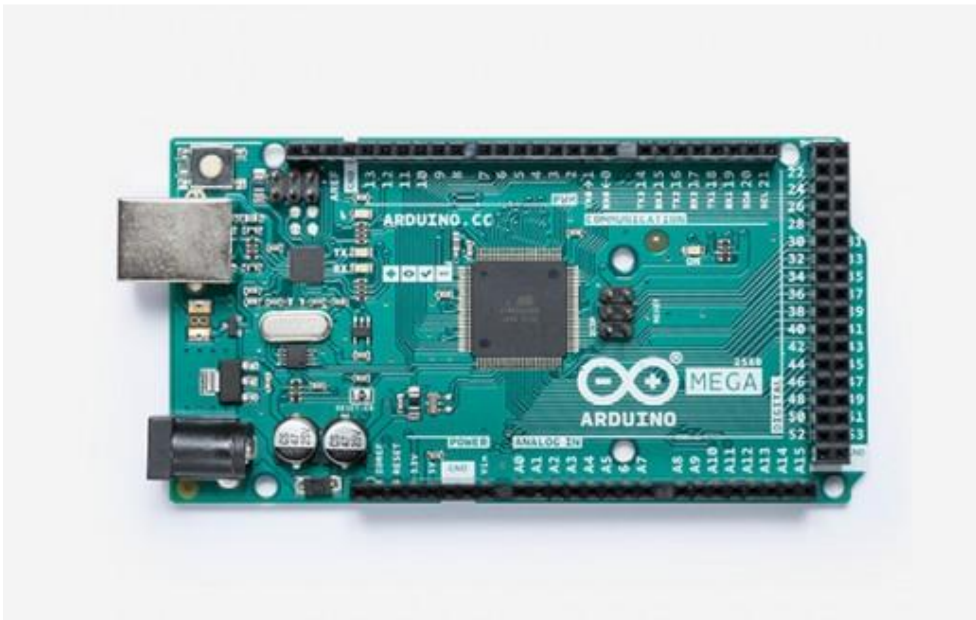


Figure 80: Arduino Mega 2560

6.2.3 AmpFlow E-30 motor

For our main torqueing motor, we've chosen the AmpFlow E30-150. This motor would supply the necessary torque at a modest price that fits our budget. The motor has a peak voltage usage of 24 Volts at 5600 RPM. However, the motor can also operate at a lower RPM when using a lower Voltage. For example, at 12 Volts, the motor will operate at 2800 RPM [91]. This allows for variable torque while using a smaller battery which would lower the overall weight of our design. This model of motor is also lighter than all the other DC motors provided by AmpFlow. The mechanical engineers have justified the selection of this motor in greater detail in technical memos that are required as part of their unique milestones. As a team we have agreed that our main role as the ECE team is to solve the problem of how to power and how to control the motor as selected by the mechanical engineer team. Although we have provided research of our own for motor selection, it was realized that the topic is deeply mechanical and have accepted the final motor selection as presented here. The documentation to reference the mechanical engineer's justification is a current work in progress but available upon request. The AmpFlow motor is pictured in Figure 81.

Figure 81: AmpFlow E30-150

6.2.4 OLED Display

We will be using an OLED Display Module that can be connected to our Arduino in order to display the status of the board as a torqueing program is being ran. The display only requires 4 pin connections to the Arduino and would provide a useful diagnostic tool while the program is running, all while being relatively cheap [94]. We will be using the Geekcreit 0.96-inch 4 pin white I2C OLED module as our display. This model is basic and would fit well into the budget, while being simple to connect. After proper connection a message can be displayed to the user like, for example in Figure 82, which will verify any settings the user may set. As a bonus, this will be a helpful debugging tool during the development of our design! The pin connections for our board to the display is detailed by Table 26:

Table 26 - Arduino to OLED PIN connections

OLED	Arduino
GND	GND
SCL	2560 pin 21
VCC	5V
SDA	2560 pin 20

Figure 82: Geekcreit 0.96-inch 4 pin white I2C OLED

6.2.5 Ultrasonic Distance Sensor

For our distance sensor we chose the Parallax PING Ultrasonic Sensor as our distance sensor. This sensor is lightweight, easily mountable, and relatively low cost. The sensor uses sonar to send an ultrasonic pulse from the unit and distance-to-target is determined by measuring the time required for the echo return. Output from the Ping sensor is a variable-width pulse that corresponds to the distance to the target [95]. It provides precise, non-contact distance measurements within a 2 cm to 3 m range. It also has a very low power consumption, operating at 5 volts and 20 mA. The three-pin header also makes the sensor good for connecting to the Arduino and breadboard testing. The ultrasonic sensor, pictured in Figure 83, will use predefined measures of expected distance and use this information for decision-making. The ultrasonic sensor will be helpful in detecting certain failures that will mainly be considered as unlikely edge cases. For example, if distance is detected to be out of range, an event that should not occur based on the given design, it could mean that the carriage has come off its track, or that there is some failure in alignment. In those cases, it would be imperative that the user is alerted immediately. In addition to its usefulness, the replacement of this component will low cost. The position of the ultrasonic sensor is expected to be in alignment of where the bolt is positioned. The idea will be to detect the slight rise from the head of the bolt. The difference in distance this creates is expected to be within the threshold of the sensor’s sensitivity. In the case that the sensor fails to work for its use case, it will be repurposed to simply check that the carriage is not disengaged from the torque tightening system.

Figure 83: Parallax PING sensor

6.2.6 Rotary Encoder

We selected the WGCD 8 Pc KY-040 360 Degree Rotary Encoder Module, pictured in Figure 84, as our sensor in order to monitor the data given back from our motors. This will give us torque monitoring by sensing the velocity of the motor and giving feedback to the user. The knob on the module allows for rotor control so the rotary encoder can be counted in the positive direction and the reverse direction during rotation of the output pulse frequency [96]. The sensor is also light weight, reasonably priced, and easily connectable to our Arduino.

Figure 84: WGCD 8 Pc KY-040 360 Degree Rotary Encoder Module

The rotary encoder module will refer to predefined values for torque when monitoring data and giving feedback about the data. For example, one of the required guidelines in applying torque is that only 30% shall be applied to each bolt in a specified pattern, and then with a second iteration of the same pattern apply 60% and then finally on the final iteration of the specified pattern apply the remaining torque which will result in 100% of the defined torque. The monitoring of data becomes extremely useful here and makes programming this behavior a task that will be well defined and easy to debug.

6.2.7 Push Button Switch

A push button encoder is being used to engage the components of our prototype. This button uses three wires to connect to the board. The first goes from one leg of the pushbutton through a pull-up resistor to the 5-volt supply. The second goes from the corresponding leg of the pushbutton to ground. The third connects to a digital i/o pin which reads the button's state [97].

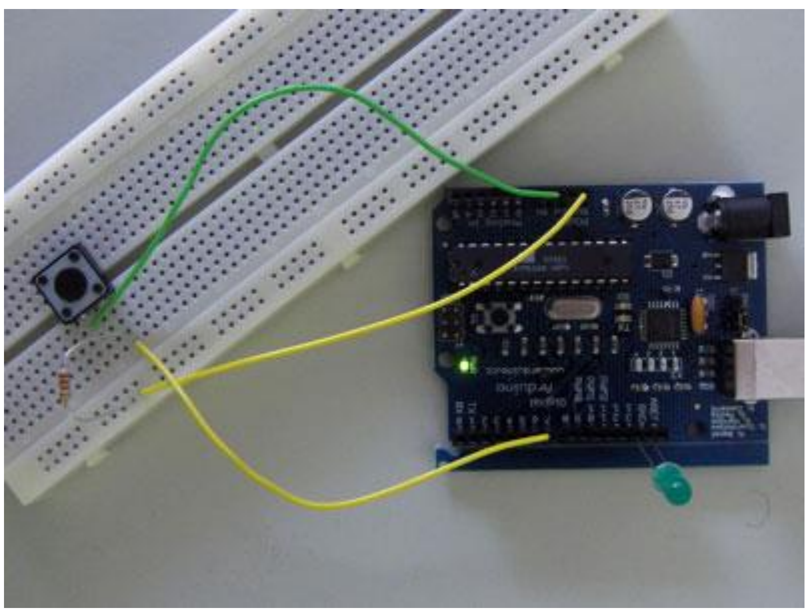


Figure 85: Push Button Connected to Arduino

Above, Figure 85 pictures the push button as it will be connected to our control device. This little button is very powerful as it allows us to define behavior with movement and it provides instructions that are readable by the machine. These instructions are saved and repeatable. Without this button many of lines of code must be written and tested and rewritten until the desired position is properly met. A push button encoder is a way of automating some of the process for the programmers involved.

6.3 Hardware Design Overview

The control device, in this case our Arduino Mega 2560, is the brains of our project. The Arduino will have many inputs and outputs as mentioned in the section above. In order to have the microcontroller communicate with those inputs and outputs we will have to wire them together. In the next few sections we will be demonstrating how we plan to integrate them to make sure we provide the proper control to the robot.

6.3.1 Breadboard Layout

In Figure 86**,** we illustrate how our breadboard looks like in the Fritzing simulation software.

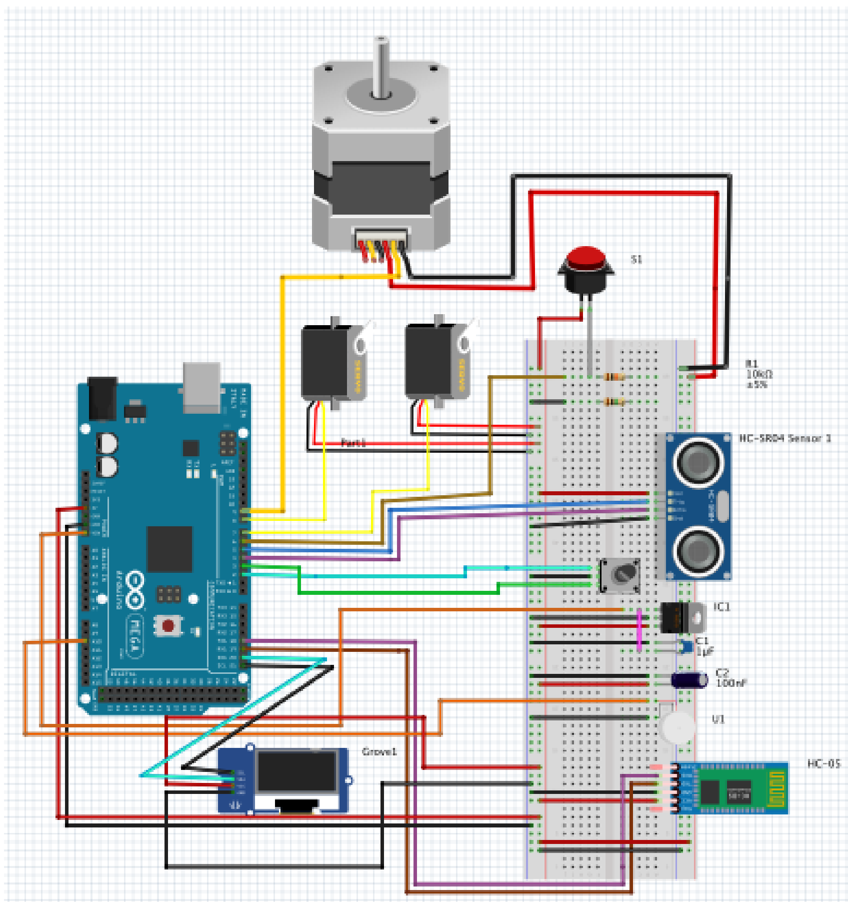


Figure 86 - Breadboard Design

As of now breadboard contains all the main inputs and output, but it is subject to change due to additions in the next semester.

6.3.2 Schematic Layout

After the breadboard has been successfully designed and tested, we want to map it to a schematic design, as shown in Figure 87**.**

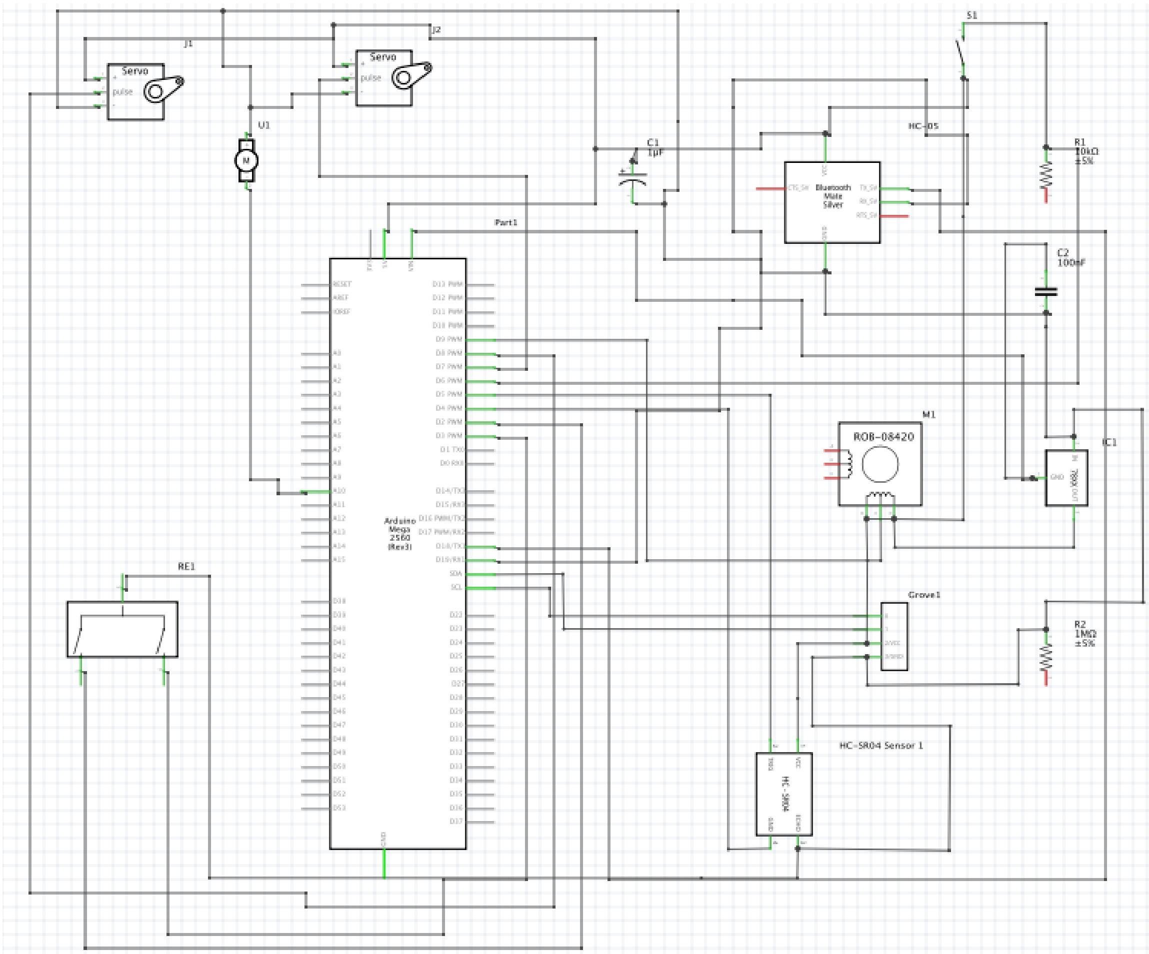


Figure 87 - Schematic Diagram

The schematic shown above illustrates the electronic circuit in an abstract and graphical symbol rather than pictures. This helps us identify the architecture of the circuit and its components.

6.3.3 PCB Layout

After carefully selecting our main components and designing the electrical circuit for our project we will produce a PCB. This will be the most important part of the robot since it will have all the necessary components that will allow it to perform the necessary task. In Figure 88, we can see the PCB that will be sent out to be fabricated next semester.

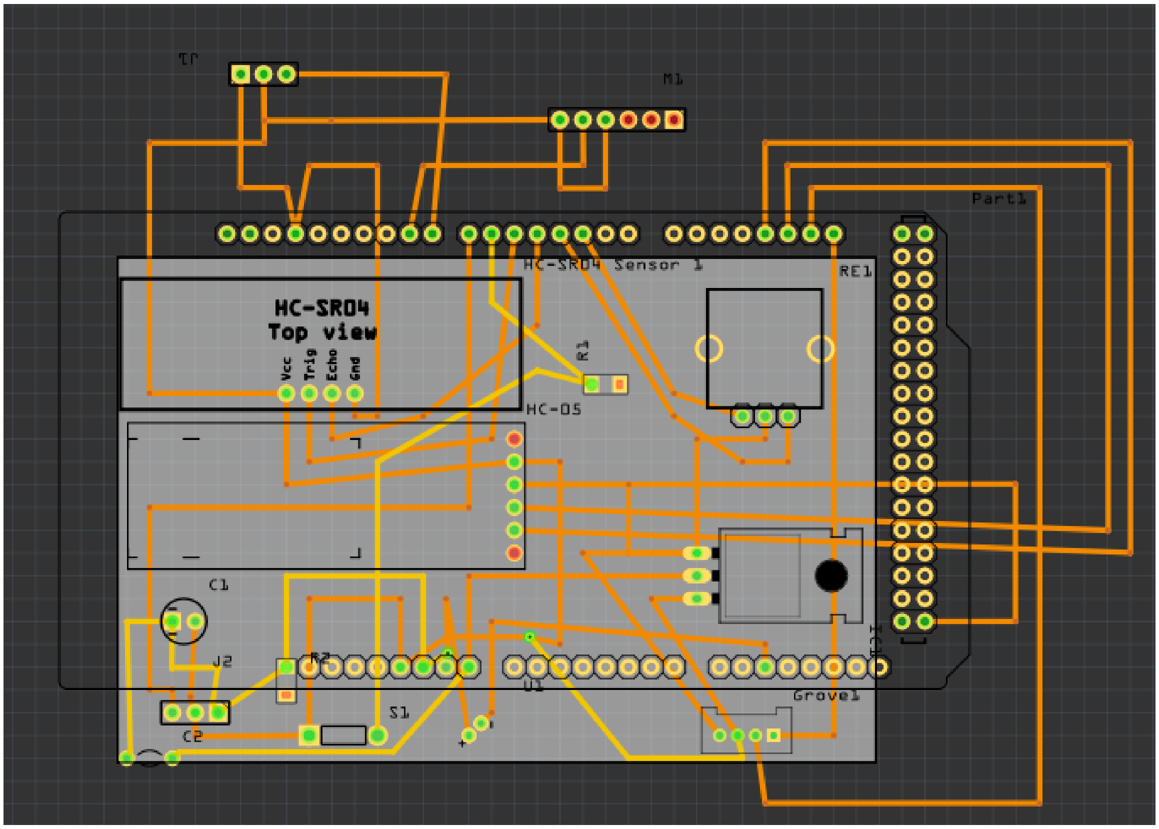


Figure 88 - PCB Design

Once we sent out and receive the fabricated PCB, we will test it multiple times with the input and outputs to make sure all the components and connections in the board are working properly.

7. Administrative Content

Engineering and design efforts require careful and concise administrative planning to be successful. Certain administrative tasks must be created and fulfilled to facilitate and distribute the orderly and timely creation of the system. Each group member is expected to serve as an administrator to the project, handling and acquiring full administrative responsibilities such as adhering to milestones, balancing budgets and finances, management, and maintaining organization of the project.

Administrative responsibilities are extremely important when it comes to engineering. Having milestones ensure that specific tasks for a project are getting completed in a timely fashion and having a certain budget set ensures that there is no overspending on a project. For engineering products to be successful, administrative responsibilities are key.

This section of the report discusses the administrative management portion of the project. It discusses the planned schedule and timeline of the project in the Milestones section. Additionally, it will show and discuss the financial and budget plans as well as the overall cost of the Robotic Flange Assembly in the Budget and Finance section. This section of the report will examine and compare the planned completion dates with the actual completion dates as well as the planned budget with the actual spent budget.

7.1 Milestones Conclusion

Milestones should represent a clear sequence of events that incrementally build up until the project is complete. Milestones show important achievement in a project; they are a way of knowing how the project is advancing with zero duration. Milestone start and end dates depend on the actual task's start and end dates; task association is a major feature of a milestone.

When the schedule is being planned, having several project milestones at once is beneficial. From there, we can estimate the completion date and compare it with the actual completion date. The milestones dates were set based on corresponding tasks in accordance with each of the team member's responsibilities. For this project, the milestone dates for Senior Design I were set based on the concrete due dates of each corresponding task, solely based on the entire groups research and familiarization with the system's components. The milestone dates for Senior Design II will be set in the future, solely based on the design and functionality of the system developed by all group members.

Displayed in Table 27 below compares the initial predicted milestone deadlines with the actual milestone completion dates. Additional tasks have been added to the Milestone Deadline Comparisons Table as they displayed high importance and responsibility for being set as a milestone. As illustrated below, the comparisons between the two sets of dates are extremely correlated with one another. Additional changes have been represented with italicized font.

Table 27 - Milestone Deadline Comparisons

Senior Design I Task List	Predicted Due Date	Actual Completed Date
Project Ideas	08/24/18	08/22/18
Project Selection	09/14/18	09/10/18
Assign Member Roles	09/28/18	09/27/18
Initial Divide & Conquer	09/14/18	09/14/18
Divide & Conquer Revision	09/28/18	09/26/18
60-page Submission	11/02/18	11/02/18
100-page Submission	11/16/18	11/16/18
Final Document	12/03/18	12/02/18
Order & Test Parts	01/07/19	12/03/18
Component Research	11/16/18	11/16/18
Acquire Components	11/19/18	11/16/18
Circuit Design & Integration	11/26/18	11/28/18
Senior Design II Task List	**Predicted Due Date**	**Actual Completed Date**
PCB Layout Design	12/03/18	12/02/18
Build Test Prototype	01/07/19	11/20/18
Testing & Redesign	TBA	TBA
Finalize Prototype	TBA	TBA
Peer Presentation	TBA	TBA
Final Report	TBA	TBA
Final Presentation	TBA	TBA

As displayed in Table 27 above, almost all the milestone completed dates were done before the predicted due date. Therefore, the milestone achievements for the robotic flange assembly have consistently been done on time most of the time. Component research, acquiring the components, circuit design and integration, and PCB layout design were all additions to the milestones table as well. Each of these milestones discovered throughout the Senior Design I process served as great importance to the completion of the robotic flange assembly and were necessary to be set as a milestone. Each of these milestones are extremely critical and important in helping determine the success or failure of the robotic flange assembly.

7.2 Budget and Finance Conclusion

For the robotic flange assembly, Siemens granted us a sponsored donation of $1200. The sponsored donation of $1200 must be distributed across four teams consisting of sixteen people; the electrical and computer engineers (four people), the mechanical engineers (six people), the industrial engineers (three people), and the computer science programmers (three people). Having a budget of only $1200 being dispersed amongst sixteen people creates a huge design and budget constraint for the creation and implementation of the project. Not having enough finances can result in having to use fewer durable materials, the possibility of not having the system completed in time, or having to reduce the number of components for the system so that budgets and finances can be met, therefore possibly affecting the overall design goals of the robotic flange assembly.

To construct, design, program, and wire the robotic flange assembly, many components are needed throughout each discipline. Some of the major necessary components for the electrical and computer engineers include a PCB, power supply, sensors, servos, a robotic programming software, and much more. These necessary components will be used to power and program the robotic flange assembly to perform all necessary functionalities.

Displayed below in Table 28 is an updated, more detailed version of Table 1. This table details the specific components needed for the robotic flange assembly, the quantity needed for each component, and the total prices of each component based on the quantity. These prices will be added up together, totaled, and compared with the total sponsored donation of $1200. For the components and their prices that are being estimated, the overall cost will be adjusted accordingly as the project system's requirements change or get updated throughout the course of its implementation.

Table 28 - Budget for ECE Components

Component	Individual Cost	Quantity	Total Cost
Arduino Mega - Atmega328P	$38.50	1	$38.50
Printed Circuit Boards	$5.00	4	$20.00
18-V Motors	$15.95	2	$31.90
Continuous Motor	$30.65	2	$61.30
Torque Motor	$15.49	1	$15.49

Motor Pack	$26.99	1	$26.99
AmpFlow E-30 Motor	$119.45	1	$119.45
Ultrasonic Distance Sensor	$4.00	1	$4.00
Vibration Sensor	$0.60	1	$0.60
Standard Servo	$13.99	1	$13.99
Micro Servo	$10.99	1	$10.99
Continuous Servo	$16.95	2	$33.90
Push Button	$8.59	1	$8.59
Resistors	$0.90	2	$1.80
Capacitors	$0.45	2	$0.90
Power Supply	$50.00	1	$50.00
5-V Voltage Regulator	$0.78	1	$0.78
Rotary Encoder	$22.00	1	$22.00
OLED Display	$20.00	1	$20.00
Wireless Connection	$80.00	1	$80.00
Miscellaneous	$10.00	5	$50.00
Total Cost Before Tax	**-**	**-**	**$611.18**
Total Cost After Tax	**-**	**-**	**$653.96**
Remaining Budget			**$546.04**

As displayed in the table above, the total cost for just the electrical and computer engineering components amount to a total of **$653.96**. The miscellaneous section was added to account for any additional components we may need for our system but have not accounted for. Having an overestimated budget accounting for possible necessary components ensures that we will still meet our budget and not go over the sponsored donation of $1200.

Taking into consideration that the total price of $653.96 only corresponds with the electrical and computer engineering components gives the rest of the group (twelve members) a remainder of $546.06 to use. The remainder of that price will go towards the mechanical engineers, as the computer science programmers and industrial engineers do not have any necessary components to purchase. Since the mechanical engineers will only have a budget of $546.06 to use, this may cause an overhead budget amongst the group members, thus resulting in using our own funds and money to purchase any other necessary components for the system. Until the final design and components are 100% decided amongst all disciplines, it is unknown if an overhead budget will happen for the group members.

7.3 Division of Labor

Since the robotic flange assembly is an interdisciplinary project that merges four different types of disciplines to work together, having tasks divided up amongst members is crucial. Since this is an interdisciplinary project, it is hard to keep up with which discipline is doing what, especially when the tasks go into further details. In a large scope, there are four main divisions of labor that correspond with the four different disciplines working on the robotic flange assembly. These four divisions include software and simulation, power and control, mechanical and physical design, and ergonomics for safety and measurability. For the electrical engineers, computer engineers, and computer science programmers, software and simulation and power and control are the main divisions of labor for those disciplines. For the mechanical engineers, power and control and mechanical and physical design are the main divisions of labor for those disciplines. For the industrial engineers, ergonomics is the main division of labor for that discipline.

For the electrical and computer engineering portion of the robotic flange assembly, there are many different divisions of labor to account for. Underneath the power and control and software and simulation divisions lie tasks in much more detail. These tasks include the PCB layout assembly, testing motor controls, power configuration, and control devices. The CpE and EE members Viviana, Cassidy, Tony, and Alana will be responsible for each of the mentioned divisions respectively.

8. List of Acronyms

Table 29 is a list of acronyms that have been used throughout this document.

Table 29 - List of Acronyms

Acronym	Full Text
AC	Alternating Current
AI	Artificial Intelligence
API	American Petroleum Institute
ASME	American Society of Mechanical Engineers
ASTM	American Society for Testing and Materials
CPU	Central Processing Unit
DB	Dual Bearing
DC	Direct Current
DM	Direct Mount
ECE	Electrical and Computer Engineering
EPROM	Erasable Programmable Read-Only Memory
ESA	European Sealing Association
FSA	Fluid Sealing Association
HMI	Human Machine Interface
HRI	Human-Robot Interaction
I/O	Input and Output
ID	Inside Diameter
IEEE	Institute of Electrical and Electronic Engineers
MCU	Microcontroller
ME	Mechanical Engineering
NPS	Nominal Pipe Size
OD	Outside Diameter
OLED	Organic Light-Emitting Diode
PCB	Printed Circuit Board
PLC	Programmable Logic Controller
POC	Person of Contact
PROM	Programmable Read-Only Memory
PSI	Pounds per Square Inch
PSIG	Per Square Inch Gauge
RAM	Random-Access Memory

ROM	Read Only Memory
ROS	Robotic Operating System
RPM	Rotations Per Minute
SB	Seal Bearing
SDR	Software Defined Radio
SLAM	Simultaneous Localization and Mapping
VREP	Virtual Robot Experimentation Platform
USB	Universal Serial Bus

Citations

[1] L.Wei EEL4914 Microsoft Word document titled "2018-19 I-Design Projects 8_07 update.docx" College of Engineering and Computer Science, University of Central Florida Orlando, FL, Aug. 9, 2018.

[2] Shenzhen Edraw Software Co., LTD. [Web-Based]. Nanshan District, Shenzhen City, Guangdong Province, 2004-2018. Available: https://www.edrawsoft.com/edraw-max.php [Accessed 14 Sept. 2018]

[3] What is Piping. (2016). *Meaning of Pipe Schedule / Schedule Numbers.* [online] Available at: http://www.whatispiping.com/pipe-schedule [Accessed 15 Oct. 2018]

[4] "Matthew Davis. (2018). *Data Sheet for Steel Australian & DIN Standard Flanges.* [online] Available at: http://www.matthewdavis.com.au/downloads/pds_Flange_Data_Sheet.pdf [Accessed 12 Oct. 2018]

[5] "Project Materials. (2017). *Flange Pressure Temperature Rating Chart.* [online] Available at: https://blog.projectmaterials.com/flanges/flange-rating-chart/ [Accessed 11 Oct. 2018]

[6] Petersen Products. (2018). *ASME ANSI Flange Sizes*. [online] Available at: https://www.petersenproducts.com/ASME-ANSI-Flange-Sizes-s/1987.htm Accessed 8 Oct. 2018].

[7] Werner Solken. (2018). *Torque Tightening.* [online] Available at: http://www.wermac.org/flanges/flanges_torque-tightening_torque-wrenches.html [Accessed 21 Oct. 2018]

[8] Woodward, A. Bower Motorsport Media. (2016). *Beginner Lug Nut Torque and Proper Tightening*. [online] Available at: https://www.ladiesoffroadnetwork.com/beginner-lug-nut-torque-and-proper-tightening/ [Accessed 3 Oct. 2018]

[9] Khan Academy. (2018). *Torque.* [online] Available at: https://www.khanacademy.org/science/physics/torque-angular-momentum/torque-tutorial/a/torque [Accessed 11 Oct. 2018]

[10] Ixthus Instrumentation. (2016). *Types of Torque Sensors.* [online] Available at: https://www.ixthus.co.uk/blog/detail.php?aid=13&did=Types-of-Torque-Sensor [Accessed 23 Oct. 2018]

[11] Sensors and Actuators. (2017). *How Torque Sensor Works.* [online] Available at: http://sensors-actuators-info.blogspot.com/2012/06/how-torque-sensor-works.html [Accessed 30 Oct. 2018]

[12] "Definition and Details of Flanges - Torque Tightening." *Explore the World of Piping*, Werner Sölken, www.wermac.org/flanges/flanges_torque-tightening_torque-wrenches.html.

[13] Rilco Industrial Controls, Inc. (2016). *These arms don't get tired: The beginning of the robotic arm.* [online]. Available at: https://www.rilcoind.com/mobile/blogpost.aspx?ID=1006 [Accessed Oct 24, 2018].

[14] Acieta (2018). *Assembly Robotics.* [online]. Available at: https://www.acieta.com/automation-application/assembly-robotics/ [Accessed Oct 25, 2018].

[15] Instructables (2018). *Complete Motor Guide For Robotics.* [online] Available at: https://www.instructables.com/id/Complete-Motor-Guide-for-Robotics/ [Accessed Oct 19, 2018].

[16] Grainger (1994-2018). *Motor Mounting Supplies.* [online] Available at: https://www.grainger.com/category/motor-mounting-supplies/motor-supplies/motors/ecatalog/N-19eh [Accessed: Oct 19, 2018].

[17] RobotShop (2018). *How to Make a Robot – Lesson 7: Using Sensors.* [online]. Available at: https://www.robotshop.com/community/tutorials/show/how-to-make-a-robot-lesson-7-using-sensors [Accessed Oct 20, 2018].

[18] RoboRobotics: Robotics and Animatronics (2011-2012). *How to Mount a Servo Motor.* [online] Available at: http://roborobotics.com/Animatronics/animatronic-mechanical-design/how-to-mount-servo-motors.html [Accessed Oct 21, 2018].

[19] SuperDroid Robots. *Motor and Wheel Mounting Methods.* [online]. Available at: https://www.superdroidrobots.com/shop/custom.aspx/motor-and-wheel-mounting-methods/95/ [Accessed Oct 28, 2018].

[20] SparkFun Electronics. *PCB Basics.* [online]. Available at: https://learn.sparkfun.com/tutorials/pcb-basics/all?print=1 [Accessed Oct 22, 2018].

[21] Safety + Health (2018). *Robots in the workplace.* [online]. Available at: https://www.safetyandhealthmagazine.com/articles/16789-robots-in-the-workplace [Accessed Oct 23, 2018].

[22] TechTarget (1999-2018). *Collaborative Robot (cobot).* [online]. Available at: https://whatis.techtarget.com/definition/collaborative-robot-cobot [Accessed Oct 24, 2018].

[23] Phys.org (2003-2018). *Sharing the workplace with robots? New tools help designers create safer machines.* [online]. Available at: https://phys.org/news/2018-05-workplace-robots-tool-safer-machines.html [Accessed Oct 23, 2018].

[24] GT Engineering. *EN ISO 12100: Risk Assessment.* [online]. Available at: http://www.gt-engineering.it/en/en-iso-standards/risk-assessment-en-12100 [Accessed Oct 26, 2018].

[25] A.Owen-Hill, "What is The Best Programming Language For Robotics," March, 2016. [Online]. Available: https://blog.robotiq.com. [Accessed Oct. 20, 2018].

[26] Y.Pyo, "Wiki," *hls_lfcd_lds_driver* Available: http://wiki.ros.org/hls_lfcd_lds_driver [last edited Apr. 4, 2018]. [Accessed Oct. 22, 2018]

[27] Robotis, "360 Laser Distance Sensor LDS-01," [903-0258-000] Specification Data, Available: http://www.robotis.us/360-laser-distance-sensor-lds-01-lidar/. [Accessed Oct. 22, 2018]

[28] TeraRanger Duo, by TeraBee Available at: http://www.teraranger.com/product/teraranger-duo/ [Accessed Oct. 22, 2018]

[29] Engineering ToolBox, (2009). *Electric Motors - Power and Torque vs. Speed*. [online] Available at: https://www.engineeringtoolbox.com/electrical-motors-hp-torque-rpm-d_1503.html [Accessed Oct. 22, 2018].

[30] Dejuan, *Arduino Tutorials*, "DIY Arduino Robot Arm with Smartphone Control". Available at: https://howtomechatronics.com/tutorials/arduino/diy-arduino-robot-arm-with-smartphone-control/ [Accessed Oct. 21, 2018]

[31] Engineering ToolBox, (2009). *Electric Motors - Power and Torque vs. Speed*. [online] Available at: https://www.engineeringtoolbox.com/electrical-motors-hp-torque- rpm-d_1503.html [Accessed Oct. 22, 2018].

[32] DeWalt, (2018). *20V MAX* XR® TOOL CONNECT™ COMPACT HAMMERDRILL*. [online] Available at: https://www.dewalt.com/products/power-tools/connected-products/20v-max-xr-tool-connect-compact-hammerdrill- kit [Accessed Oct. 30, 2018].

[33] DeWalt, (2018). *20V MAX* 3.0 AH COMPACT BATTERY*. [online] Available at: https://www.dewalt.com/products/accessories/batteries-and-

chargers/batteries/20v-max-30-ah-compact-battery/dcb230 [Accessed Oct. 30, 2018].

[34] Amazon (2018) [online] Available at https://www.amazon.com/s/ref=sr_st_price-asc-rank?keywords=20-volt+battery+dewalt&rh=n%3A228013%2Ck%3A20-volt+battery+dewalt&qid=1541140035&sort=price-asc-rank [Accessed Nov. 1, 2018].

[35] learn.parallax.com (2018) *Build a Servo Power Supply* [online] Available at https://learn.parallax.com/tutorials/robot/elev-8/add-servos-elev-8-v3/build-servo-power-supply [Accessed Nov. 1, 2018].

[36] Parallax Inc., “PIR Sensor,” #555-28027 datasheet, March. 24, 2014

[37] REUK.co.uk, “Project of the Day – Multiple PIR Sensors to Control Multiple Lights. [Online]. Available http://www.reuk.co.uk/wordpress/project-of-the-day-multiple-pir-sensors-to-control-multiple-lights/ [Accessed Nov. 1, 2018].

[38] Atmel, “ATmega48A/PA/88A/PA/168A/PA/328/P,” megaAVR® datasheet, Nov. 2015 [Revised Oct. 2018].

[39] Explainthatstuff!, “Inverters”. [Online]. Available https://www.explainthatstuff.com/how-inverters-work.html [Accessed Nov. 1, 2018].

[40] Wikibooks, “AC to DC converter”. [Online]. Available https://en.wikibooks.org/wiki/Electronics_Handbook/Devices/Converter/AC_to_DC_Converter [Accessed Nov. 1, 2018].

[41] Dronetrest, “Power Distribution Boards - How to choose the right one”. [Online]. Available https://www.dronetrest.com/t/power-distribution-boards-how-to-choose-the-right-one/1259 [Accessed Nov. 1, 2018].

[42] M.Liss , *20mm nuts and bolts m3-m16.* Jan, 13, 2015 [Online] Available at: https://www.thingiverse.com/thing:633534/files [Accessed Nov 10, 2018].

[43] TigerTek(1983). “Difference Between Servo Motor and Stepper Motor”[Online] Available at: http://www.tigertek.com/servo-motor-resources/differences-between-servo- stepper-motors.html [Accessed Nov 7, 2018].

[44] Shah, A. (2018). *Intel on the cheap: Chip maker ships $15 Quark maker board*. [online] PCWorld. Available at: https://www.pcworld.com/article/3055372/internet-of-things/intel-on-the-cheap-chip-maker-ships-15-iot-developer-board.html [Accessed 2 Nov. 2018].

[45] Yadav, D. and Singh, A. (2006). *Microcontrollers*. New Delhi: New Age International (P) Ltd. Publishers.

[46] The Mag Pi. (2015). *Raspberry Pi 3: Specs, Benchmarks & Testing.* [online] Available at: https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/ [Accessed 30 Oct. 2018]

[47] Texas Instruments. (2018). *MSP430FR6989.* [online] Available at: http://www.ti.com/product/MSP430FR6989 [Accessed 30 Oct. 2018]

[48] Store.arduino.cc. (2018). *Arduino Mega 2560 Rev3*. [online] Available at: https://store.arduino.cc/usa/arduino-mega-2560-rev3 [Accessed 6 Nov. 2018].

[49] "Field Programmable Gate Array (FPGA)" xilinx.com, Nov. 22, 2017. [Online]. Available: https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html [Accessed Nov. 16, 2018]. (FPGA definition)

[50] Asim Qureshi, "10 Best FPGA Boards For Engineers And Hobbyists" wonderfulengineering.com, 2015. [Online]. Available: https://wonderfulengineering.com/10-best-fpga-boards-for-engineers-and-hobbyists/ [Accessed Nov. 16, 2018]. (best FPGAs)

[51] "Comparing an FPGA to a Microcontroller, Microprocessor or an ASIC" element14.com, Feb 22, 2018. [Online]. Available: https://www.element14.com/community/groups/fpga-group/blog/2018/02/22/comparing-an-fpga-to-a-microcontroller-microprocessor or-an- asic [Accessed Nov. 16, 2018]. (Microcontroller vs. FPGA)

[52] "Picking The Right Tool For The Job: MCU, SBC or FPGA?" mouser.com, August 13, 2014. [Online]. Available: https://www.mouser.com/blog/picking-the-right-tool-for-the-job- mcu-sbc-or-fpga [Accessed Nov. 16, 2018]. (MCUvs.SBCvs.FPGA)

[53] "What is the difference between CPLD and FPGA?" quora.com, 2013. [Online]. Available: https://www.quora.com/What-is-the-difference-between-CPLD-and-FPGA [Accessed Nov. 16, 2018]. (FPGA image)

[54] "Block diagram (SBD) for Single Board Computer, Human Machine Interface application equipment" electronicproducts.com, November 6, 2013. [Online]. Available: https://www.electronicproducts.com/Board_Level_Products/Single_Board_Computer/ HMI_Human_Machine_Interface.aspx (SBC image)

[55] "Microprocessor and Microcontroller – The difference" circuitstoday.com, December 5, 2011. [Online]. Available:

http://www.circuitstoday.com/microprocessor-and- microcontroller [Accessed Nov. 16, 2018]. (microcontroller image)

[56] "Block diagram (SBD) for Single Board Computer, Human Machine Interface application equipment" electronicproducts.com, November 6, 2013. [Online]. Available: https://www.electronicproducts.com/Board_Level_Products/Single_Board_Computer/ HMI_Human_Machine_Interface.aspx (SBC image)

[57] Anaheim Automation (2018). *Encoders | Optical and Magnetic, Incremental and Rotary*. [online] Available at: https://www.anaheimautomation.com/manuals/forms/encoder-guide.php [Accessed 20 Nov. 2018].

[58] Fritzing.org. (2018). *Fritzing*. [online] Available at: http://fritzing.org/home/ [Accessed 18 Nov. 2018].

[59] Autodesk.com. (2018). *EAGLE | PCB Design Software | Autodesk*. [online] Available at: https://www.autodesk.com/products/eagle/overview [Accessed 6 Nov. 2018].

[60] Easyeda.com. (2018). *EasyEDA - Online PCB Design & Circuit Simulator*. [online] Available at: https://easyeda.com [Accessed 28 Nov. 2018].

[61] TryEngineering, "What is an engineering standard? What are the advantages and disadvantages of using standards in engineering?" (2018). [Online]. Available: http://tryengineering.org/ask-expert/what-engineering-standard-what-are-advantages-and-disadvantages-using-standards. [Accessed: Nov. 2, 2018].

[62] ASME, "Standards & Certifications FAQ," (2018). [Online]. Available: https://www.asme.org/about-asme/standards/about-codes-standards. [Accessed: Nov. 2, 2018].

[63] Engineering Standards: An Introduction for Electrical and Computer Engineering Students, March, 2017. [Online]. Available: https://eujournal.org/index.php/esj/article/viewFile/9028/8613. [Accessed: Nov. 3, 2018].

[64] ANSI News and Publications, "ANSI Announces Improved Search Engine for Standards," (2018). [Online]. Available: https://www.ansi.org/news_publications/news_story?menuid=7&articleid=d1ab780e-f192-441a-9558-df47fa0a2123. [Accessed: Nov. 3, 2018].

[65] NIST, "Standard Knowledge for Robots," May 19, 2015. [Online]. Available: https://www.nist.gov/news-events/news/2015/05/standard-knowledge-robots. [Accessed: Nov. 4, 2018].

[66] ria, “Robotics Online,” (2008-2018). [Online]. Available: https://www.robotics.org/robotics/technical-report-ria-tr-r15-606-2016-for-robots-and-robotic-devices-collaborative-robots. [Accessed: Nov. 4, 2018].

[67] M. Bélanger-Barrette, Robotiq, “Which ISO Standards Are Made for Collaborative Robots,” Aug. 25, 2015. [Online]. Available: https://blog.robotiq.com/which-iso-standards-are-made-for-collaborative-robots. [Accessed: Nov. 5, 2018].

[68] ANSI Webstore, “Electronic Component Standards,” (2018). [Online]. Available: https://webstore.ansi.org/industry/electronic-components#Other%20Electronic%20Components%20Standards. [Accessed: Nov. 5, 2018].

[69] E. Chao, and J. Henshaw, *Controlling Electrical Hazards*, U.S. Department of Labor Occupational Safety and Health Administration, 2002 (Revised). [online document]. Available: https://www.osha.gov/Publications/osha3075.pdf. [Accessed: Nov. 5, 2018].

[70] ANSI Webstore, “Human System Interaction Ergonomics Standards,” (2018). [Online]. Available: https://webstore.ansi.org/industry/ergonomics/human-system-interaction-ergonomics. [Accessed: Nov. 5, 2018].

[71] ANSI Webstore, “Software Standards,” (2018). [Online]. Available: https://webstore.ansi.org/industry/software. [Accessed: Nov. 5, 2018].

[72] ANSI Webstore, “Software Engineering,” (2018). [Online]. Available: https://webstore.ansi.org/industry/software. [Accessed: Nov. 5, 2018].

[73] NPES Standards, “Standards: What Are They And Why Are They Important?” May 2005. [Online]. Available: http://www.npes.org/pdf/Standards-WhatAreThey.pdf. [Accessed: Nov. 6, 2018].

[74] IEEE Standards Association, “IEEE Standards Activities in the Robotics and Automation Space,” Nov. 9, 2018. [Online]. Available: https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/other/robotics.pdf. [Accessed: Nov. 11, 2018].

[75] IGI Global, “What is Design Constraints,” (1988-2018). [Online]. Available: https://www.igi-global.com/dictionary/design-of-hexapod-walking-robots/46586. [Accessed: Nov. 11, 2018].

[76] C. Basu, “What Are The Differences Between Time Constraints & Resource Constraints in a Project?” (2018). [Online]. Available:

https://smallbusiness.chron.com/differences-between-time-constraints-resource-constraints-project-35756.html. [Accessed: Nov. 12, 2018].

[77] P. Wenman, "What are environmental constraints?" Mar. 16, 2018. [Online]. Available: https://www.quora.com/What-are-environmental-constraints. [Accessed: Nov. 12, 2018].

[78] American Society for Engineering Education, "Social Constraints: A Critical Component of Global Humanitarian-based Projects," June 23-26, 2018. [Online]. [Accessed: Nov. 13, 2018].

[79] Encyclopedia, "Engineering Design Ethics," (2005). [Online]. Available: https://www.encyclopedia.com/science/encyclopedias-almanacs-transcripts-and-maps/engineering-design-ethics. [Accessed: Nov. 14, 2018].

[80] D. Firesmith: "Engineering Safety Requirements, Safety Constraints, and Safety-Critical Requirements", in *Journal of Object Technology*, vol. 3, no. 3, March-April 2004, pp. 27-42.
Available: http://www.jot.fm/issues/issue_2004_03/column3. [Accessed: Nov. 14, 2018].

[81] United States Department of Labor, "Occupational Safety and Health Administration," [Online]. Available: https://www.osha.gov/dts/osta/otm/otm_iv/otm_iv_4.html. [Accessed: Nov. 15, 2018].

[82] Engineering Strategy News, "The Importance of Sustainable Design Considerations in Engineering," Aug. 6, 2012. [Online]. Available: https://engineeringstrategynews.com/the-importance-of-sustainable-design-considerations-in-engineering/. [Accessed: Nov. 15, 2018].

[83] argos-sim.info, 2018. [Online]. Available: http://argos-sim.info/ [Accessed Nov. 16, 2018]. (ARGoS Website)

[84] v-rep.eu, 2018. [Online]. Available: http://www.v-rep.eu/ [Accessed Nov. 16, 2018 (V- REP website)

[85] gazebosim.org, 2018. [Online]. Available: http://gazebosim.org/ [Accessed Nov. 16, 2018] (Gazebo website)

[86] "[V-REP, Gazebo or ARGoS? A Robot Simulators Comparison]" lenkaspace.net, Jan 2018. [Online]. Available: http://lenkaspace.net/tutorials/programming/robotSimulatorsComparison [Accessed Nov. 16, 2018]. (Simulator comparison)

[87] "Gazebo : Introduction" robocomp.github.io, 2018. [Online]. Available: https://robocomp.github.io/web/gsoc/2018/akash_kumar_singh/post2 [Accessed Nov. 16, 2018]. (Gazebo image)

[88] "ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems" argos- sim.info, Nov. 16, 2012. [Online]. Available: http://argos-sim.info/stuff/Pinciroli:SI2012.pdf [Accessed Nov. 16, 2018]. (ARGoS images)

[89] "Writing code in and around V-REP" coppeliarobotics.com. [Online]. Available: http://www.coppeliarobotics.com/helpFiles/en/writingCode.htm [Accessed Nov. 16, 2018]. (V-REP images)

[90] "Virtual Robot Experimentation Platform V-REP" v-rep.eu. [Online]. Available: http://www.v-rep.eu/assets/v-repoverviewpresentation.pdf [Accessed Nov. 16, 2018]. (V-REP flowcharts)

[91] "AmpFlow" ampflow.com, [Online]. Available at http://www.ampflow.com/three_inch_high_performance_motors.htm [Accessed Dec. 3, 2018].

[92] "8 x AA battery holder with 5.5mm/2.1mm Plug and On/Off Switch" http://www.voc-electronics.com, 2018. [Online]. Available: http://www.voc-electronics.com/a-39643984/tools/8-x-aa-battery-holder-with-5-5mm-2-1mm-plug-and-on-off-switch/ [Accessed Dec. 3, 2018].

[93] "ARDUINO MEGA 2560 REV3" store.arduino.cc, 2018. [Online]. Available: 93]

[94] "OLED I2C Display Arduino Tutorial" startingelectronics.org, Oct. 17, 2017. [Online]. Available: https://startingelectronics.org/tutorials/arduino/ modules/OLED-128x64-I2C-display/ [Accessed Dec. 3, 2018].

[95] "Parallax PING Ultrasonic Sensor" robotshop.com, 2018. [Online]. Available: https://www.robotshop.com/en/parallax-ping-ultrasonic-sensor.html [Accessed Dec. 3, 2018].

[96] "WGCD 8 Pc KY-040 360 Degree Rotary Encoder Module Brick Sensor Board with Knob Cap for Arduino" amazon.com, Mar. 1, 2018. [Online]. Available: https://www.amazon.com/WGCD-KY-040-Degree-Encoder-Arduino/dp/B07B68H6R8/ref=sr_1_3?s=electronics&ie=UTF8&qid=1543796306&sr=1-3&keywords=rotary+encoder+arduino [Accessed Dec. 3, 2018].

[97] "Pushbutton" www.arduino.cc. [Online]. Available: https://www.arduino.cc/en/tutorial/pushbutton [Accessed Dec. 3, 2018].