ModRob: An Inexpensive and Modular Robotic Arm for Hobbyists

Albert Anwar
Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI

Engineering Honors Capstone Report
Project Advisor: Prof. Grace Hsia, Entrepreneurship/Leadership Lecturer, Center for Entrepreneurship, University of Michigan, Ann Arbor, MI

Submitted on: 4/25/22

Table of Contents

Abstract

Today's robotics industry is applications-centric, where companies provide custom solutions for each customer. This business model results in a high cost for an industrial robot arm that cannot easily adapt to different tasks. To overcome these challenges, this project has investigated the market need for an inexpensive, mass-producible robot arm that would take less time to set up than a custom-built robot arm. The results of the investigation indicated that there was at least some willingness to pay more than $500 on a robotic arm that would take less time to set up. Based on the results, a prototype of a modular arm that could be set up in minutes was built using inexpensive hardware and open-source software. Using this prototype, a 4-joint robotic arm assembly could be built for less than $200, excluding packaging, shipping, and labor costs. This prototype has the potential to bring cheaper, more adaptable robots to the fields of education and research, as well as potential for being used by hobbyists and for rapid prototyping. An additional value proposition is its potential to record specific movements through kinetic-only physical manipulation, without any programming necessary. More work is needed to validate the results obtained from customer discovery and market evaluation, as well as making the prototype robust enough to put on the market.

Introduction

Today's robotics industry is applications-centric, where companies provide custom solutions for each customer. This business model requires high investment in time and money for a single industrial robotic arm model, and the highly customized solution results in low demand from other customers. The result is that the cost per robot arm remains high (can be anywhere from $25,000-$400,000[1]) and becomes limited in other potential applications. The robotics industry is moving towards lower-cost solutions, including robots that can be mass produced, modular robots, and developing software frameworks that can make reprogramming a robot for other applications easier.

A comparison of some existing robotic arm and modular robot solutions are shown in Table 1. These solutions can be either costly, complex to use, or not suitable to perform common robotic arm tasks. Virtually all solutions do not have the ability to customize the number of joints in a robotic arm solution on the fly, within minutes. A potential gap in the market therefore exists for a lower cost, easy-to-use, easy-to-assemble robotic arm solution where the number of joints used can be easily customized to adapt to any task.

Table 1. Comparison of Existing Robotic Arm and Modular Robot Solutions

Product Name or Product Line	Can Perform Common Robotic Arm Tasks?	Customizable number of joints on the fly?	Easy to Use?	Cost	Manufacturer
Cubelets[2] [3]	No	No	Yes, but also requires some programming	$149.00 for a basic kit	Modular Robotics
InterbotiX[4]	Yes	No	No, requires programming and using complex software frameworks	$549.95 for cheapest model	Trossen Robotics
DOBOT[5]	Yes	No	Yes	$2,699.00 for cheapest model	Trossen Robotics
Mirobot[6]	Yes	No	Yes, but also requires some programming	$1,680.00 for cheapest kit	WLKATA
UR[7]	Yes	No, customizable but fixed on order	Yes, but through using an additional product on a subscription basis	N/A, Custom Quote Required	Universal Robots

Objectives

As a result of the high costs in both time and money in engineering an industrial robot arm solution, the goal of the project was to investigate the market need and desire to purchase a cheaper robotic arm that takes less time to set up for executing different tasks, and provide a potential solution. To complete this goal, three objectives were set up for the project:

1. To perform customer discovery and assess a desire for a modular, mass-producible robot arm.
2. To provide a robot arm model that is general enough for mass production and higher volume sales.
3. To provide a robot arm model that is easy to assemble and configure for different tasks.

The second and third objectives are contingent on the results of the first. If there is no market desire for an inexpensive robotic arm with less setup time, then providing a prototype would not be worthwhile and would need to pivot away from the second and third objectives.

Methods

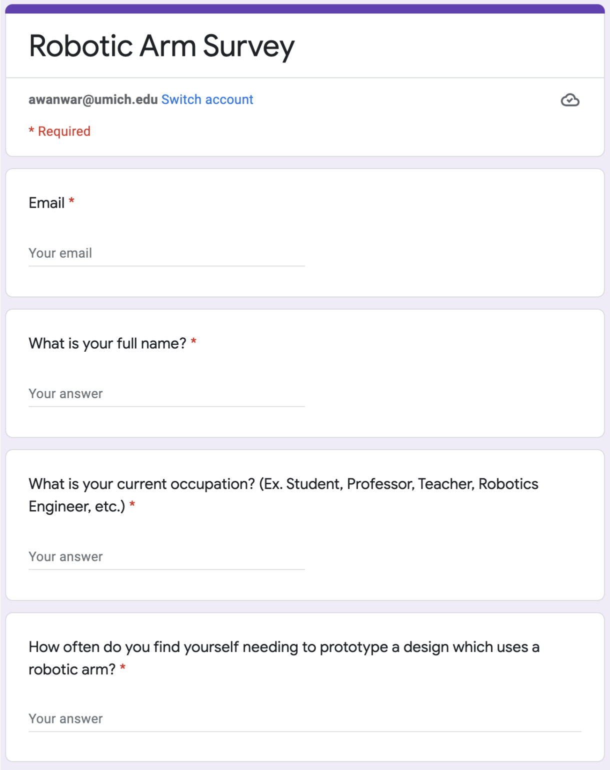
There were two main components to the project. The first is customer discovery and market evaluation, and the second is building the prototype. Each of them is described in detail below.

Customer Discovery and Market Evaluation

Customer discovery and market evaluation were arguably the most important steps of the project. The purpose behind this step was to estimate the market demand and desire (need) to purchase an inexpensive, adaptable robotic arm that is easy to set up. It also served as a guide to develop and market different features or products to customer segments. The worthwhileness to construct and features of a prototype would depend on the results of this portion of the project. To perform customer discovery the following steps were taken:

1. Provide a survey to assess the current process of building and prototyping robotic arms. The survey benchmarks customer willingness to spend less time and/or money during the prototyping and building process. It also gathers some information about who the users are such that different customer profiles could be built, each of which could be marketed to separately.
2. Process the data from the survey to build different customer profiles, if applicable. This is done by summarizing the data into quantitative or discrete results, and using pivot tables to give a top-level overview and find trends within the results.
3. Perform a literature review, using library databases to assess the overall market for educational robots and the potential for inexpensive modular robotic arms.

The Customer Discovery Survey



Robotic Arm Survey

awanwar@umich.edu Switch account

* Required

Email *

Your email

What is your full name? *

Your answer

What is your current occupation? (Ex. Student, Professor, Teacher, Robotics Engineer, etc.) *

Your answer

How often do you find yourself needing to prototype a design which uses a robotic arm? *

Your answer

Figure 1. The customer discovery survey.

The first step in the process of customer discovery and market evaluation was to provide a survey for those who have existing experience in building and prototyping robotic arms. The purpose of the survey was to investigate the current process of building and prototyping the arms, discovering what customer's pain points are, how important those pain points are, and what they would do to resolve such pain points. The survey collected some data about current occupation so that different customer profiles could be built from the data. It also looked at how much they have spent in terms of time and money in prototyping the arms. The survey asked how much respondents would be willing to pay (in dollar amounts) for a cheaper robotic arm that would take less time to assemble and set up.

The survey was created using Google Forms, with email addresses collected so the form does not get filled out twice by the same person and skew the results. The survey was handed to select people within the University of Michigan network who had experience in robotics. No monetary reward was given at the end of the survey. The survey questions are as follows:

1. What is your full name?
2. What is your current occupation? (Ex. Student, Professor, Teacher, Robotics Engineer, etc.)
3. How often do you find yourself needing to prototype a design which uses a robotic arm?
4. If you have ever needed to prototype a design which uses a robotic arm, how did you go about prototyping the design(s)? Please provide as much detail about the process as you can.

5. If you have ever needed to prototype a design which uses a robotic arm, which parts of the prototyping process are you most dissatisfied about? Are alleviating those dissatisfactions important to you?
6. If you have ever needed to prototype a design which uses a robotic arm, how much time did you spend assembling the prototype? How much time did you spend programming the prototype? About how much did the prototype cost, in US dollar amounts?
7. In what situations in your life would you use a robotic arm that can rapidly and easily be assembled and programmed to perform a variety of different tasks?
8. On a scale of 1 to 10, with 1 being not helpful in any way, and 10 being an absolute necessity, how much value to your life would a robotic arm that can rapidly and easily be assembled and programmed to perform a variety of different tasks provide?
9. How much would you be willing to pay for a robotic arm that can rapidly and easily be assembled and programmed to perform a variety of different tasks, in US dollar amounts?
10. Are there any other person(s) that you know of that have either spent time creating solutions that incorporate a robotic arm, or would need a robotic arm that can quickly adapt to performing different tasks? If so, try to name and give contact information for at least three such people.
11. Are there any other comments you would like to add regarding your experience with robotic arms?

The survey can be found at the following URL: https://forms.gle/FHqzkFFvopBHGxgDA

Building Customer Profiles

After the data from the survey was collected, one or more customer profiles needed to be built to characterize how likely and how much someone from a customer profile would spend on a cheaper, less time consuming to assemble robotic arm.

The first step was to summarize the results of the questions of the survey, and to produce more quantitative results. The purpose of this step was to generate results that can be more easily analyzed, and that can indicate certain trends without needing to dive deep into the responses. It also allowed the results to be more easily organized for a top-level overview. Customer profiles were also built during this step .

The second step was to generate pivot tables to discover trends in the results. The pivot tables were generated using Microsoft Excel. Pivot tables allowed us to easily compare and summarize different results from the survey, such as the money spent on building a robotic arm for each customer profile.

Literature Review

In addition to performing the customer discovery survey and summarizing and analyzing its results, a literature review was also performed to provide a top-down assessment of the robotics industry and its future, and to assess the viability of modular robots as a strategy to implement inexpensive robots. To perform the literature review, sources from the University of Michigan Library database were used. Relevant search terms to modular robotics and inexpensive robots were used to search for sources. The following information was recorded for each source: Platform Searched, Keywords Searched, Type of Work (such as Book, Article, Thesis/Dissertation), Year Published, Title of Work, Author, URL of Work, Summary of Work, and Notes. The Notes section can summarize extra context for the work being recorded, and was not necessarily required for each work. The literature review was kept as a Microsoft Excel spreadsheet.

Building the Prototype

The second project component was to build the prototype, based on the results of customer discovery. As a result of customer discovery, the modular approach to building a robotic arm seemed like a viable strategy to allow customers to easily assemble and configure their arm. The prototype arm would consist of three main designs: the joint of the arm to actuate movement, joint connectors to link the joints together, and a controller that would coordinate the movement of each joint. Each of these designs would be a product that could be sold so that the customer could use these parts to assemble their arm system.

Aims and Objectives

The goal of this type of design was to allow any user to quickly assemble a prototype arm within minutes. The arm should be able to be assembled by selecting a number of joints, and then linking each joint together with the joint connectors, ideally by snapping on a joint to each end of the connector to make the assembly process simple. The software would be able to recognize the number of joints connected, and coordinate entire arm movements based off of this information.

A key design goal was to have the robot arm assembly be adaptable in performing different tasks. A step towards achieving this goal would be allowing the user to add or remove joints or connectors based on what is needed to perform a specific task. This was achieved easily with the modular approach, as it would only take a couple of minutes for a user to add or remove a joint or replace a connector, and have the software recognize the new number of joints being used. On the other hand, a robotic arm currently on the market would require the user to spend lots of time and/or money trying to add or remove a joint, and in some cases would require the user to purchase a completely new arm assembly instead.

Another advantage of this modular approach was the ease of which parts can be replaced. If a joint or connector breaks, the modularity of the design allows the joint or connector to be replaced with the exact same part for minimum cost and replacement time. This was opposed to products currently on the robotic arm market; if a joint breaks, it may be quite expensive and/or time consuming to replace a part since the complete arm assembly is tightly integrated.

System Architecture

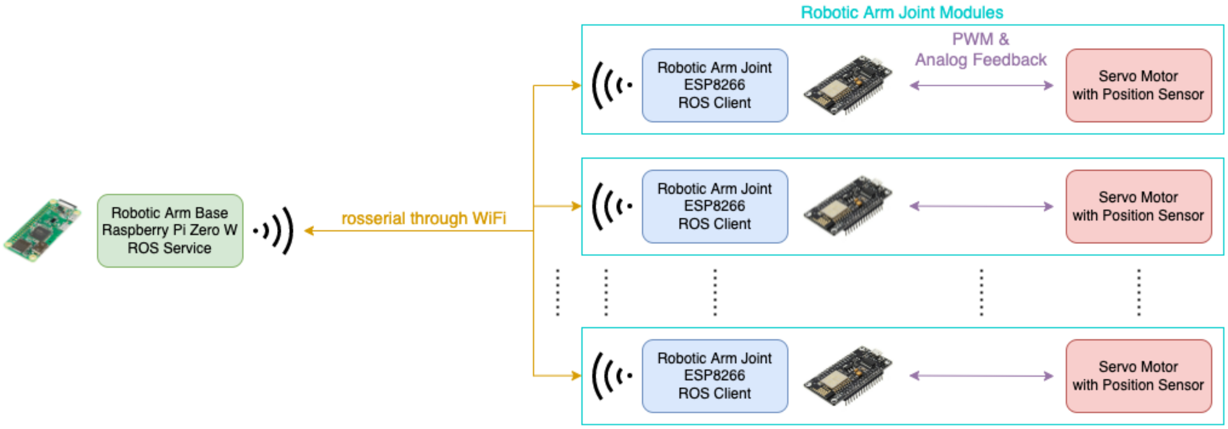


Figure 2. System architecture for the robotic arm prototype.

With the modular approach in mind, the system architecture of the robot can be summarized in Figure 2. Each joint would communicate wirelessly to the controller, which allows modularity to be more easily achieved because wires from each joint do not need to be routed to the controller in order for the joint to communicate with the controller. Since each joint could be placed anywhere in the total robot arm assembly and connected together through variable connector lengths, having a wired communication meant that length of the wire connecting the joint to the controller would be unknown. If the wire length was made long to prepare for a worst case scenario, this could not only result in a degradation of reliable communication between the joint and the controller, but the wire could also interfere with the whole robot's operation if the wires were not properly managed and kept out of the way.

For the scope of this project, the prototype was meant to perform two tasks: first, calibrate each arm joint to help eliminate any manufacturing discrepancies between parts; and second, to record a single arm position and allow the user to return to that position on the push of a button.

The Arm Controller

There were two parts of designing the arm controller: the hardware, and the software. Each of these parts are described in detail below.

Hardware

The arm controller was made up of a Raspberry Pi Zero W, along with wires, push buttons, and a breadboard to allow for user control. There were three push buttons used in the controller: one used to commence the calibration task, one to record the current position of the arm assembly, and another that would either relax every joint upon being pushed, or tell each joint to move the arm to the recorded position when pushed. Each push button is placed on the breadboard, wired on one end to 3.3V, and on the other end to a current-limiting resistor (220Ω) in series with a GPIO pin of the Raspberry Pi Zero W. When a button was pushed, the corresponding GPIO pin would detect a rising edge, and execute software in response to the button push. A breadboard diagram of the circuit is shown below in Figure 3.

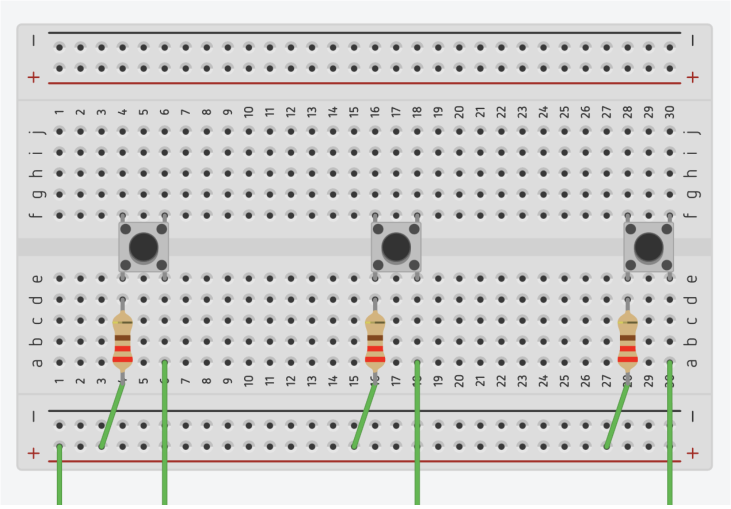


Figure 3. Breadboard diagram of the arm controller's button circuit.

Software

The arm controller was loaded with the default Raspbian operating system, which includes the open-source Linux operating system. The system was also configured for wireless operation, specifically using WiFi at 2.4GHz. The Raspberry Pi Zero W was configured as a wireless access point to allow arm joints to connect wirelessly to it. On top of the operating system, the arm controller ran a software package called Robot Operating System, or ROS. ROS is an open-source software package meant for robotic systems, and one of its capabilities is to package messages to be sent to different parts of the robot. Additionally, a package called rosserial was used to serialize the messages being sent, which is useful for wireless communication. The controller ran a ROS node, which in this case was software that I've written in Python 3 to publish data to the arm joint and subscribe to data that the arm joint publishes. The ROS node would also handle action upon a button press. When sending data to the arm joint, the ROS package would package the data into messages for communication, and rosserial would serialize the messages to be sent over WiFi. When receiving data from the arm joint, rosserial would deserialize the messages, and then ROS would unpack those messages such that the new data can be used by the ROS node.

Whenever a button was pressed, the ROS node would use a library called pigpio to detect and debounce the button press from any of the three buttons. The button debouncing ensured that the button does not accidentally trigger button presses more than once when the user intended to only press the button once. When the calibration button was pressed, the arm controller published a command to the arm joint to perform a calibration sequence, and then waited for another button press to occur. This sequence was handled completely by the arm joint and will be discussed in that section.

When the button to record the current position of the arm assembly was pressed, the controller published a command to the arm joint through ROS to grab the current angle of the servo motor. The controller would subscribe to the response, waiting for the arm joint to send the angle back to the controller. The controller would then store the angle of each joint in a global variable, until that variable is overwritten, or the controller is turned off.

The button that controlled joint movement or relaxation used a state variable to record whether the joints are currently locked to a certain position, or relaxed so that they can be moved around more easily by the user. If the state was in the locked state, the button press would move the joints to a relaxed state by sending a command to each arm joint to relax. When the state was in the relaxed state, the button press would send a command to the joints to move to the angle recorded in the joint's corresponding global variables.

The controller was designed to execute the code within the ROS node upon power-up. It would take a couple of minutes for the Raspberry Pi Zero W to set up the system before it would be ready to receive user input in the form of button presses. When the controller became ready, an LED on the Raspberry Pi Zero W would begin blinking.

The Arm Joint

There were two parts of designing the arm joint: the hardware, and the software. Each of these parts are described in detail below.

Hardware

The arm joint consisted of a servo to provide the joint's motion, and an ESP8266 microcontroller chip with wireless communication capabilities running ROS, rosserial, and software that I've written in C to control the servo and to manage communication with the arm controller.

The servo used for the prototype is the Batan S1213. The motor was controlled through pulse-width modulation (PWM) and included an analog feedback wire that indicates the servo's current position. The servo ran on 5V, and the PWM and the feedback wire connected to the ESP8266. However, since the ESP8266 ran on 3.3V, voltage conversion on the feedback wire needed to occur, or else the ESP8266 could be damaged from a 5V signal on the wire. This voltage conversion was done through a resistor divider.

A battery powered both the ESP8266 and the servo on the joint. The battery used was a 4S NiMH battery with a nominal voltage of 4.8V. The battery voltage was filtered through capacitors to reduce noise on the power lines, and directly powered the servo motor. The battery's power lines also ran through a LD1117 low dropout regulator (LDO) to provide a 3.3V power supply to the ESP8266.

Software

The ESP8266 was able to run Arduino code programmed in C. It was programmed to act as a ROS node that could react to commands given by the arm controller, publish and subscribe to data over rosserial, and manage the servo.

When the calibration command is received, the ESP8266 would tell the arm joint to move from 0˚ to 180˚ in increments of 6˚. At each increment, the value from the analog feedback wire would be stored into a table that maps the angle of the servo to the feedback value. The table was stored in EEPROM, which allowed the data from calibration to persist even after the joint is powered off, so the joint did not need to be calibrated every time it was turned on.

When the joint is told to record the current position of the servo, the ESP8266 would read the analog value from the feedback wire of the servo. It then would use the data table recorded from the calibration action, and linearly interpolate the angle of the servo based on the data. This angle would be sent to the arm controller through ROS.

Upon receiving the command to move to the latest joint position, along with the desired angle from the arm controller, the ESP8266 would tell the servo motor to move to that desired angle. It would do so through a software "Servo" object that automatically manages the PWM signal to move the servo to the desired angle. When the joint was told to relax, it would detach the PWM pin on the ESP8266 from this "Servo" object, which would allow the servo shaft to be moved freely by the user.

Upon power-up, the arm joint would begin executing the program described above.

The 3D-Printed Casing

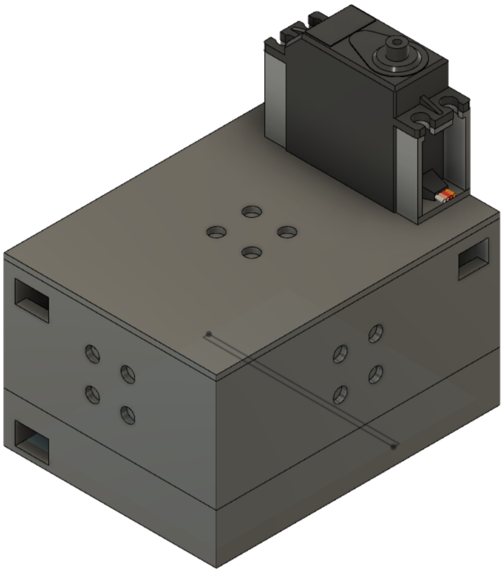


Figure 4. CAD model of the arm joint casing.

To package the arm joint concisely, a 3D-printed casing was designed and manufactured using polylactic acid (PLA). The casing was designed using a CAD software application called Fusion 360, and 3D-printed using the Prusa i3 MK3S+. The design is shown in Figure 4. There were 3 main sections of the casing which will be described.

The first section of the casing was the servo holder. This holder consisted of two brackets on either side of the servo motor, with a hollow interior to allow the servo wires to be strung through. Each bracket had two circular studs on the bottom, each 4mm in diameter, so that the brackets could be positioned and attached to the rest of the casing. The brackets were dependent on the servo and its dimensions, so if the servo motor were to change in the future, the brackets would likely need to be modified as well.

The second section of the casing was the electronics and breadboard compartment. This section, as the name implies, held all of the electronics needed to operate the joint except for the battery. This compartment included a lid 101.6mm (4 in.) in length and 76.2mm (3 in.) in width, with a raised rectangular section on the underside of the lid to align the lid with the rest of the compartment. The lid had 4 5mm-diameter holes on the north side for the servo holder to attach to. The lid also included a set of 4 5mm-diameter holes aligned in a square for connector attachments. The rest of the compartment was a 101.6mm by 76.2mm by 38.1mm (1.5 in.) open-faced box, with 4mm walls. The underside of the box also had a raised rectangular section as it formed the lid of the third section of the casing. On all sides of the box, except for the underside, there were a set of 4 5mm-diameter holes aligned in a square for connector attachments. On the north side of the east wall of the box near the servo was a rectangular hole to allow for the servo wires to connect to the rest of the electronics. Another rectangular hole on the west side of the south wall was present to allow the wires from the battery to connect to the rest of the electronics.

The last section of the casing was for the battery compartment. The lid of this compartment was built into the second section of the casing. Another open-faced box with similar dimensions to that from the second section was designed with a smaller height of 16mm. On the west side of the south wall, a rectangular hole was present to allow the wires from the battery to connect to the rest of the electronics. On the underside of the box was a set of 4 5mm-diameter holes aligned in a square for connector attachments, for a total of 6 possible places to attach a connector to the entire joint.

The three sections were designed to be hot-glued together to form the complete arm joint casing.

The Joint Connector

The joint connectors, like the casing, were designed using Fusion 360 and 3D-printed with the Prusa i3 MK3S+ using PLA. Four different connectors were designed. Each was a 25.4mm (1 in.) length by 25.4mm width rectangular prism with varying heights, and the top and bottom sides of the prism includes a set of 4 4mm-diameter studs aligned in a square to attach to the holes on the arm joint casing. The heights of the four designs were 50.8mm (2 in.), 101.6mm (4 in.), 152.4mm (6 in.), and 203.2mm (8 in.). On the 6-inch tall variant, a set of 4 4mm-diameter studs aligned in a square was also placed near the bottom for a more secure attachment to the servo. The joint connector design is shown in Figure 5.

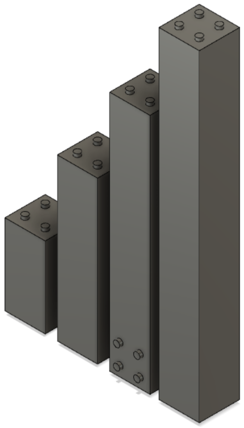


Figure 5. CAD models of arm connectors.

The ModRob Assembly

The complete robotic arm assembly designed for this project consisted of an arm controller and 3 arm joints in their casings with a joint connector attached to each servo. The servo shaft of each joint was screwed onto a servo horn. The servo horn was hot-glued to one end of the joint connector. The other end of the joint connector was hot-glued to a different arm joint's casing, allowing the arm joints to be chained together to complete the assembly.

Results

Both the customer discovery and market evaluation results, and the prototype results, will be reviewed below.

Customer Discovery and Market Evaluation

8 people have responded to the customer discovery survey. All of these people have been affiliated with the University of Michigan in some aspect. From these responses, two customer profiles were generated. The first is the high school student/teacher. This customer profile tends to work with simpler robotic arms where the technical details and capabilities are not as important. The second customer profile determined is the graduate/college robotics researcher. This customer profile is more interested in the technical details and capabilities of the robot, and will tend to have more experience in programming the robot.

After determining the customer profiles, as well as summarizing and quantifying the responses, pivot tables were generated using Microsoft Excel. The pivot tables generated are shown below.

Table 2. Money Spent vs. Customer Profiles Pivot Table

Count of MONEY SPENT - NEVER, <$100, $100 - $500, $500+	**Column Labels**				
Row Labels	**NEVER**	**<$100**	**$100-$500**	**$500+**	**Grand Total**
HIGH SCHOOL		1		1	2
GRADUATE COLLEGE ROBOTICS	1	1	1	3	6
Grand Total	**1**	**2**	**1**	**4**	**8**

Table 3. Willingness to Spend vs. Money Spent Pivot Table

Count of WILLINGNESS to SPEND - NEVER, LOW (<$500), HIGH (>=$500)	**Column Labels**			
Row Labels	**NEVER**	**LOW**	**HIGH**	**Grand Total**
NEVER			1	1
<$100		2		2
$100-$500			1	1
$500+	1		3	4
Grand Total	**1**	**2**	**5**	**8**

Table 4. Time Spent vs. Money Spent Pivot Table

Count of TIME - NEVER, HOURS, DAYS, WEEKS, MONTHS, YEARS	**Time Spent**					
Money Spent	**NEVER**	**HOURS**	**DAYS**	**WEEKS**	**MONTHS**	**Grand Total**
NEVER	1					1
<$100		1	1			2
$100-$500				1		1
$500+				3	1	4
Grand Total	**1**	**1**	**1**	**4**	**1**	**8**

Table 5. Willingness to Spend vs. Customer Profiles Pivot Table

Count of WILLINGNESS to SPEND - NEVER, LOW (<$500), HIGH (>=$500)	**Willigness to Spend**			
Archetypes	**NEVER**	**LOW**	**HIGH**	**Grand Total**
HIGH SCHOOL		1	1	2
GRADUATE COLLEGE ROBOTICS	1	1	4	6
Grand Total	**1**	**2**	**5**	**8**

In addition to the survey, a literature review was performed using 26 works from the University of Michigan Library database. Keywords searched include "modular robotics," "cheaper robots," and "democratization of robots". The review includes books, patents, articles, conference papers, and other types of works published from the 2010s stretching back to the mid-1980s. The complete review can be found in Appendix A.

Building the Prototype

The prototype was built and assembled according to the methods described above. The robot arm can be successfully calibrated, and its position can be recorded and moved upon a button press. An image of the prototype is shown in Figure 6.

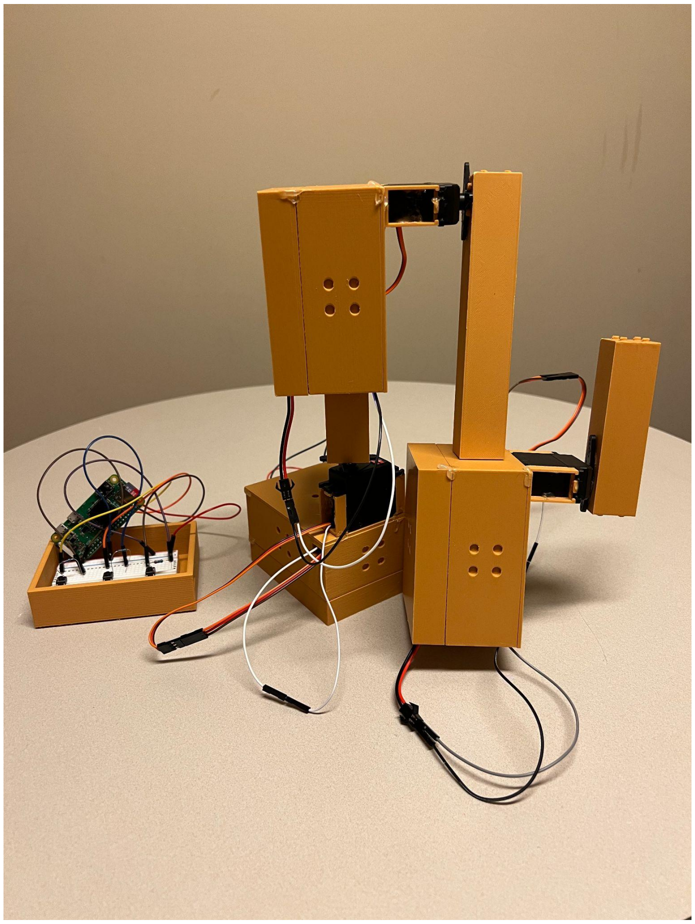


Figure 6. The complete ModRob prototype assembly.

Discussion

Both the results of the customer discovery and market evaluation section and the prototype section will be discussed below.

Customer Discovery and Market Evaluation

The customer discovery survey provided valuable insights into potential customer segments and their behaviors. As mentioned in the results section, two customer profiles were built: the first is the high school student/teacher, and the second is the college/graduate robotics researcher. One differentiator between the two groups is that the college/graduate robotics researcher tends to spend more time and money prototyping and building a robotic arm. This is likely because this group tends to pay more attention to technical details of the arm, which takes more time to determine, and also could lead to higher spending if, for example, they needed to use a servo with more torque. In addition, those who work with robotic arms on the college/graduate level seem to be willing to spend more money on an inexpensive robotic arm that would take less time to assemble and program. This contributes to a trend that the time spent and the money spent building and assembling a robotic arm are positively correlated.

The results of the survey can be used to inform a potential market strategy for addressing the market. Each controller, joint, and connector could be sold separately, as it would allow users to assemble a robotic arm customized to their needs. It would also allow for an easy and inexpensive replacement if a part breaks since the user does not need to repurchase an entire system or kit. Additionally, since it appears that early users or buyer fall into two distinct willingness to pay segments, one less than $100 that corresponds to the high school student/teacher profile, and the other greater than $500 that corresponds to the college/graduate robotics researcher, two different kits could be helpful to address the needs of both customer profiles. A less expensive and simpler kit (<$100 total price) can be marketed to the high school profile and a more expensive kit ($500+ total price) that allows for more advanced capabilities and finer control can be marketed to the college robotics researcher customer profile.

If marketing to any of the two customer segments, then the timing appears to revolve around a standard US two-semester bifurcated academic year, which starts in September and ends in April for Graduate/College Robotics Researchers and June for K-12. As a result, it would be key for expanded survey data collection and any user testing to take this timeline into account. The May to August timeframe should be used for wrapping up future prototype feedback surveys and finalizing any additional feature sets for prototypes, including reviewing the prototype through focus groups.

The literature review performed also yielded a valuable insight into the overall market potential for modular robots. It indicates that inexpensive, modular robots are an emerging technology given recent publication dates and have the potential to impact a wide variety of fields, including the educational and medical spaces.

Building the Prototype

The prototype arm has been able to demonstrate basic functionality of a robotic arm at low cost. Each arm joint costs ~$43, and the arm controller costs ~$18. A full arm assembly could cost less than $200, excluding packaging, shipping, and labor costs. These costs can be lowered significantly through replacing the breadboard with custom printed circuit boards, using injection molded cases instead of 3D-printed ones, and through high volume production that will take advantage of economies of scale. An example of preliminary unit economics for a kit targeted to the college/graduate robotics researcher is shown in Table 6.

Table 6. Preliminary Unit Economics of a Graduate Kit

Total Estimated Costs	$300
Arm Joint (x4)	$172
Arm Controller	$18
Arm Connectors	$5
Packaging	$25
Shipping	$30
Labor Costs	$50
Sales Price	$500-$1000
Gross Profit	$200-$700

The modular nature of the ModRob system allows the arm controllers, joints, and connectors to be used to build solutions for a wide variety of applications. Some fields where this product could make an impact include the medical field and the educational space. Additionally, the system could help expedite the robot prototyping process within the manufacturing space, and reduce the time to market for other robot arm solutions that may be built, lowering the average cost for a single arm solution.

Future Work

Both the customer discovery and market evaluation, and the prototype can be expanded on and improved in the future. They will be discussed in detail below.

Customer Discovery and Market Evaluation

Since the responses from the customer discovery survey come from those affiliated with the University of Michigan, future work can include expanding the customer discovery survey to more participants outside of the University of Michigan. This would help make the results from the survey more statistically significant and could lead to more customer profiles or provide additional insights into existing customer profiles.

Similarly, the literature review can also be expanded to validate current results. More sources from outside of the University of Michigan Library database and the use of more varied search terms can help provide additional insights into current market and robotics research trends. Expanding the robotic arms technology analysis would be helpful for assessing feature and price competitiveness of the ModRob system.

Another way to expand on customer validation is to demonstrate the prototype to a select focus group to gather feedback. This can help identify important features of the system that need to be included or improved upon, as well as unimportant features of the system that are not worthwhile to include. The focus group can also test the ease of assembling a system from the controllers, joints, and connectors, and test the time it takes to get a robotic arm system to work from these parts.

Further research is also needed to identify distribution channels for arm controllers, joints, connectors, and kits. An example would be to follow the Raspberry Pi's distribution model, which sells their products both through their website, as well as through electronics distributors such as AdaFruit.

Improving the Prototype

The prototype could be improved in many ways. One way is to improve the robustness of the prototype. This could be done by using an injection-molded casing and using custom printed circuit boards, which would reduce the costs of the product as well as give it a more professional look.

Another way to improve the prototype would be to replace the electronic components inside. The servo motor used is relatively cheap and small, and does not provide much torque or range of motion (currently 180˚). The servo motor can be replaced to provide additional features at a greater cost, and could potentially be suited for a more expensive robotic arm kit directed to the college/graduate robotics researcher customer segment. Both the ESP8266 and the Raspberry Pi Zero W could potentially be upgraded to increase computing power, or reduce power consumption. The battery and its related electronics could also be replaced to provide a longer battery life, shorter charge time, and better safety features in case of damage.

In addition to improving the hardware, the software can also be improved to provide optional greater control of the system as well as more features. Lots of possibilities exist in this space, and would be constrained by what the hardware allows. One possibility would be to provide customizable PID loops for finer control of the servo. Another would be to save entire arm motions and control what motions can be played at any time. In the future, the robot arm system may be able to connect to an app to provide an easy-to-use interface for control and simulation features. Lastly, a camera can also be used to allow the robot to perform both forward and inverse kinematics.

And last but not least, more components to the system could be designed, produced, and sold. Like improving the software, there are multiple possibilities here. For instance, a detachable base can stabilize a robotic arm system that uses more than a few joints. Multiple different end effectors can be designed for a multitude of applications. Furthermore, the parts do not need to be limited by usage in an arm system. The modular concept and the described business strategy can be applied to a vehicular-based robotic system such as a rover, for instance.

Conclusion

In contrast to existing applications-centric robotics solutions, the ModRob prototype has shown potential to provide a more general solution to the robotics industry at a substantially lower cost by leveraging its modular design. Through its potential applications, such as in the medical, educational, and manufacturing spaces, the average cost of a robotic arm system can be reduced. As the robotics industry is trending towards lower-cost robotic solutions, the ModRob system can aid in this trend while offering greater adaptability, greater customizability, easier use and assembly, and in the future, more advanced and complex features.

Acknowledgements

I want to thank my project advisor Prof. Grace Hsia for mentoring me throughout the project and providing valuable insights and connections for the project customer discovery and market evaluation. I also want to thank Rachel Armstrong-Ceron and Prof. Jason McCormick for reviewing and supporting my project proposal.

References

[1] K. Thayer, “What is the real cost of an industrial robot arm?,” Engineering360, 27-Apr-2017. [Online]. Available: https://insights.globalspec.com/article/4788/what-is-the-real-cost-of-an-industrial-robot-arm.
[2] https://modrobotics.com/cubelets-discovery-set/
[3] M. Cartagena, C. Bonilla, J. Menendez and M. Cardona, "Design and Development of a Modular Robotic Kit “CubeBot” for Educational Purposes," 2018 International Conference on Research in Intelligent and Computing in Engineering (RICE), 2018, pp. 1-7, doi: 10.1109/RICE.2018.8509071.
[4] https://www.trossenrobotics.com/pincherx-100-robot-arm.aspx
[5] https://www.trossenrobotics.com/dobot-mg400-industrial-desktop-robot
[6] https://www.wlkata.com/product/wlkata-mirobot-education-kit
[7] https://www.universal-robots.com/products/ur3-robot/

Appendix A: Literature Survey

Platform Searched	Keywords Searched	Type of Work	Year Published	Title	Author	URL	Summary	Notes
University of Michigan Library	modular robotics	Book	2021	Modular Robots: Theory and Practice	Guilin Yang, I-Ming Chen	https://link.springer.com/book/10.1007/978-981-16-5007-9	For high volume/low variety, custom robots are preferred. For low volume/high variety, general-purpose robots are preferred. For mid volume/mid variety, Modular Reconfigurable Robot Systems (MRSSs) can find usefulness. Hardware design (of the joints) is the most challenging aspect (and the most studied). Software design for modular robots are yet to be studied intensively.	Seems like a good advanced reading for implementation of modular robot systems.

University of Michigan Library	modular robotics	Thesis/Dissertation	2005	A framework for self-reconfiguration planning for unit-modular robots	Carl A. Nelson	https://www.proquest.com/docview/305389203?parentSessionId=vamnVBMnIQiEcJ%2BnS8s%2FB92L%2FRus4g9eLWDSr%2FX8UW0%3D&accountid=14667	Discusses overview of lattice-type robots, which are more well-studied, and chain-type robots, which are potentially more complex and less well studied. Uses graph theory to provide a self-reconfiguration algorithm. Does not appear to mention applications of such robots.	May be more applicable to the software side, or as a stretch goal. Unlikely to use self-reconfiguration in a reasonable time frame.
University of Michigan Library	modular robotics	Book	2015	Self-Sufficiency of an Autonomous Reconfigurable Modular Robotic Organism	Raja Humza Qadir	https://link.springer.com/book/10.1007/978-3-319-10289-4	Mentions robots used in specialized tasks, but lacking in the ability to adapt to varying conditions, such as in search and rescue, and exploration environments. Mentions that ideally greater autonomy would be involved, especially in swarm robotics applications. Focuses on self-sufficient, modular robotic systems (to	

							which it claims there has been no prior research from the self-sufficiency perspective) and looks to biological organisms as a basis for the work.	
University of Michigan Library	modular robotics	Patent	1997	Modular robot	Todd A. Ferrante	https://www.osti.gov/biblio/871220/	Describes a modular, walking robot using fluids to actuate the movement. Uses include exploration of environments. The patent also distinguishes wheeled robots vs. walking robots and mentions advantages and disadvantages of both systems.	
University of Michigan Library	modular robotics	Technical Article	1996	Modular robotics overview of the `state of the art`	R. L. Kress, J. F. Jansen, W. R. Hansel	https://www.osti.gov/biblio/403929/	Mentions applications of modular robotics for hazardous environments, such as nuclear waste sites. The ability to maintain and repair away from hazardous conditions as well as reeconfiguring the robot for different tasks are	

							drivers for modular robotics research. Focuses on kinematic issues as well as control algorithms.	
University of Michigan Library	modular robotics	Technical Article	2015	Design and Fabrication of an Elastomeric Unit for Soft Modular Robots in Minimally Invasive Surgery	Iris De Falco, Giada Gerboni, Matteo Cianchetti, Arianna Menciassi	https://www.jove.com/t/53118/design-fabrication-an-elastomeric-unit-for-soft-modular-robots	Details a soft robotic manipulator for use in minimally invasive surgeries.	
University of Michigan Library	modular robotics	Conference Paper	1994	A modular dexterous robot for glove box applications	James P. Karlen, Keith A. Kowalski, Paul H. Eismann	https://www.osti.gov/biblio/10191474/	Details a modular dextrous robot for scientific glove box applications, where isolation of a substance or material from the lab environment is needed.	
University of Michigan Library	modular robotics software	Conference Paper	2010	Modular Countermine Payload for Small Robots	Herman Herman, Doug Few, Roelof Versteeg, Jean-Sebastien Valois, Jeff McMahill, Michael Licitra, Edward Henciak	https://www.osti.gov/biblio/978367/	Describes a modular payload that can allow the robot to perform multiple tasks. Uses a payload controller unit (PCU) that interfaces with different sensors depending on the mission's needs. Depending on the mission, different payloads are attached.	

University of Michigan Library	modular robotics software	Book	2021	Software Engineering for Robotics	Ana Cavalcanti, Brijesh Dongol, Rob Hierons, Jon Timmis, Jim Woodcock (Editors)	https://link.springer.com/book/10.1007/978-3-030-66494-7	A compilation of papers from a conference. Includes papers on verifiability of autonomous robotics, software languages used to program robots, and discussions on regulation and ethics of robotic systems.	
University of Michigan Library	modular robotics software	Book	2007	Software Engineering for Experimental Robotics	Davide Brugali	https://link.springer.com/book/10.1007/978-3-540-68951-5	Mentions that most R&D in robotic software are based on proprietary architectures invented from scratch each time. They are developed to solve a specific class of problems that are not often reusable. Book discusses software engineering in view of robotic applications.	
University of Michigan Library	modular robotics software	Book	2020	Engineering Autonomous Vehicles and Robots: The DragonFly Modular-based Approach	Shaoshan Liu	https://onlinelibrary.wiley.com/doi/book/10.1002/9781119570516	Describes in depth a modular approach to land-based robotic vehicles.	Good for something like an exploration type of vehicle, but not as applicable for a robotic

								arm.
University of Michigan Library	modular robotics software	Thesis/Dissertation	2014	Software architecture and development for controlling a Hubo humanoid robot	Manas Paldhe	https://www.proquest.com/docview/1617518740?parentSessionId=WVBRQNbtDi4kZCkK6jS7uy%2F%2BYCUAbgollOljumSOe34%3D&accountid=14667	Describes a (series of) software packages to provide open-sourced, real-time operation for controlling a humanoid robot. Mentions the pitfalls of Robot Operating System (ROS) (uses TCP so no real-time operation), and Microsoft Robotics Developer Studio (MRDS) (not open-sourced, so cannot be optimized to hardware).	
University of Michigan Library	modular robotics software	Thesis/Dissertation	2015	Evaluating engineering learning and gender neutrality for the product design of a modular robotic kit	Ansh Verma	https://www.proquest.com/docview/1881321037	Describes the impact of gender-neutral robotic designs and its impact on children. They also describe a gesture-based user interface used to control the robots.	
University of Michigan Library	modular robotics software	Conference Paper	2002	Intelligent Control of Modular Robotic Welding Cell	H. B. Smartt, K. L. Kenney, C. R. Tolle	https://www.osti.gov/biblio/910640/	Describes the use of a modular robot in welding cells, where each element has local rules of behavior but no	Seems quite applicable to the goal of a modular robot with a user-friendly interface, but for specific applications (welding).

							overarching knowledge of the welding process. It also formulates a human/machine interface that treats the human as an element in the entire system.	
University of Michigan Library	democratization of robots	Article	2016	Intelligent Automation, Collaborative Robots Can Solve a Big Challenge	Patricia Moody	https://www.industryweek.com/technology-and-iiot/robotics/article/22008052/intelligent-automation-collaborative-robots-can-solve-a-big-challenge	Mentions current robotic applications, and the rise of collaborative robots and how it can make the workplace safer. Mentions that the democratization of robots could be possible. Mentions the possibility of robots sharing information across plants.	
University of Michigan Library	robot democratization	Technical Article	2014	How robots could assemble themselves a la origami - and not cost too much	Pete Spotts	https://www.proquest.com/docview/1551794708?parentSessionId=FffoOyE%2FGjeXVpRR1LKBEZ0%2FZ74GfV7In14QbRmamZ8%3D&pq-origsite=primo&accountid=14667	Describes robots that could be shipped flat and then auto-assembled for search-and-rescue operations. Price tag is from a few tens to a few hundreds of dollars (costs around	

							$100). Takes 4.5 minutes to assemble.	
University of Michigan Library	cheaper robots	Article	2015	Cheaper Robots, Fewer Workers	New York Times	https://www.proquest.com/docview/1712363649?pq-origsite=primo		
University of Michigan Library	cheaper robots	Article	2017	Cheaper robots	Latin Trade	https://go.gale.com/ps/i.do?p=AONE&u=umuser&id=GALE|A513759807&v=2.1&it=r		
University of Michigan Library	cheaper robots	Article	2013	Barron's: Cheaper Robots, Pricier Stocks	Jack Hough	https://www.proquest.com/docview/2102171164?pq-origsite=primo		
University of Michigan Library	cheaper robots	Technical Article	2004	Cost-adjusted surgical robotic performance: The plea for a cheaper robot	E. J. Hanly MD, S. L. Bachman MD, J. Zand MD, M. R. Marohn Do, M. A. Talamini MD	https://www.sciencedirect.com/science/article/pii/S0022480404003610		
University of Michigan Library	cheaper robots	Technical Article	2018	Is the pace of technology development a threat or opportunity for sustainability? The case of remanufactured industrial robots	Lisa Melander, Sofia Lingegård	https://www.sciencedirect.com/science/article/pii/S2212827118304943		
University of Michigan Library	cheaper robots	Article	1985	You Could Be Replaced by Smarter, Cheaper Robot: [Home Edition]	Edward Cornish	https://www.proquest.com/docview/292018089?pq-origsite=primo		
University of Michigan Library	cheaper robots	Thesis/Dissertation	2009	A reconfigurable cooperative control system for rapid deployment of multi-robot systems	Stephen Sodokan Nestinger	https://www.proquest.com/docview/304856904?pq-origsite=primo		
University of Michigan Library	cheaper robots	Technical Article	2021	Aspects of Industrial Applications of Collaborative Robots	Peter Zentay, Lajos Kutrovacz, Mark Ottlakan, Tibor Szalay	https://link.springer.com/chapter/10.1007/978-3-030-88458-1_1		
University of Michigan Library	cheaper robots	Article	2019	Robots and International Economic Development	Robert D. Atkinson	https://go.gale.com/ps/i.do?p=AONE&u=umuser&id=GALE|A		

						639986287&v=2.1&it=r		
University of Michigan Library	cheaper robots	Technical Article	2020	U.S. Robots and their Impacts in the Tropics: Evidence from Colombian Labor Markets	Adriana D. Kugler, Maurice Kugler, Laura Ripani, Rodimiro Rodrigo	https://www.nber.org/papers/w28034		