# Final Project: 3 Arm Army

Ryan Craig
Dan Nguyen
Rachel Platt
Alex Urbanski

## Idea

The idea for our project is to assemble and control a mechanical arm to be an assistant for small tasks that require a hand which someone may not be able to do or someone may want an easier time doing. An example of this is eating and stirring a bowl of cereal or soup. Using the adjustable arm you could make changes between actions on the fly with the Microbit attached to any of the user's limbs.

## User Sensor

The training of the Microbit would be done on the computer. There will be instructions in the GUI on the placement of the Microbit that will instruct the user on how to wear the Microbit. We will then have an application that allows users to see all the motions or gestures the arm could perform. When a user wants to train a gesture that they will perform they can press record and the gesture will be trained and linked to the arm motion. Right now we are looking at the possibility of using the accelerometer. A minimum of 5 examples for the gesture will be asked for but the user will be encouraged to train more as the motion will become more accurate.

## Data Extraction and Processing

We will be using a python script that we wrote to extract the data from the Microbit before sending it to to another python program that will read the data. We are going to use the GUI to grab the accelerometer data from the Microbit when we are recording actions.

Once we have a hold of our data we will be using python scripts with the DTW library so that we can classify different actions that the user is making using dynamic time warping. When the user is making actions we will check a certain time frame of values against the different classes that the user has recorded and see if the action qualifies as any of the classes.

## Arm

The arm will be controlled by a Raspberry Pi. Currently we have a 4-degrees of motion robotic arm. With our current setup we are using a Raspberry Pi to control the the motor drivers. The motor drivers take in a Pulse Width Modulation (PWM) wave for each motor to control it.

On the Pi there is a custom program to drive the arm. It takes OSC messages and make a motion happen either linearly or exponentially as it keeps receiving the same command. This allows us to have more control at the beginning of the motion incase a user wants to move the arm slightly. Allowing it to move exponentially will allow the hand to move more quickly as the user is sending the same gesture probably meaning they want to make a significant change in position.

## Tying It Together

To tie the project together we are using OSC messages between each application. The python Microbit scripts deals with getting the values from the band and allows you to decide what port and address to send the data to. That data is going into another python script to see if the value matches any of the user defined classes. Once we classify the actions using python and the dtw library, we can send the correct OSC signal to the arm. This is where the Pi will process the message and finally send the correct PWM to the motor controllers making the arm move in the desired fashion.

## Feasibility

The project was feasible because the materials and coding aspects were easy to break down into smaller chunks. Most parts could be worked on individually before being tied back together to complete the project. We believe the hardest parts of this project was getting the hardware to work. We started this part early and had plenty of resources to get this to work. We have a few electrical engineering friends that were more than willing to help us push through the hurdles. Also given we used Raspberry Pi and SparkFun parts, there was a massive community that had probably experienced our issues before.

## Meeting Schedule

Currently we have two meeting times. During class on Mondays 1:00 to 3:30, and Sundays 2:00 till 4:00. In these meetings we usually decide what needs to be done individual or in a pair as all four of our schedules don't match up perfectly. These are the stand up times that we use to get on the same page then move work time.

## Technologies We Will Use

For the hardware we simplified our scale down by using an existing claw. This claw has 5 degrees of motion but currently we can only access 4 of them. This is because we were given a modified version of this claw and the motors need a motor driver to control the speed and direction. We have two of these motor drivers which will allow us to run 4 total motors. If we decide we want the 5th motor we would need to purchase another motor driver.

For language tools, Python will be our main language. Given we are using a Raspberry Pi and a regular computer we will be able to quickly and effectively create software to communicate together using OSC messages.

We will also be using the Micro Bits for the user input. With the use of 2 microbits one will be attached to the user and the other will be connected to the computer. They will communicate together using the radio frequency to send the accelerometer data to Wekinator.

We will also be using Github which will store our open source software and the documentation on how to set up the closed source elements that were mentioned above.

## Risks to failure

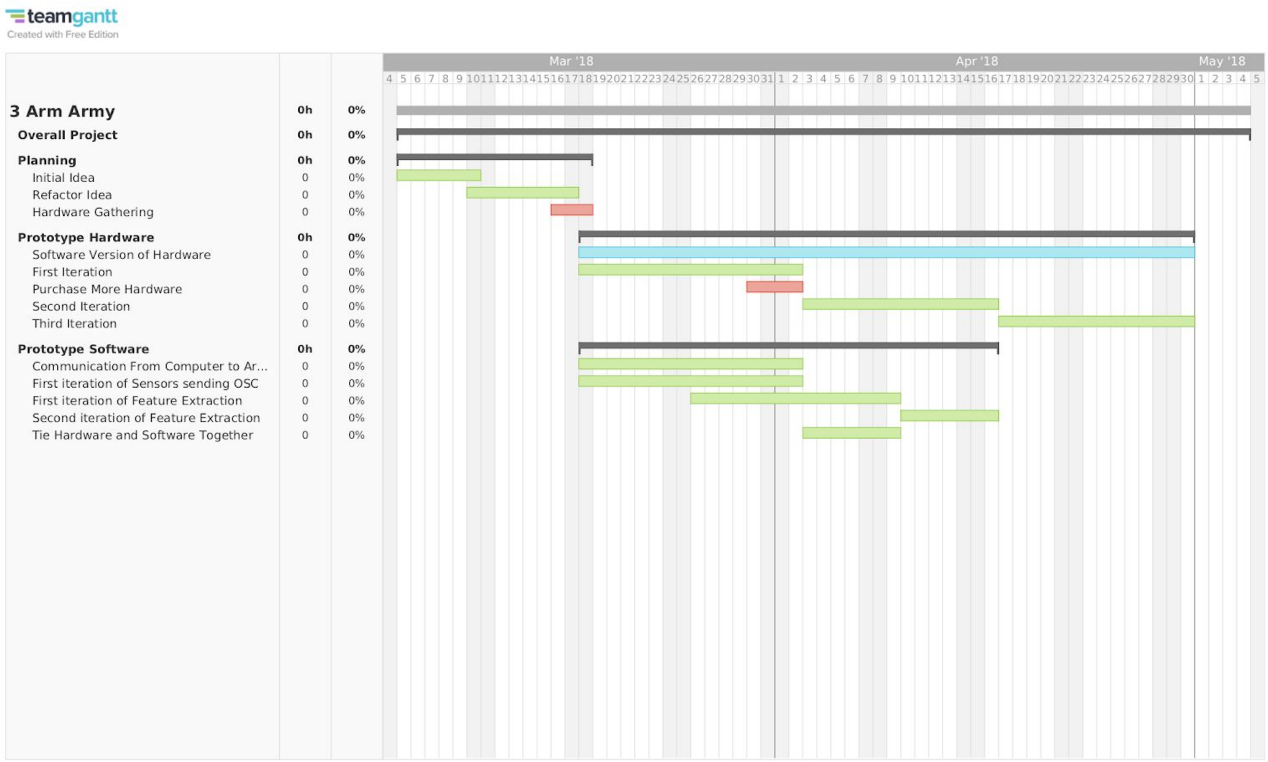
One risk in relation to hardware, if the hardware loses control and the user is not able to regain control quickly we need to have some sort of kill switch to shut the power off to the arm to prevent any major injury. To mitigate this risk we will be adding in a switch from the power to the arm. Once this switch is toggled the arm will receive no power disabling it from moving before the problem is fixed. Another risk to failure is having the arm not be able to pick up certain objects. Because of this, we will be incorporating necessary components for the user tasks into our project.

## Ethical Question

One ethical question about this project is that given the nature of creating a tool to help people complete everyday tasks, there is high expectation for reliability and accuracy. If our project is labeled as an accessibility tool, then it's ability to actually help a user complete a task needs to be taken very seriously-- as a malfunction could lead to a user to not be able to complete a task, or worse, damage to their property or an injury. Because of this, we need to make sure that our project has a well defined use scope and performs as intended.

## The Timeline

Our current schedule is listed below but is bound to change. The live update can be found here.



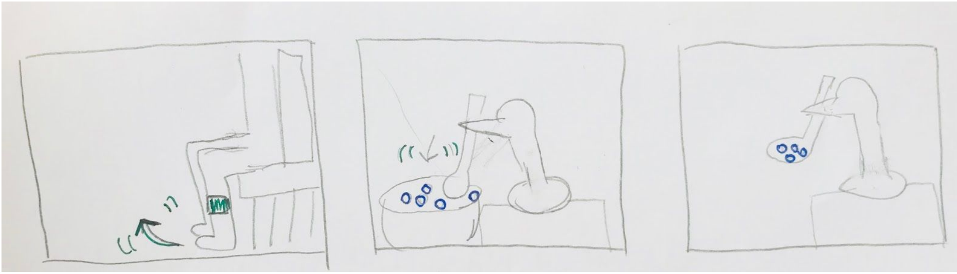

# User Stories

Training the System

1. The user selects a motion they want to train a gesture for
    a. User selects “Open / Close”
2. I use my Microbit and do one of the pre made gestures (double tap, clench, hand expand, hand to the left, hand to the right)
    a. I Clench my fist
3. Now when I clench my fist the claw opens / closes

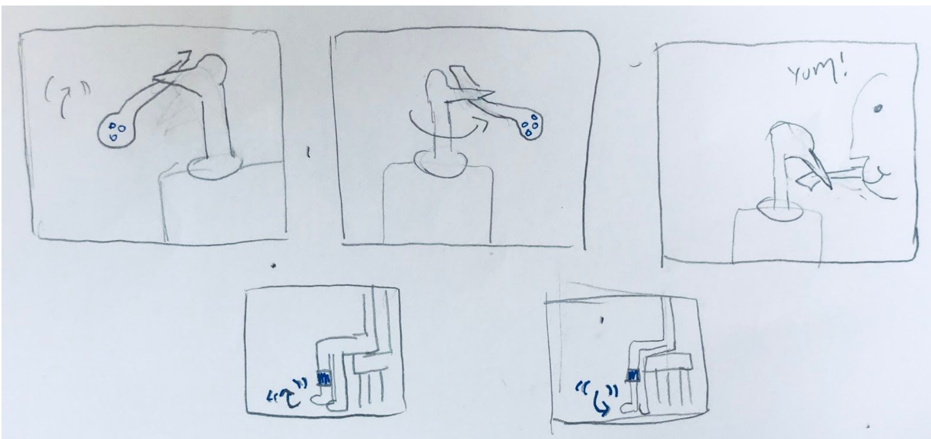

Obtaining a spoonful of cereal

1. User has already trained the gestures of the arm
2. User uses the pretrained gesture to move the spoon into the cereal and get a spoonful



Bringing cereal to mouth

3. User uses pretrained gesture to bring the cereal closer to their mouth
4. Once fairly close, user uses joystick controls for fine tuning




Screwing things in that require light

5. I train the system using the application by selecting “shine light” (The claw has a built in light) using the expand fingers gesture
6. I am screwing something in where I need someone to hold a flashlight so I can see
7. I spread my fingers and light shines from the mouth of the claw

Holding a circuit for soldering

1. I need to solder my circuit board but it keeps moving
2. I open my claw by clenching my right fist
    a. The claw opens
    b. I put the circuit board in
3. I clench my right fist again
    a. The claw closes
4. I can use both my hands to solder the circuit board

# Hardware

| Item | Quantity | Price | Notes |
|---|---|---|---|
| Claw | 1 | $40.00 | Link to Claw<br>We have access to a claw already. |
| Motor Drivers | 2 | $5 each | Link to Motor Driver<br>We have access to two. We need 3 if we want to drive all 5 motors. |
| Raspberry Pi 2 Model B | 1 | $35 | Link to Pi<br>Have access to this |
| Breakout Expansion | 1 | $11 | Link to Breakout Expansion |

## Challenges to Overcome

As we have just started developing and looking at hardware we face a challenge of having an Arduino create custom set motions for the robotic arm real time. At first we just want to be able to train custom gestures to set arm motions like stirring a pot or adding salt to a dish. This is reasonable but the later of having custom arm motions would require more work. We decided to go with Arduino because we already have a mechanical arm that is controlled by an Arduino.

Incase something goes wrong with the claw we need to have a kill switch. This will be added the the electrical component  but will need to be assets were the best location is for this switch. This will need to be decided once we have the hardware and electrical components to decide what would be best for the user.

Not everyone in our group has coded for an Arduino before so this may present a challenge since we need to control the robot arm. The robot arm that we have is already connected to an Arduino so that eases the burden, but we will still have to figure out how to program the arm to do set actions and receive OSC messages. Our first idea of this is to have direct serial communication to a computer. The computer would process the OSC messages then forward them over the serial line to the Arduino. This would be in the first iteration of the hardware, looking forward we would try and get a wireless communication for the second and third hardware iteration.

## Project Uses and Training

The arm should be able to perform simple tasks for the user so that they can use it for trivial tasks while they use their other arms for more complicated tasks. Some of the ideas that we came up with for using the third arm include brushing your teeth. The arm could also be used to stir a soup or a pot of noodles. In addition it could be used to feed the user food, or hold the user's vape close to their mouth. Most of these uses would use a sort of claw grip to function, but we thought of the possibility of adding other attachments that could be used in place of a grip. We thought that you could use an attachment that would have some sort of rubber like a stylus. Then you could use the arm to swipe through your Tinder or scroll through social media feeds. You could also have something like a brush attachment so that you could pet animals with the arm.

To train the arm we would most likely have a set of tasks that the user could select from in a UI. After the user selects the task that they want to complete we would have a recording system where the user could record themselves performing a gesture with an accelerometer like a microbit. We would then map the user's gesture to the selected action. This seems like the most realistic starting point. If we accomplish this system, we could possibly add a feature where the user is able to create custom tasks by moving the accelerometer and having a 1 to 1 mapping of the arm.

# Future Work

Currently we are having issue with having the set motions to work on the PI. Given our motors didn't have encoders on them we needed to create our own relative location logic. Moving forward we would like to debug and improve the default motions code. The game logic loop is set up and working its just location updating code isn't working.

# Video Demonstration

## Bibliography


Edwards, A L, et al. “Application of Real-Time Machine Learning to Myoelectric Prosthesis Control: A Case Series in Adaptive Switching.” *Prosthetics and Orthotics International.*, U.S. National Library of Medicine, Oct. 2016, www.ncbi.nlm.nih.gov/pubmed/26423106.

Edwards, Ann L. “Application of Real-Time Machine Learning to Myoelectric Prosthesis Control: A Case Series in Adaptive Switching.” *Application of Real-Time Machine Learning to Myoelectric Prosthesis Control: A Case Series in Adaptive Switching*, 30 Sept. 2015, journals.sagepub.com/doi/abs/10.1177/0309364615605373.

“Machine Learning For Hand Prosthetics .” Powerpoint that we found. Lots of technical information. They have a website on the slides http://www.neurobotics.org/

Molteni, Megan. “Want a True Bionic Limb? Good Luck Without Machine Learning.” *Wired*, Conde Nast, 3 May 2017, www.wired.com/2017/05/want-true-bionic-limb-good-luck-without-machine-learning/.

Scully, Leah. “Could Machine Learning Seal the Deal for Lifelike Bionic Limbs?” *Machine Design*, 12 May 2017, www.machinedesign.com/motion-control/could-machine-learning-seal-deal-lifelike-bionic-limbs.