# Robot Arm

Website: <https://github.com/voltAG3D/CENG-317--Daniel>

Names: Daniel Shelepinsky, Ali Khaliq and Matthew Gelfand

Course: Computer Engineering Technology

Date: April 23rd 2018

# Declaration of Joint Authorship

The robotic arm project consist of three group members (Daniel Shelepinsky, Ali Khaliq and Matthew Gelfand) and is a joint effort on the completion of the project. The work for the project has been divided equally among the group members. Ali Khaliq and Matthew Gelfand has worked on the hardware aspect of the project; which includes connecting an Arduino with a Raspberry Pi, controlling the servo motors. Daniel Shelepinsky has worked on the design and functionality of the mobile application. Daniel Shelepinsky has also worked on the database in terms of setting it up, connecting it with the app and maintaining it. Testing and maintenance for the hardware and software has been tested by all three members equally and all project updates and changes have been checked and approved by all members.

# Approved Proposal

September 2017 and reapproved January 2018

Prepared by Daniel Shelepinsky, Ali Khaliq and Matthew Galfand

## Executive Summary

As a student in the Computer Engineering Technology program, I will be integrating the knowledge and skills I have learned from our program into this Internet of Things themed capstone project. This proposal requests the approval to build the software app that will connect to a hardware as well as to a mobile device application. In the app, we will have incorporated a controller that will be used to control the hardware. The database will store the coordinates of the robot arm and the angles that the motor and arms need to be when it returns to rest position. The mobile device functionality will include some very basic test functions and commands to move the arm. It will create logs and save that data, so the developer can later look it at and make improvements to the hardware or software and other users can use these logs to reproduce experiments. In the winter semester I plan to form a group with the following students (Daniel Shelepinsky, Ali Khaliq and Matthew Galfand), who are in a different course. The hardware will be completed in CENG 317 Hardware Production Techniques independently and the application will be completed in CENG 319 Software Project. These will be integrated together in the subsequent term in CENG 355 Computer Systems Project as a member of a 3 student group.

# Abstract

The purpose of the Robot Arm project is to create a robot arm that has many purposes but for the main purpose would be an assembly line or sorting. The main components are servo motors, color sensor, mobile app, and the robotic arm will all be explained in great detail. The arm itself will be controlled by four analog feedback servo motors. One at the base of the arm to control left and right in a 180 ° Than two more for the arm itself to extend it and retract. The last motor controls the grip which will open and close it. Than all servo motors will report back to the database their positions. Which the user can edit through the mobile application. The arm itself is made out of acrylic parts. It will be connected to a Raspberry Pi 3 and an Arduino. The Pi being handling all software aspects of the project and the Arduino handling the PWM. The main focus of this technical report is to give the users a full understanding of the project and how it came to be.

# Table of contents

Declaration of Joint Authorship

Proposal

Abstract

Illustrations or Diagrams

1. Introduction
2. Project Description
   1. Requirements specifications
      * 2.1.1 Mobile Application Concept
      * 2.1.2 Application Screen
      * 2.1.3 Hardware Interfaces
      * 2.1.4 Software Interfaces
      * 2.1.5 Communication Interfaces
   2. Build Instructions
      * 2.2.1 Introduction
      * 2.2.2 Bill of Materials/Budget
      * 2.2.3 Time Commitment
      * 2.2.4 Mechanical Assembly
   3. Specific Requirements
      * 2.3.1 Database
      * 2.3.2 Application
      * 2.3.3 Hardware
   4. Progress Reports
   5. Problem Encountered
3. Conclusion
4. Appendix
5. Reference

# Introduction

Our hands can only do the same task over and over without stopping to take a break. A robot arm when programmed and built correctly can fix this issue and allow the user to repeat the same task over and over and does not need to stop for breaks like a human arm does. If any injuries were to happen on the job. It would happen to the robot arm and not the human/user which would mean reduced work place injuries where the arm can be applied to. The arm itself is cheap and easy to setup and replace parts if the arm does break down. This would mean production in a company would not fall as much as if a human would get injured. The arm can be automated or controlled from a distance with a mobile application which would again leave the user free of harm.

The Arm consists of four motors and a color sensor. The four motors will control the arms and movement going from left to right and extension/retraction. The color sensor is used to detect objects by color and can be programmed to ignore certain objects or pick them up and sort them depending on what the user wishes the arm to do. If the user does not like to use a mobile application or automation they can use buttons/controller that will move the arm if they so prefer.

# Project Description 2

## Requirements specifications 2.1

### Mobile application Concept 2.1.1

The Concept for the mobile application is to control the Robot arm from afar with either a phone, iPad, computer etc. The application will allow the user to move and report positions of the arm and possibly debugging if the arm ever gets in trouble/stuck. The application also includes a database which stores information about the arm and positions/ presets the user sets for it. The database currently in use is SQL may switch to firebase.

### Applications Screen 2.1.2

Connection Screen: This screen will include two types of connection but towards the end we will decide on one type. So far, the user can select either connection via Wi-Fi. The pi itself is a server which will give the user a port to be able to connect too. Another way is Bluetooth.

Control Screen: This Screen includes 4 bars which will allow the user to control each motor of the arm. Once moved the application will also report back showing what position they are in and report it to the database.

## Hardware Interfaces 2.1.3

## Software Interfaces 2.1.4

# Build Instructions 2.2

## Introduction

The robot arm can be used for many things as long as it is programmed and built accordingly. Our arm will be used for basic use and testing. Meaning that it will be able to move with 4 motors pick up and drop items and sort items through color. The Arm will include a phone application to control the arm and store information into a database if the user so wishes to have.

## Budget

First you will need a raspberry pi or something to be able to control the motors.

* <https://www.amazon.ca/gp/product/B01CCF6V3A/ref=oh_aui_detailpage_o02_s00?ie=UTF8&psc=1> This includes everything that will be needed for the arm to work GPIO pins, WiFi, Bluetooth etc. This goes for $99 CAD

Next on the list is the motors analog feedback micro servos bought from adafruit

* <https://www.adafruit.com/product/1449> These go for $10 CAD a piece which is $40 CAD. 4 wires that come with these motors. 1 white for the feedback and the usually power, ground and signal.

Here you have a choice of either getting the arm chassis laser cut or buy it. I got it laser cut via the Humber prototype lab but in case you wish to buy it you can from

* <https://shop.mime.co.uk/> Which is around 50-70 $CAD

## Time Commitment

* Buying the materials and awaiting for them to arrive will take about 1 week – 2 weeks depending on where you order I would suggest amazon and ad fruit as they are pretty quick with their orders.
* Assembly is pretty easy when steps are followed: <http://www.instructables.com/id/Pocket-Sized-Robot-Arm-meArm-V04/>. The parts are easy to break but are easy to replace as well as long as you have the AutoCad files. 1 hour – 3 hours
* Wiring can be hard as the raspberry pi has a floating voltage which can do damage to your motors if you are not careful or if you have another power source. Before hooking up the motors to the arm be sure to test the motors and see if they work. 1 hour - 2 hours
* Coding is the hardest part out of the project as it is the backbone. It connects the arm to the pi and the database as well as the mobile application which overall will move the arm.

## Mechanical Assembly

Once you have all your parts you may proceed. Side note, always double check your wiring.

Step One: confirm which GPIO pins you wish to use. I used 12,13,18,19 for PWM and 2 GPIO pins for each motor: 16, 20; 21, 5; 6, 26 and 23, 24.

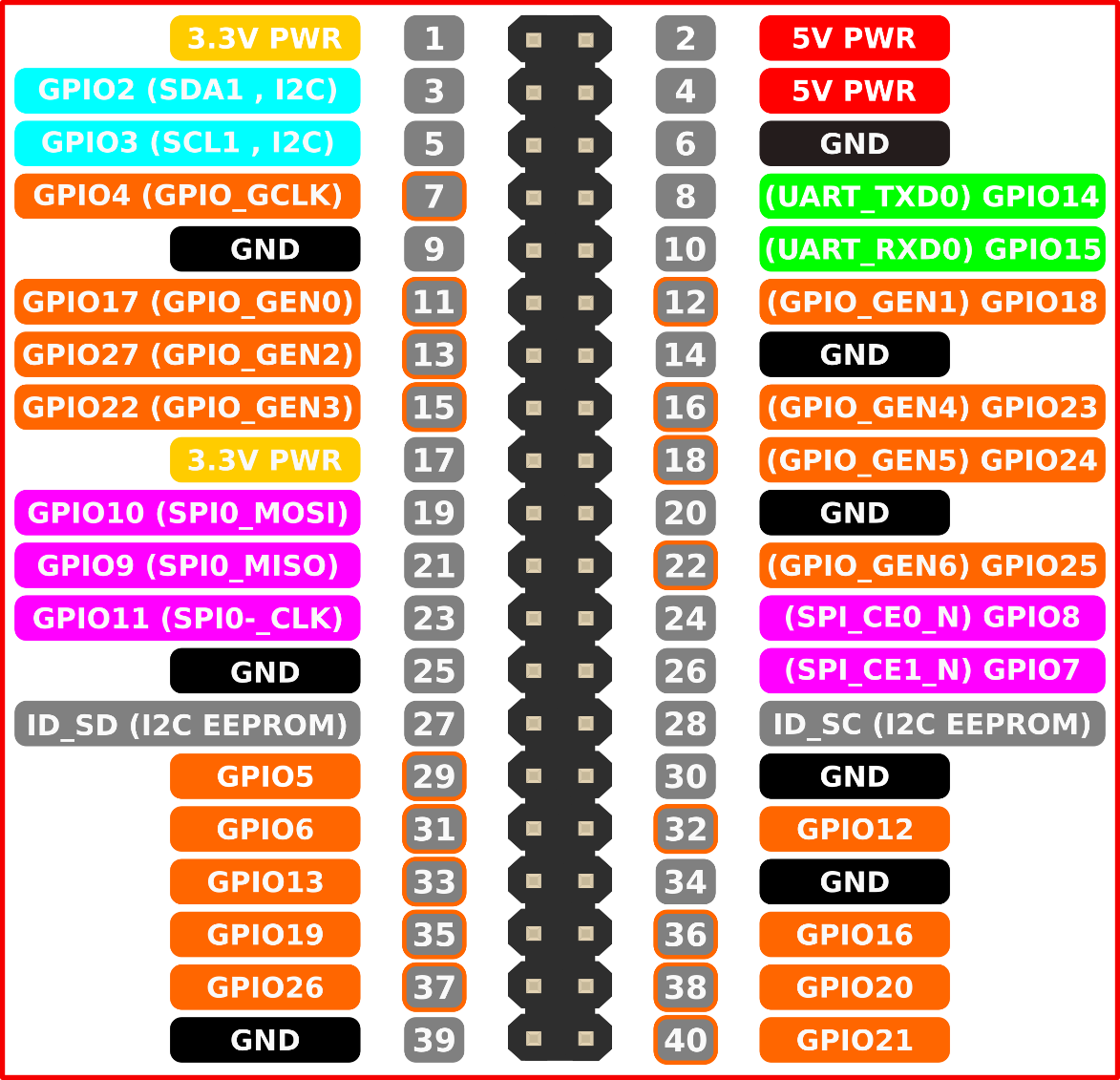
Step Two: Once you have your pins selected connect them to your motors. Then test your motors by running a simple script with either C or Python. The Code I used for the arm is here <https://pastebin.com/nCWX3j4F>. I also suggest adding LEDS to show that they are getting power in case. Before going to step 3 really make sure every motor you have works.

Step Three: Build the Arm using the guide given by <http://www.instructables.com/id/Pocket-Sized-Robot-Arm-meArm-V04/>.

Step Four: The arm is ready to be used mechanically.

## basicImage dump

Basic way to connect a motor to a raspberry pi.

GPIO pins used by a raspberry pi 3. Some will have to be set through code.

# Project Breakdown 2.3

## Database 2.3.1

The Database will be firebase. The reason behind this is because it is free and a very friendly database for newcomers to work with. It only has 1 table which includes and will store the following information: Claw motor, elbow motor, arm motor and base motor.

## Application 2.3.2

The overall concept of the mobile application for the project is to control the four servo motors of the robotic arm and keep logs on the user’s actions. The first screen of the application will be the login screen which will require an email and password to create and login to an account. The accounts are used to hold individual logs of the user’s controller usage. When logged in, the user is presented with two option; controller and user logs.

The first option of the application is to control the motor with four sliders that will control the arms movements. When the application first begins, the sliders will be set to a default (centered) position. When exiting the application, the robotic arm will return to its default position after all created movements.

During the controller screen, the application will have logs of the user’s usage and will be presented in the user log screen. The user logs will contain the log of all the users usage of the application based on the users in the database. In conclusion, the login screen, controller screen, and user logs are the three key features of the application.

## Hardware 2.3.3

Robotic Arm: The arm was created by ME industries. I have taken that design and added a database along with a phone application and switched out the motors for analog feedback servos.

Analog feedback servos: These servos are just like any other but have another wire that is wire which reports back on the position it is in with analog.

Raspberry pi3: This is the main component as it will be sending the signals to the arm in order to move it.

Wires, Resistors, Buttons: You will need wires in order to connect the arm together and have it work from the pi. As it cannot send signals without these. The buttons are optional if the user wishes to control from a controller.

# Progress Reports 2.4:

### Progress Report 1:

Hello Sir,

Over the pass weeks not much has actually happened in regards to work besides writing up the report and coming up with ideas for what to do.

- We have gotten a new arm. ( laser cut instead of 3d printed)

- finished the starting points of the report.

- Come up with ideas to make it unique.

- Working on Hardware build instructions

Problem/Question: If possible can i still switch projects/Sensors? The sensor to  which we would switch to is a TCS3200 color sensor. Our reasons to switching to this is because there have been many robot arms done in the past and we would like something more unique.

Financial: We have bought a new Sensor which is the TCS3200, Screws and bolts and an Arduino.

Regards,

Daniel Shelepinsky

N01019434

### Progress Report 2:

Dear Kristian

The integration between our groups is going along fine so far as the application, database and pi can talk to each other to an extent. The next step(s) is to get the application to control the motors themselves, redo the wiring of the arm, possibly make a PCB and or get a box-base to hold it all under so it isn’t in view. There has been a minor setback and that is that we tightened the motor to the claw too much and it broke as a result but that has already been replaced we just need to attach the new claw.

Regards,

Daniel Shelepinsky

# Problems encountered 2.5

During the creation of the arm we had to go through a great many tests to ensure things have worked. Such as motors now these gave us a few problems because we thought that micro servos were the same as normal servos and fed it the same voltage as regular servos which burnt them out. Which set us behind as we had no backup’s motors to fill the slots. After testing the new motors we have learned there limits and how to use them with code. Which took a bit of time as we are new to working with raspberry coding and pins but once we learned them we could progress further. Now further down the line we have encountered problems of both the hardware and software side always. The software was a harder issue as we have not done connections with Bluetooth before and had to learn some key basics as you have to allow permissions in your manifest files before you can proceed. Android studio does not allow the simulator to have Bluetooth which made it even harder to test. So we set up the raspberry as a server which gave out a port that would allow the user from the phone to connect.

## Conclusion 3

# Appendices 4

# References 5.

1. (2017, October 14). Pocket Sized Robot Arm MeArm V0.4. Retrieved March 26, 2018, from <http://www.instructables.com/id/Pocket-Sized-Robot-Arm-meArm-V04/>