Technological University Dublin
Blanchardstown

School of Informatics and Engineering

ROBOT ARM

by

Ethan Lawlor

A report submitted in partial fulfillment of the
requirements for:

ORDINARY BACHELOR OF ENGINEERING DEGREE
IN MECHATRONICS

SUPERVISOR: MARY LOOBY
SUBMISSION DATE: 28/04/2024

ABSTRACT

This report details the design and implementation of a versatile desktop robot arm, part of the Mechatronics program at Technological University Dublin. Aimed at enhancing efficiency in small-scale automation, the project addresses challenges in task diversification, user-friendly customization, and seamless integration of critical components like servos, Arduino Uno, and 3D-printed parts.

The introduction highlights the project's goal, emphasizing the need for improved efficiency in desktop environments. The background section discusses limitations of existing solutions and introduces the proposed robot arm. Scope and objectives outline a focus on task versatility, user-friendly customization, and an educational component.

The literature review explores motor choices, motion algorithms, control systems, and Bluetooth integration. The project design chapter covers mechanical specifications, electrical architecture, software control flow, and critical success factors.

Materials and methods provide an overview of implementation, including software and hardware prototyping, experimentation, and future plans for semester 2. Results and discussion sections present data and interpretations, addressing project objectives. Conclusions and recommendations synthesize the project's significance, evaluate success, and suggest future directions.

The report includes a bibliography and appendices containing project planning, extracts from the project diary, design drawings, software code, relevant standards, and additional technical appendices. Overall, the report showcases the desktop robot arm's evolution, offering insights into small-scale automation and robotics for readers interested in mechatronics.

DECLARATION

The work submitted in this report is the results of the candidate's own investigations and has not been submitted for any other award. Where use has been made of the work of other people it has been fully acknowledged and referenced.

Student Name

Ethan Lawlor

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1 Introduction

1.1 Problem: Enhancing Accessibility in Small-Scale Automation

Addressing the need for improved efficiency in small-scale automation within desktop environments, this chapter introduces the project's goal: designing a versatile desktop robot arm to cater to diverse user needs.

Key Challenges:

Task Diversification: Meeting the challenge of handling diverse tasks with precision, from soldering to small cleaning operations in desktop environments.

User-Friendly Customization: Overcoming the lack of user-friendly customization in existing solutions to allow users, regardless of technical expertise, to adapt the robot arm to unique requirements.

Integration of Components: Ensuring seamless communication and functionality among critical components like servos, Arduino Uno, and 3D-printed parts to optimize the robot arm's performance.

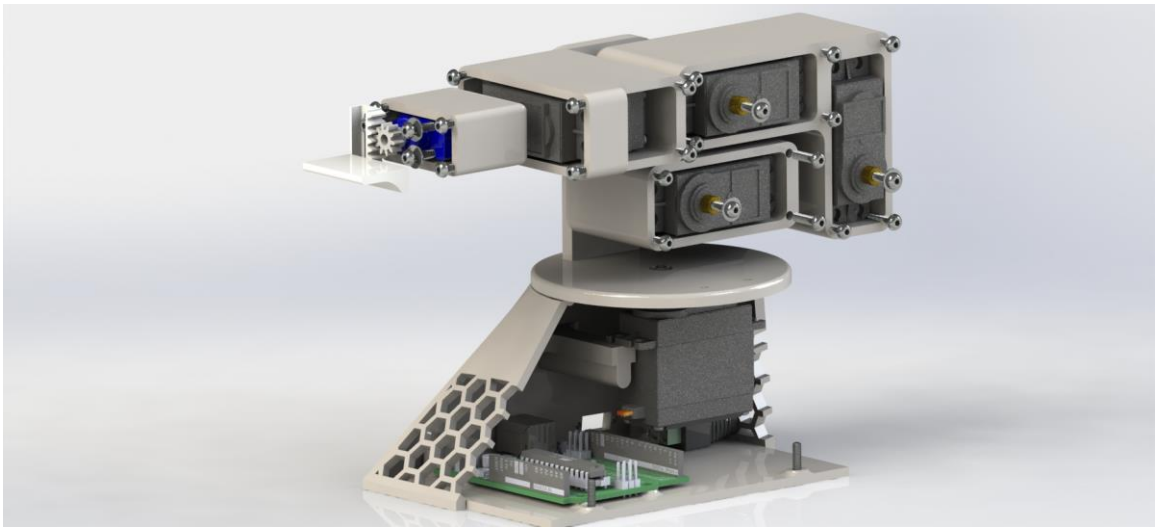


Figure 1.Conceptual CAD drawing showcasing the integration of servos, Arduino Uno, and 3D-printed parts in the proposed desktop robot arm.

Solution Approach:

Visual representation of the proposed solution, illustrating the integration of servos for controlled movements, Arduino Uno for automation programming, and 3D-printed parts for cost-effective and customizable mechanical components.

Envisioning an accessible desktop robot arm, this project not only addresses challenges but also serves as a learning platform, demystifying complexities for individuals regardless of their automation background.

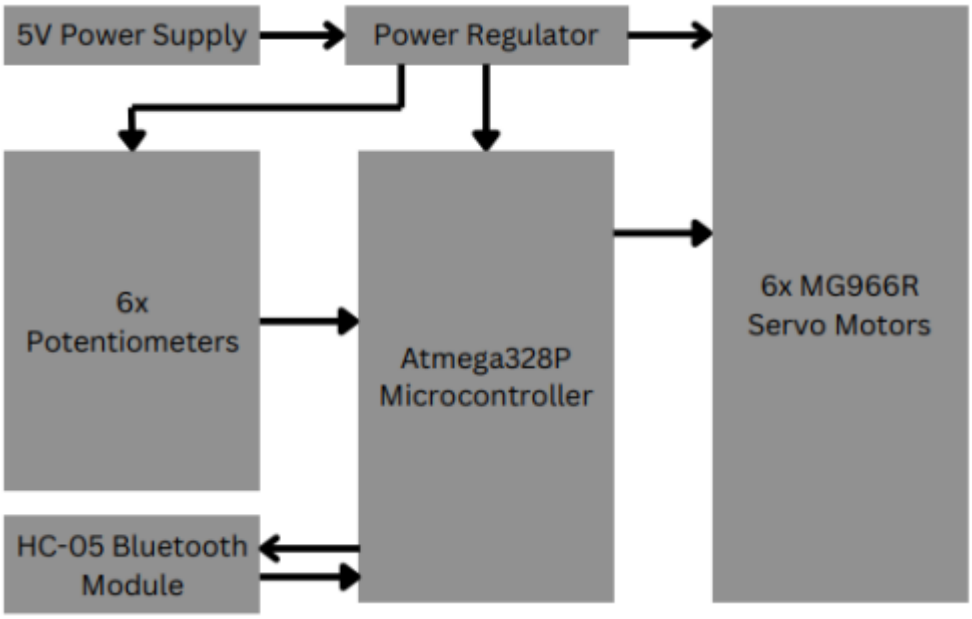


Figure 2.Overall block diagram illustrating the integrated components and workflow of the desktop robot arm project.

1.2 Background

While automation technologies have evolved, applying robotic assistance in desktop environments lacks innovation. Existing systems tailored for industrial settings fall short in versatility for small-scale tasks. The project introduces a desktop robot arm to bridge this gap, designed to handle diverse tasks efficiently.

Automation in Desktop Environments:

Covering a spectrum of tasks from intricate soldering to simple cleaning operations, existing solutions lack adaptability. The desktop robot arm integrates servos, Arduino Uno, and 3D-printed parts to ensure a harmonious collaboration for seamless task execution.

Challenges in Small-Scale Automation:

Small-scale automation demands precision, adaptability, and user-friendly customization. This project provides a solution enhancing efficiency in desktop tasks while empowering users to customize the robot arm according to specific requirements.

Related Work and Research:

Building upon collective knowledge, the project incorporates lessons learned from related work to create a solution emphasizing adaptability, ease of use, and hands-on learning opportunities.

1.3 Scope and Objectives

Scope:

Task Versatility: Excel in a variety of small-scale tasks in desktop or workspace settings, including soldering, cleaning, and precision activities.

User-Friendly Customization: Focus on providing an accessible platform for users to customize and adapt the robot arm, catering to varying technical expertise levels.

Educational Component: Serve as an educational tool, offering hands-on learning experiences in engineering and automation for students and DIY enthusiasts.

Objectives:

Design and Fabrication: Develop a robust and versatile desktop robot arm through the integration of servos, Arduino Uno, and 3D-printed parts, ensuring precision and adaptability.

User-Friendly Interface: Implement an interface allowing individuals, regardless of technical proficiency, to easily program and control the robot arm for different tasks.

Educational Integration: Create comprehensive documentation, including CAD models and circuit schematics, to facilitate understanding and modification, encouraging a community of learners, makers, and innovators.

Task Optimization: Optimize the robot arm's performance in executing small-scale tasks, emphasizing efficiency and accuracy.

Out of Scope:

Industrial-Scale Automation: Focus on desktop and small-scale tasks, not designed for large-scale industrial automation.

Complex Manufacturing Processes: While enhancing efficiency in various tasks, it does not address complex manufacturing processes outside the scope of desktop applications.

Specialized Research Applications: Not delving into specialized research applications, aiming to cater to a broad audience seeking a versatile desktop companion.

1.4 Document Overview

Navigating the evolution of the desktop robot arm project, this document begins with a comprehensive exploration of the background in Chapter 1, establishing the context and motivations. Each subsequent chapter provides balanced coverage, ensuring a detailed exploration of the project.

Chapter 2 Literature Review

2.1 Motors: Navigating Choices in Robotic Actuation

In the expansive realm of robotic projects, the perpetual dichotomy persists in the selection of actuation mechanisms—specifically, the prevalent preference for either servo motors or stepper motors (Motion Solutions, n.d.). A contemporary shift in this landscape favors the adoption of the latest servo motors, distinguished by their heightened torque capabilities. This choice is rooted in the quest for precision and dynamism, where the newer servo motors emerge as the optimal solution for orchestrating intricate and agile movements.

```
// *** smoothing ***

switch1Smoothed = (switch1 * 0.05) + (switch1Prev * 0.95);

switch1Prev = switch1Smoothed;

// *** end of smoothing ***
```

Figure 3. James Bruton Smoothing algorithm.

2.2 Motion Algorithm: Harmonizing Movement with James Bruton

James Bruton, the mastermind behind XRobots, introduces a sophisticated motion algorithm designed to imbue motors with a seamless and aesthetically pleasing trajectory. This algorithm tactically blends 97% of the last value with 3% of the new value in each loop iteration, orchestrating a deliberate deceleration as the motor approaches the culmination of its movement. The finesse of this algorithm lies in its ability to deliver not just functionality but a visual elegance in robotic motion (XRobots, 2021).

Figure 4. Kelton Serra's 1/3rd Scale Model Controller

2.3 Control: Kelton Serra's Ingenuity in Robotic Governance

In the domain of robotic control systems, Kelton Serra from Build Some Stuff offers a nuanced and effective approach. Serra advocates for the construction of a 1/3 scale model mirroring the robot's design, wherein rotary encoders replace the actual motors. The result is a symbiotic relationship where the robot mirrors the movements of its scaled-down counterpart. Beyond this inventive control mechanism, Serra's meticulous attention extends to cable management within the robot arm, showcasing a commitment to both functionality and aesthetics (Build Some Stuff, 2023).

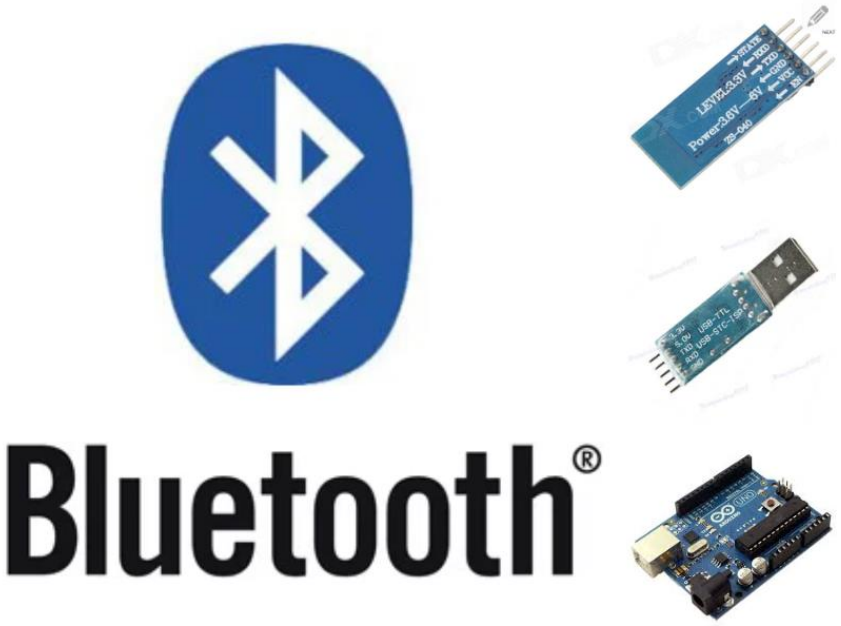


Figure 5.AT Command Mode of HC-05 and HC-06 Bluetooth Module with Shah Saifur Rahman

2.4 Bluetooth Integration: Crafting Connectivity with Dejan Nedelkovski

A pervasive trend among makers involves the integration of wireless control functionality into robot arms. This augmentation provides these devices with graphical interfaces featuring intuitive sliders for controlling motor angles. Noteworthy in this realm is Dejan Nedelkovski, the innovative mind behind HowToMechatronics. His contribution stands out for its comprehensible system, leveraging MIT App Inventor and HC-05 Bluetooth Modules to seamlessly integrate Bluetooth functionality into the robotic ecosystem (HowToMechatronics, 2018).

2.4 Practical Bluetooth: Using HC-05 Module from Autodesk Instructables

This section explores practical Bluetooth integration using the HC-05 Module, referencing Autodesk Instructables. It details the steps and considerations for incorporating Bluetooth functionality into the robotic system, extending the project's connectivity capabilities (Rahman, 2017).

Chapter 3 Project Design

This chapter delves into the comprehensive design specifications of the project, providing a outline of its operational parameters. The essence of the design is encapsulated through detailed mechanical specifications, an electrical block diagram, and a software flowchart. The agreed-upon Critical Success Factors are detailed within this section.

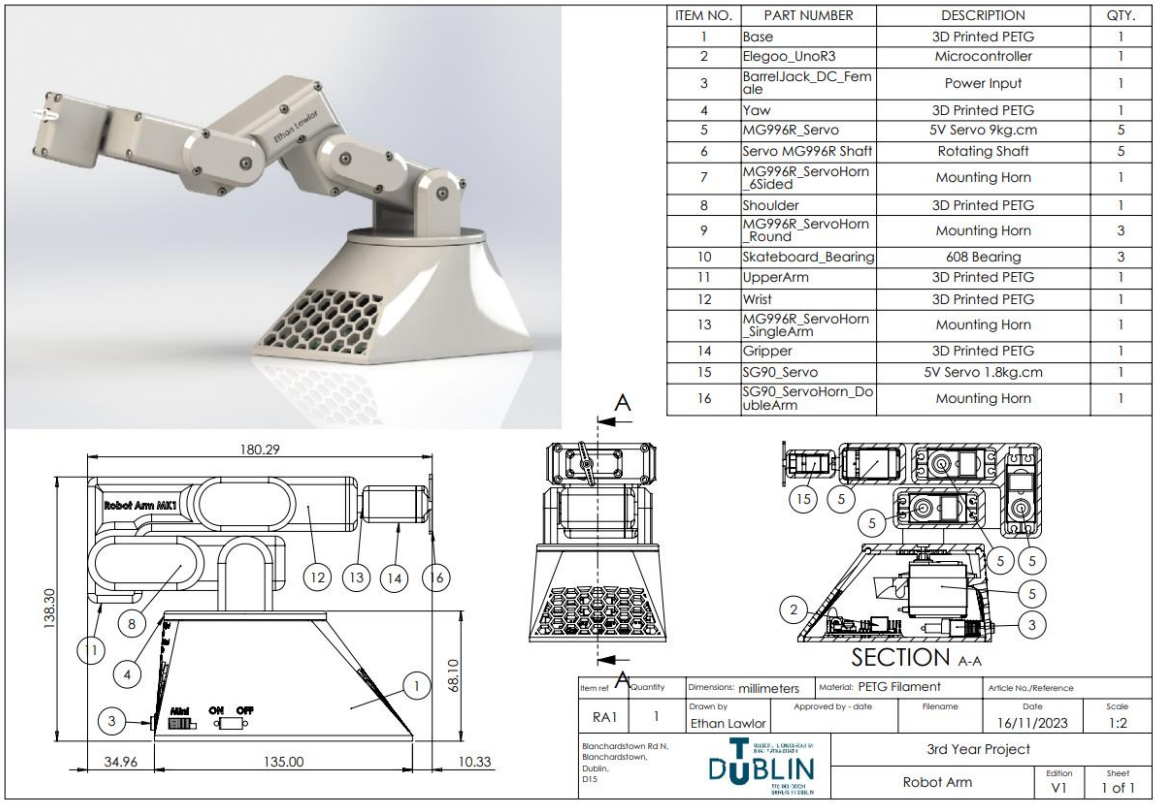


Figure 6.Solidworks Assembly BOM

3.1 Mechanical Design

The mechanical design is a pivotal aspect of the project, dictating its form and functionality. Detailed CAD drawings, including subsections, will be presented to elucidate the intricate details of the robotic arm. These drawings will provide insights into the structural elements, component placements, and the overall architecture.

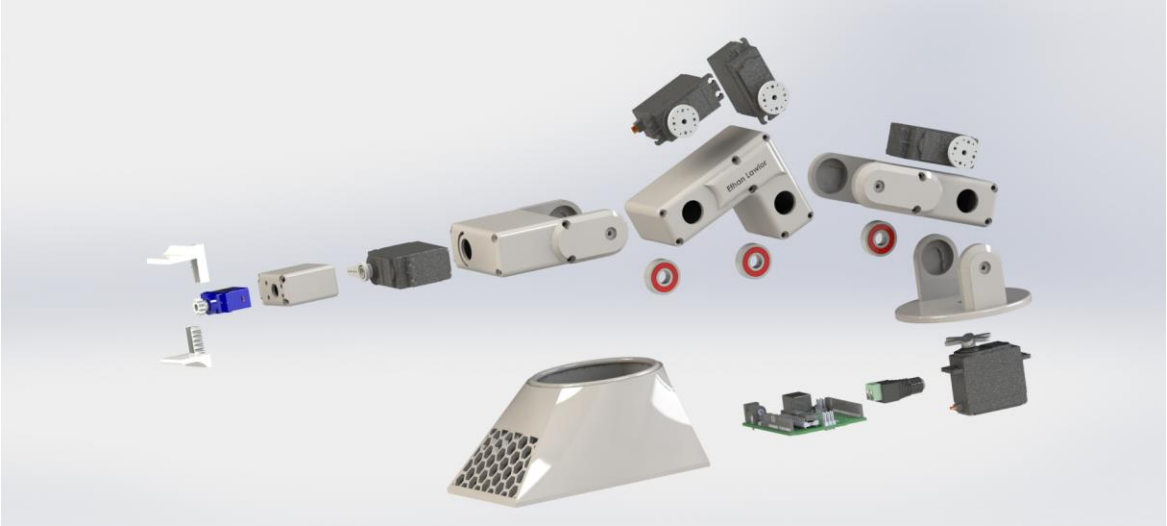


Figure 7.Exploded View of 3D CAD Model

The robot will have 6 degrees of freedom served by 5 MG66R Servo motors and 1 SG90 Servo. The footprint is 135mm by 90mm meaning it is small enough to keep on an average sized desk.

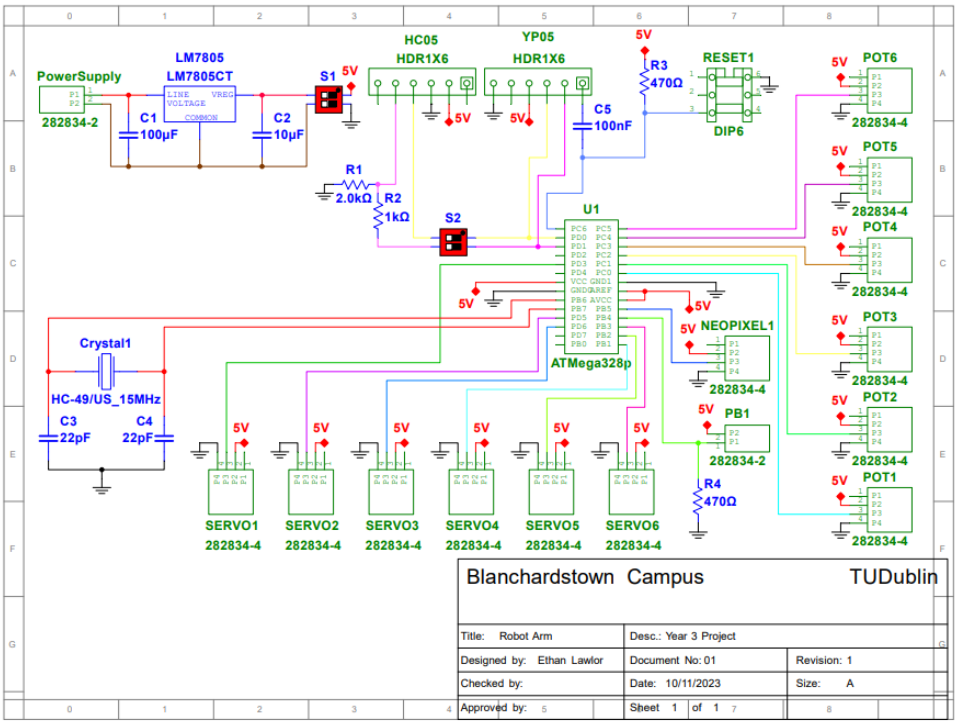


Figure 8. Electrical Schematic

3.2 Electrical Architecture

An integral facet of the project lies in its electrical architecture, governing the interactions between components. The electrical block diagram will be presented to offer a visual representation of the connections, interfaces, and dependencies among various electrical elements. In this section, supporting CAD drawings illustrating the electrical components and their arrangements should be seamlessly integrated.

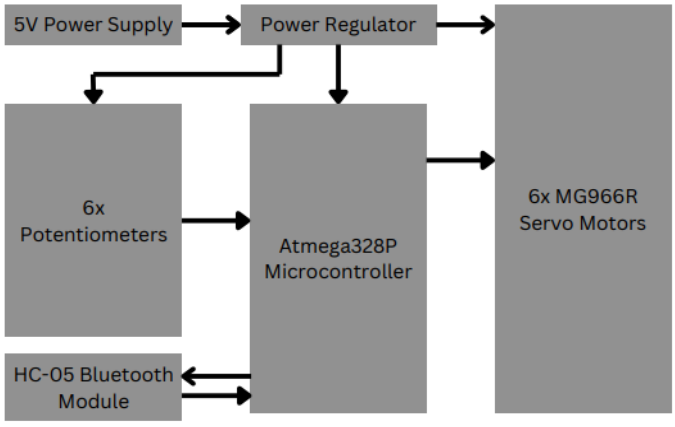
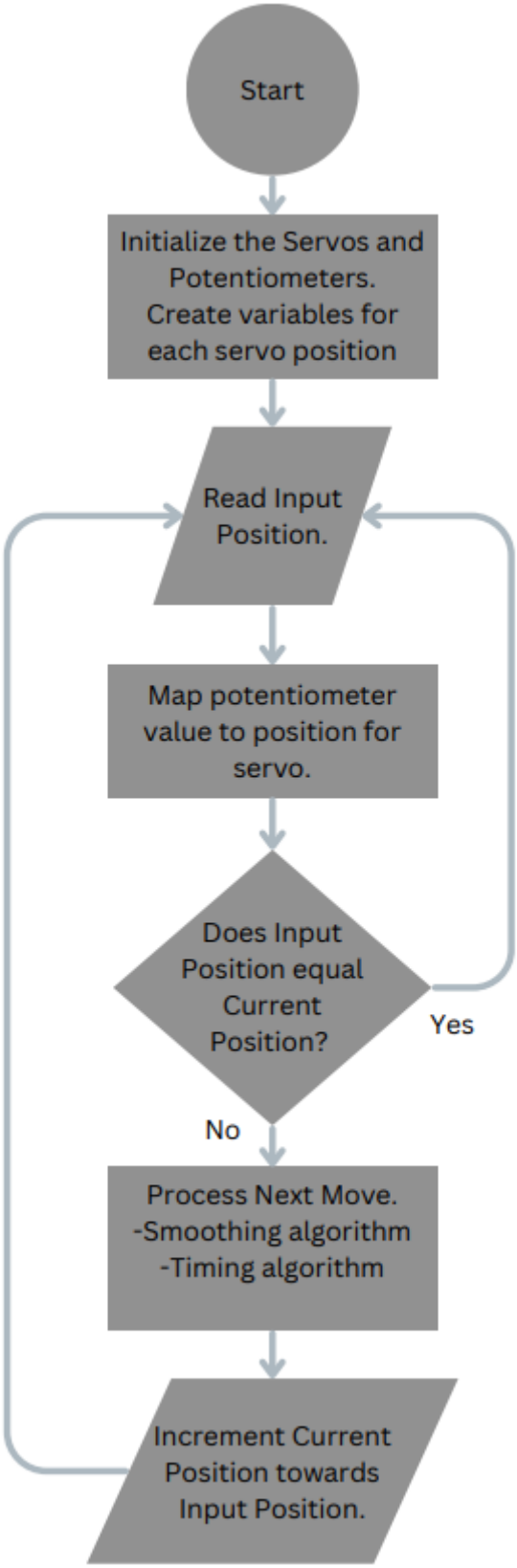


Figure 9.Block Diagram

3.3 Software Control Flow

The control mechanism of the project is orchestrated through a software flowchart. This graphical representation elucidates the sequence of operations and decision points in the control system. It provides a holistic view of how the software governs the project's behavior. Supporting documents, such as the actual software flowchart and relevant code snippets, should be embedded within this section to enhance comprehension.



3.4 Critical Success Factors

Critical Success Factors (CSFs) represent the key benchmarks that, when achieved, ensure the project's success. At the very minimum the robot arm should be able to pick up and move a small mass from one point to another on the a desk.

Chapter 4 Materials and Methods

4.1 Project Implementation Overview

In the journey of bringing our desktop robot arm to life, several crucial steps have been undertaken to translate concepts into tangible progress.

4.1.1 Software Prototyping

Programming the Arduino Uno was a pivotal aspect of the project. The initial software prototyping focused on basic automation functionalities, ensuring seamless communication between the servo motors and the control system. This iterative process involved experimenting with different code snippets to refine the robot arm's responsiveness.

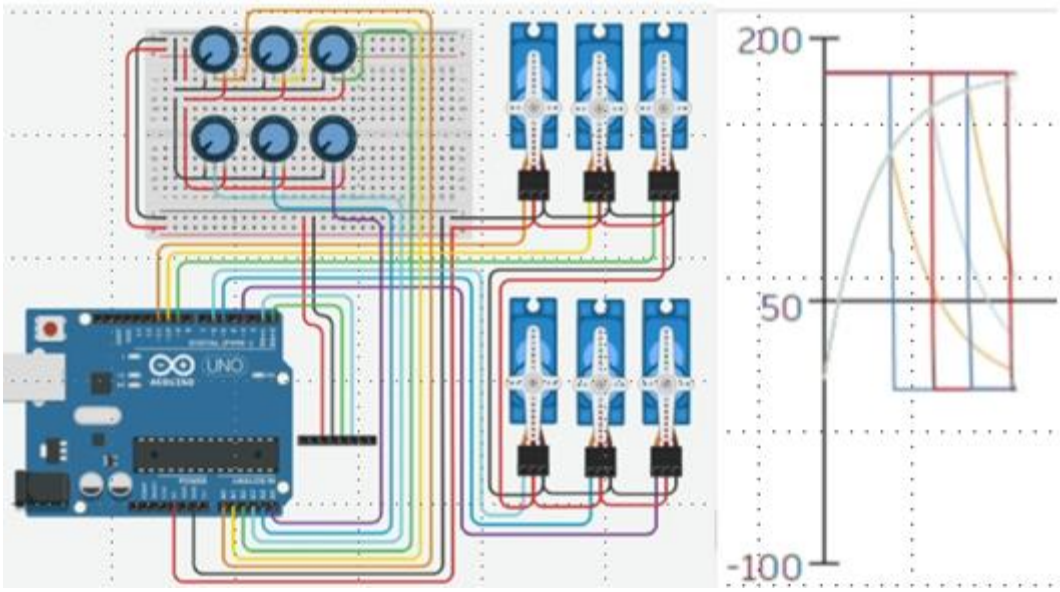


Figure 10.Software Simulation

4.1.2 Hardware Prototyping

The foundational step involved testing the current consupmtion for the robot arm. Using a combination table top power supply we called for the servos to make controlled movements, and we successfully measured a value of 2.8Amps at peak. This value aligns with the expected value drawn from datasheets.

Figure 11.Hardware Prototyping

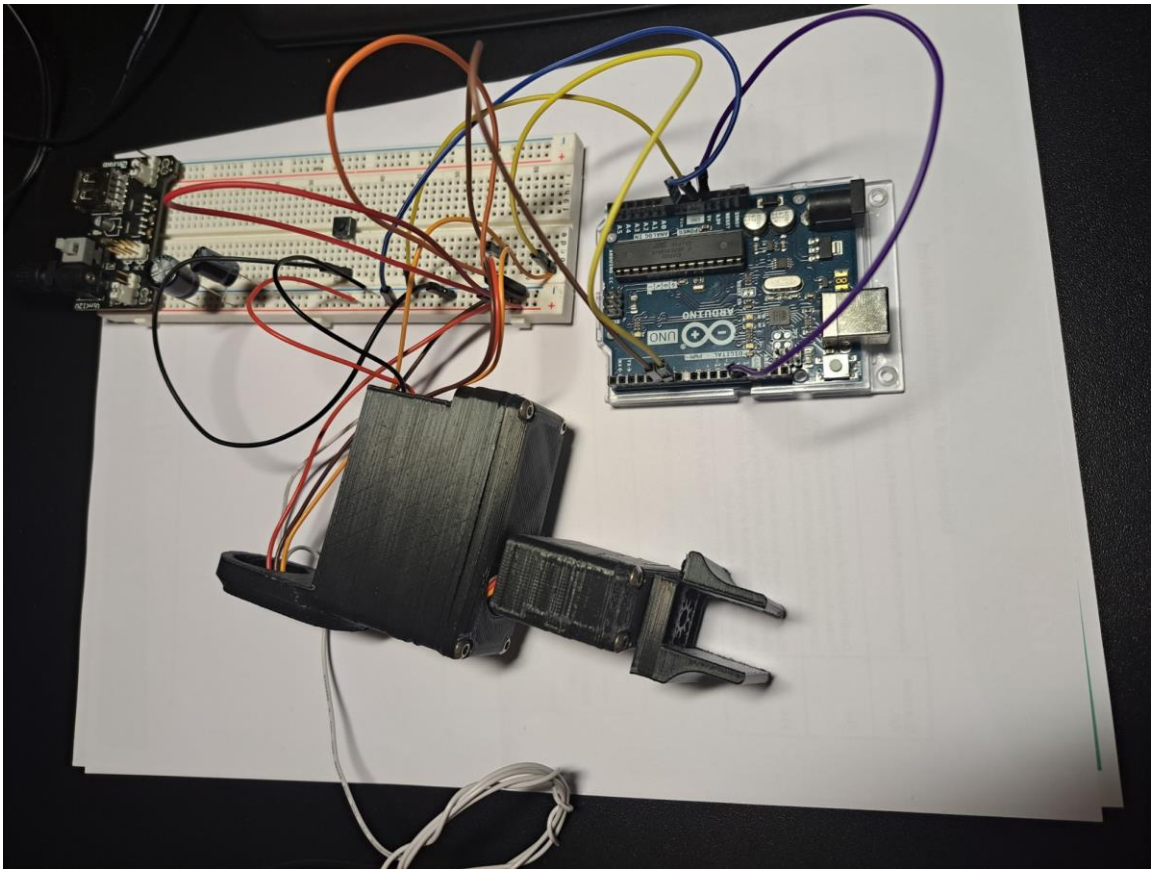


Figure 12.Experimental Print and Joint Movement

4.1.3 Experimentation and Concept Investigation

To validate our design choices, experimentation was conducted. The robot arm underwent trial print and assembly to check the tolerances on various mating components to make sure movement was not hindered by any unforeseen design flaws.

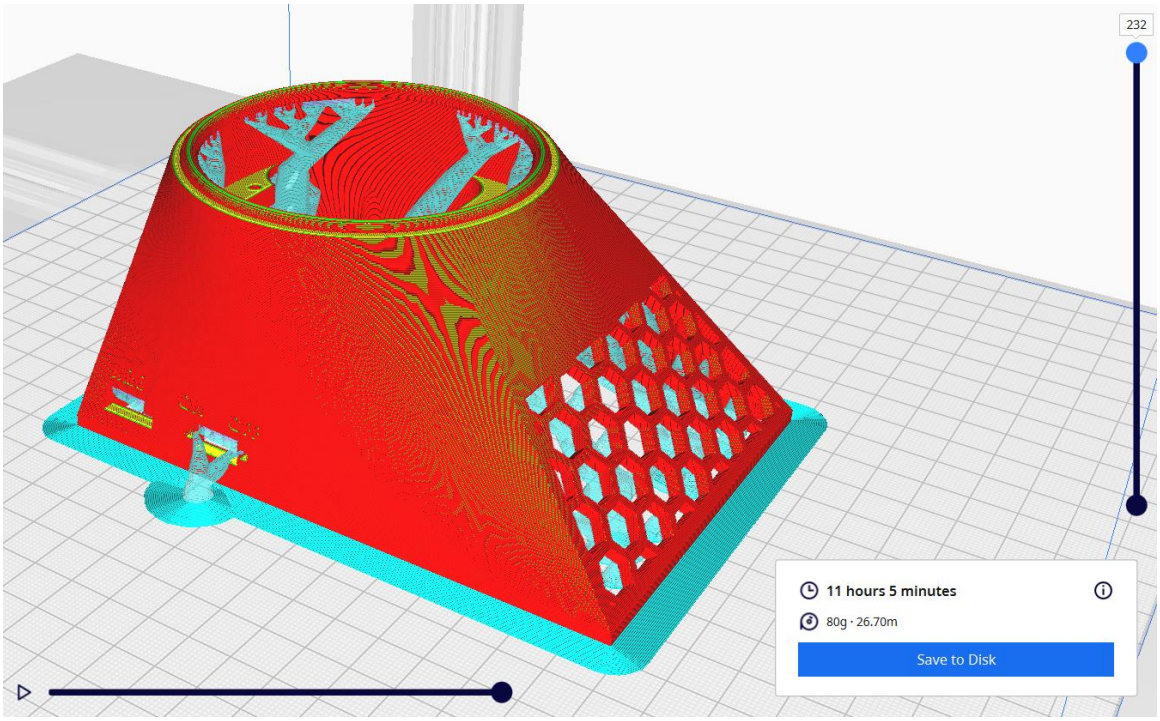


Figure 13.Cura Slicing Software Writing the G-Code for the 3D Printer.

4.2.1 Future Hardware

Looking ahead, we plan to implement the use of a 3D printer and its slicing software to manufacture the remaining pieces required for the build. Then assembly of all the components begins, this includes the integratration of the electrical wiring throughout the assembly.

4.2.2 Software Refinements

In the software realm, refinements are on the horizon. With the aim being to optimize the control algorithm, introducing more sophisticated motion planning to enhance the robot arm's efficiency and fluidity in task execution.

4.2.3 Timeline Plan

Our Semester 2 journey is mapped out in a Gantt chart, providing a visual roadmap. Tasks such as hardware modifications, software refinements, and testing phases are strategically planned to ensure a systematic and well-paced progression.

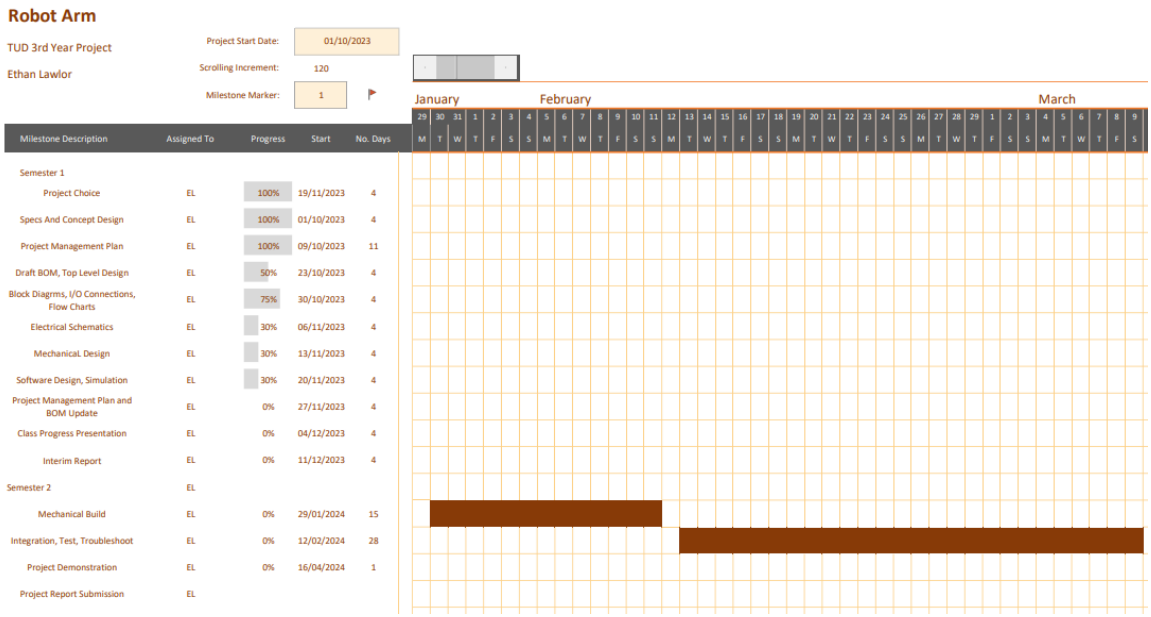


Figure 14.Gantt Chart

4.2.4 Explanation of Future Plans

The decisions driving our future plans are rooted in user feedback, technological advancements, and a commitment to creating a versatile and user-friendly desktop robot arm. We anticipate overcoming challenges and reaching new milestones as we refine and expand the project.

4.3 Summary

This chapter encapsulates the strides made in hardware and software development, showcasing our robot arm's evolution. Looking forward, Semester 2 promises exciting conclusions, solidifying our commitment to innovation in small-scale automation.

Chapter 5 Results and Discussion

5.1 Results

The project achieved several key milestones in developing the desktop robot arm:

- **3D Printing Completion:** The entire structure of the robot arm was successfully 3D printed, meeting the design specifications and maintaining high dimensional accuracy.

Figure 15.All Parts 3D Printed.

- **Servo Motor Programming:** All six servo motors were effectively programmed to perform specific movements using the ATmega microcontroller unit (MCU), demonstrating precise control and synchronization.

- **Custom PCB Development:** A custom PCB was developed using the same ATmega chip, which streamlined the electrical connections and improved the overall reliability of the system.

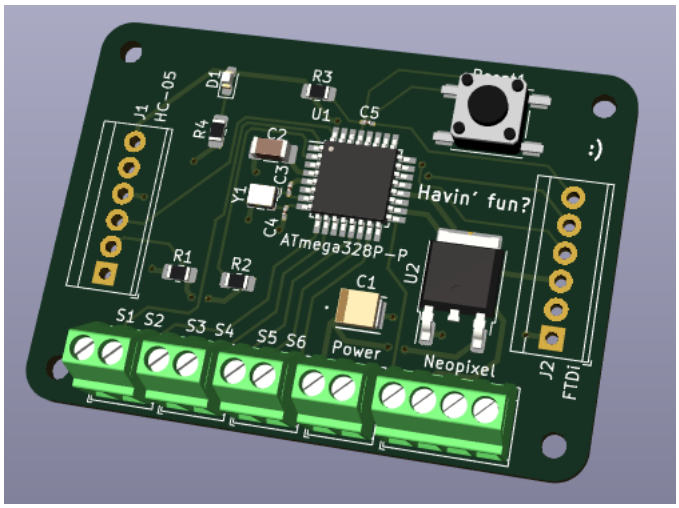


Figure 16.Custom PCB designed in KiCAD.

- **Mobile App Development:** A mobile application was created using MIT App Inventor, allowing for intuitive and responsive control of the robot arm from a smartphone.

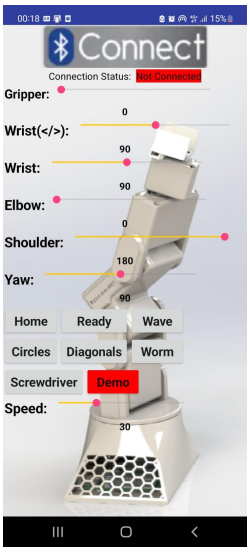


Figure 17.Custom Android App designed in MITAppInventor.

- **Coordinated Movements:** With the combined successes mentioned above, the robot arm was able to be command to move, understand that command and complete it establishing coordinated movements.

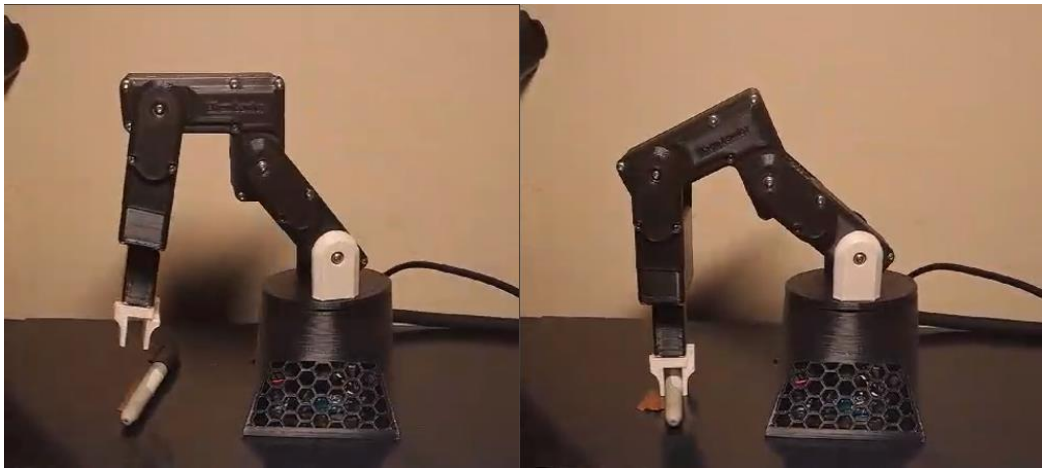


Figure **18**.It Lives**.**

These achievements confirm the technical viability of the project components integrated within the robot arm.

5.2 Discussion

The project's results underscore its success in integrating various technologies into a functional desktop robot arm:

Performance of Servo Motors: The servo motors performed well, with movements that were both precise and reliable. This is crucial for tasks requiring high accuracy and repeatability.

Functionality of the Custom PCB: The custom PCB facilitated efficient electrical routing and component organization, enhancing the system's stability and ease of troubleshooting.

Usability of the Mobile App: The mobile app proved to be user-friendly, providing a straightforward interface for controlling the robot arm. Its performance was robust, with minimal latency and high reliability in command execution.

These outcomes demonstrate the project's effectiveness in achieving its goals of creating a versatile and user-friendly desktop robot arm. Future work could focus on further optimizing the design for energy efficiency and exploring the integration of additional sensors for expanded capabilities.

Chapter 6 Conclusions and Recommendations

6.1 Conclusions

This desktop robot arm project has successfully integrated 3D printing, custom electronics, and software to provide a functional tool for small-scale automation. It illustrates an effective approach to combining diverse technologies to enhance efficiency and adaptability in automation tasks.

- **Project Importance:** The project showcases how combining mechanical design, precise electronic controls, and intuitive software can produce a functional automation tool. This robot arm can be a model for future automation solutions across various industries.
- **Synthesis of the Project:** The integration of 3D printed components, programmed servo motors, and a user-friendly mobile app created a coherent system. This model demonstrates how various elements can work together to solve complex problems in automation.
- **Global Perspective:** The technologies used here could be adapted for broader applications in manufacturing, healthcare, and education, promoting more accessible and customizable automation solutions.

6.2 Recommendations

To expand the project's impact, the following areas could be explored:

- **Integration of Advanced Sensors:** Adding touch, pressure, and proximity sensors could enable the robot arm to perform more delicate and responsive operations.
- **Development of AI Algorithms:** Implementing machine learning could allow the robot arm to improve its operations based on experience, enhancing performance with minimal supervision.
- **Expansion to IoT:** Enabling IoT connectivity could facilitate remote operations and coordination among multiple units.
- **Sustainability Considerations:** Future models should incorporate sustainable materials and energy-efficient components to reduce environmental impact.
- **Educational Kits:** Creating educational kits from this project could provide practical learning tools for aspiring engineers and makers.

Future Directions

Building on these recommendations could significantly enhance the robot arm’s capabilities, turning it into a smarter automation tool and a platform for further technological innovations.

Figure **19**.Fin**.**

In summary, this project not only fulfills its initial goals but also sets a robust groundwork for future advancements in robotic technology. It demonstrates a scalable and adaptable approach to automation that has significant potential for practical applications and educational purposes. This work paves the way for innovations that could extend the capabilities of desktop automation, enhancing both the efficiency and accessibility of robotic solutions across various sectors.

Bibliography

Bruton, J., 2021. *github.com.* [Online]
Available at: https://github.com/XRobots/ServoSmoothing
[Accessed 17/12/2023 December 2023].

Build Some Stuff, 2023. *How I Built a 3D Printed Robot Arm from Scratch (Arduino Based).* [Online]
Available at: https://www.youtube.com/watch?v=5toNqaGsGYs
[Accessed 17 12 2023].

HowToMechatronics, 2018. *DIY Arduino Robot Arm with Smartphone Control.* [Online]
Available at: https://howtomechatronics.com/tutorials/arduino/diy-arduino-robot-arm-with-smartphone-control/?utm_content=cmp-true
[Accessed 17 12 2023].

Motion Solutions, n.d. *Servo Motor vs Stepper Motor: Which is right for your application?.* [Online]
Available at: https://www.motionsolutions.com/servo-motor-vs-stepper-motor-right-application/
[Accessed 17 12 2023].

Rahman, S. S., 2017. *AT Command Mode of HC-05 and HC-06 Bluetooth Module.* [Online]
Available at: https://www.instructables.com/AT-command-mode-of-HC-05-Bluetooth-module/
[Accessed 17 12 2023].

XRobots, 2021. *ServoSmoothing.* [Online]
Available at: https://github.com/XRobots/ServoSmoothing
[Accessed 17 Decemeber 2023].

XRobots, n.d. [Online].

Appendix A Project Planning

Robot Arm

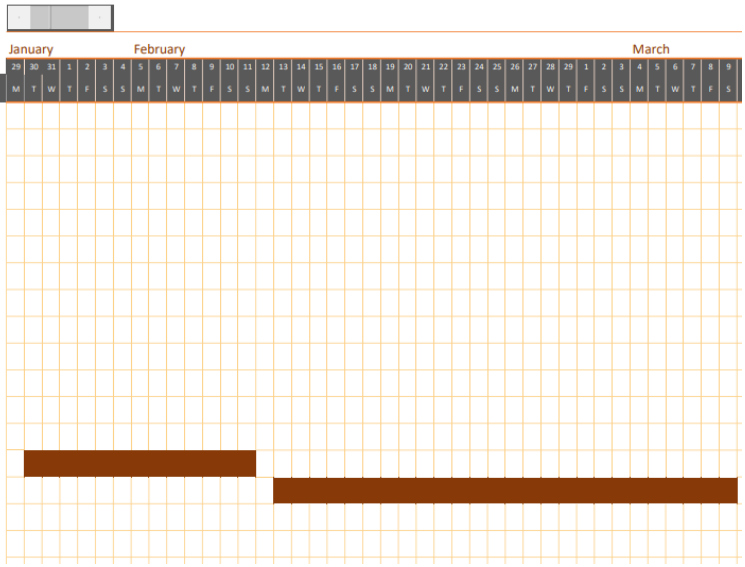
TUD 3rd Year Project

Ethan Lawlor

Project Start Date: 01/10/2023

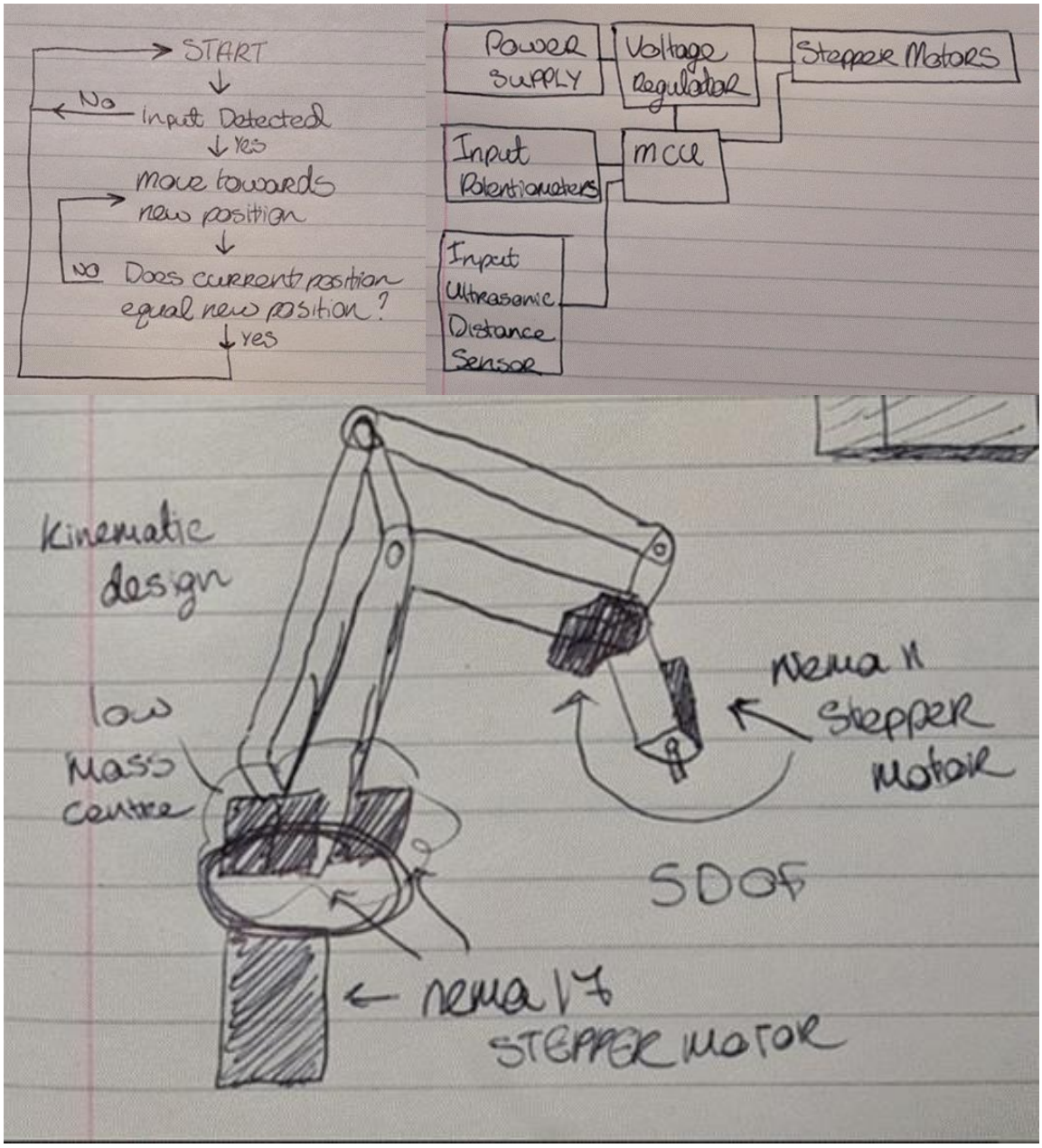
Scrolling Increment: 120

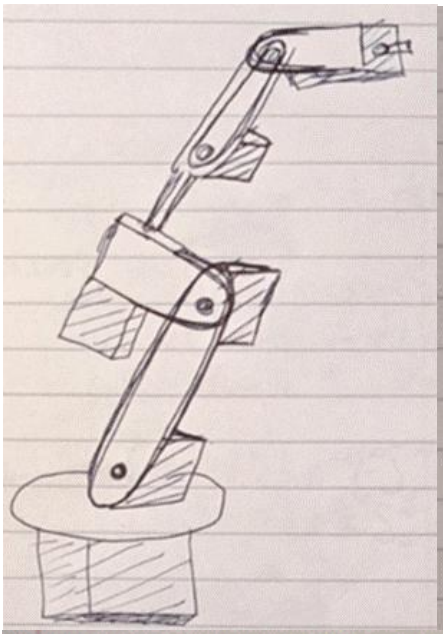
Milestone Marker: 1

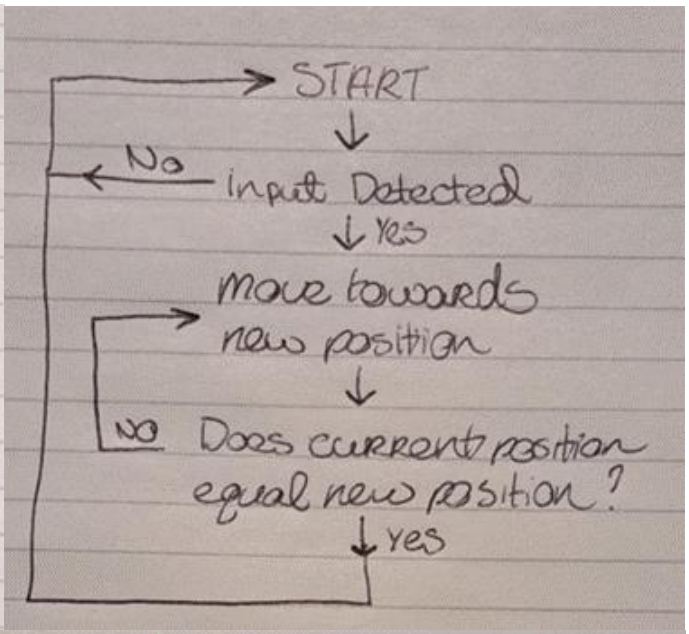


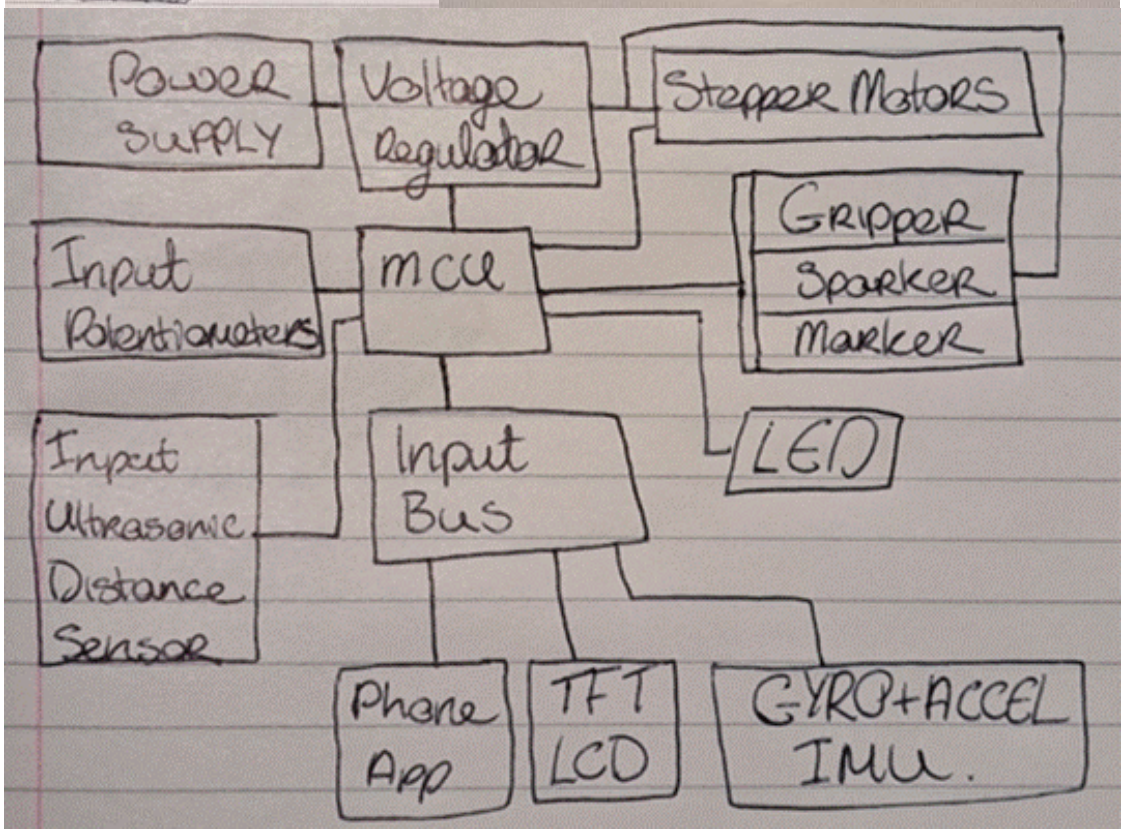
Milestone Description	Assigned To	Progress	Start	No. Days
Semester 1				
Project Choice	EL	100%	19/11/2023	4
Specs And Concept Design	EL	100%	01/10/2023	4
Project Management Plan	EL	100%	09/10/2023	11
Draft BOM, Top Level Design	EL	50%	23/10/2023	4
Block Diagrms, I/O Connections, Flow Charts	EL	75%	30/10/2023	4
Electrical Schematics	EL	30%	06/11/2023	4
Mechanical Design	EL	30%	13/11/2023	4
Software Design, Simulation	EL	30%	20/11/2023	4
Project Management Plan and BOM Update	EL	0%	27/11/2023	4
Class Progress Presentation	EL	0%	04/12/2023	4
Interim Report	EL	0%	11/12/2023	4
Semester 2	EL			
Mechanical Build	EL	0%	29/01/2024	15
Integration, Test, Troubleshoot	EL	0%	12/02/2024	28
Project Demonstration	EL	0%	16/04/2024	1
Project Report Submission	EL			

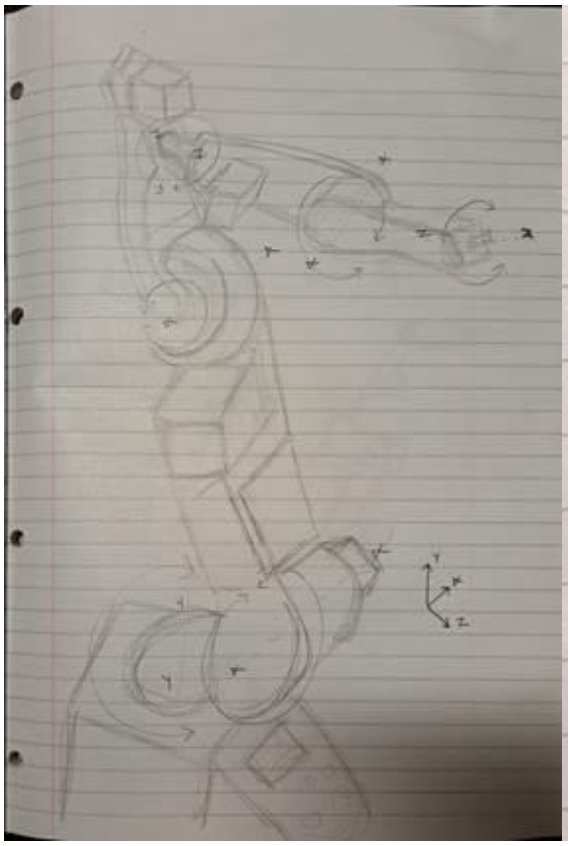
Appendix B Salient Extracts from Project Diary

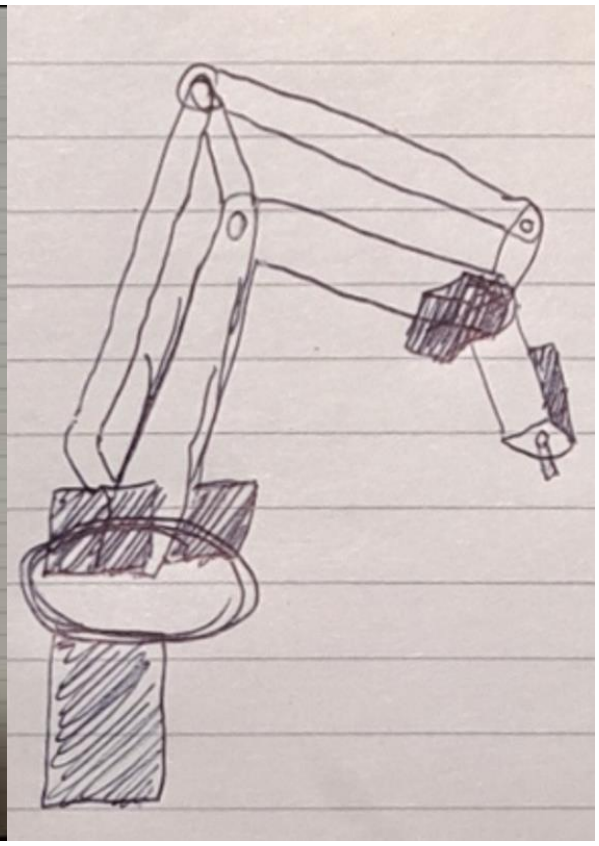


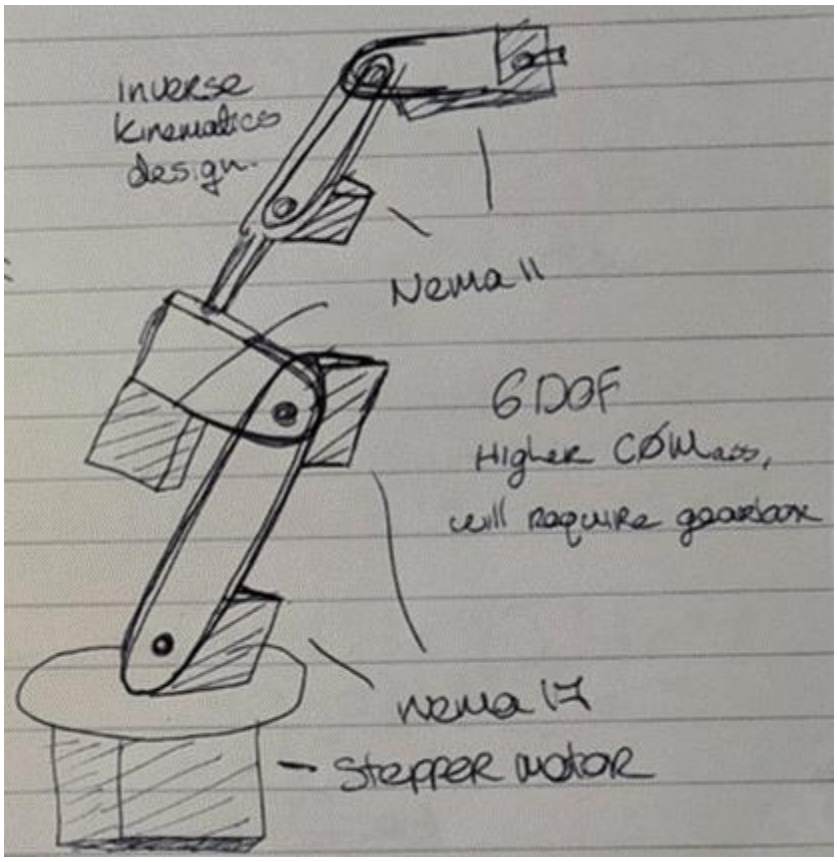



START
No
Input Detected
Yes
move towards new position
No
Does current position equal new position?
Yes

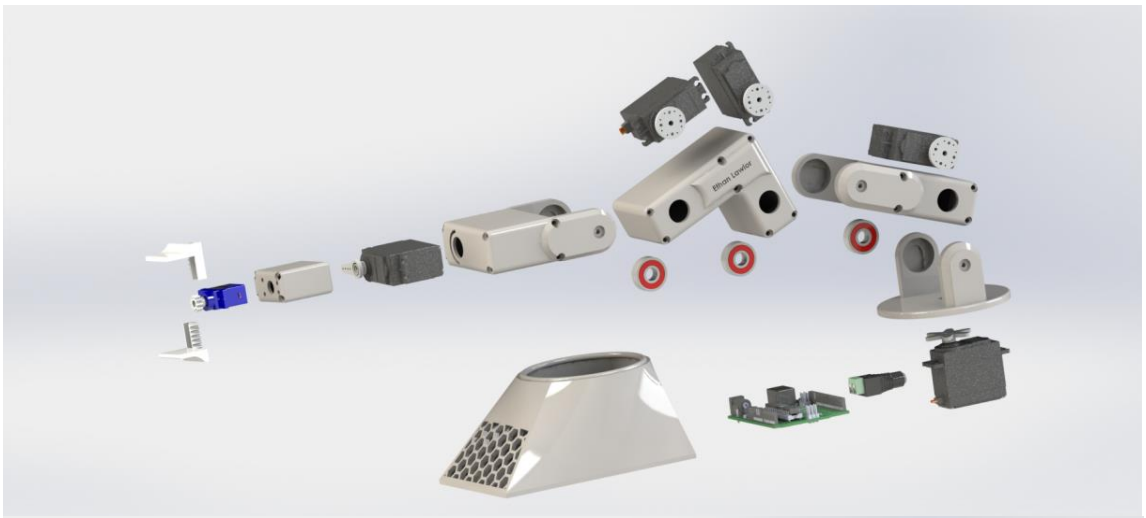

Power Supply
Voltage Regulator
Stepper Motors
Gripper
Sparker
Marker
Input Potentiometers
MCU
LED
Input Ultrasonic Distance Sensor
Input Bus
Phone App
TFT LCD
GYRO+ACCEL IMU.

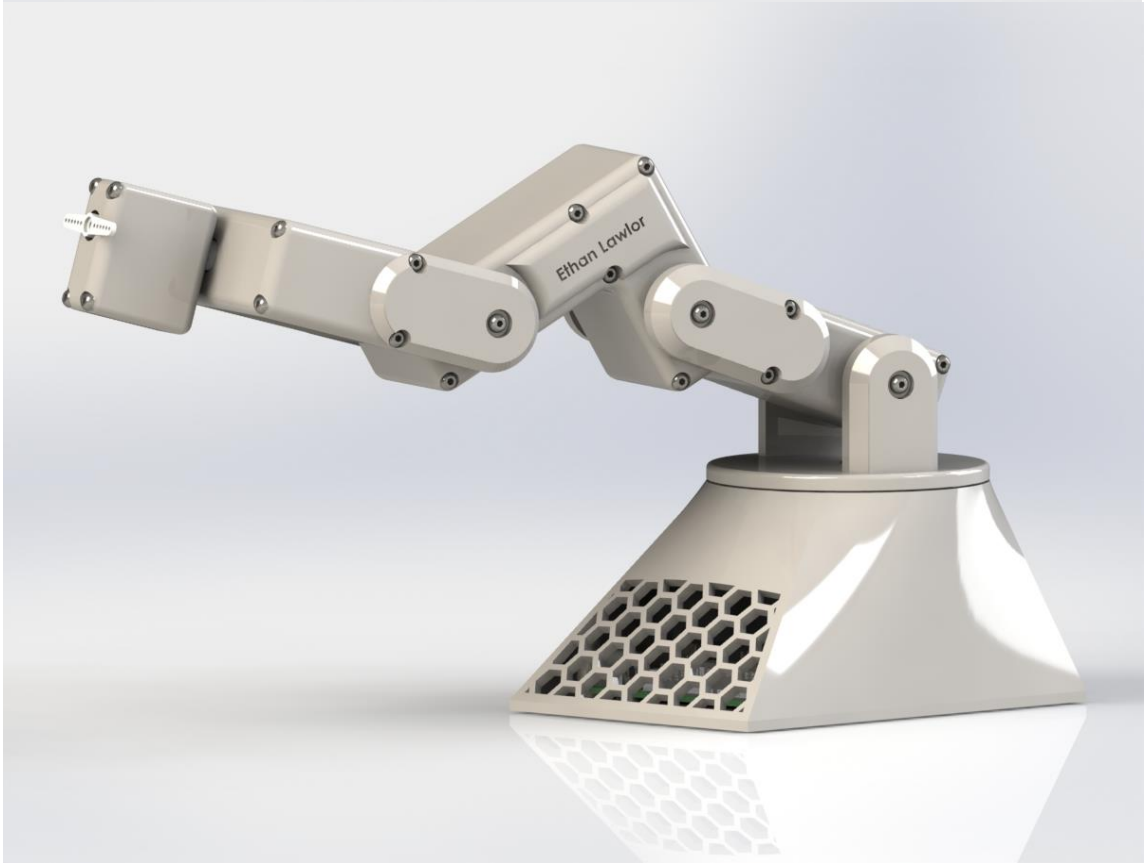


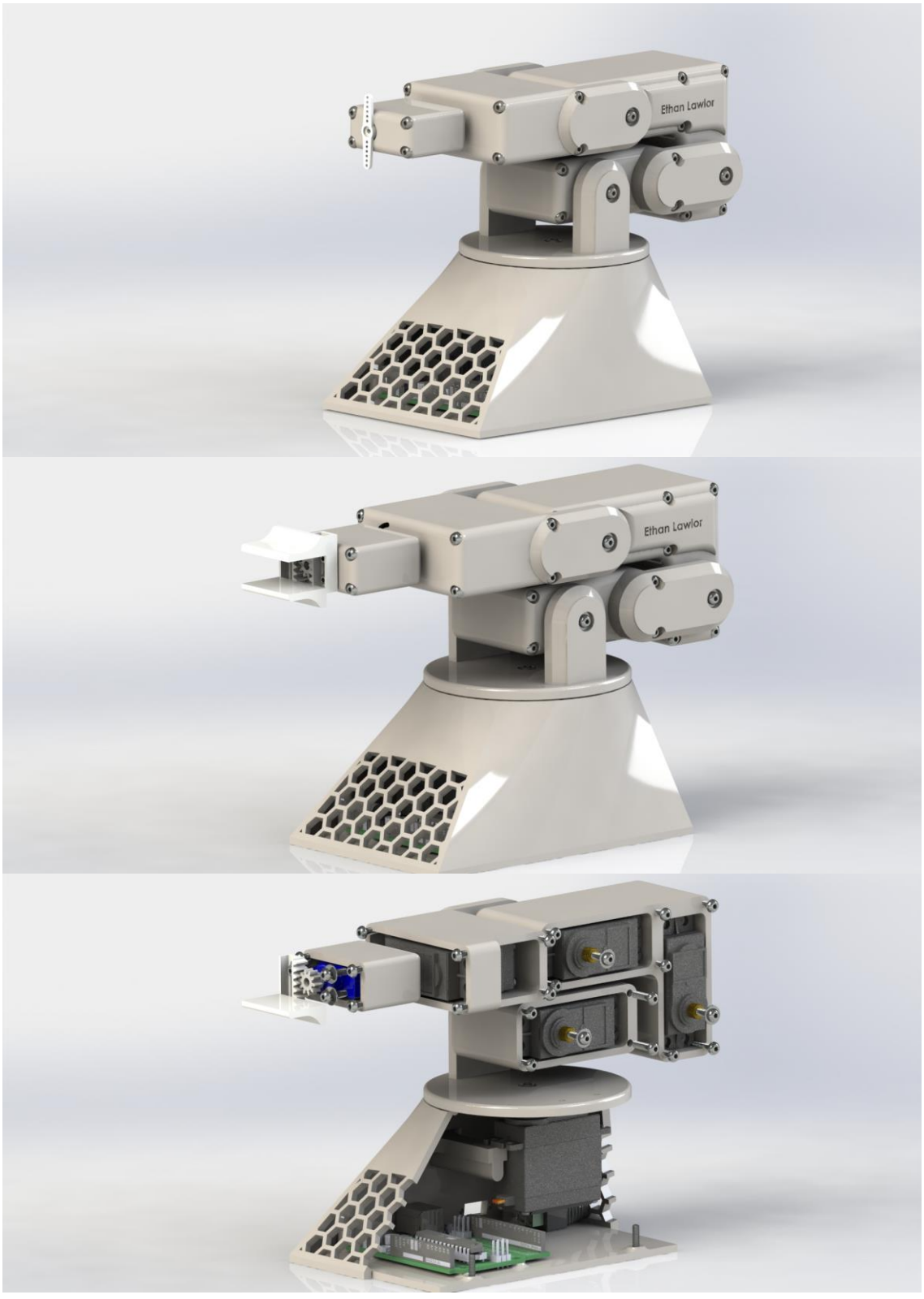



inverse
Kinematics
design.
Nema 11
6 DOF
Higher COM mass,
will require gearbox
Nema 17
- Stepper motor

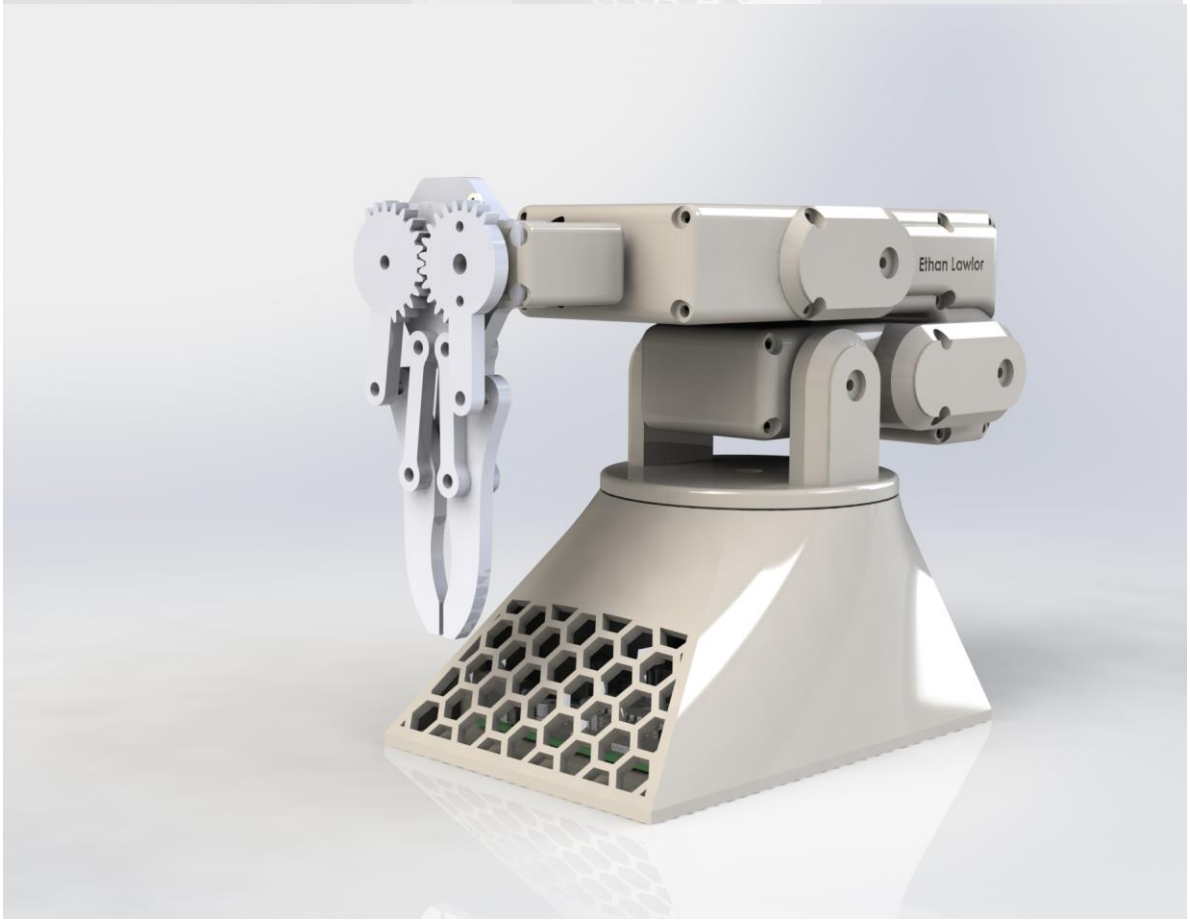
Appendix C Design Drawings & Component Specifications


Ethan Lawlor

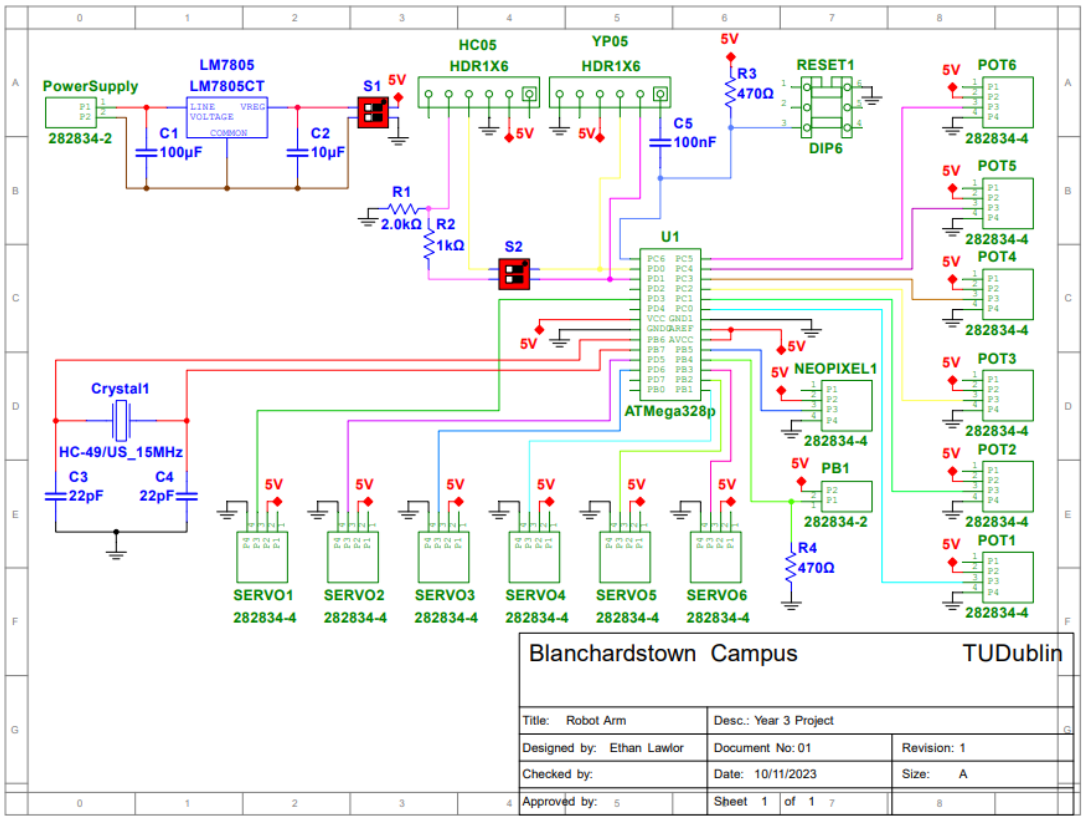

Ethan Lawlor

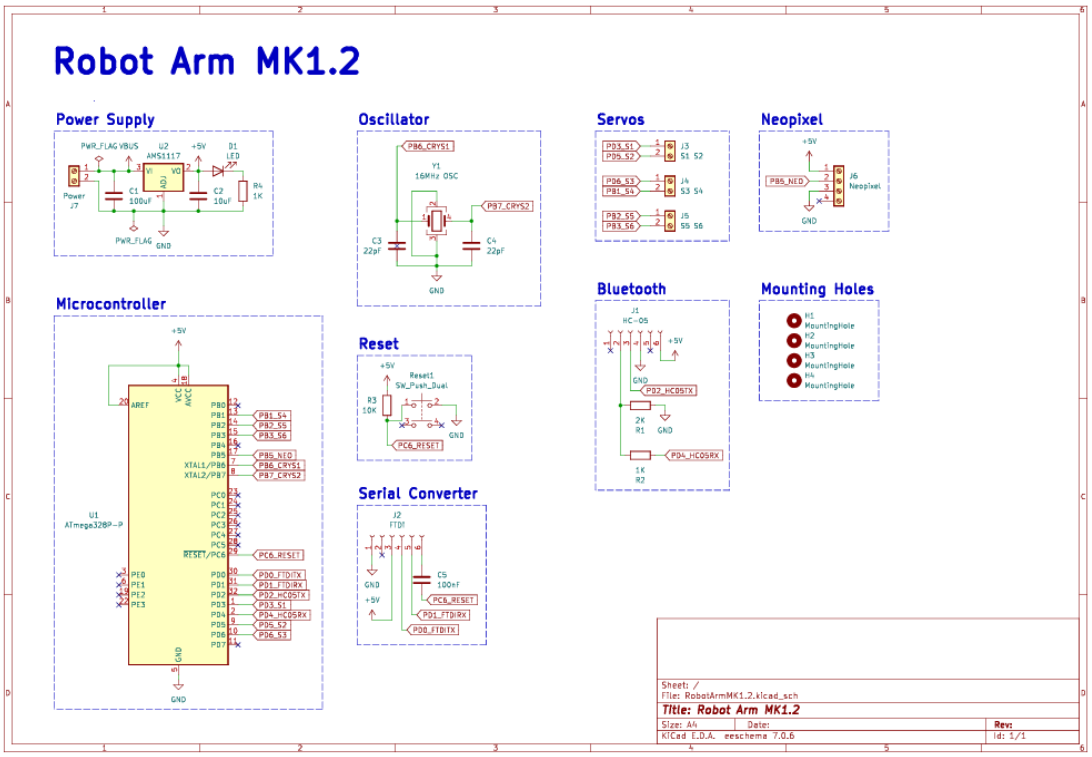

Ethan Lawlor
Ethan Lawlor

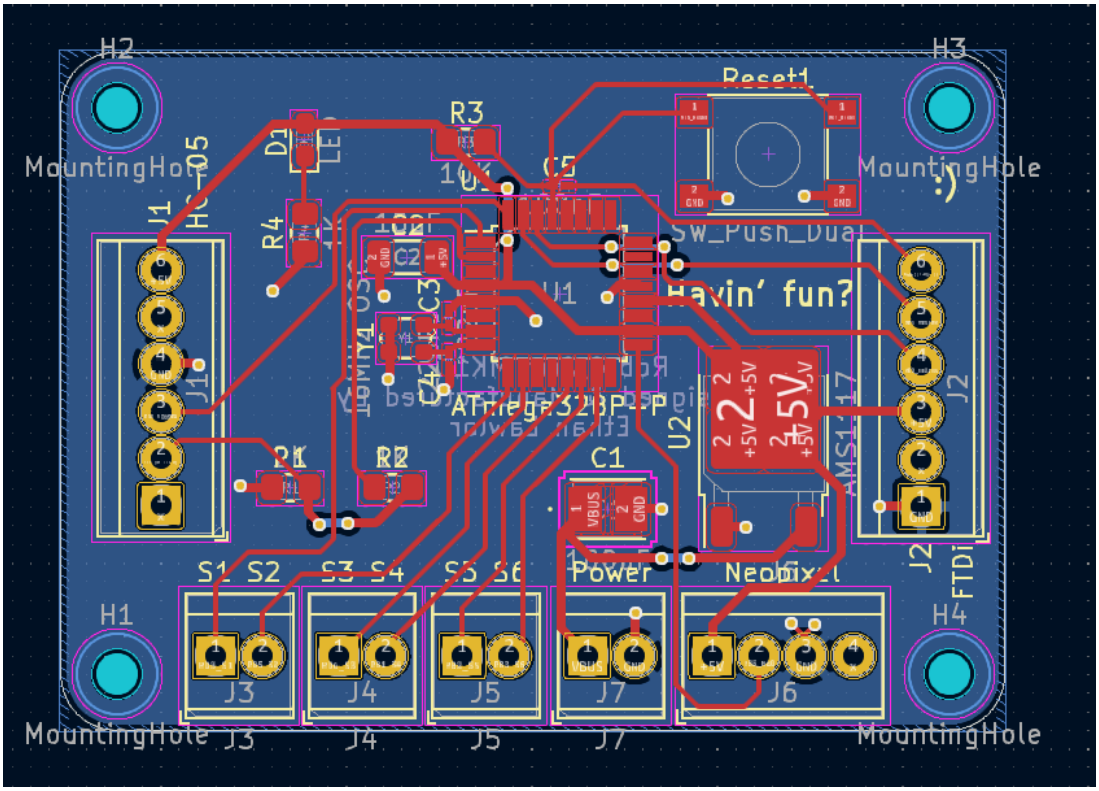

Ethan Lawlor


Ethan Lawlor

Appendix D Schematics

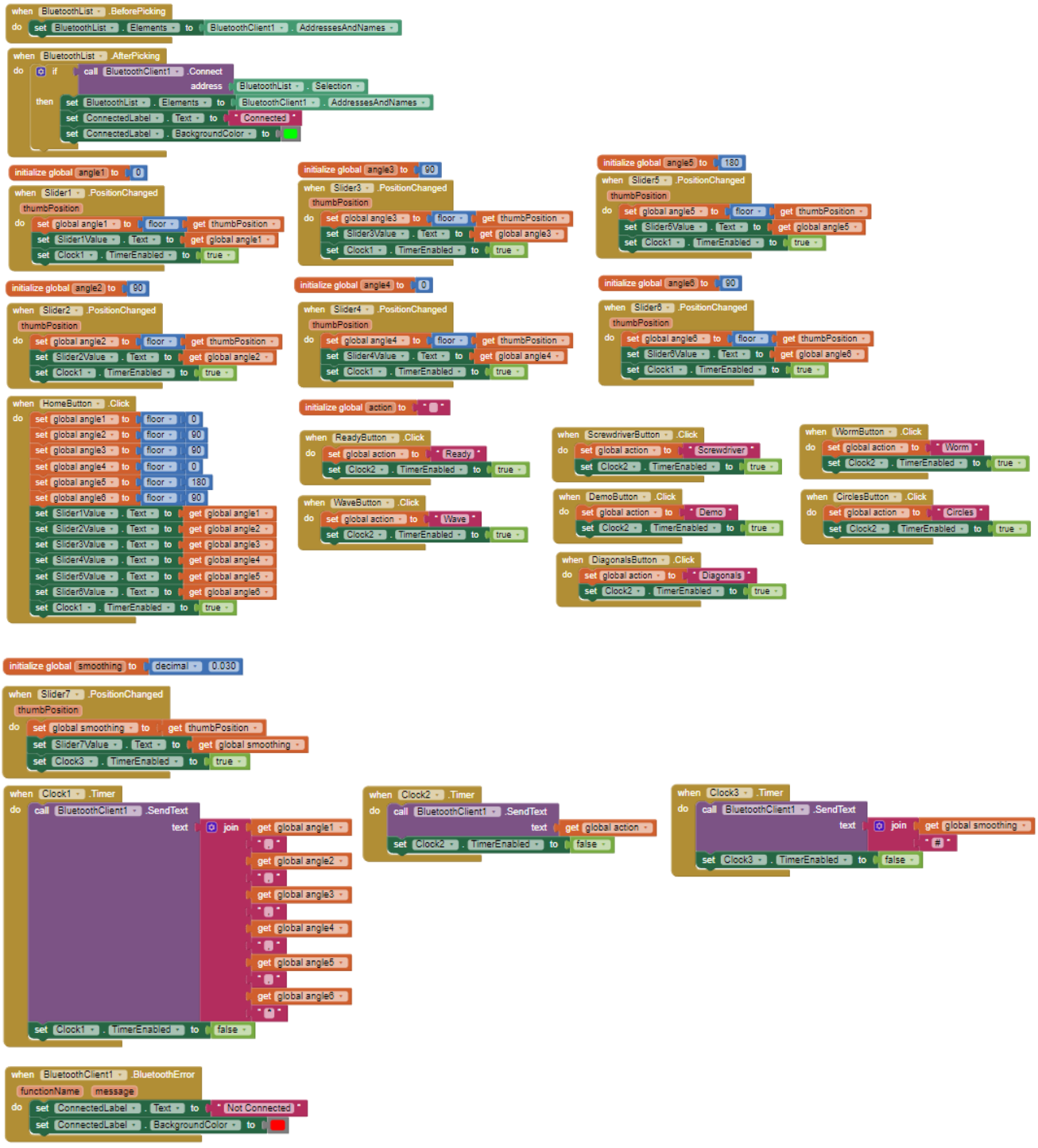


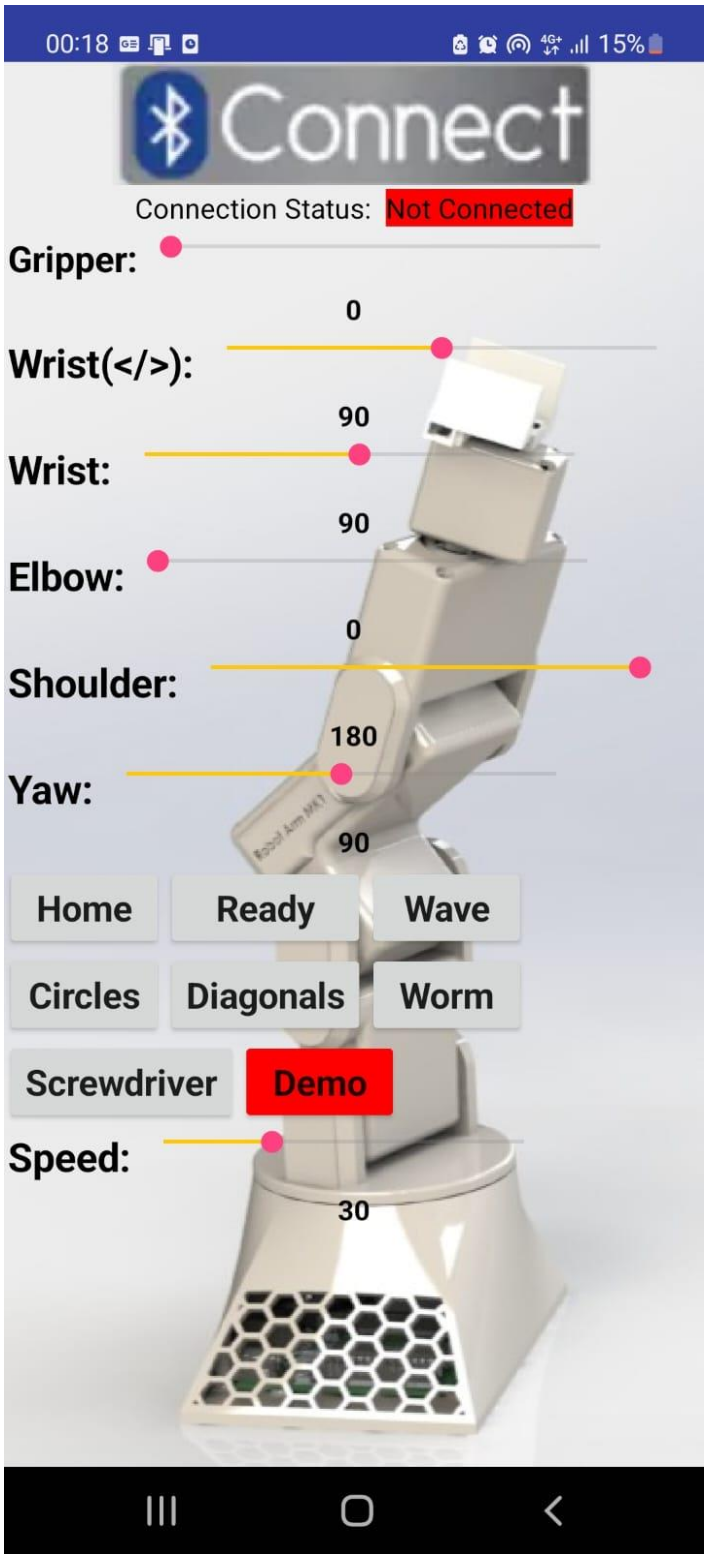



H2
H3
MountingHole
MountingHole
Reset1
R3
D1
C5
U3
R4
C2
C3
C4
U1
SW_Push_Dual
Havin' fun?
ATmega328P-P
U2
R1
R2
C1
J1
J2
S1 S2 S3 S4 S5 S6
Power
Neopixel
FTDI
H1
H4
J3
J4
J5
J7
J6
MountingHole
MountingHole

Appendix E List of Software Code – MITApp Inventor

https://gallery.appinventor.mit.edu/?galleryid=967702b1-feb8-411e-8451-febcac57ca8f




00:18
15%
Connect
Connection Status: Not Connected
Gripper:
0
Wrist(</>):
90
Wrist:
90
Elbow:
0
Shoulder:
180
Yaw:
90
Home
Ready
Wave
Circles
Diagonals
Worm
Screwdriver
Demo
Speed:
30

Appendix F List of Software Code - Arduino

```
// Ethan Lawlor

// B00149346

// Functionality of Code:
// Commands from Serial Monitor
// Commands from Bluetooth - MIT App Inventor
// Set individual servo angles - 0,0,0,0,0,0*/180,180,180,180,180,180*
// Send groups of servo angles through keywords – Demo/Home/Ready/Wave
// Set smoothing value - 0#/100#

#include <SoftwareSerial.h>

SoftwareSerial btSerial(2, 4);

#include <Servo.h>
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;
Servo servo5;
Servo servo6;

int gripper;
int servo1pos;
float grippersmoothed;
float gripperPrev;

int switch2;
int servo2pos;
float switch2smoothed;
float switch2Prev;

int switch3;
int servo3pos;
float switch3smoothed;
float switch3Prev;

int switch4;
int servo4pos;
float switch4smoothed;
float switch4Prev;
```

```
int switch5;
int servo5pos;
float switch5smoothed;
float switch5Prev;

int switch6;
int servo6pos;
float switch6smoothed;
float switch6Prev;

const int servoPin1 = 3;  // PWM pin for servo1
const int servoPin2 = 5;  // PWM pin for servo2
const int servoPin3 = 6;
const int servoPin4 = 9;  // PWM pin for servo4
const int servoPin5 = 10;
const int servoPin6 = 11;  // PWM pin for servo6

const int open = 180;
const int close = 50;

char character;
String receivedData;
String angle_1;
String angle_2;
String angle_3;
String angle_4;
String angle_5;
String angle_6;

float smoothing = 0.03;
int servoDelay = 5;

int commaIndex;

void setup() {

  Serial.begin(9600);
  btSerial.begin(9600);

  pinMode(servoPin1, OUTPUT);
  pinMode(servoPin2, OUTPUT);
  pinMode(servoPin3, OUTPUT);
  pinMode(servoPin4, OUTPUT);
  pinMode(servoPin5, OUTPUT);
  pinMode(servoPin6, OUTPUT);
```

```
servo1.attach(servoPin1);
servo2.attach(servoPin2);
servo3.attach(servoPin3);
servo4.attach(servoPin4);
servo5.attach(servoPin5);
servo6.attach(servoPin6);

Serial.println("<Arduino Ready>");
delay(1000);

Serial.println("");
Serial.print("Homing...");

smoothing = 0.010;
Serial.print("");
Serial.print("Smoothing = ");
Serial.println(smoothing, 3);

servo6.write(90);
switch6 = 90;
servo6pos = 90;
switch6smoothed = 90;
switch6Prev = 90;

servo5.write(180);
switch5 = 180;
servo5pos = 180;
switch5smoothed = 180;
switch5Prev = 180;

servo4.write(0);
switch4 = 0;
servo4pos = 0;
switch4smoothed = 0;
switch4Prev = 0;

servo3.write(90);
switch3 = 90;
servo3pos = 90;
switch3smoothed = 90;
switch3Prev = 90;

servo2.write(90);
switch2 = 90;
```

```
  servo2pos = 90;
  switch2smoothed = 90;
  switch2Prev = 90;

  servo1.write(0);
  gripper = 0;
  servo1pos = 0;
  grippersmoothed = 0;
  gripperPrev = 0;

  ready();
  home();

  smoothing = 0.03;
  Serial.println("");
  Serial.print("Finished Homing...");
  Serial.print("");
  Serial.print("Smoothing = ");
  Serial.println(smoothing);

  Serial.println("");
  Serial.println("<Robot Arm Ready>");
  delay(1000);
}

void loop() {

  readSerialData();

}  // End of Loop

void readSerialData() {

  while (Serial.available()) {
    character = Serial.read();
    receivedData = receivedData + character;
    checkAndExecute();
  }

  while (btSerial.available()) {
    character = btSerial.read();
    receivedData = receivedData + character;
    checkAndExecute();
  }
}
```

```
void checkAndExecute() {

  if (receivedData == "Home") {
    receivedData = "";
    home();
  }

  if (receivedData == "Ready") {
    receivedData = "";
    ready();
  }

  if (receivedData == "Wave") {
    wave();
    receivedData = "";
  }

  if (receivedData == "Screwdriver") {
    receivedData = "";
    screwdriver();
  }

  if (receivedData == "Circles") {
    receivedData = "";
    circles();
  }

  if (receivedData == "Diagonals") {
    receivedData = "";
    diagonals();
  }

  if (receivedData == "Worm") {
    receivedData = "";
    worm();
  }

  if (receivedData == "Demo") {
    receivedData = "";
    demo();
  }
```

```
 else if (character == '#') {

   Serial.println("");
   Serial.println(receivedData);
   receivedData = receivedData.substring(0, receivedData.length() - 1);
// Remove the last character '*'
   int receivedStr = receivedData.toInt();
   float smoothingRaw = map(receivedStr, 0, 100, 1, 90);
   smoothing = smoothingRaw / 1000;
   Serial.print("");
   Serial.print("Smoothing = ");
   Serial.println(smoothing, 3);

   receivedData = "";

 }

 else if (character == '*') {
   Serial.println("");
   Serial.println(receivedData);
   receivedData = receivedData.substring(0, receivedData.length() - 1);
// Remove the last character '*'
   commaIndex = receivedData.indexOf(',');
   angle_1 = receivedData.substring(0, commaIndex);
   receivedData = receivedData.substring(commaIndex + 1);

   commaIndex = receivedData.indexOf(',');
   angle_2 = receivedData.substring(0, commaIndex);
   receivedData = receivedData.substring(commaIndex + 1);

   commaIndex = receivedData.indexOf(',');
   angle_3 = receivedData.substring(0, commaIndex);
   receivedData = receivedData.substring(commaIndex + 1);

   commaIndex = receivedData.indexOf(',');
   angle_4 = receivedData.substring(0, commaIndex);
   receivedData = receivedData.substring(commaIndex + 1);

   commaIndex = receivedData.indexOf(',');
   angle_5 = receivedData.substring(0, commaIndex);
   angle_6 = receivedData.substring(commaIndex + 1);

   gripper = angle_1.toInt();
   switch2 = angle_2.toInt();
   switch3 = angle_3.toInt();
```

```
    switch4 = angle_4.toInt();
    switch5 = angle_5.toInt();
    switch6 = angle_6.toInt();

    receivedData = "";

    moveServos();
  }
}

void moveServos() {

  Serial.println("");
  Serial.print("Moving...");
  Serial.print("");
  Serial.print("Smoothing = ");
  Serial.println(smoothing, 3);

  while (servo1pos != gripper || servo2pos != switch2 || servo3pos !=
switch3 || servo4pos != switch4 || servo5pos != switch5 || servo6pos !=
switch6) {

    readSerialData();

    // **   smoothing **  //

    grippersmoothed = (gripper * smoothing) + (gripperPrev * (1 -
smoothing));
    gripperPrev = grippersmoothed;
    servo1pos = round(grippersmoothed);

    switch2smoothed = (switch2 * smoothing) + (switch2Prev * (1 -
smoothing));
    switch2Prev = switch2smoothed;
    servo2pos = round(switch2smoothed);

    switch3smoothed = (switch3 * smoothing) + (switch3Prev * (1 -
smoothing));
    switch3Prev = switch3smoothed;
    servo3pos = round(switch3smoothed);

    switch4smoothed = (switch4 * smoothing) + (switch4Prev * (1 -
smoothing));
    switch4Prev = switch4smoothed;
    servo4pos = round(switch4smoothed);
```

```
    switch5smoothed = (switch5 * smoothing) + (switch5Prev * (1 -
smoothing));
    switch5Prev = switch5smoothed;
    servo5pos = round(switch5smoothed);

    switch6smoothed = (switch6 * smoothing) + (switch6Prev * (1 -
smoothing));
    switch6Prev = switch6smoothed;
    servo6pos = round(switch6smoothed);

    // ** end of smoothing  **  //

    servo1.write(servo1pos);
    servo2.write(servo2pos);
    servo3.write(servo3pos);
    servo4.write(servo4pos);
    servo5.write(servo5pos);
    servo6.write(servo6pos);

    delay(servoDelay);
  }
}

void home() {

  Serial.println("");
  Serial.println("Home");
  gripper = close;
  switch2 = 90;
  switch3 = 90;
  switch4 = 0;
  switch5 = 180;
  switch6 = 90;
  moveServos();
}

void ready() {

  Serial.println("");
  Serial.println("Ready");
  gripper = 0;
  switch2 = 0;
  switch3 = 135;
  switch4 = 90;
```

```
  switch5 = 135;
  switch6 = 90;
  moveServos();
}

void wave() {

  Serial.println("");
  Serial.println("Wave");
  gripper = close;
  switch2 = 0;
  switch3 = 135;
  switch4 = 90;
  switch5 = 135;
  switch6 = 90;
  moveServos();

  gripper = open;
  switch2 = 180;
  switch6 = 105;
  moveServos();

  gripper = close;
  switch2 = 0;
  switch6 = 75;
  moveServos();
}

void circles() {
  Serial.println("");
  Serial.println("Circles");

  // Initial positions to start drawing circles
  switch6 = 90;
  switch5 = 130;  // Base position - slightly rotate if needed for better
motion
  switch4 = 90;   // Elbow - mid position
  switch3 = 90;   // Wrist - starting position
  moveServos();
  delay(1000);  // Wait a bit before starting the circles

  float previousValue = smoothing;
  smoothing = previousValue * 25;

  for (int i = 0; i < 5; i++) {                    // Draw 5 circles
```

```
    for (int angle = 0; angle < 360; angle += 10) {  // Complete a circle
in steps
      // Simulate circular motion by adjusting elbow and wrist
      switch4 = 90 + 10 * cos(angle * PI / 180);  // Elbow - adjust for
circular motion
      switch3 = 90 + 10 * sin(angle * PI / 180);  // Wrist - adjust for
circular motion

      // Optionally, slightly rotate the base to enhance the circular
drawing effect
      switch5 = 90 + 10 * cos(angle * PI / 180);  // Base - slight
rotation
      switch6 = 90 + 30 * sin(angle * PI / 180);  // Base - slight
rotation

      moveServos();
    }
  }
  smoothing = previousValue;
}

void worm() {
  Serial.println("");
  Serial.println("Worm");

  switch6 = 90;
  switch5 = 130;
  switch4 = 90;
  switch3 = 90;
  moveServos();
  delay(1000);

  float previousValue = smoothing;
  smoothing = 0.50;

  for (int i = 0; i < 5; i++) {  // Draw 5 circles
    for (int angle = 0; angle < 360; angle += 10) {

      switch4 = 90 + 30 * cos(angle * PI / 180);
      switch3 = 90 + 30 * sin(angle * PI / 180);
      switch5 = 90 + 15 * sin(angle * PI / 180);

      moveServos();
    }
  }
```

```
  smoothing = previousValue;
}

void diagonals() {
  Serial.println("");
  Serial.println("Diagonals");

  switch6 = 90;
  switch5 = 130;  // Base position - slightly rotate if needed for better
motion
  switch4 = 90;   // Elbow - mid position
  switch3 = 90;   // Wrist - starting position
  moveServos();
  delay(1000);  // Wait a bit before starting the circles

  float previousValue = smoothing;
  smoothing = 0.50;

  for (int i = 0; i < 5; i++) {  // Draw 5 circles
    for (int angle = 0; angle < 360; angle += 10) {

      switch3 = 90 + 15 * sin(angle * PI / 180);
      switch6 = 90 + 30 * sin(angle * PI / 180);
      moveServos();
    }
  }
  smoothing = previousValue;
}

void screwdriver() {

  Serial.println("");
  Serial.println("Screw Driver");
  gripper = 180;
  switch2 = 90;
  switch3 = 180;
  switch4 = 140;
  switch5 = 45;
  switch6 = 0;
  moveServos();

  switch3 = 160;
  switch4 = 120;
  switch5 = 30;
  moveServos();
```

```
  gripper = 0;
  moveServos();

  smoothing = 0.01;
  delay(500);

  switch4 = 140;
  switch5 = 45;
  moveServos();

  switch3 = 70;
  switch4 = 90;
  switch5 = 110;
  switch6 = 90;
  moveServos();
  delay(1000);

  gripper = 0;
  switch2 = 90;
  switch3 = 120;
  switch4 = 90;
  switch5 = 110;
  switch6 = 90;
  moveServos();

  gripper = 180;
  moveServos();

  smoothing = 0.03;
}

void demo() {
  home();
  ready();
  home();
  ready();
  home();
  ready();
  home();
  diagonals();
  worm();
  circles();
}
```