ENGINEERING

McMaster University

Project three – Revenge of the Recycling System:

Design a System for Sorting and Recycling Containers

ENGINEER 1P13 – Integrated Cornerstone Design Projects

Tutorial 07

Team Thurs-02

Ayush Patel (patea202)

Rayyan Suhail (suhailr)

Peter Hull (hullp1)

Mohammad Hadi (hadim4)

Tony Han (han18)

Submitted: March 6, 2022

Table of Contents

Academic Integrity Statement

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Ayush Patel 400381354

Ayush

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Rayyan Suhail 400357184

Rayyan S

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Tony Han 400368851 Click or tap here to enter text.

X ____________

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Mohammad Hadi 400383319

X ________________

The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Peter Hull 400368981

X ________________

Executive Summary

Project three challenged us to design and create a system that sorts recycling containers into their appropriate containers. To achieve this, we were split into two sub-teams; modeling and computing, both of which will be discussed in more detail in the prevailing paragraphs. The terrestrial drone was available to be fitted with various types of sensors, which would sort the containers through differentiation. Our main objective was to make this system efficient whilst using the least amount of material possible.

On the computation side, the goal was to produce an efficient program that can sort and transfer a wide variety of containers into the right bin. The process of producing the final code went from making flowcharts and pseudocode to continuous refinement of code using the Q-labs environment. We were given the freedom to choose from a variety of sensors. We did short research on several types of sensors by comparing their working principle, advantages, and disadvantages and understanding their attributes. We chose the color sensor because its working is simple and efficient and works well in short range. We first worked on running one cycle efficiently. Functions were made which would perform the subtasks of dispensing containers, transferring containers into Q-bot using Q-arm, moving the Q-bot near an assigned bin, dumbing containers, and returning home in chronological order. In the end, the conditions for dispensing conditions were added and vigorous refinement of code with test cases was done to conclude the making of the code.

The main objective for the modeling sub-team was to create a mechanism that would rotate a recycling hopper to deposit containers into a recycling bin, it had to fit into a certain area on the robot. We were given a choice of two actuators: linear and rotary; we selected the linear actuator due to its simple implementation. Our initial design was based around the idea of a scissor-lift which in brief consisted of two arms aligned diagonally on the y-plane using slider rails and having an opposed motion to lift the baseplate. After 3D-printing the first set of rails and arms we observed that the lift force produced from only one set was enough to raise and lower the baseplate when fully loaded. We decided to dismiss the printing of the second set and conserve material. The M4 screws were used in attaching our mechanism to the baseplate and hopper, similarly magnets were used to attach the hopper to the baseplate. Overall, the design and mechanism worked coordinated with the program and completed their tasks; there were still improvements to be made, the main one being, replacing minute connecting parts with more accessible ones.

After looking into the workings of computation and modeling teams, we can interpret that we were successfully able to integrate the physical and the computational/logical components into producing a functioning mechanism. Our system was able to effectively transfer, and sort containers based on container material type using the principles of assembly modeling and smart devices.

References

[1] WatElectronics, “Light dependent resistor (LDR) - working principle and its applications,” WatElectronics.com, 06-Sep-2021. [Online]. Available: https://www.watelectronics.com/light-dependent-resistor-ldr-with-applications/#:~:text=Working%20Principle%20of%20LDR&text=These%20devices%20depend%20on%20the,light%20its%20resistance%20will%20decrease. [Accessed: 18-Jan-2022].

[2] N. *, “Light dependent resistor : Circuit diagram, types, working & applications,” ElProCus, 02-Apr-2021. [Online]. Available: https://www.elprocus.com/ldr-light-dependent-resistor-circuit-and-working/. [Accessed: 20-Jan-2022].

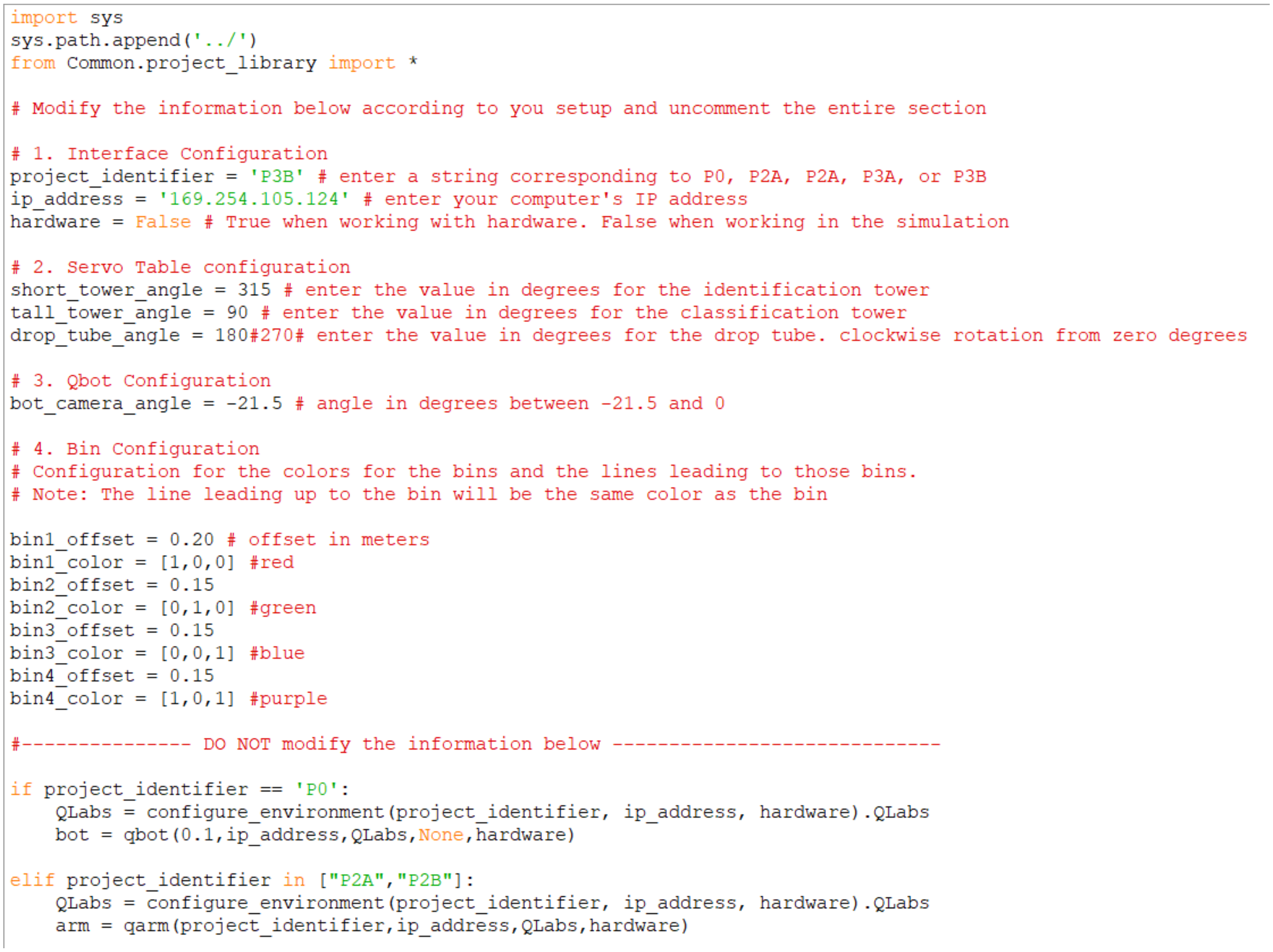
[3] D. Jost, “What is an IR sensor?,” Fierce Electronics, 30-Jul-2019. [Online]. Available:https://www.fierceelectronics.com/sensors/what-ir-sensor#:~:text=Active%20IR%20sensors%20have%20two,is%20detected%20by%20the%20receiver.&text=Passive%20infrared%20(PIR)%20sensors%20only,emit%20it%20from%20an%20LED. [Accessed: 18-Jan-2022].

[4] 2 years ago, Shawn, S. S. author's posts, and S. author's posts, “Types of distance sensors and how to select one?,” Latest Open Tech From Seeed, 29-Jun-2021. [Online].Available:https://www.seeedstudio.com/blog/2019/12/23/distance-sensors-types-and-selection-guide/. [Accessed: 20-Jan-2022].

[5] “Detection based on ‘light‘what is a colour sensor?,” *KEYENCE*. [Online]. Available:https://www.keyence.ca/ss/products/sensor/sensorbasics/color/info/#:~:text=A%20colour%20sensor%20is%20a,detection%20object%20with%20a%20receiver. [Accessed: 21-Jan-2022]

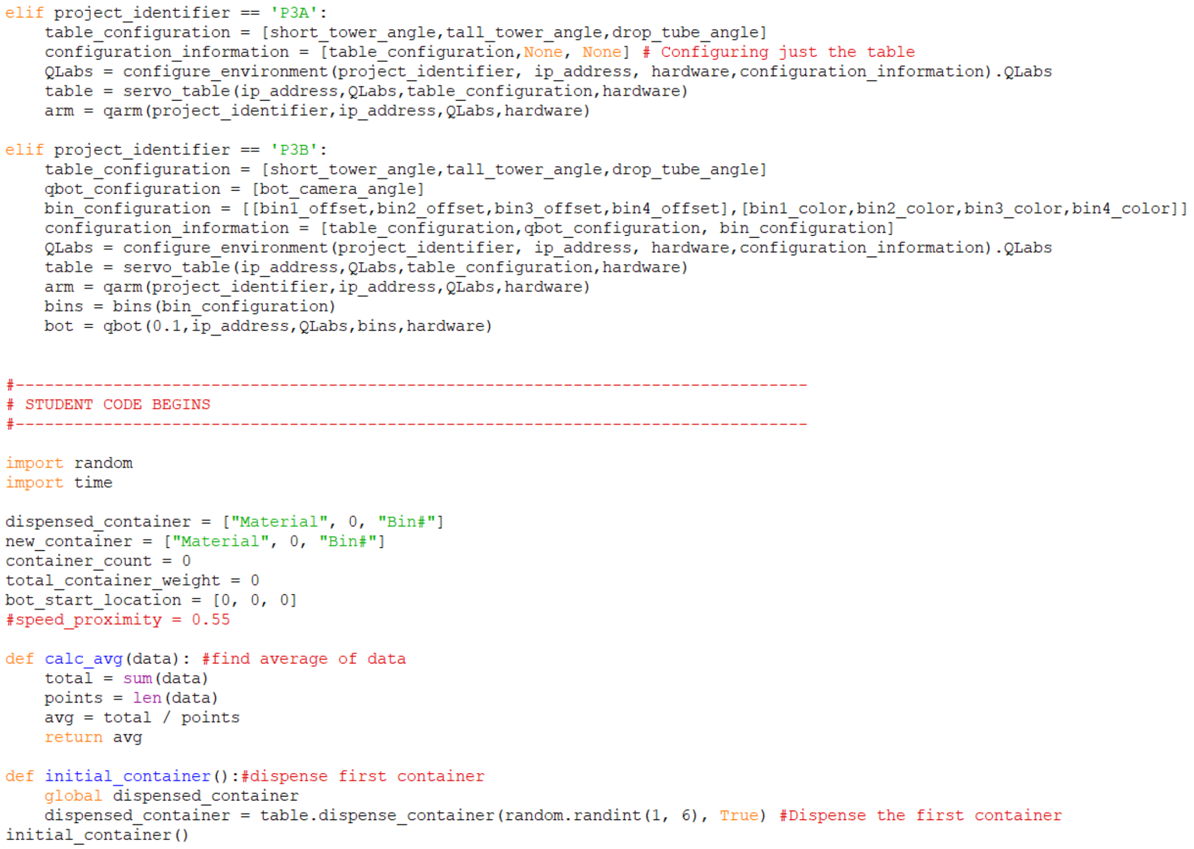
[6] R. Burnett, “Understanding how ultrasonic sensors work,” Understanding How Ultrasonic Sensors Work Comments, 04-Mar-2021. [Online].
Available: https://www.maxbotix.com/articles/how-ultrasonic-sensors-work.htm. [Accessed: 21-jan- 2022]

Appendix A – Supporting documents



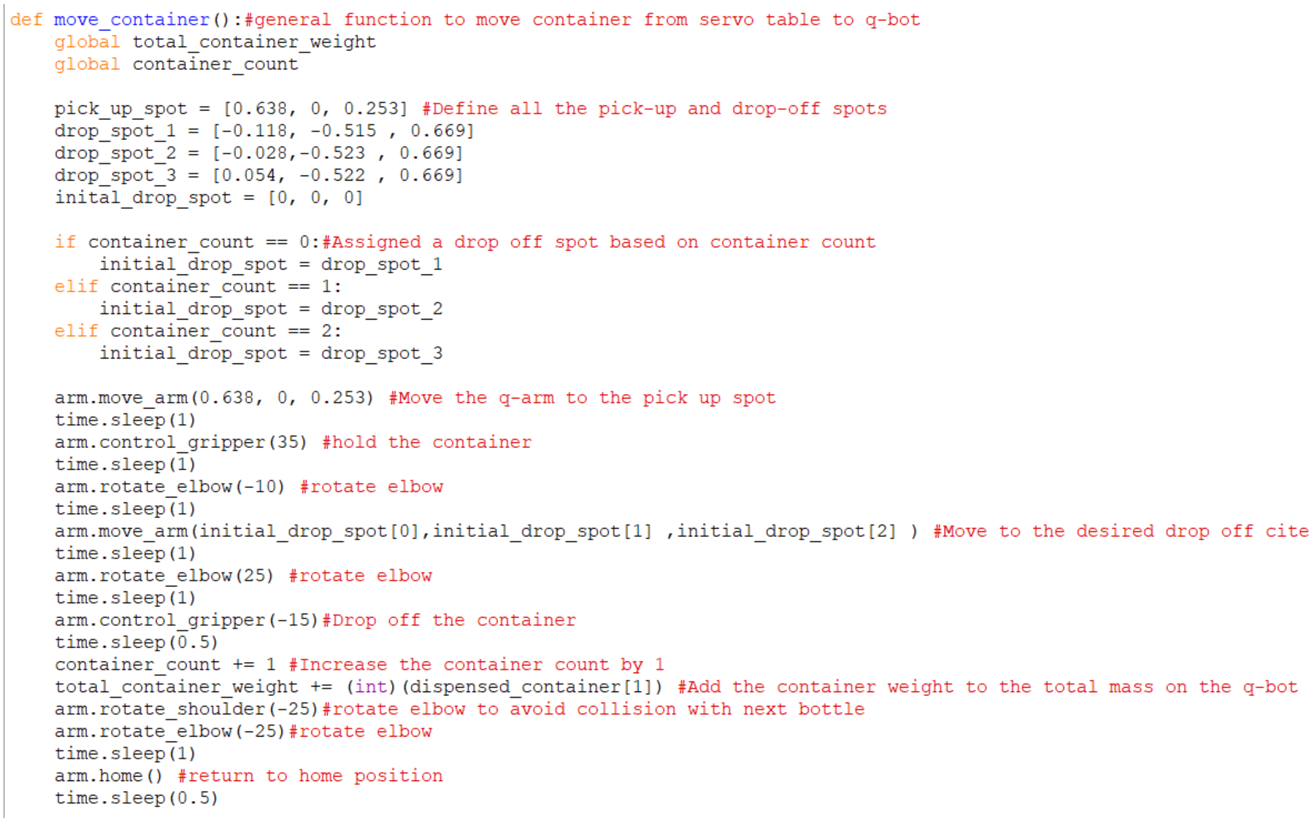
```
import sys
sys.path.append('../')
from Common.project_library import *

# Modify the information below according to you setup and uncomment the entire section

# 1. Interface Configuration
project_identifier = 'P3B' # enter a string corresponding to P0, P2A, P2A, P3A, or P3B
ip_address = '169.254.105.124' # enter your computer's IP address
hardware = False # True when working with hardware. False when working in the simulation

# 2. Servo Table configuration
short_tower_angle = 315 # enter the value in degrees for the identification tower
tall_tower_angle = 90 # enter the value in degrees for the classification tower
drop_tube_angle = 180#270# enter the value in degrees for the drop tube. clockwise rotation from zero degrees

# 3. Qbot Configuration
bot_camera_angle = -21.5 # angle in degrees between -21.5 and 0

# 4. Bin Configuration
# Configuration for the colors for the bins and the lines leading to those bins.
# Note: The line leading up to the bin will be the same color as the bin

bin1_offset = 0.20 # offset in meters
bin1_color = [1,0,0] #red
bin2_offset = 0.15
bin2_color = [0,1,0] #green
bin3_offset = 0.15
bin3_color = [0,0,1] #blue
bin4_offset = 0.15
bin4_color = [1,0,1] #purple

#--------------- DO NOT modify the information below -----------------------------

if project_identifier == 'P0':
    QLabs = configure_environment(project_identifier, ip_address, hardware).QLabs
    bot = qbot(0.1,ip_address,QLabs,None,hardware)

elif project_identifier in ["P2A","P2B"]:
    QLabs = configure_environment(project_identifier, ip_address, hardware).QLabs
    arm = qarm(project_identifier,ip_address,QLabs,hardware)
```

Figure 1. Code



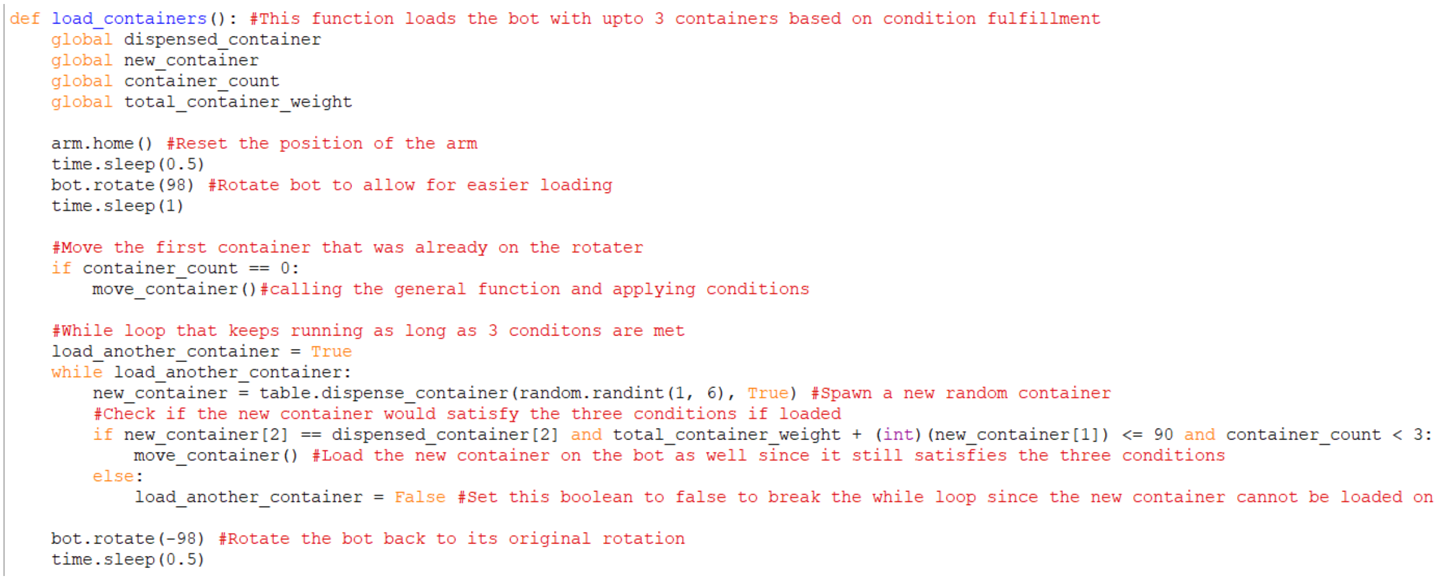
```
elif project_identifier == 'P3A':
    table_configuration = [short_tower_angle,tall_tower_angle,drop_tube_angle]
    configuration_information = [table_configuration,None, None] # Configuring just the table
    QLabs = configure_environment(project_identifier, ip_address, hardware,configuration_information).QLabs
    table = servo_table(ip_address,QLabs,table_configuration,hardware)
    arm = qarm(project_identifier,ip_address,QLabs,hardware)

elif project_identifier == 'P3B':
    table_configuration = [short_tower_angle,tall_tower_angle,drop_tube_angle]
    qbot_configuration = [bot_camera_angle]
    bin_configuration = [[bin1_offset,bin2_offset,bin3_offset,bin4_offset],[bin1_color,bin2_color,bin3_color,bin4_color]]
    configuration_information = [table_configuration,qbot_configuration, bin_configuration]
    QLabs = configure_environment(project_identifier, ip_address, hardware,configuration_information).QLabs
    table = servo_table(ip_address,QLabs,table_configuration,hardware)
    arm = qarm(project_identifier,ip_address,QLabs,hardware)
    bins = bins(bin_configuration)
    bot = qbot(0.1,ip_address,QLabs,bins,hardware)


#--------------------------------------------------------------------------------
# STUDENT CODE BEGINS
#--------------------------------------------------------------------------------

import random
import time

dispensed_container = ["Material", 0, "Bin#"]
new_container = ["Material", 0, "Bin#"]
container_count = 0
total_container_weight = 0
bot_start_location = [0, 0, 0]
#speed_proximity = 0.55

def calc_avg(data): #find average of data
    total = sum(data)
    points = len(data)
    avg = total / points
    return avg

def initial_container():#dispense first container
    global dispensed_container
    dispensed_container = table.dispense_container(random.randint(1, 6), True) #Dispense the first container
initial_container()
```

Figure 2. Code



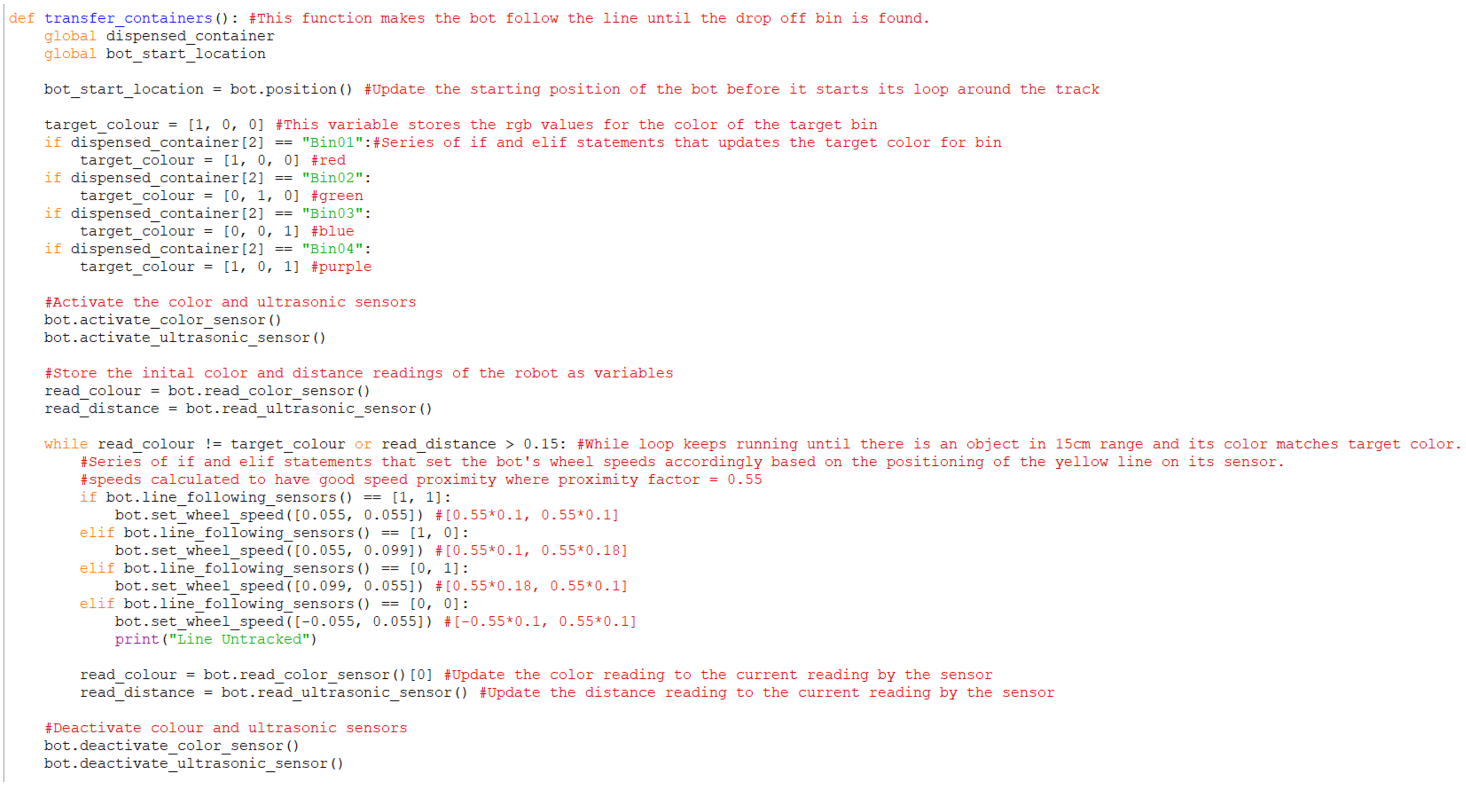
```
def move_container():#general function to move container from servo table to q-bot
    global total_container_weight
    global container_count

    pick_up_spot = [0.638, 0, 0.253] #Define all the pick-up and drop-off spots
    drop_spot_1 = [-0.118, -0.515 , 0.669]
    drop_spot_2 = [-0.028,-0.523 , 0.669]
    drop_spot_3 = [0.054, -0.522 , 0.669]
    inital_drop_spot = [0, 0, 0]

    if container_count == 0:#Assigned a drop off spot based on container count
        initial_drop_spot = drop_spot_1
    elif container_count == 1:
        initial_drop_spot = drop_spot_2
    elif container_count == 2:
        initial_drop_spot = drop_spot_3

    arm.move_arm(0.638, 0, 0.253) #Move the q-arm to the pick up spot
    time.sleep(1)
    arm.control_gripper(35) #hold the container
    time.sleep(1)
    arm.rotate_elbow(-10) #rotate elbow
    time.sleep(1)
    arm.move_arm(initial_drop_spot[0],initial_drop_spot[1] ,initial_drop_spot[2] ) #Move to the desired drop off cite
    time.sleep(1)
    arm.rotate_elbow(25) #rotate elbow
    time.sleep(1)
    arm.control_gripper(-15)#Drop off the container
    time.sleep(0.5)
    container_count += 1 #Increase the container count by 1
    total_container_weight += (int)(dispensed_container[1]) #Add the container weight to the total mass on the q-bot
    arm.rotate_shoulder(-25)#rotate elbow to avoid collision with next bottle
    arm.rotate_elbow(-25)#rotate elbow
    time.sleep(1)
    arm.home() #return to home position
    time.sleep(0.5)
```

Figure 3. Code



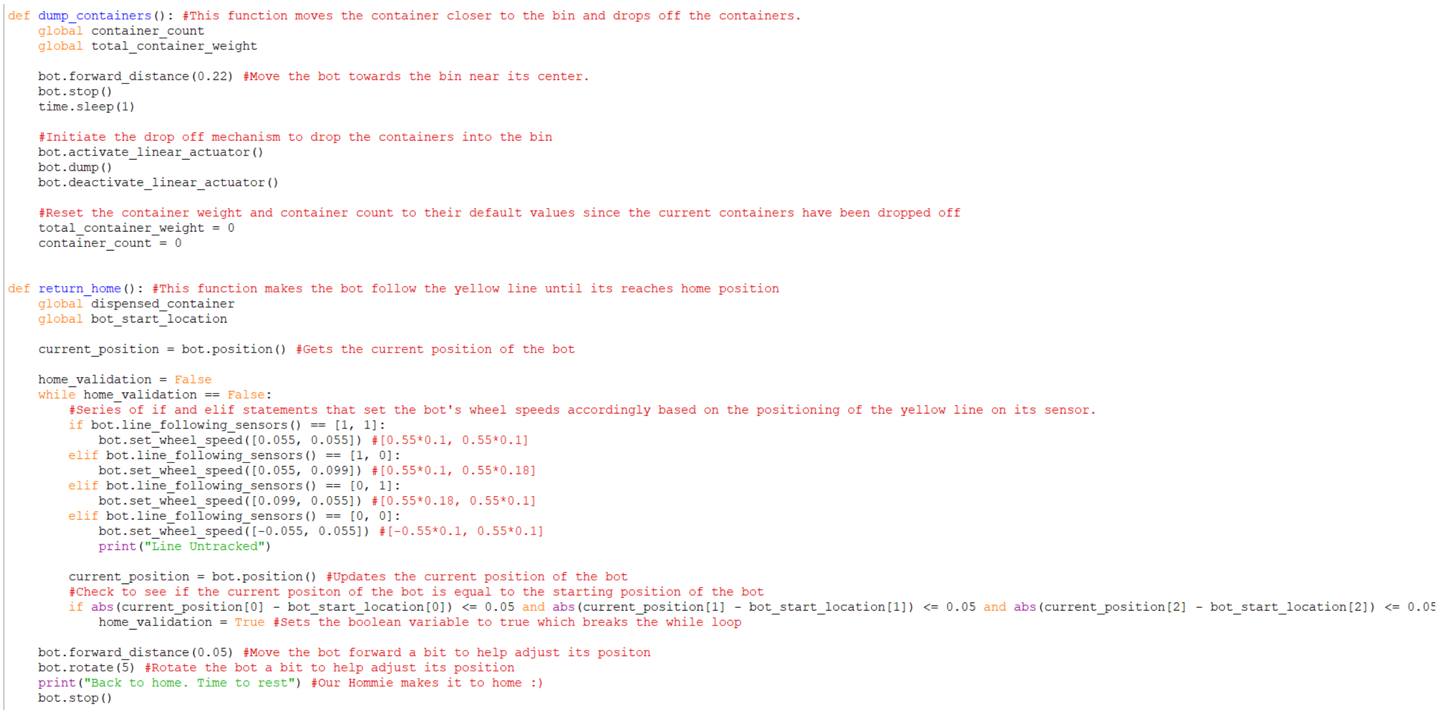
```
def load_containers(): #This function loads the bot with upto 3 containers based on condition fulfillment
    global dispensed_container
    global new_container
    global container_count
    global total_container_weight

    arm.home() #Reset the position of the arm
    time.sleep(0.5)
    bot.rotate(98) #Rotate bot to allow for easier loading
    time.sleep(1)

    #Move the first container that was already on the rotater
    if container_count == 0:
        move_container()#calling the general function and applying conditions

    #While loop that keeps running as long as 3 conditons are met
    load_another_container = True
    while load_another_container:
        new_container = table.dispense_container(random.randint(1, 6), True) #Spawn a new random container
        #Check if the new container would satisfy the three conditions if loaded
        if new_container[2] == dispensed_container[2] and total_container_weight + (int)(new_container[1]) <= 90 and container_count < 3:
            move_container() #Load the new container on the bot as well since it still satisfies the three conditions
        else:
            load_another_container = False #Set this boolean to false to break the while loop since the new container cannot be loaded on

    bot.rotate(-98) #Rotate the bot back to its original rotation
    time.sleep(0.5)
```

Figure 4. Code



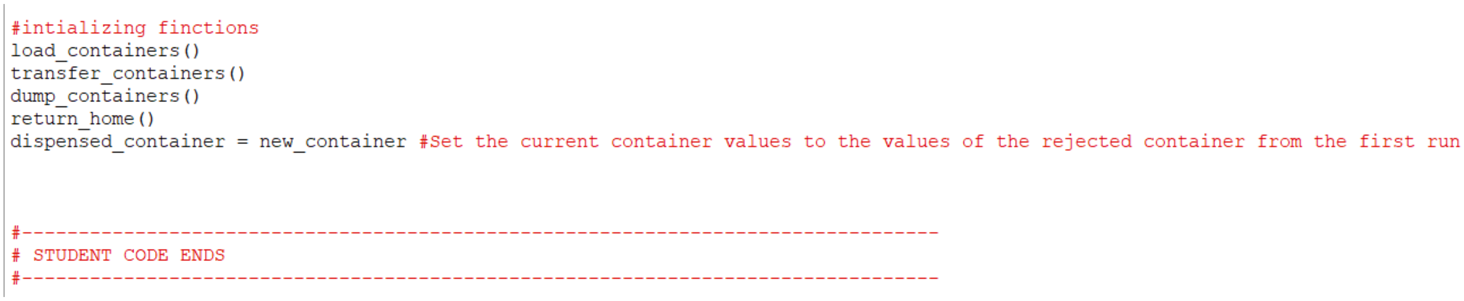
```
def transfer_containers(): #This function makes the bot follow the line until the drop off bin is found.
    global dispensed_container
    global bot_start_location

    bot_start_location = bot.position() #Update the starting position of the bot before it starts its loop around the track

    target_colour = [1, 0, 0] #This variable stores the rgb values for the color of the target bin
    if dispensed_container[2] == "Bin01":#Series of if and elif statements that updates the target color for bin
        target_colour = [1, 0, 0] #red
    if dispensed_container[2] == "Bin02":
        target_colour = [0, 1, 0] #green
    if dispensed_container[2] == "Bin03":
        target_colour = [0, 0, 1] #blue
    if dispensed_container[2] == "Bin04":
        target_colour = [1, 0, 1] #purple

    #Activate the color and ultrasonic sensors
    bot.activate_color_sensor()
    bot.activate_ultrasonic_sensor()

    #Store the inital color and distance readings of the robot as variables
    read_colour = bot.read_color_sensor()
    read_distance = bot.read_ultrasonic_sensor()

    while read_colour != target_colour or read_distance > 0.15: #While loop keeps running until there is an object in 15cm range and its color matches target color.
        #Series of if and elif statements that set the bot's wheel speeds accordingly based on the positioning of the yellow line on its sensor.
        #speeds calculated to have good speed proximity where proximity factor = 0.55
        if bot.line_following_sensors() == [1, 1]:
            bot.set_wheel_speed([0.055, 0.055]) #[0.55*0.1, 0.55*0.1]
        elif bot.line_following_sensors() == [1, 0]:
            bot.set_wheel_speed([0.055, 0.099]) #[0.55*0.1, 0.55*0.18]
        elif bot.line_following_sensors() == [0, 1]:
            bot.set_wheel_speed([0.099, 0.055]) #[0.55*0.18, 0.55*0.1]
        elif bot.line_following_sensors() == [0, 0]:
            bot.set_wheel_speed([-0.055, 0.055]) #[-0.55*0.1, 0.55*0.1]
            print("Line Untracked")

        read_colour = bot.read_color_sensor()[0] #Update the color reading to the current reading by the sensor
        read_distance = bot.read_ultrasonic_sensor() #Update the distance reading to the current reading by the sensor

    #Deactivate colour and ultrasonic sensors
    bot.deactivate_color_sensor()
    bot.deactivate_ultrasonic_sensor()
```

Figure 5. Code



```
def dump_containers(): #This function moves the container closer to the bin and drops off the containers.
    global container_count
    global total_container_weight

    bot.forward_distance(0.22) #Move the bot towards the bin near its center.
    bot.stop()
    time.sleep(1)

    #Initiate the drop off mechanism to drop the containers into the bin
    bot.activate_linear_actuator()
    bot.dump()
    bot.deactivate_linear_actuator()

    #Reset the container weight and container count to their default values since the current containers have been dropped off
    total_container_weight = 0
    container_count = 0


def return_home(): #This function makes the bot follow the yellow line until its reaches home position
    global dispensed_container
    global bot_start_location

    current_position = bot.position() #Gets the current position of the bot

    home_validation = False
    while home_validation == False:
        #Series of if and elif statements that set the bot's wheel speeds accordingly based on the positioning of the yellow line on its sensor.
        if bot.line_following_sensors() == [1, 1]:
            bot.set_wheel_speed([0.055, 0.055]) #[0.55*0.1, 0.55*0.1]
        elif bot.line_following_sensors() == [1, 0]:
            bot.set_wheel_speed([0.055, 0.099]) #[0.55*0.1, 0.55*0.18]
        elif bot.line_following_sensors() == [0, 1]:
            bot.set_wheel_speed([0.099, 0.055]) #[0.55*0.18, 0.55*0.1]
        elif bot.line_following_sensors() == [0, 0]:
            bot.set_wheel_speed([-0.055, 0.055]) #[-0.55*0.1, 0.55*0.1]
            print("Line Untracked")

        current_position = bot.position() #Updates the current position of the bot
        #Check to see if the current positon of the bot is equal to the starting position of the bot
        if abs(current_position[0] - bot_start_location[0]) <= 0.05 and abs(current_position[1] - bot_start_location[1]) <= 0.05 and abs(current_position[2] - bot_start_location[2]) <= 0.05
            home_validation = True #Sets the boolean variable to true which breaks the while loop

    bot.forward_distance(0.05) #Move the bot forward a bit to help adjust its positon
    bot.rotate(5) #Rotate the bot a bit to help adjust its position
    print("Back to home. Time to rest") #Our Hommie makes it to home :)
    bot.stop()
```

Figure 6. Code



```
#intializing finctions
load_containers()
transfer_containers()
dump_containers()
return_home()
dispensed_container = new_container #Set the current container values to the values of the rejected container from the first run


#----------------------------------------------------------------------------------
# STUDENT CODE ENDS
#----------------------------------------------------------------------------------
```

Figure 7. Code

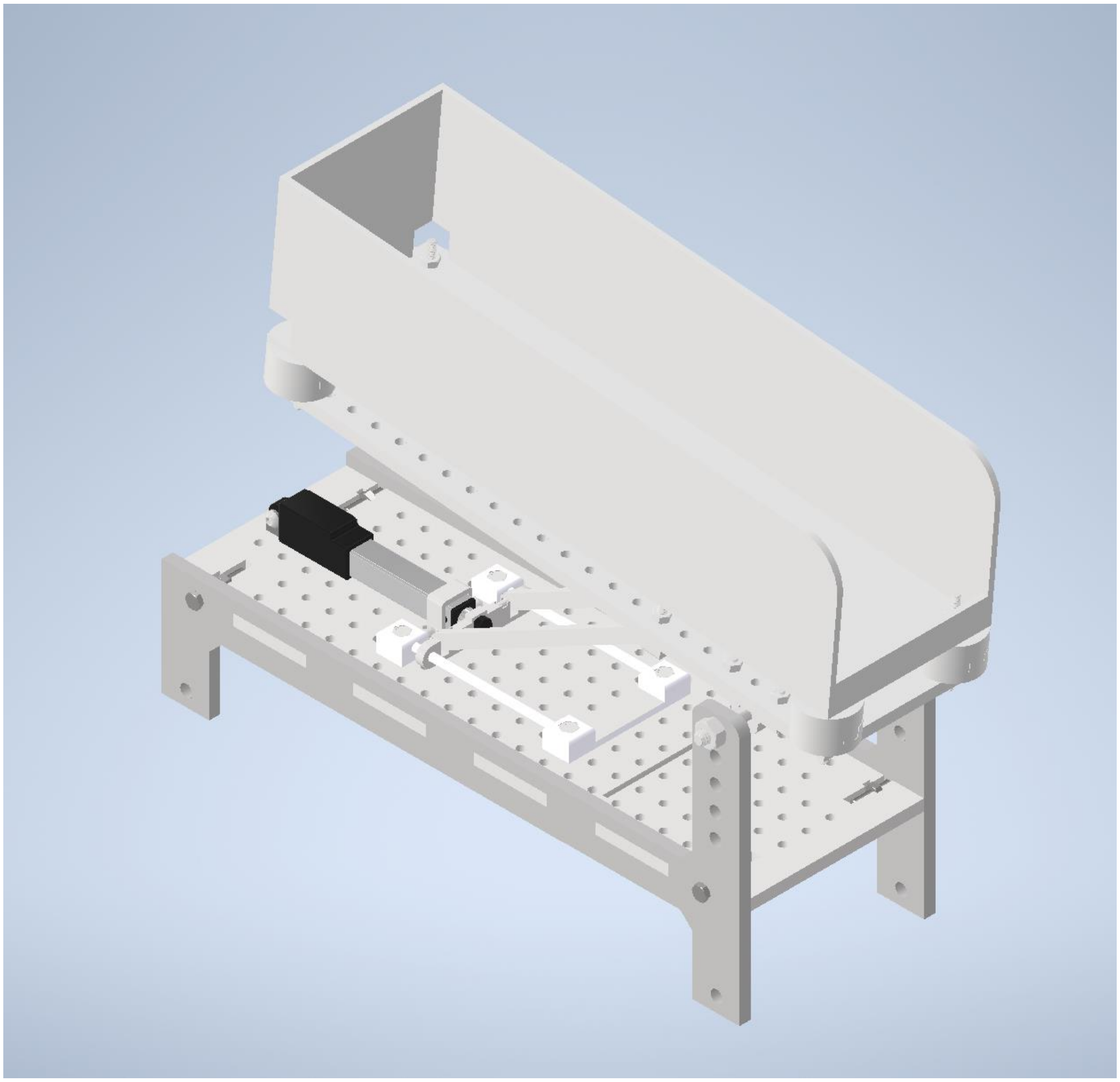


Figure 8. Overview of model

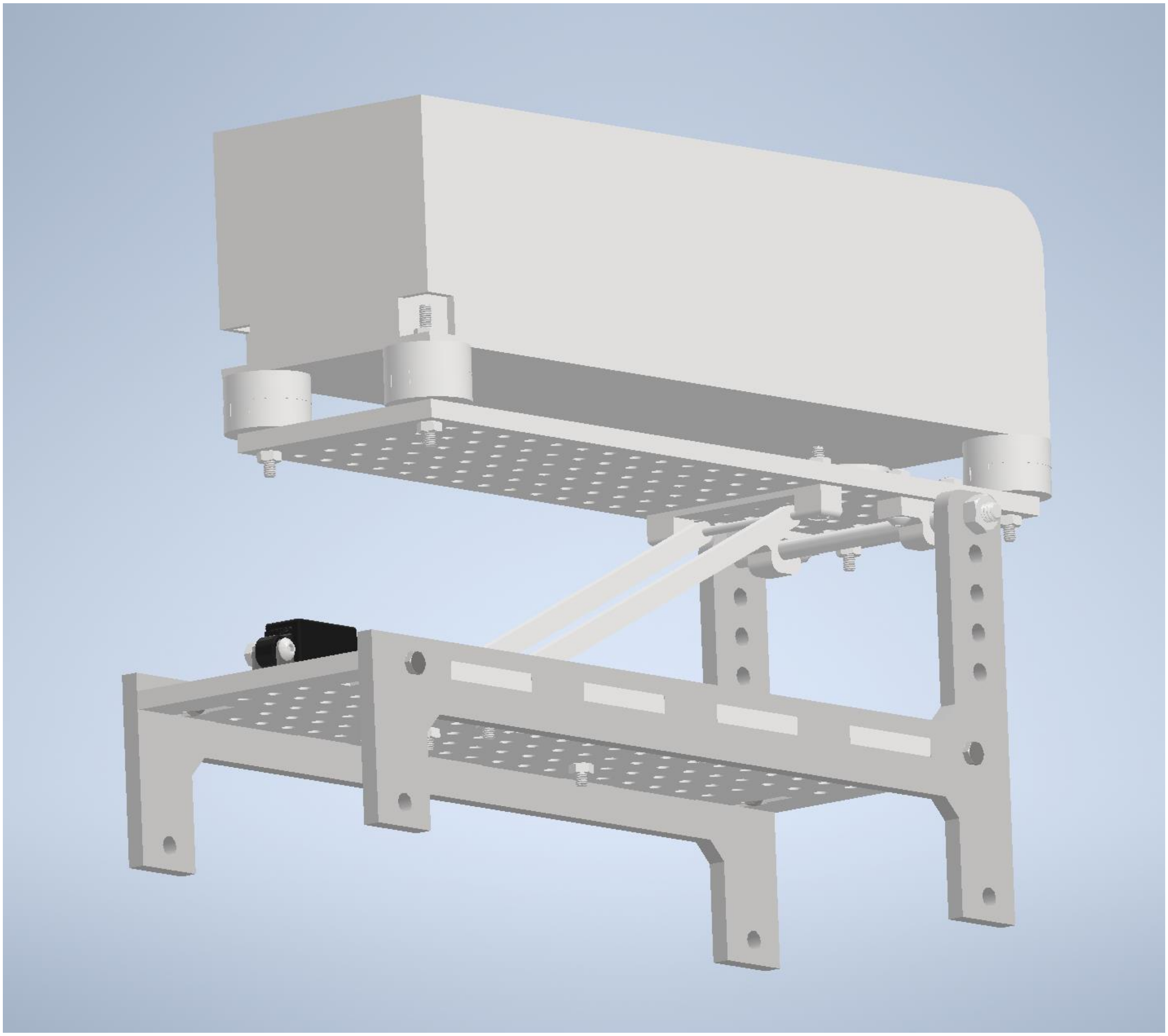


Figure 9. Underside of model

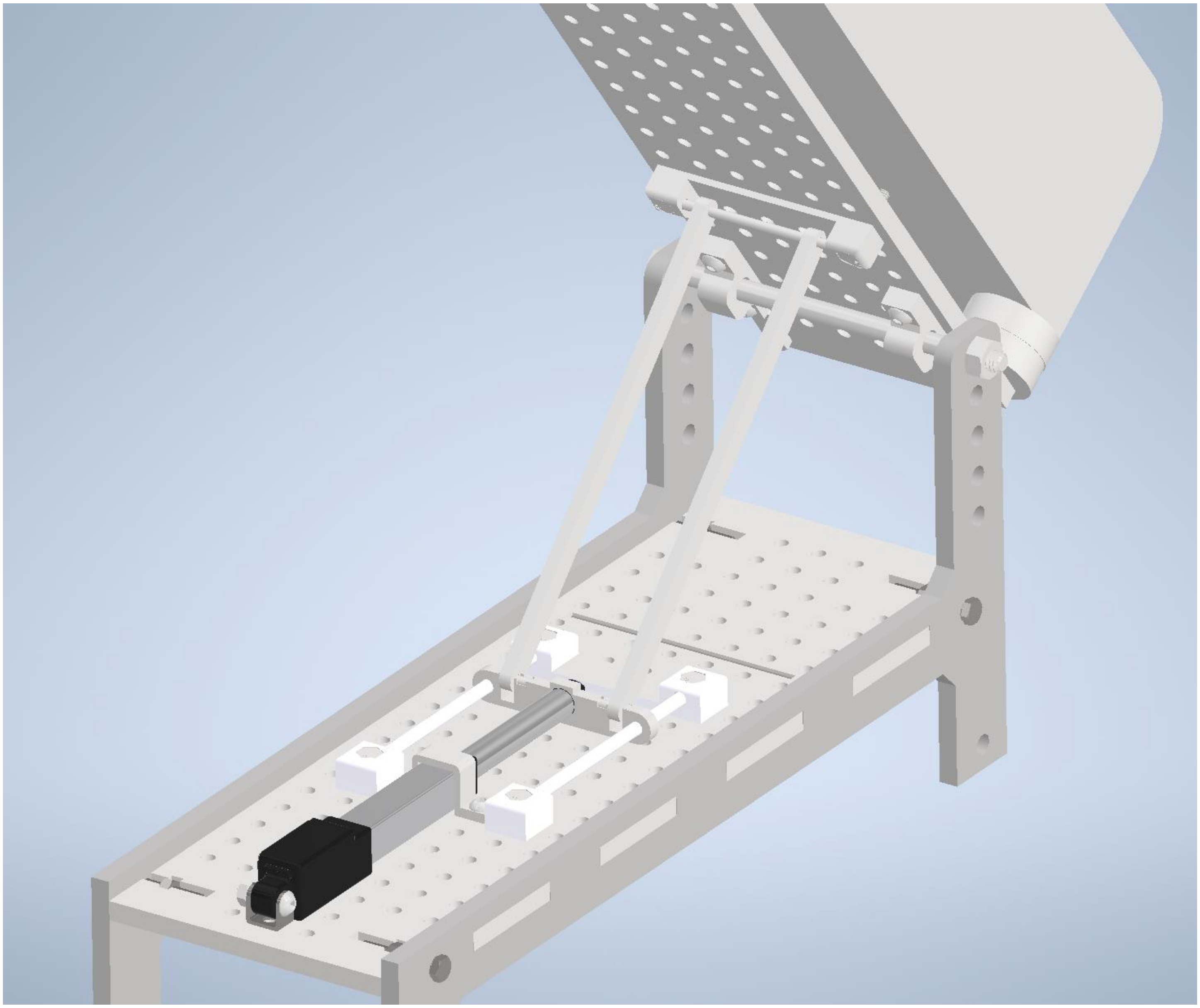


Figure 10. Mechanism in open position.

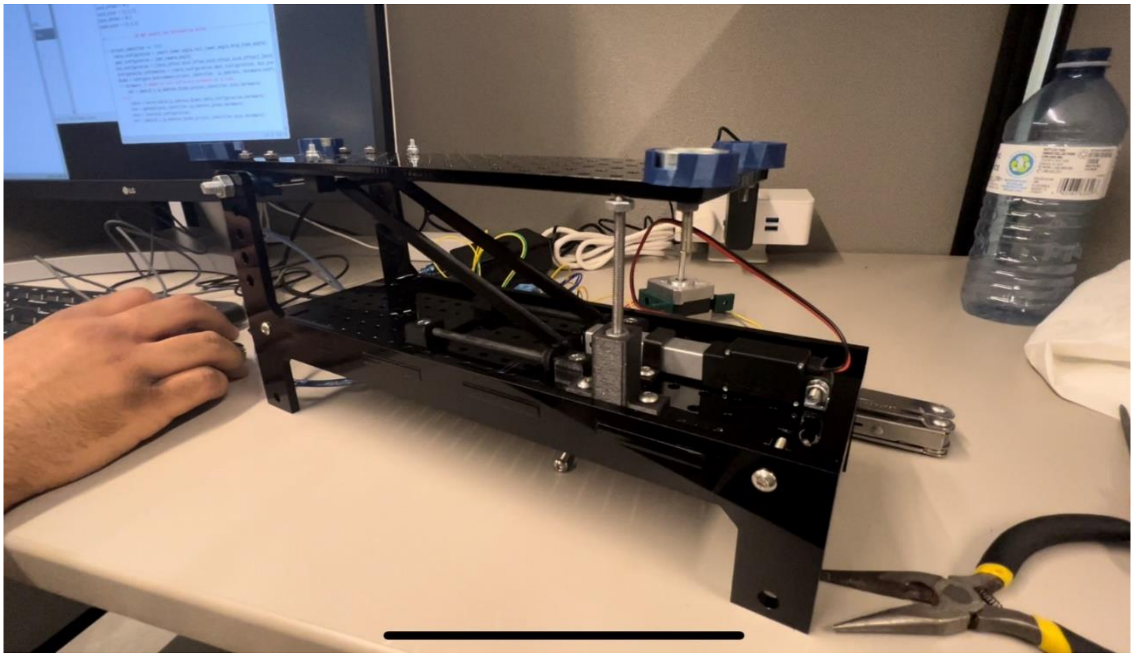


Figure 11. Mechanism prototype in closed position

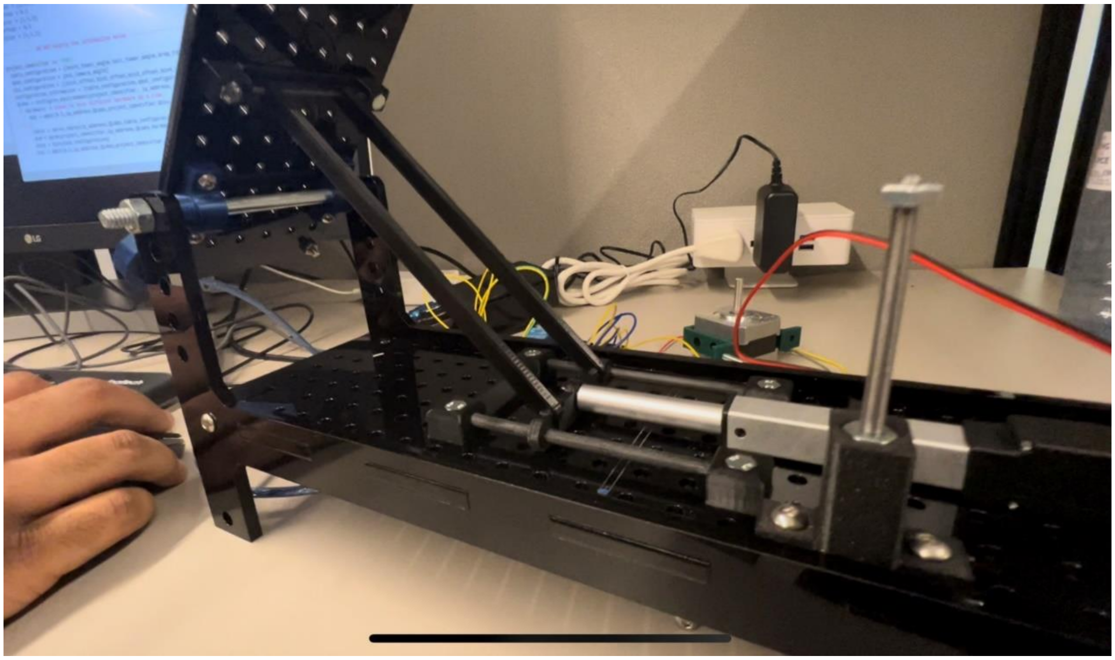


Figure 12. Mechanism prototype in open position.

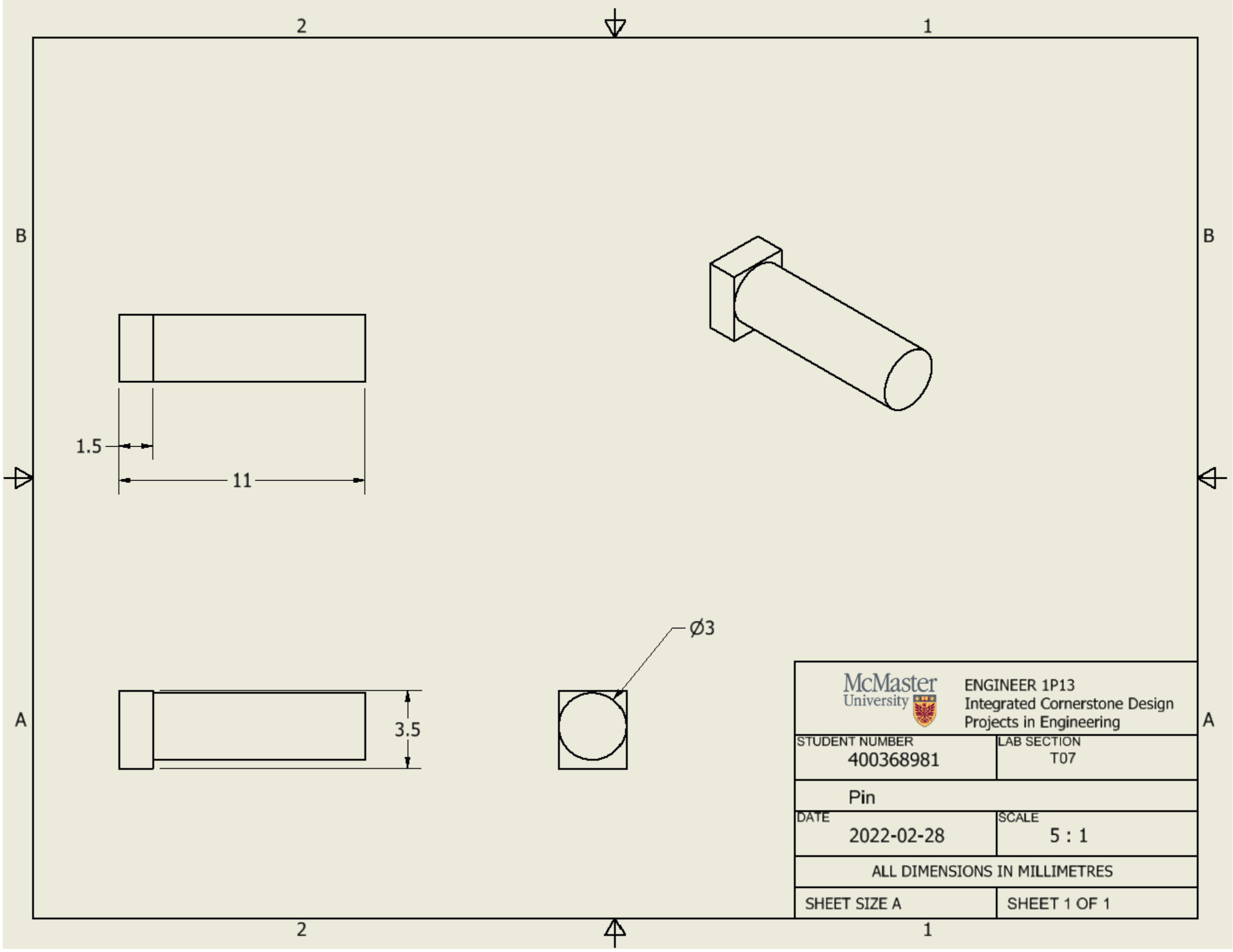
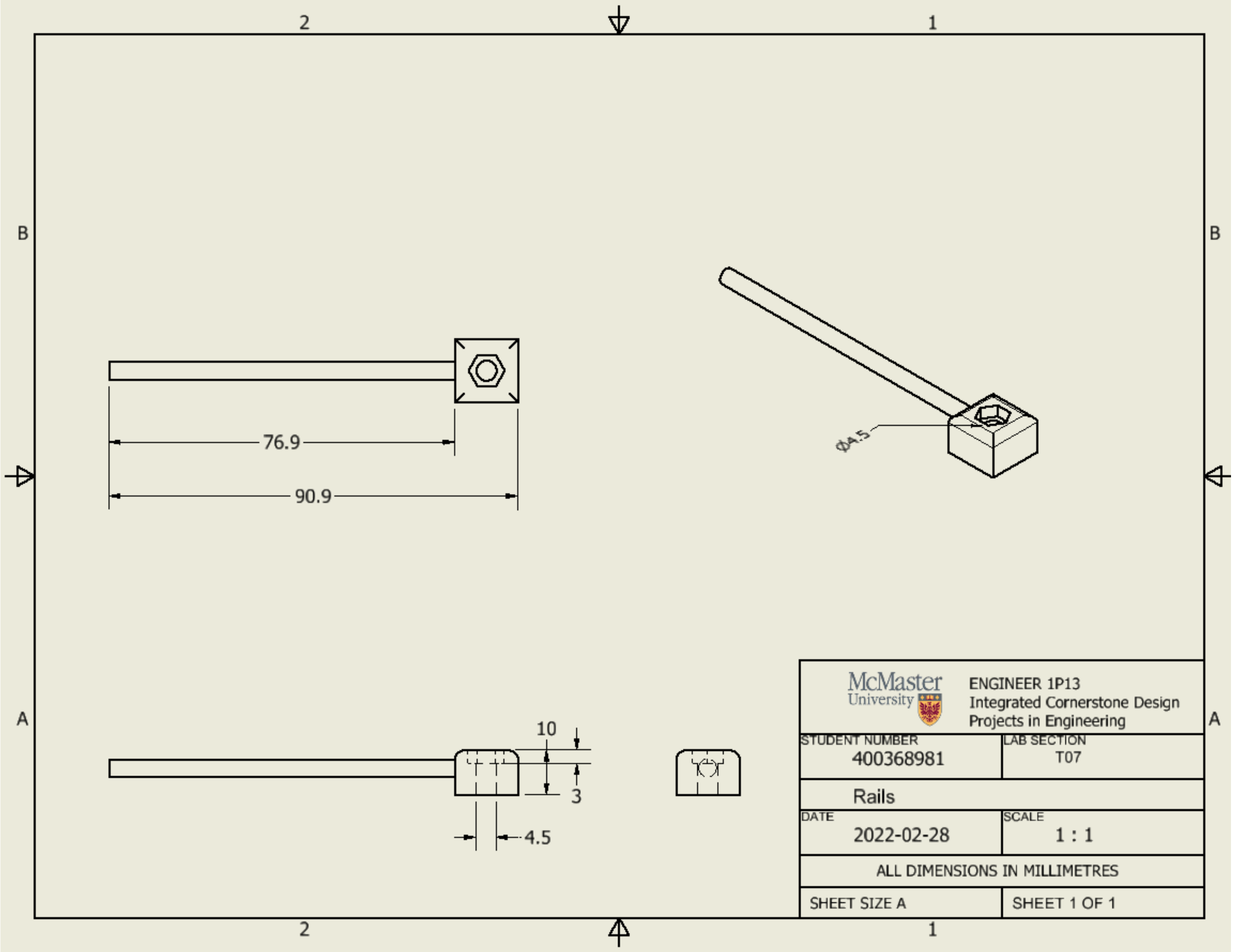


Figure 13. Engineering Drawing of Pin part.



McMaster University
ENGINEER 1P13
Integrated Cornerstone Design
Projects in Engineering

STUDENT NUMBER
400368981

LAB SECTION
T07

Rails

DATE
2022-02-28

SCALE
1 : 1

ALL DIMENSIONS IN MILLIMETRES

SHEET SIZE A

SHEET 1 OF 1

Figure 14. Engineering Drawing of Rail part.

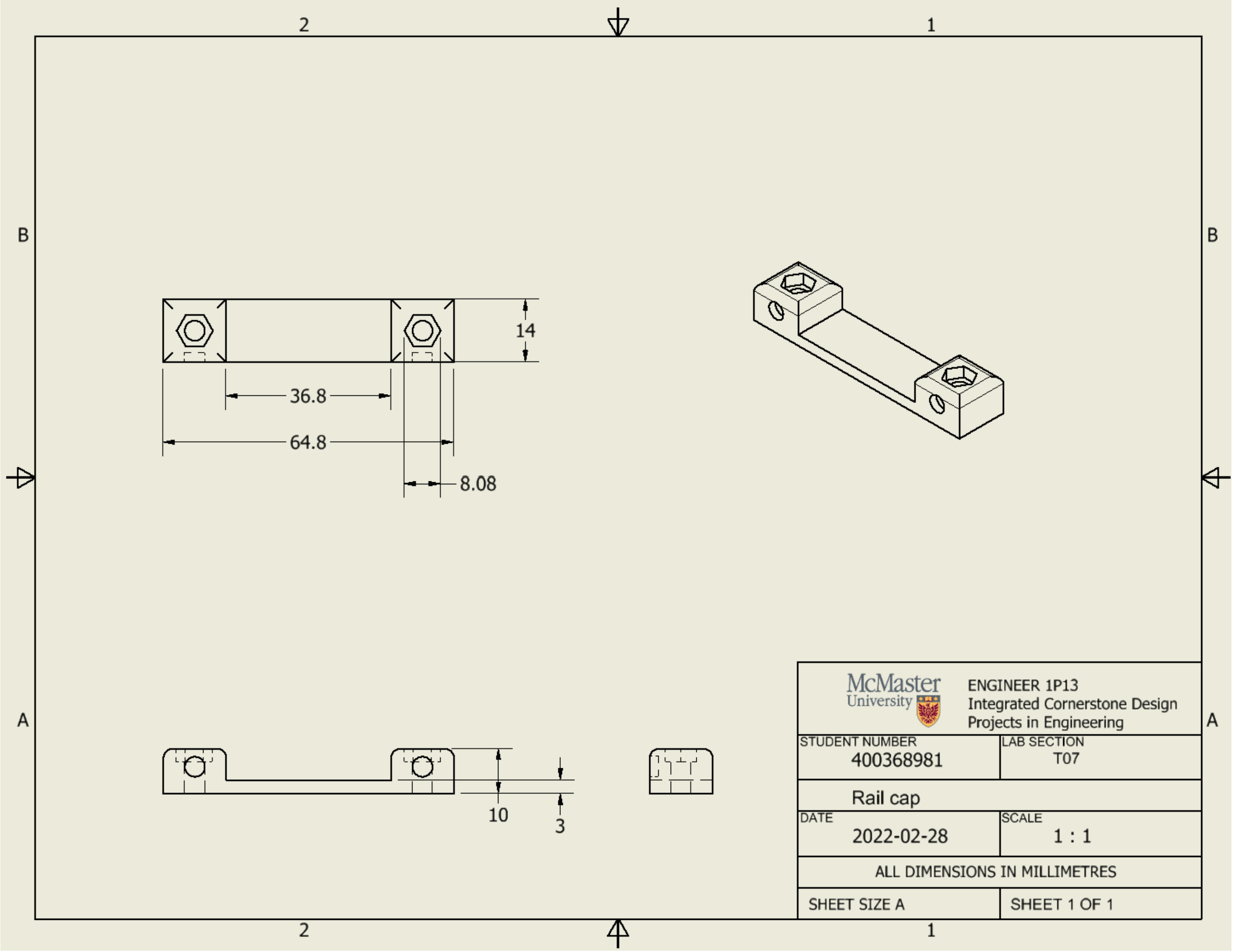


Figure 15. Engineering Drawing of Rail cap part

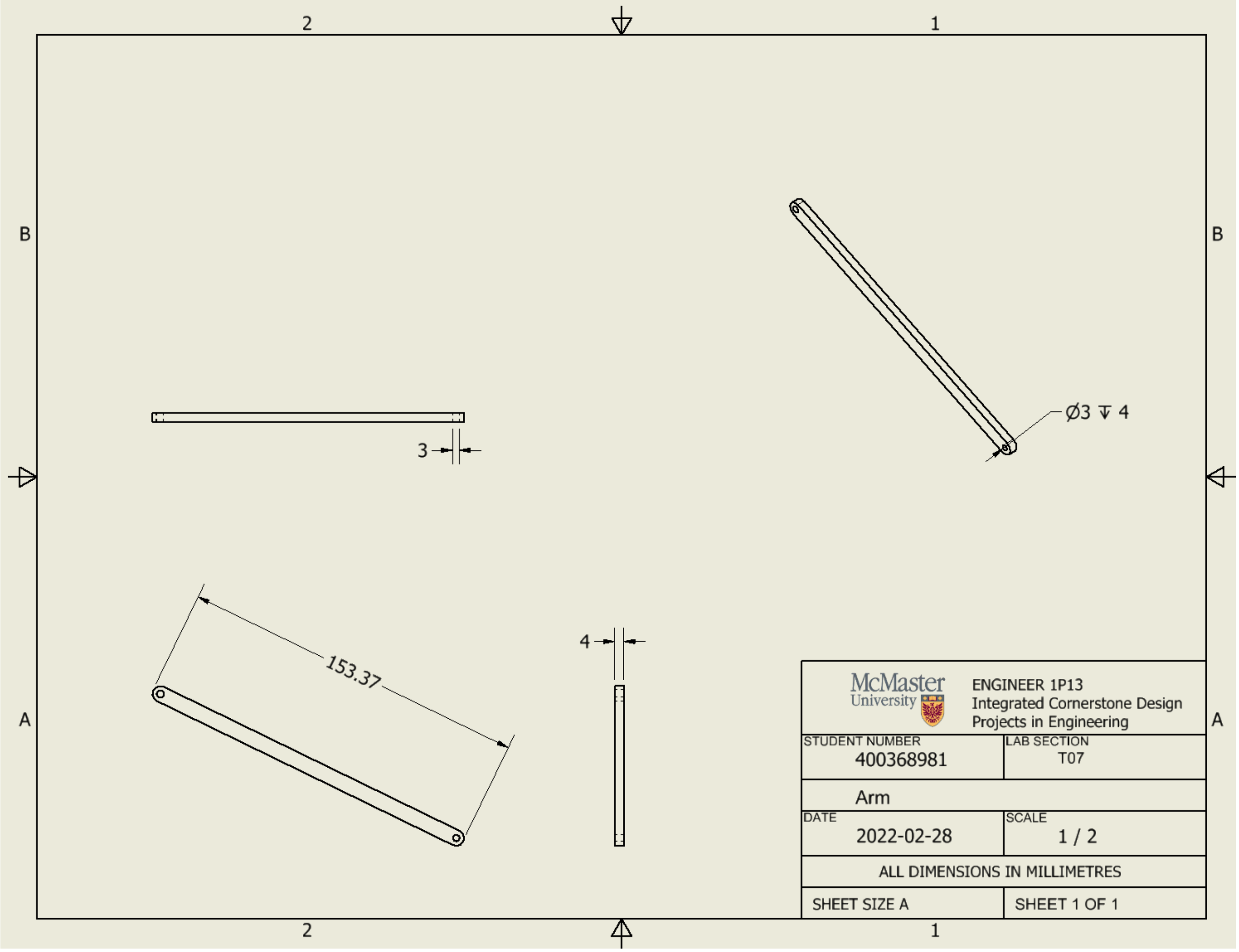


Figure 16. Engineering Drawing of Arm part.

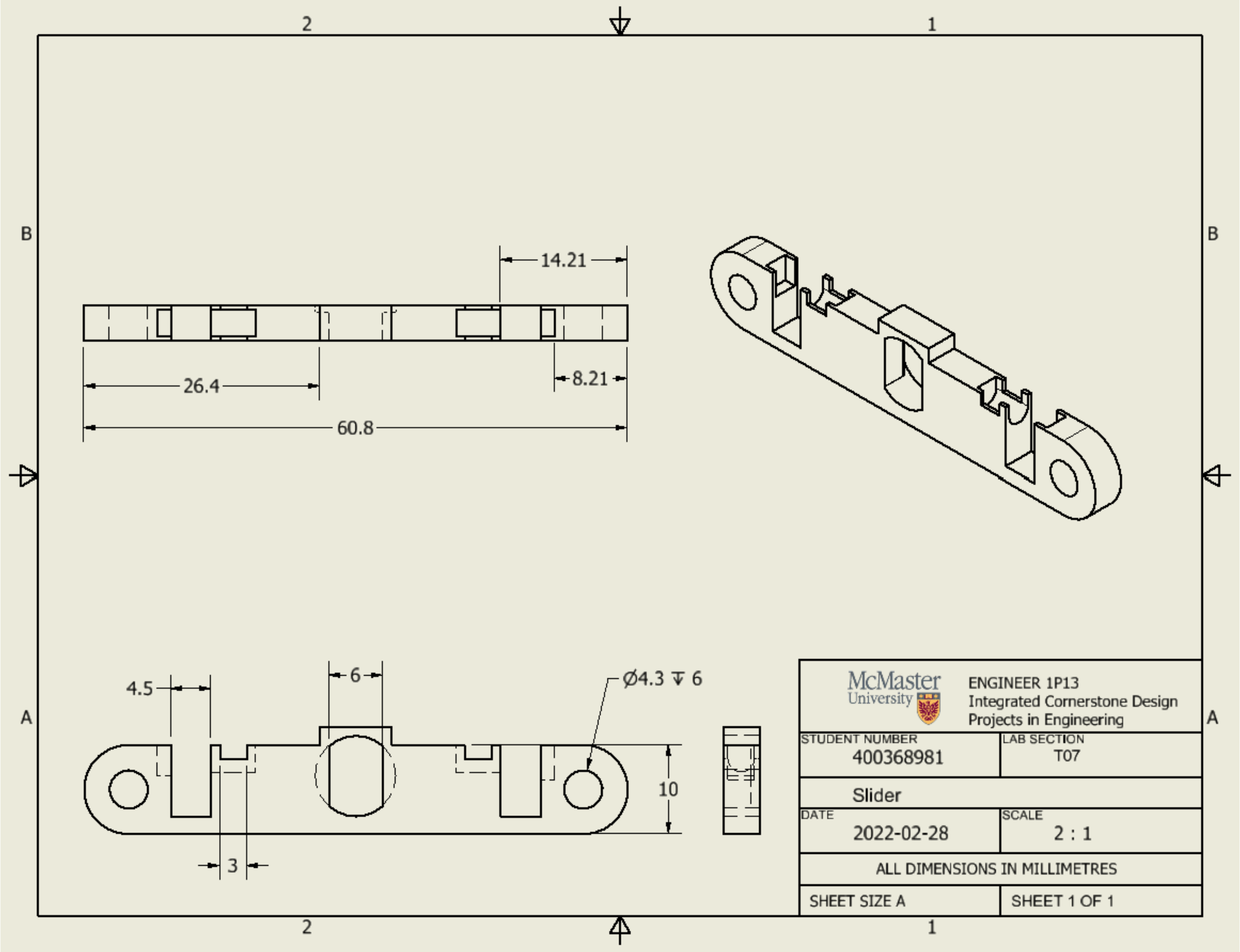
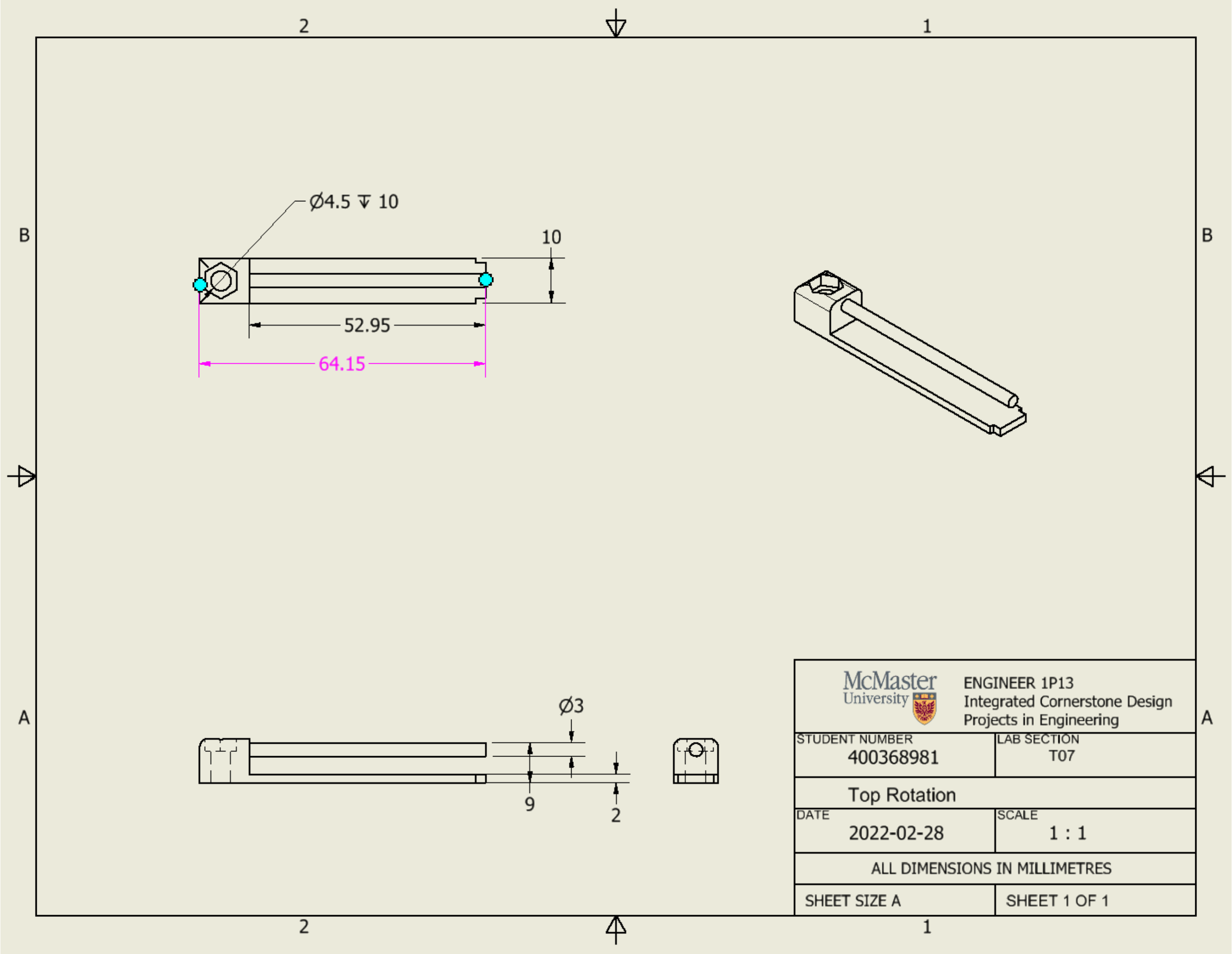


Figure 17. Engineering Drawing of Slider part.



ENGINEER 1P13
Integrated Cornerstone Design
Projects in Engineering

STUDENT NUMBER 400368981	LAB SECTION T07
Top Rotation	
DATE 2022-02-28	SCALE 1 : 1
ALL DIMENSIONS IN MILLIMETRES	
SHEET SIZE A	SHEET 1 OF 1

Figure 18. Engineering Drawing of Top Rotation part.

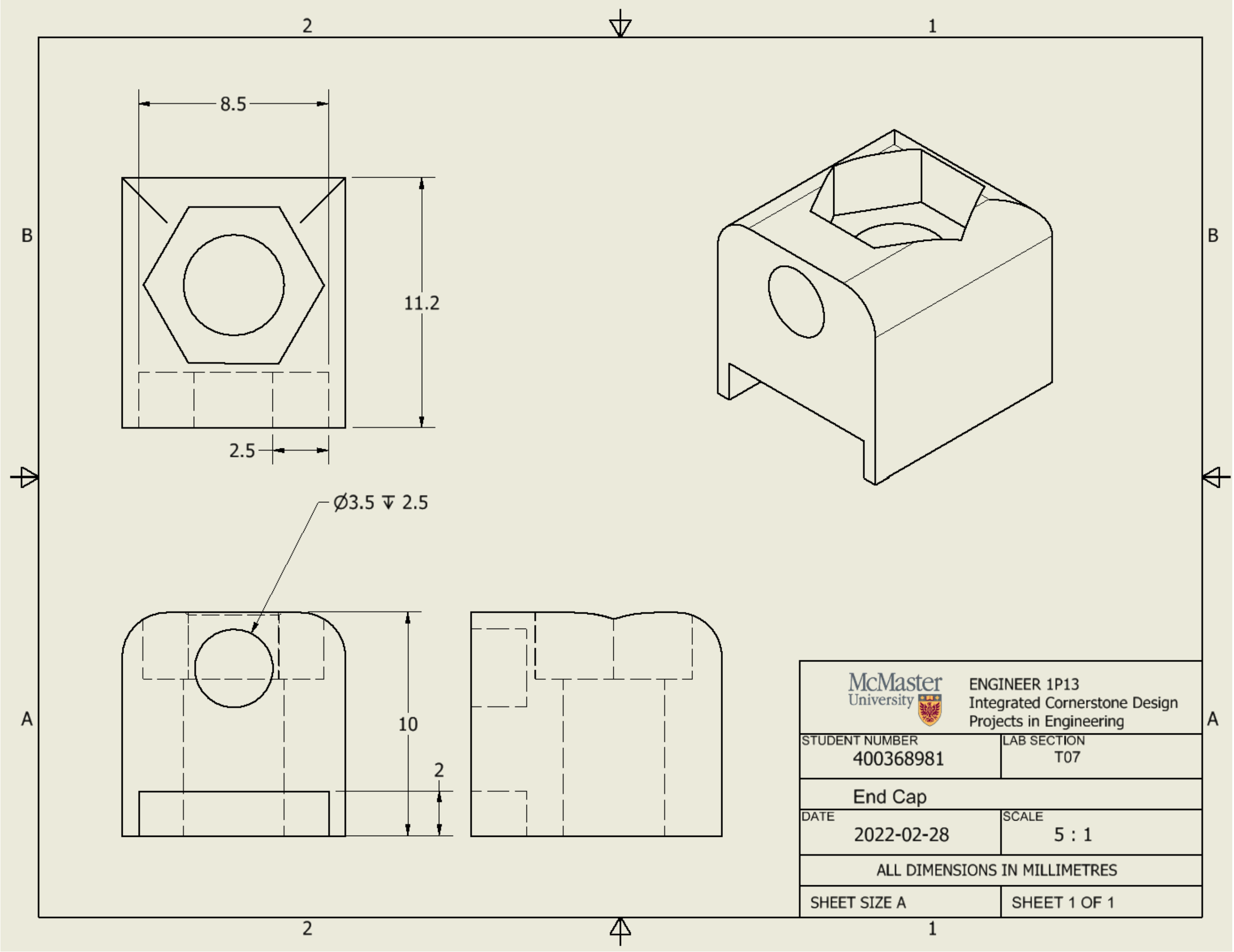
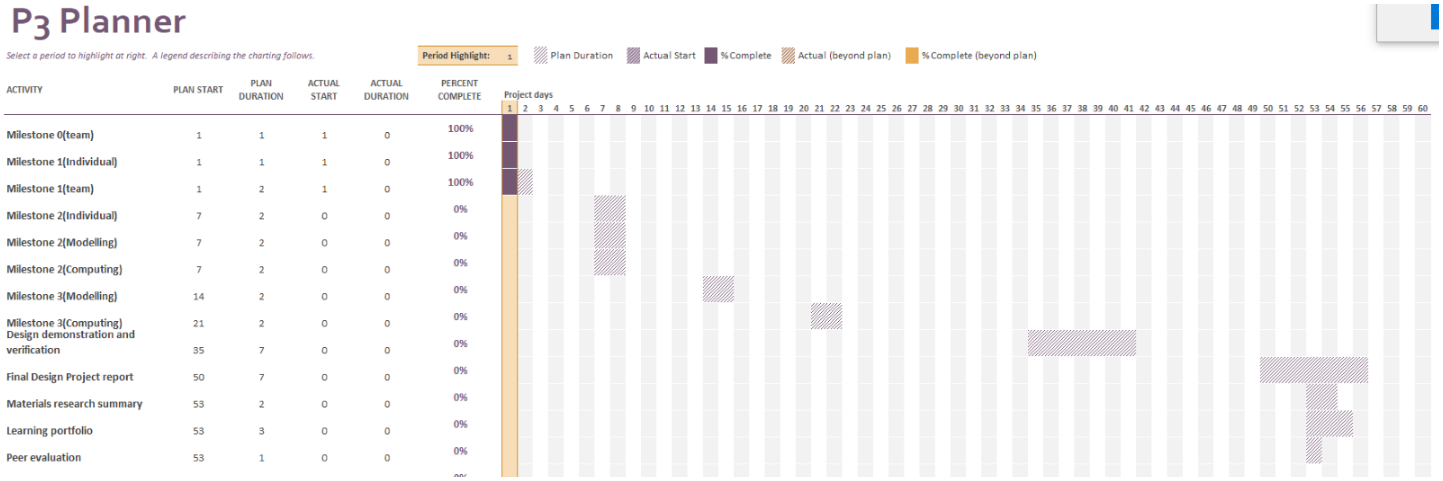


Figure 19. Engineering drawing of End Cap part.

Appendix B – Project Schedule


P3 Planner

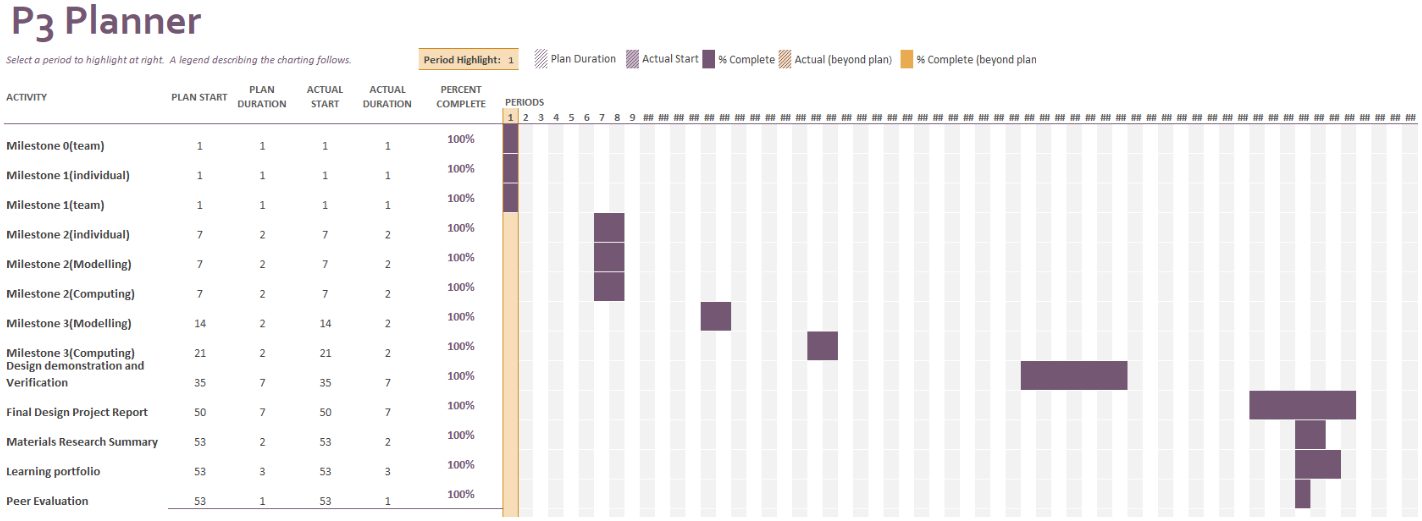
Select a period to highlight at right. A legend describing the charting follows.

Period Highlight: 1

▨ Plan Duration ▨ Actual Start ■ % Complete ▨ Actual (beyond plan) ■ % Complete (beyond plan)

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Milestone 0(team)	1	1	1	0	100%
Milestone 1(Individual)	1	1	1	0	100%
Milestone 1(team)	1	2	1	0	100%
Milestone 2(Individual)	7	2	0	0	0%
Milestone 2(Modelling)	7	2	0	0	0%
Milestone 2(Computing)	7	2	0	0	0%
Milestone 3(Modelling)	14	2	0	0	0%
Milestone 3(Computing)	21	2	0	0	0%
Design demonstration and verification	35	7	0	0	0%
Final Design Project report	50	7	0	0	0%
Materials research summary	53	2	0	0	0%
Learning portfolio	53	3	0	0	0%
Peer evaluation	53	1	0	0	0%

Figure 20. Preliminary Gantt Chart


P3 Planner

Select a period to highlight at right. A legend describing the charting follows.

Period Highlight: 1

▨ Plan Duration ▨ Actual Start ■ % Complete ▨ Actual (beyond plan) ■ % Complete (beyond plan

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Milestone 0(team)	1	1	1	1	100%
Milestone 1(individual)	1	1	1	1	100%
Milestone 1(team)	1	1	1	1	100%
Milestone 2(individual)	7	2	7	2	100%
Milestone 2(Modelling)	7	2	7	2	100%
Milestone 2(Computing)	7	2	7	2	100%
Milestone 3(Modelling)	14	2	14	2	100%
Milestone 3(Computing)	21	2	21	2	100%
Design demonstration and Verification	35	7	35	7	100%
Final Design Project Report	50	7	50	7	100%
Materials Research Summary	53	2	53	2	100%
Learning portfolio	53	3	53	3	100%
Peer Evaluation	53	1	53	1	100%

Figure 21. Final Gantt Chart

Appendix C – Weekly Meetings -coordinator

Table 1. Logbook of Additional Meetings

Date	Meeting purpose	Additional details
Thurs Jan 13th Afternoon Approx. 20m	Milestone 1 completion	Completed the remaining sections of our milestone 1 and 0 worksheets.
Mon Jan 17th Evening Approx. 1hr	Refining concept sketches	Modelling team worked on refining their concept sketches while working together.
Fri Jan 21st Morning Approx. 30m	Model selection	Modelling sub team decided which mechanism they would proceed with
Fri Feb 11th Afternoon Approx. 1hr	Working on mechanism	Worked on the function of our mechanism after completing some 3D printed parts individually.
Fri Mar 4th Afternoon Approx. 20m	Final deliverable	Split up sections of the final deliverable to be worked on individually.

Meeting minutes (Agenda items not included as our manager never created one)

Date of meeting: Thursday January 20th

Attendance

Role	Name	Mac ID	Attendance (Yes/No)
Manager	**Mohammad Hadi**	**Hadim4**	**Yes**
Administrator	**Ayush Patel**	**Patea202**	**Yes Yes**
Adminstrator	**Rayyan Suhail**	**Suhailr**	
Coordinator	**Peter Hull**	**Hullp1**	**Yes**
Coordinator 2	**Tony Han**	**Han18**	**Yes**
Guest			

Meeting Minutes

1. . Modelling sub team: Sketches are all done, working on deciding the design.
   a. Using pro-con list to make the decision
   b. Leaning towards linear actuator, rotary actuator designs were difficult
2. . Computing sub team:
   a. Made pseudocode and flowchart to gain better understanding of the system.
   b. Ayush worked on pseudocode and Hadi worked on flowchart

Post-Meeting Action Items

Modelling sub team choose a design

Computing sub team worked on flowcharts and pseudocode

Date of meeting: Thursday January 27th

Attendance

Role	Name	Mac ID	Attendance (Yes/No)
Manager	**Mohammad Hadi**	**Hadim4**	**No**
Administrator	**Ayush Patel**	**Patea202**	**Yes Yes**
Adminstrator	**Rayyan Suhail**	**Suhailr**	
Coordinator	**Peter Hull**	**Hullp1**	**Yes**
Coordinator 2	**Tony Han**	**Han18**	**Yes**
Guest			

Meeting Minutes

1. . Modelling sub team: Decided on a linear actuator design
   a. Creating a scissor lift design
   b. Might be hard to implement but could pay off
2. . Computing sub team:
   a. Completed sensor research
   b. Work was split up to make pseudocode or flowcharts for individual functions
   c. They have chosen a sensor which detects colour. Color sensor has simple functioning and it works well in short ranges.

Post-Meeting Action Items

Start creating inventor model for modelling team

Computing sub team Justify chosen sensor

Date of meeting: Thursday February 3rd

Attendance

Role	Name	Mac ID	Attendance (Yes/No)
Manager	**Mohammad Hadi**	**Hadim4**	**No**
Administrator	**Ayush Patel**	**Patea202**	**Yes Yes**
Adminstrator	**Rayyan Suhail**	**Suhailr**	
Coordinator	**Peter Hull**	**Hullp1**	**Yes**
Coordinator 2	**Tony Han**	**Han18**	**Yes**
Guest			

Meeting Minutes

1. . Modelling sub team: Model is done beginning to 3D print
    a. Printed at home but some parts didn't work out
    b. Looks like it will work together had to adapt design to use two sliders in order to facilitate a scissor
2. . Computing sub team:
    a. Started ramping up process towards making real code and our aim was to complete 1 cycle
    b. Functions were made in a systematic manner and testing was done for individual functions

Post-Meeting Action Items

Modelling sub team get hardware and finish mechanism

Computing sub team aim to complete one cycle

Date of meeting: Thursday Febuary 10th

Attendance

Role	Name	Mac ID	Attendance (Yes/No)
Manager	**Mohammad Hadi**	**Hadim4**	**No**
Administrator	**Ayush Patel**	**Patea202**	**Yes Yes**
Adminstrator	**Rayyan Suhail**	**Suhailr**	
Coordinator	**Peter Hull**	**Hullp1**	**Yes**
Coordinator 2	**Tony Han**	**Han18**	**Yes**
Guest			

Meeting Minutes

1. . Modelling sub team: Preliminary model done
    a. Used only one half of the scissor model worked well decision made to drop the second level as its unnecessary waste
    b. Still haven't gotten hardware
    c. Some room for improvement on the rotating top part as it's a bit loose
2. . Computing sub team:
    a. We continued to refine the code and added some comments on which parts were facing problems.
    b. Refinement was done in the coordinates and bin colors were changed for easier sensing.

Post-Meeting Action Items

Modelling sub team get hardware and refine the top rotation piece

Computing sub team refinement was process and adds conditions

Date of meeting: Thursday February 17th

Attendance

Role	Name	Mac ID	Attendance (Yes/No)
Manager	**Mohammad Hadi**	**Hadim4**	**No**
Administrator	**Ayush Patel**	**Patea202**	**Yes**
Adminstrator	**Rayyan Suhail**	**Suhailr**	
Coordinator	**Peter Hull**	**Hullp1**	**Yes**
Coordinator 2	**Tony Han**	**Han18**	**Yes**
Guest			

Meeting Minutes

1. . Modelling sub team: Mechanism is Done!
   a. Some refinement could be done but overall working condition
   b. Be ready for design review with drawings and assembly
2. . Computing sub team:
   a. Conditions and functions were added for dispensing conditions.
   b. Tried while and for loops to see which worked better
   c. Be ready for design review

Post-Meeting Action Items

Modelling sub team completes engineering drawings.

Computing sub team works on final refinement and commenting

Appendix E – Worksheets

Table 2. Team Milestones:

Document	Link
Milestone 0	https://mcmasteru365.sharepoint.com/:w:/s/msteams_3f8bad-Thurs-02/EUqRMhQFF7NNkxlsqfnfzwkBUQYVw1F5AWkq6aCylfLVZA?e=UIiW92
Milestone 1	https://mcmasteru365.sharepoint.com/:w:/s/msteams_3f8bad-Thurs-02/EQQj_VaxP6VOldw4l_sNJBUB6V0OMg2JMkh387-AoK-DMg?e=EEhbDH
Milestone 2	https://mcmasteru365.sharepoint.com/:w:/s/msteams_3f8bad-Thurs-02/EV3xdcDQpI5JieSZshN1sAUBJ16OZzdD4S7lGrT1e8Thaw?e=SjytwT
Milestone 3	https://mcmasteru365.sharepoint.com/:w:/s/msteams_3f8bad-Thurs-02/EdPerVTi_3JEqnMrPV36TYIBUPdwj7Khr-8ZuiWuMhXnWA?e=NBtkYX