Design Project 3 – Revenge of the Recycling System

ENGINEER 1P13 – Integrated Cornerstone Design Projects in Engineering

Tutorial T10

Team Fri-09

GRIFFIN LARKE (larkeg)

HABEL KINGSON (kingsonh)

FAHD ZADA (zadaf)

PARVESHWARA VANAPILLI NURSIMULU (vanapilp)

Submitted: March 4th, 2024

Course Instructors: Dr. McDonald, Dr. Doyle, Dr. Ebrahimi, Dr. Fleisig, Dr. Hassan, Dr. Zurob

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.

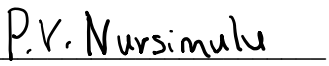
Fahd Zada 400516007

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.

Habel Kingsley 400528728

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.

Parveshwara Vanapilli Nursimulu 400431169



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.

Griffin Larke 400497452

Executive Summary

The goal behind this project is to develop an efficient and automated recycling system to address environmental concerns. The idea of the recycling system is to create a smart and functional mechanism that could be controlled by a computer program, providing an automated solution for sorting and recycling containers based on their materials. The solution involves a collaborative approach between the Modelling Sub-Team and the Computer Sub-Team to design both a physical and Python-based control program.

Going into the mechanism design, the mechanism is designed to connect to the base plate and connecting plate of the Q-bot. The mechanism incorporates an actuator, which can be both linear and rotary. The team decided to use a rotary actuator because it provided flexibility and ease of use, although the design process was not easy. The Modelling Sub-Team was responsible for creating a comprehensive assembly model in Autodesk Inventor, showcasing the integration of all components.

As for the Computation Sub-Team, they were tasked with programming a Python computer program that controls the Q-bot and controls its actions. The program takes the type of container that is being placed on the hopper from the user, and based on the input, the Q-bot transfers the containers to the correct bin in the Recycling Station. Additionally, to differentiate between bins, the program activates sensors that are on the Q-bot which are used to know which container the bot is in front. The Q-bot follows a predetermined path on the floor, stops at the correct bin using the sensor data, deposits the container, and returns to its home position. The cycle repeats until the user kills the program.

References

References for citations used in Executive Summary

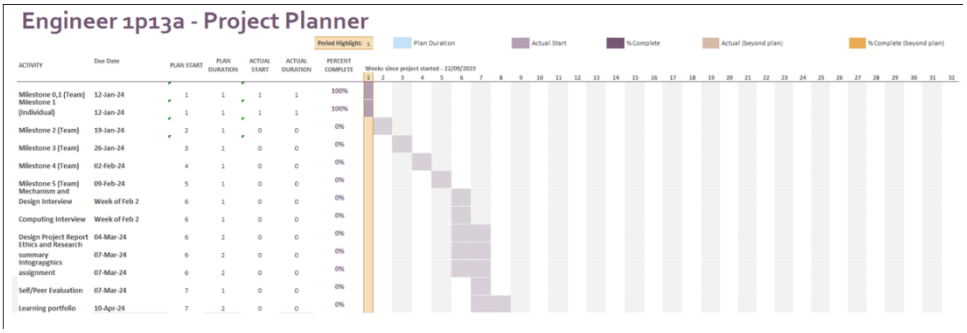
[1] “Project 3 Module,” class notes for ENG1P13, Department of Engineering, Winter, 2024.

Appendices

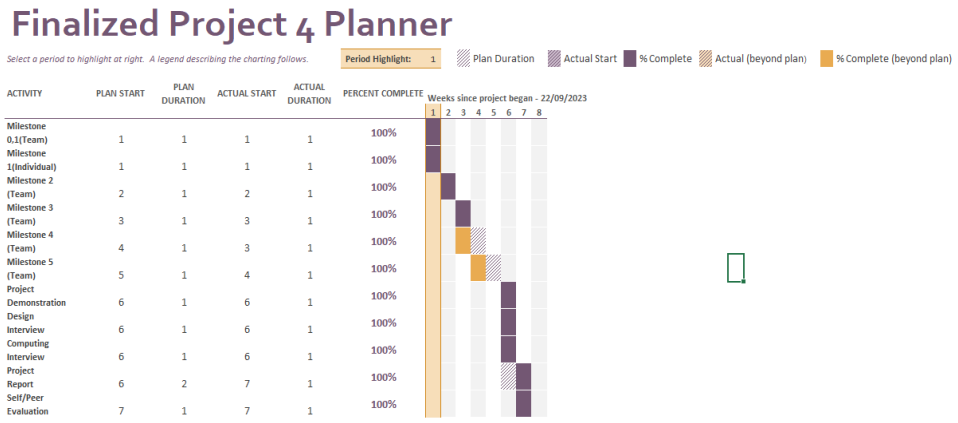
Table of Figures

Appendix A: Project Schedule

Preliminary Gantt Chart (Manager)



Final Gantt Charts (Administrator 1)



Logbook of Additional Meetings and Discussions

24th January: Modelling Sub team meeting at Thode Makerspace for printing and assembly of original prototype and discuss what will need to be added to final prototype.

30th January: Discussion about how to get fasteners (M5 x 40mm Screw).

Appendix B: Scheduled Weekly Meetings

Weekly Design Studio Agenda's (Manager)

Jan 26

Agenda Items

1. Attendance & Updates
2. Issues from past week
3. Meeting with reviewers
4. Discuss feedback from reviewers
5. Action Items for next meeting

Meeting Minutes

1. Everybody here, all required work finished
2. No issues everything went well
3. Reviewer meeting:
    - Modeling team:
    - Fix tolerances
    - Fix interferences
    - Computing team:
    - Cycle through multiple containers
    - Identify containers

Post-Meeting Action Items

Computing team:

- Cycle through multiple containers
- Identify containers

Modeling team:

- Attach rack to connecting plate
- Fix tolerances and interference is assembly

Feb 2

Agenda Items

6. Attendance & Updates
7. Issues from past week
8. Meeting with reviewers
9. Discuss feedback from reviewers
10. Action Items for next meeting

Meeting Minutes

4. Everybody here, all required work finished
5. No issues everything went well
6. Reviewer meeting:
    - Modeling team:
    - Fix tolerances
    - Fix interferences
    - Computing team:
    - Cycle through multiple containers
    - Identify containers

Post-Meeting Action Items

Computing team:

- Make it work for multiple containers
- Integrate with modeling team
- Write code for physical environment

Modeling team:

- Start on engineering drawings
- Integrate with computing team

Feb 9

Agenda Items

11. Attendance & Updates
12. Issues from past week
13. Meeting with reviewers
14. Discuss feedback from reviewers
15. Action Items for next meeting

Meeting Minutes

7. Everybody here, all required work finished
8. No issues everything went well
9. Reviewer meeting:
    - Both teams all good
    - Start working on hardware environment

Post-Meeting Action Items

Computing team:

- Set up hardware template

Modeling team:

- All done

Both:

- Prepare for interview next week

Weekly Design Studio Meeting Minutes

Week 2 Milestone 2 19th January:

Informal TA Check-in: Receive feedback on mechanism design sketches and workflow of computer program.
Comments: Specify what constraints to be used in assembly, and torque specifications of rotary actuator.
For Computing: clarify team roles and explain importance of container attributes and functionality within code.

Week 3 Milestone 3 26th January:

Informal TA Check-in:
Demonstrate first functional mechanism prototype, and final prototype assembly in Inventor.
Mentor Comments: Try to remove all interferences that are in the assembly. Go ahead to start printing.
For computing: Everything in order, continue working on developing the code

Week 4 Milestone 4 2nd February:

Dedicated Project Work Time:

Modelling Sub team tested physical mechanism.

Computing sub team: Everything seems to be in order, continue to develop code for complete container drop-off.

Week 5 Milestone 5 9th February:

Design Review, and computing sub team tested code in physical environment in preparation for design interview.

Weekly Design Studio Post-Meeting Action Items:

Week 2 Milestone 2 19th January:

1. Complete mechanism assembly and test out function. Adjust based on errors that arise. [Modelling Sub team]
2. Finish complete transfer, load and drop off functions. [Computation Sub team]
3. Test to ensure one functional cycle [Computation Sub team]

Week 3 Milestone 3 26th January:

1. Print Rack and Mechanism housing. [Modelling Sub team]
2. Constrain rack and connecting plate [Modelling Sub team]
3. Make code functional for multiple cycles, with different bins and containers [Computation Sub team]
4. Write a function to identify containers properties [Computation Sub team]
5. Write main function [Computation Sub team]

Week 4 Milestone 4 2nd February:

1. Ensure that all container cycles are working and develop physical environment code [Computation Sub team]

Appendix C: Comprehensive List of Sources

[1] “There’s a Recyclable Among Us,” P3 Project Module, pp. 3–44, class notes for ENGINEER 1P13, Department of Engineering, McMaster University, Winter, 2021.

[2] “Quanser Interactive Labs’’ Quanser Consulting Inc, Markham, ON, 2021. (https://www.quanser.com/)

[3] “Autodesk Inventor Professional 2021.” Autodesk, San Rafael, CA, 2021. (https://www.autodesk.com/)

[4] “Mechanisms: Types and Application,” class notes for ENGINEER 1P13, Department of Engineering, McMaster University, Winter, 2021.

[5] “P3 Python Library Documentation,” class notes for ENGINEER 1P13, Department of Engineering, McMaster University, Winter, 2021

Appendix D: Supporting Documents

Screenshots of your solid model, prototype, and computer program

Computer Program:

Dispense Container Function:

```
bin_id = []
total_weight = []

#(Habel Kingson - kingsonh)
def dispense_container():

    global bin_id, total_weight

    container = random.randint(1,6)
    container_info = table.dispense_container(container,True)
    bin_id.append(container_info[2])
    total_weight.append(container_info[1])
    time.sleep(1)
    print(bin_id, total_weight)
```

Load Container Function:

```
# (Griffin Larke - larkeg)
def load_container(): # function to take containers from table to hopper
    global current_containers # stores number of containers currently in hopper
    global hopper_full # true/false if hopper is full or not, and can another container be added

    dropoff_locations = [(0.02, -0.578, 0.53),(0.02, -0.5, 0.532),(0.02, -0.412, 0.511)] # list of coordinates of dropoff locations in hopper

    if current_containers < 3: #only loads container if requirements are met. will add weight and bin requirements later
        arm.home()
        time.sleep(1)
        arm.move_arm(0.624, 0.0, 0.283)# coordinates of pickup location on table
        time.sleep(1)
        arm.control_gripper(43)
        time.sleep(1)
        arm.rotate_shoulder(-5)
        time.sleep(1)
        arm.rotate_base(-10)
        time.sleep(1)
        arm.rotate_shoulder(-40)
        time.sleep(1)
        arm.rotate_base(-80)
        time.sleep(1)

        arm.move_arm(*dropoff_locations[current_containers]) # coordinates of dropoff location depending on number on containers currently in hopper
        time.sleep(1)
        arm.rotate_shoulder(1)
        time.sleep(1)
        arm.control_gripper(-43)
        time.sleep(1)
        if current_containers == 0: # controls arm to push containers into standing upright, depending on their position in hopper. not fully working
            arm.rotate_elbow(-8)
            time.sleep(1)

        else: # different controls for containers other than the first one
            arm.rotate_elbow(-5)
            time.sleep(1)
            arm.rotate_shoulder(14)
            time.sleep(1)
            arm.rotate_shoulder(-20)

        arm.home()
        current_containers += 1
    else:
        hopper_full = True #if no other containers can be added, hopper is full
```

Deposit Function:

```
#(Griffin Larke - larkeg)
def deposit_container():

    global hopper_full #global variables that will be used in function
    global current_containers

    bot.set_wheel_speed([0.05,0.05]) # move q-bot to desired location
    bot.activate_stepper_motor() # activate hopper control mechanism
    time.sleep(3.3)
    bot.set_wheel_speed([0.0,0.0])
    time.sleep(0.5)
    bot.rotate_hopper(15) #rotate hopper
    time.sleep(0.5)
    bot.rotate_hopper(30)
    time.sleep(0.5)
    bot.rotate_hopper(45)
    time.sleep(0.5)
    bot.rotate_hopper(60)
    time.sleep(0.5)
    bot.rotate_hopper(90)
    time.sleep(0.5)
    bot.rotate_hopper(0)
    bot.deactivate_stepper_motor()
    bot.deactivate_color_sensor()
    print("returning home")
    hopper_full = False
    return_home()
```

Transfer Function:

```
#(Griffin Larke - larkeg)
def transfer_container(bin_id): #function to drive Q-bot to specified bin (bin_id) and drop off containers. activates return_home when done

    bot.activate_line_following_sensor()# activate required sensors
    bot.activate_color_sensor()
    bot.activate_stepper_motor()
    home_position = (1.5, 0, 0)

    global hopper_full #global variables that will be used in function
    global current_containers

    moving = True # true/false if q-bot is following line or not
    bin_rgbs = [[1,0,0],[0,1,0],[0,0,1],[0,1,1]] # list of simplified rgb values of bins
    target_rgb = bin_rgbs[int(bin_id)-1]

    print(target_rgb)

    if hopper_full == True: # only starts dropping off containers if hopper is full
        moving = True

    while moving == True: # while loop to keep checking if statements, this loop makes it follow the line and check bin color

        if bot.line_following_sensors() == [1,1]: # if both on line, goes straight
            bot.set_wheel_speed([0.05,0.05])

        elif bot.line_following_sensors() == [1,0]: # speeds up wheel to turn
            bot.set_wheel_speed([0.05, 0.08])

        elif bot.line_following_sensors() == [0,0]:
            print("oh no")
            moving = False

        else: #bot.line_following_sensors == [0,1]:
            bot.set_wheel_speed([0.08,0.05])

        rgb_reading = bot.read_color_sensor()[0] # reads input from color sensor

        if rgb_reading == target_rgb: #checks if target bin is right color


            bot.set_wheel_speed([0,0]) #stops qbot
            moving = False # stops following line
            current_containers = 0

            if bin_id == 1: # different movement for 1st and 3rd bins as they are after a corner

                bot.rotate(-30)
                bot.set_wheel_speed([0.05,0.05])
                time.sleep(4)

            elif bin_id == 3:

                bot.rotate(-30)
                bot.set_wheel_speed([0.05,0.05])
                time.sleep(4)

            deposit_container()
```

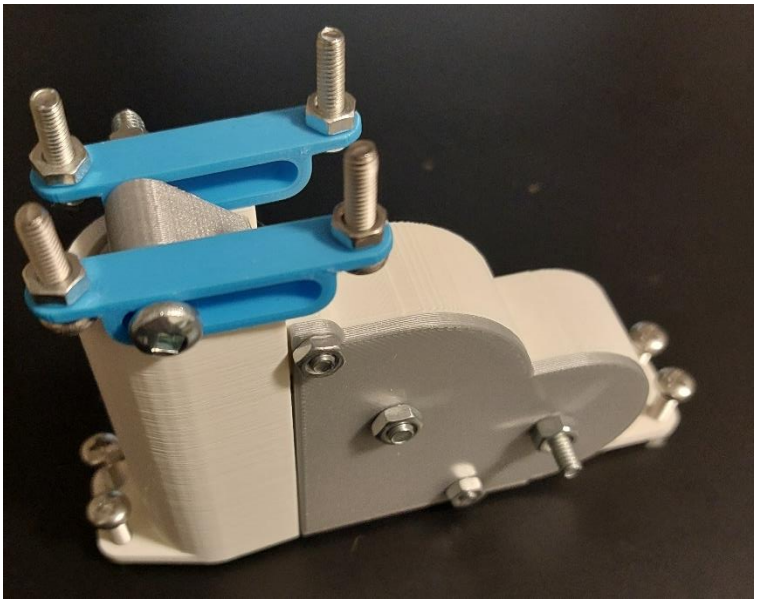
Return Home Function:

```
#(Griffin Larke - larkeg)
def return_home(): # drives the q-bot along the line until it reaches starting point


    while bot.position()[1] < -0.01 or bot.position()[0] < 1.45: # makes qbot follow line until it reaches desired coordinates

        if bot.line_following_sensors() == [1,1]: # same follow line function used in transfer_container
            bot.set_wheel_speed([0.05,0.05])

        elif bot.line_following_sensors() == [1,0]:
            bot.set_wheel_speed([0.05, 0.08])

        else: #bot.line_following_sensors == [0,1]:
            bot.set_wheel_speed([0.08,0.05])

    bot.set_wheel_speed([0,0]) # stops qbot
```

Main Function:

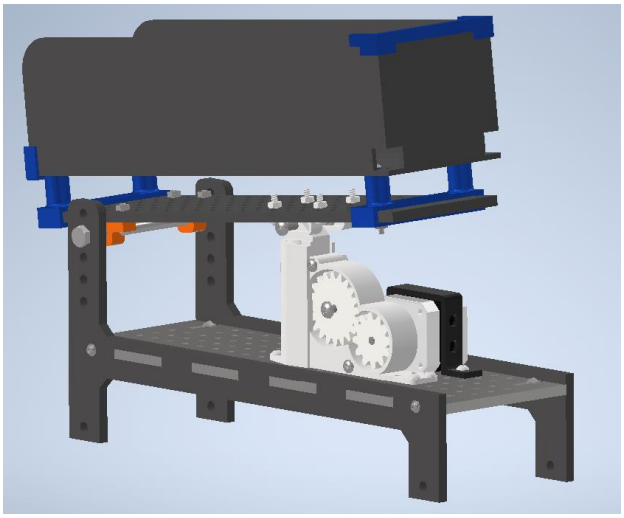
```
from itertools import count #used for indefinite loop

#(Griffin Larke - larkeg)
def main():

    global bin_id
    global total_weight

    for i in count(0):
        if i == 0:
            dispense_container()
            load_container()
            dispense_container()
        else:
            load_container()
            dispense_container()

        if bin_id[0] == bin_id[1] and sum(total_weight) < 90: #condition checking for same bin id's and weight < 90g
            load_container()
            dispense_container()
            if bin_id[0] == bin_id[1] == bin_id[2] and sum(total_weight) < 90: #checks 3rd container for the same conditions
                load_container()
            else:
                transfer_container(int(bin_id[0][-1]))
                bin_id = [bin_id[2]]
                total_weight = [total_weight[2]]

        else:
            transfer_container(int(bin_id[0][-1]))
            bin_id = [bin_id[1]]
            total_weight = [total_weight[1]]
```

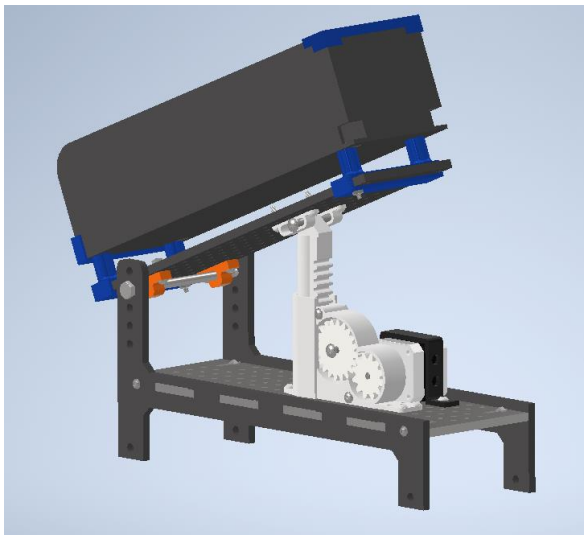
Solid Model:



Clip of Rack moving: Rack_Movement.mp4
Clip of Gears Moving Inside: Gears_Movement.mp4

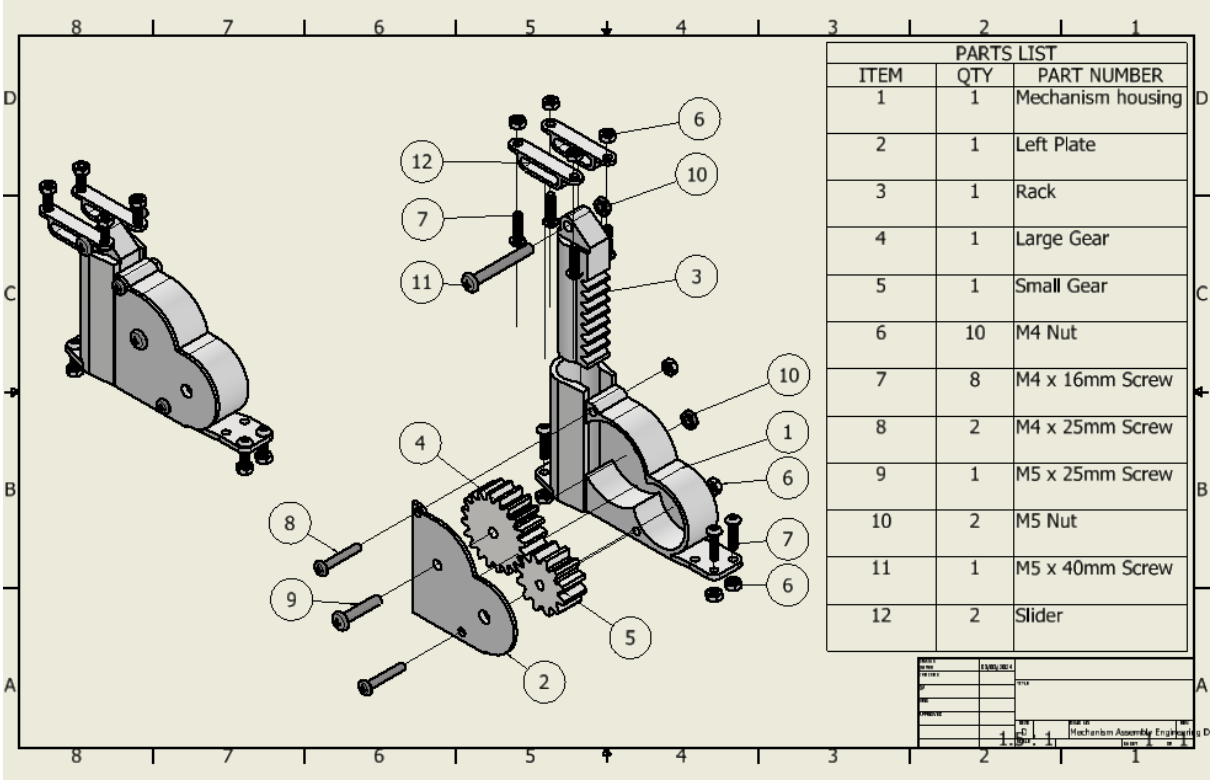
Autodesk Assembly:





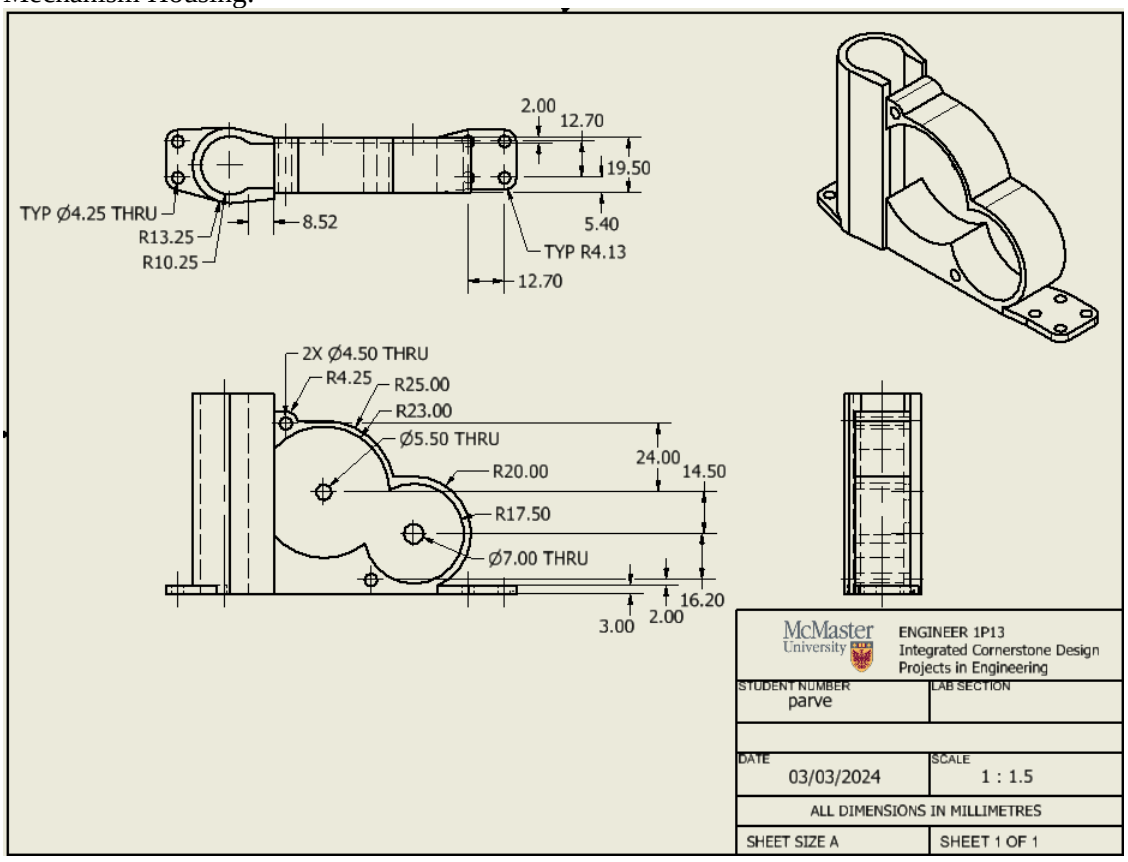
Fully dimensioned Engineering Drawings of mechanism and design

Exploded View of Mechanism Assembly:

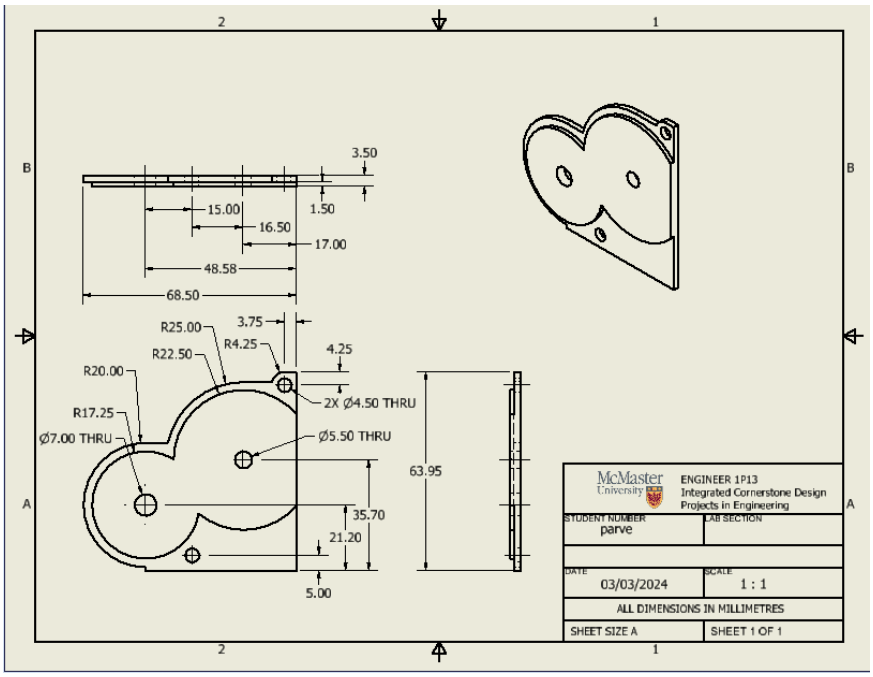


Engineering Drawings:

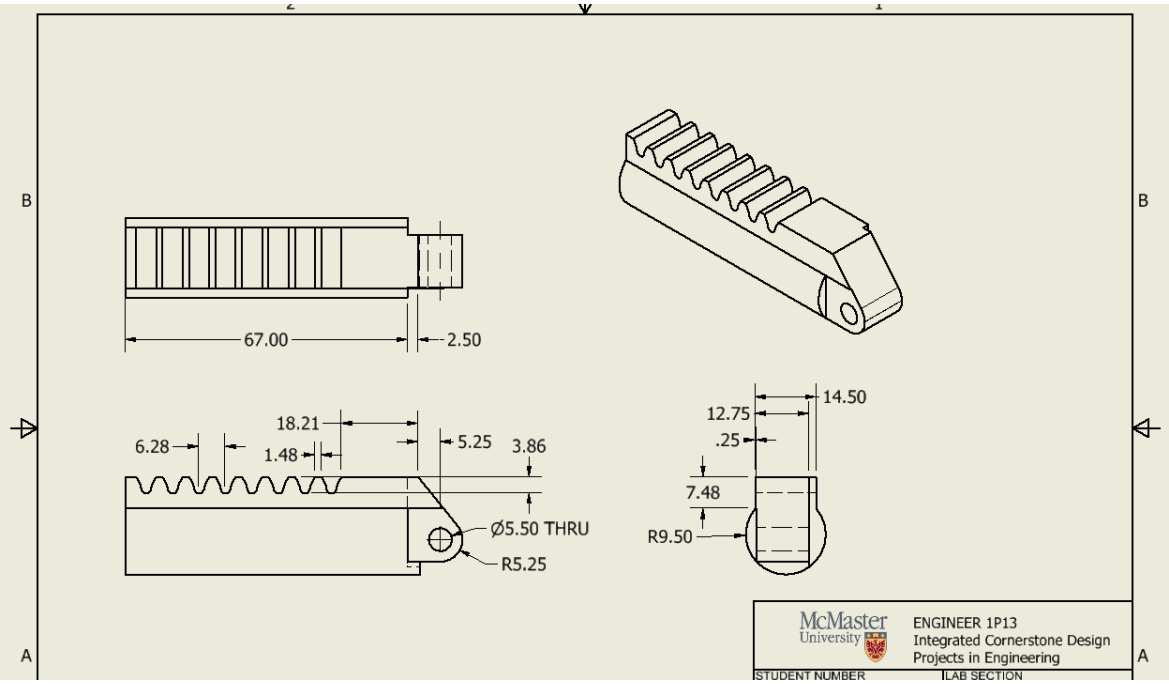
Mechanism Housing:



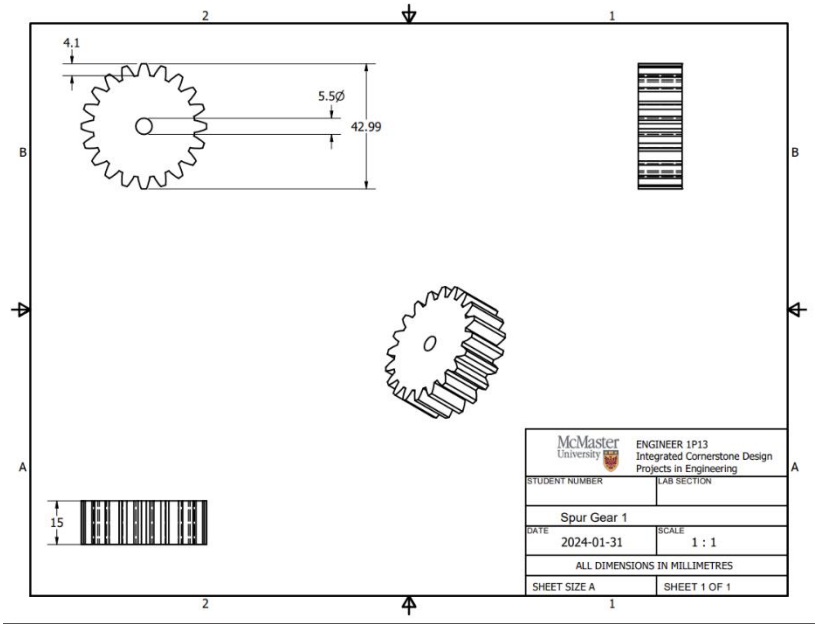
Left Plate:



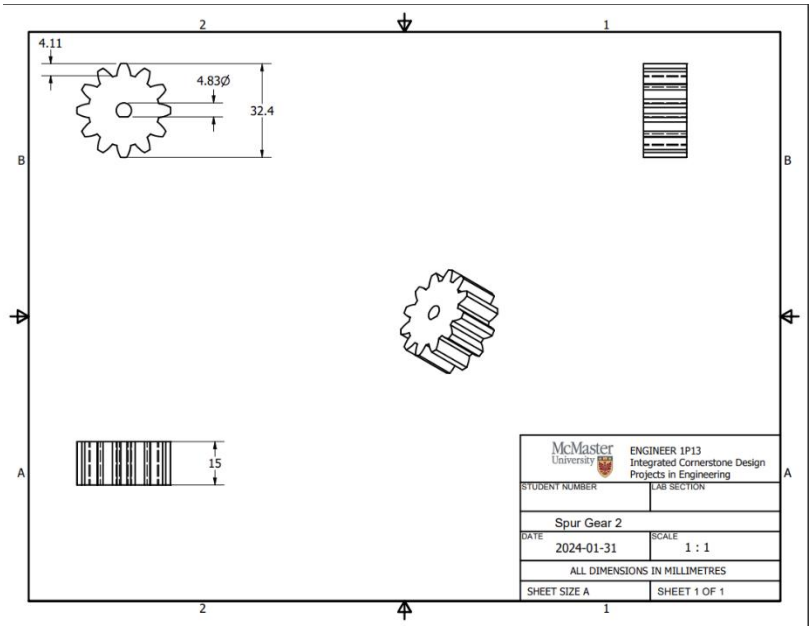
Rack:



Large Gear:



Small Gear:


4.11
4.83Ø
32.4
15
McMaster University
ENGINEER 1P13
Integrated Cornerstone Design
Projects in Engineering
STUDENT NUMBER
LAB SECTION
Spur Gear 2
DATE
2024-01-31
SCALE
1 : 1
ALL DIMENSIONS IN MILLIMETRES
SHEET SIZE A
SHEET 1 OF 1

Appendix E: Design Studio Worksheets

Individual Submissions:

P3_Milestone1-combined.pdf
Fri-09_P3_Milestones.pdf