ENGINEERING

McMaster University

Project Three – There's A Recyclable Among Us:

Designing a System for Sorting and Recycling Container

ENGINEER 1P13 – Integrated Cornerstone Design Projects

Tutorial 15

Team MON-58

Aya Selman (selmaa1)

Muhammad Saad Siddiqui (siddim68)

Mila Ilijevska (ilijevm)

Mohammad Muntazar Bhurwani (bhurwanm)

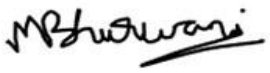
Submitted: Sunday March 7th, 2021

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.

Mohammad Muntazar Bhurwani 400296770



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.

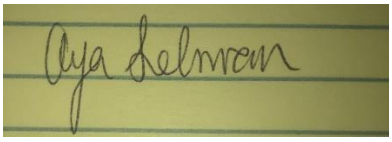
Mila Ilijevska 400312899

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.

Muhammad Saad Siddiqui 400311808

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.

Aya Selman 400318730



Executive Summary

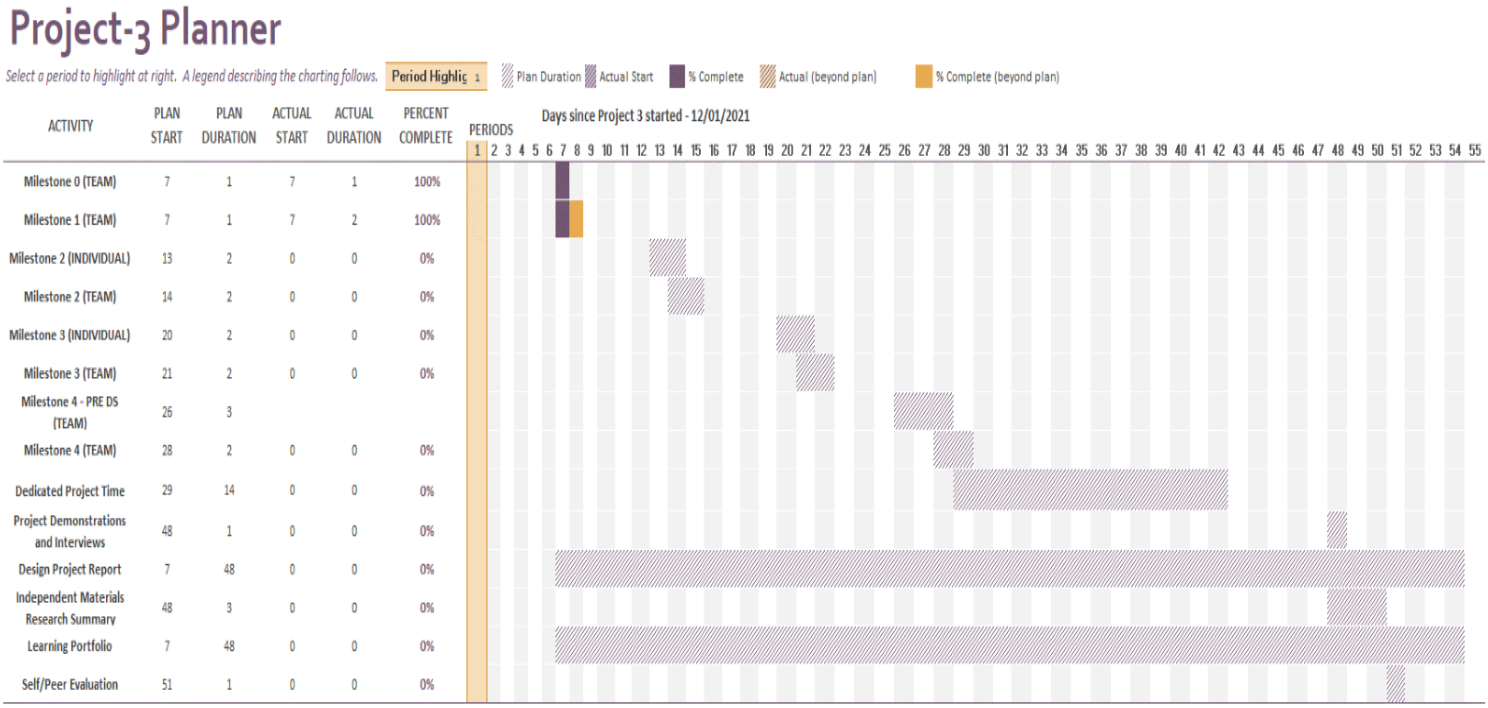
Project three's objective was to design a system for sorting and recycling different types of containers. The system included designing a hopper paired with a mechanism, and a computer program. Our goal is to speed up the recycling process and increase its accuracy using autonomation. We completed this by designing a system to identify and sort materials, while also being able to securely transport and deposit them into their respective bins. The focus of identifying containers was to determine whether they could be recycled. Having manual inspections of contaminated containers slows down the process of recycling.

The computing sub-team created the computer program in python to identify the presence of a container, determine the material of the container, and classify whether the container is recyclable. To do this, the program takes the data stimulated from sensors and compares the weight of the object and the material to see if it can be recycled. Once the container was classified, the program loaded the containers onto the Q-bot. The pick and drop off locations of the containers were made into a list. Once the Q-bot is loaded based on three conditions it moves to the recycling station. In the transfer function, we used the color sensor to detect the correct recycling bin. Once the color of the bin that corresponded to the containers on the Q-bot, the Q-bot would stop moving and be ready for deposit. The deposit function rotated the Q-bot to be in angled in such a way that the containers would fall directly into the correct bin. Once the deposit was completed the Q-bot would rotate again to face toward the yellow line and then following the trajectory of the line until it would return to its home position.

The function of the hopper was to hold the container/s dropped in by the Q-arm safely during its transportation to the recycling station. Clubbed with a designed mechanism, the hopper would rotate to deposit the containers accurately into their respective bins. Our solution included a simple but sturdy ramp-like mechanism that was connected to a linear actuator mounted on a baseplate. The actuator begins at an extended position, on sliding back to origin, the connected rods that are mated with the hopper slide to push the hopper upwards at an angle thus depositing the container. To further better the dependability of our design we added an additional rod mated to the first one allowing each rod to undergo less stress. Another factor to help reduce the amount of stress of these rods is our implementation of a solid block which allows the hopper to rest on it when at the rest position, further reducing the stress upon the two rods. To ensure stability the hopper was modified to include a safety barrier around 3 walls while the front face was removed and replaced by a sloping face to allow an easy fall for the containers. The base of the hopper is also slightly sloped for the same reason.

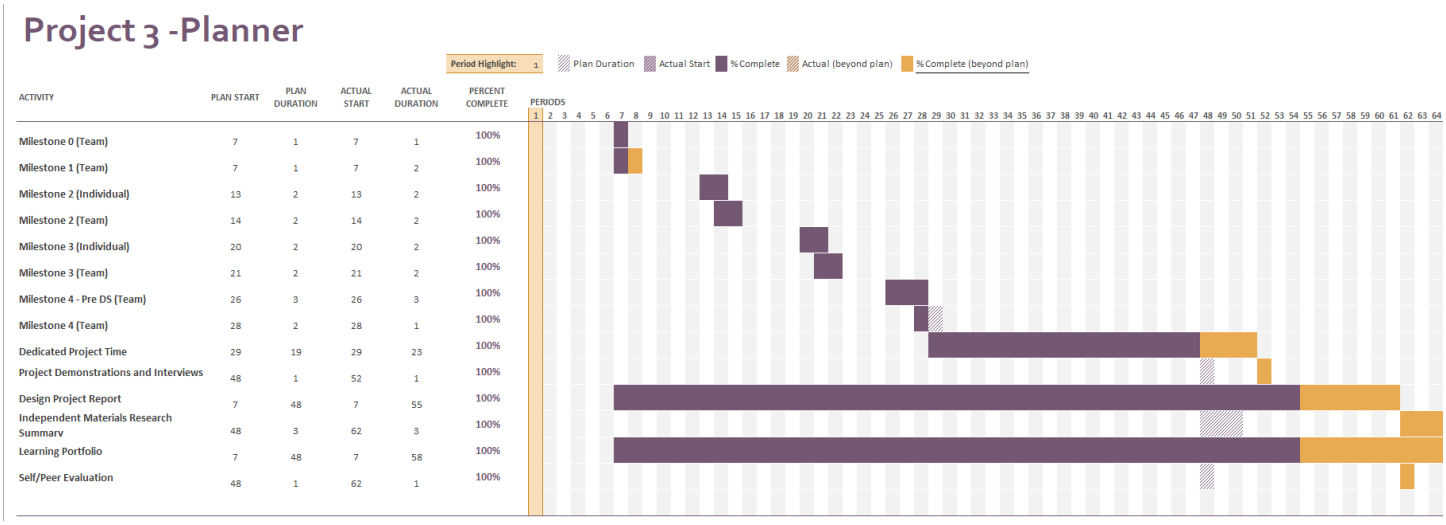
Project Schedule:

Preliminary Gantt Chart



ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Milestone 0 (TEAM)	7	1	7	1	100%
Milestone 1 (TEAM)	7	1	7	2	100%
Milestone 2 (INDIVIDUAL)	13	2	0	0	0%
Milestone 2 (TEAM)	14	2	0	0	0%
Milestone 3 (INDIVIDUAL)	20	2	0	0	0%
Milestone 3 (TEAM)	21	2	0	0	0%
Milestone 4 - PRE DS (TEAM)	26	3			
Milestone 4 (TEAM)	28	2	0	0	0%
Dedicated Project Time	29	14	0	0	0%
Project Demonstrations and Interviews	48	1	0	0	0%
Design Project Report	7	48	0	0	0%
Independent Materials Research Summary	48	3	0	0	0%
Learning Portfolio	7	48	0	0	0%
Self/Peer Evaluation	51	1	0	0	0%

Final Gantt Chart:



ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Milestone 0 (Team)	7	1	7	1	100%
Milestone 1 (Team)	7	1	7	2	100%
Milestone 2 (Individual)	13	2	13	2	100%
Milestone 2 (Team)	14	2	14	2	100%
Milestone 3 (Individual)	20	2	20	2	100%
Milestone 3 (Team)	21	2	21	2	100%
Milestone 4 - Pre DS (Team)	26	3	26	3	100%
Milestone 4 (Team)	28	2	28	1	100%
Dedicated Project Time	29	19	29	23	100%
Project Demonstrations and Interviews	48	1	52	1	100%
Design Project Report	7	48	7	55	100%
Independent Materials Research Summary	48	3	62	3	100%
Learning Portfolio	7	48	7	58	100%
Self/Peer Evaluation	48	1	62	1	100%

Logbook of Additional Meetings and Discussions:

EVENT

Modelling Meeting 2/7/2021

ATTENDANCE

Role	Name	Mac ID	Attendance (Yes/No)
Manager	Muntazar	bhurwanm	Yes
Administrator	Mila	Ilijevm	No
Coordinator	Saad	Siddim68	Yes
Subject Matter Expert	Aya	Selmaa1	No

AGENDA ITEMS

1. Putting parts together

MEETING MINUTES

1. Combining and constraining parts made individually

POST-MEETING ACTION ITEMS

- Finish the constraints on each part
- Fix the action of the linear actuator and rods

EVENT

Final Deliverables and Bonus Marks Meeting 2/15/2021

ATTENDANCE

Role	Name	Mac ID	Attendance (Yes/No)
Manager	**Muntazar**	**bhurwanm**	**Yes**
Administrator	**Mila**	**Ilijevm**	**Yes**
Coordinator	**Saad**	**Siddim68**	**Yes**
Subject Matter Expert	**Aya**	**Selmaa1**	**Yes**

AGENDA ITEMS

1. Discuss final deliverables
2. Discuss the bonus mark work

MEETING MINUTES

1. Finished Executive summary
2. Assigned independent research summary topics
    a. Light detector: Saad, Siddim68
    b. Ultrasonic: Aya, Selmaa1
    c. Inductive: Muntazar, bhurwanm
    d. Capacitive: Mila, Ilijevm
3. Ask about bonus task later

POST-MEETING ACTION ITEMS

- Finish logbook and meeting minutes
- Complete independent research summaries

Scheduled Weekly Meetings:

Weekly Design Studio Agenda's:

WEEK 1 – JANUARY 18, 2021 AGENDA ITEMS

1. Introduction with TA and group members – 1st DS
2. Discuss Why/How Ladder
3. Feedback on Objectives vs Constraints

WEEK 2 – JANUARY 25 AGENDA ITEMS

1. Status of pre-design studio milestone.
2. Status of both sub teams
3. Question about determining criteria for decision matrix.

WEEK 3 – FEBRUARY 1 AGENDA ITEMS

1. Catch up both sub teams and TA
2. Worked outside of DS to decide on modelling solution and program tasks

WEEK 4 – FEBRUARY 8 AGENDA ITEMS

1. Design Review - Present sub-team solutions

WEEK 5 – FEBRUARY 22 AGENDA ITEMS

1. Status of sub-teams – finalised and ready to present.
2. Team has completed Executive Summary.

Weekly Design Studio Meeting Minute's:

WEEK 1 – JANUARY 18, 2021

Meeting Minutes

- Note down each team member's objectives/constraints before finalising.
- Ensure there is no repetitive words.
- Simple problem statement → refined at the end of DS.

POST-MEETING ACTION ITEMS

- Complete Milestone 0 and Milestone 1
- Determine administrative responsibilities.

WEEK 2 – JANUARY 25, 2021

Meeting Minutes

1. Individual milestone completion discussion
    a. Confirm predesign studio work completion

2. Clarification of sensor use.
    a. TA clears up question about how sensors should be used, "for this milestone we will be using the sensors to determine which bin it is."

POST-MEETING ACTION ITEMS

- N/A

WEEK 3 – FEBRUARY 1, 2021

Meeting Minutes

1. Discuss about predesign studio work
    a. Both teams have finished predesign studio work

2. Modelling team has chosen a design.
    a. Modelling team has chosen a design from milestone 2

3. Computing team will utilize flowchart in Milestone 3

POST-MEETING ACTION ITEMS

- N/A

WEEK 4 – FEBRUARY 8, 2021

Meeting Minutes

1. *Modelling:* all sorted -Mechanism good, screws fall with range, holds all 3 containers improvements - strengthen rod/resting pillar for hopper, great design, recheck DOF for tiny parts of design

2. Computing: recheck path of q-bot (loop), code + simulation great, recheck occasional issues,

POST-MEETING ACTION ITEMS

- Modelling team
    - Add support for hopper when in rest
    - Add another rod to help disperse load
- Computing team
    - Fix up any small technical glitches

WEEK 5 – FEBRUARY 22, 2021

Meeting Minutes

1. Finalize designs
2. Discuss finished summary
3. Design interview

POST-MEETING ACTION ITEMS

- Team members are assigned to finish individual tasks of project report

Design Studio Worksheets:

Milestone 0

PROJECT THREE: MILESTONE 0 – COVER PAGE

Team Number: Mon-58

Please list full names and MacID's of all *present* Team Members

Full Name:	MacID:
Mohammad Muntazar Bhurwani	bhurwanm
Aya Selman	selmaa1
Mila Ilijevska	ilijevm
Muhammad Saad Siddiqui	siddim68

Insert your Team Portrait in the dialog box below

MILESTONE 0 – TEAM CHARTER

Team Number: Mon-58

Incoming Personnel Administrative Portfolio:

Prior to identifying Leads, identify each team members incoming experience with various **Project Leads**

	Team Member Name:	Project Leads
1.	Aya Selman	☐M ☒A ☒C ☐S
2.	Mohammad Muntazar Bhurwani	☐M ☒A ☒C ☐S
3.	Mila Ilijevska	☒M ☐A ☐C ☒S
4.	Muhammad Saad Siddiqui	☒M ☐A ☐C ☒S

To 'check' each box in the Project Leads column, you must have this document open in the Microsoft Word Desktop App (not the browser and not MS Teams)

Project Leads:

Identify team member details (Name and MACID) in the space below.

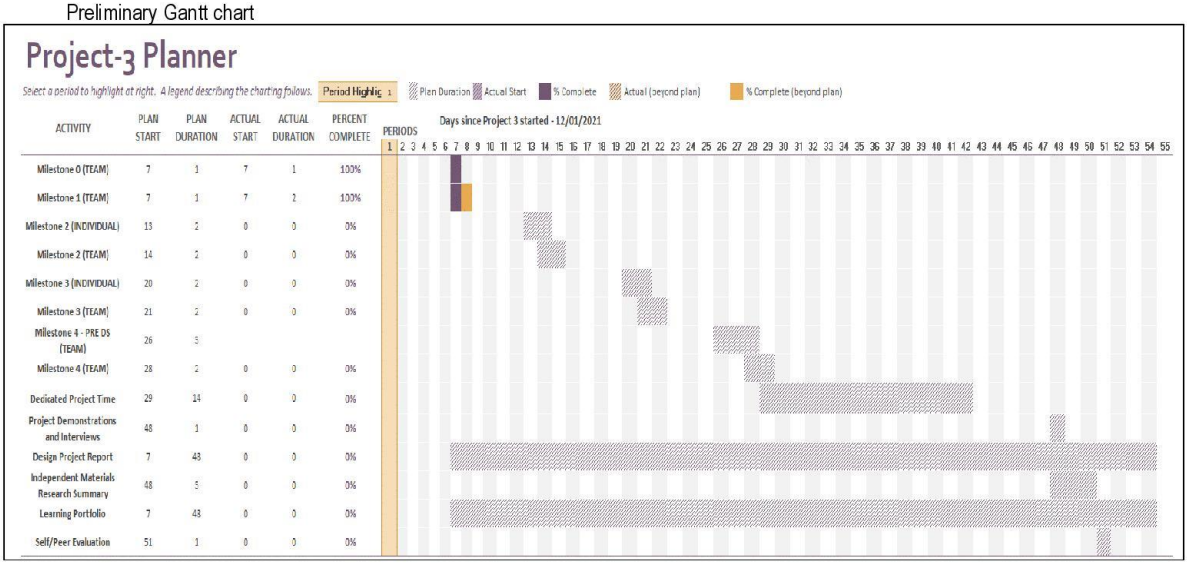
Role:	Team Member Name:	MacID
Manager	Mohammad Muntazar Bhurwani	bhurwanm
Administrator	Mila Ilijevska	ilijevm
Coordinator	Muhammad Saad Siddiqui	Siddim68
Subject Matter Expert	Aya Selman	Selmaa1

MILESTONE 0 – PRELIMINARY GANTT CHART (TEAM MANAGER ONLY)

Team Number: Mon-58

Full Name of Team Manager:	MacID:
Mohammad Muntazar Bhurwani	bhurwanm

Preliminary Gantt chart



Milestone 1:

PROJECT THREE: MILESTONE 1 – COVER PAGE

Team Number: Mon-58

Please list full names and MacID's of all *present* Team Members

Full Name:	MacID:
Aya Selman	Selmaa1
Mila Ilijevska	ilijevm
Mohammad Muntazar Bhurwani	bhurwanm
Muhammad Saad Siddiqui	Siddim68

MILESTONE 1 (STAGE 1) – WHY/HOW LADDERING

Team Number: Mon-58

1. Document both your conversation and a refined visual on a separate sheet of paper
2. Take a photo of both your rough work and refined visual
3. Insert each photo as a Picture (Insert > Picture > This Device)
4. **Do not include more than one Picture per page**

NEEDS HIERARCHY

Q. How do you design a system to sort and recycle containers?

- Identify objects using sensors, use multiple sensors for different criteria

① Why

- Ensure materials are not sent to the wrong bin
- Landfills get congested with incorrect materials, environmental issue

② Why should device be eco-friendly

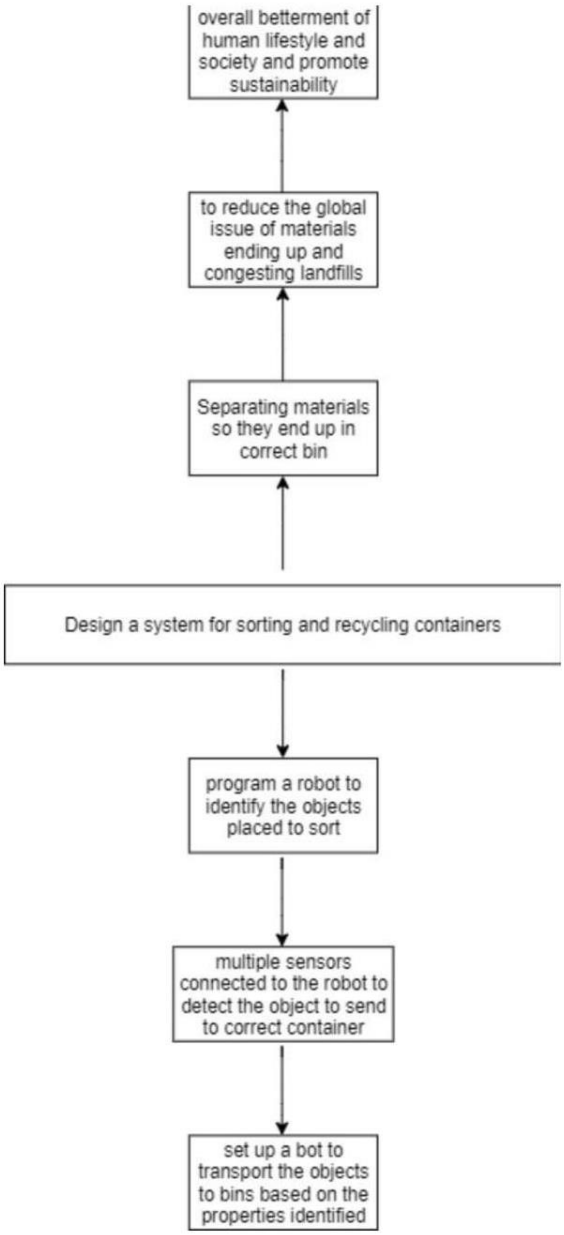
- Promote sustainability, better human lifestyle
- Pollution free, to battle enviroment issue

③ Why design a system

- Efficient, accurate, reliable

① How

- Program a robot, pick up object after sensors identify
- Set up bot to transport containers to right bins
- Identify using different sensors, based on phy/chem properties X (too solution oriented)


overall betterment of human lifestyle and society and promote sustainability
to reduce the global issue of materials ending up and congesting landfills
Separating materials so they end up in correct bin
Design a system for sorting and recycling containers
program a robot to identify the objects placed to sort
multiple sensors connected to the robot to detect the object to send to correct container
set up a bot to transport the objects to bins based on the properties identified

MILESTONE 1 (STAGE 2) – LIST OF OBJECTIVES AND CONSTRAINTS

Team Number: Mon-58

As a team, create a list of objectives and constraints in the table below. The exact number you should have depends on what information you have gathered from the Project Pack as well your previously completed needs hierarchy.

Objectives	1. Should accurately identify and classify the materials placed before the system 2. Should analyze container for contamination 3. Should verify if the container is recyclable or non-recyclable 4. Should transport and deposit the objects into their respective bins 5. Designed device should be able to deposit container into recycle bin
Constraints	• The total mass of the new container positioned in the Sorting Station as well as all the containers on the Q-bot must exceeds 90-grams. • There must be fewer than 3 containers already on the Q-bot when the Q-arm goes to pick up and load a container. • For a new container to be picked up, the new container must be transferred to the same bin as the containers already on the Q-bot. • The total mass of all containers on the Q-bot must be less than 90-grams, for a new container to be picked up. • Designed system must be environment friendly to promote sustainability. • The baseplate mounted to the Q-bot must connect to the device at 2 locations

MILESTONE 1 (STAGE 3) – REFINED PROBLEM STATEMENT

Team Number: Mon-58

Initial Problem Statement

6. Write the initial problem statement in the space below. This will have been defined in a previous lecture, prior to your scheduled Design Studio.

Design a system for sorting and recycling containers.

Refined Problem Statement

7. Write the refined problem statement below. Kindly refer to the Refined Problem Statement rubric provided on Avenue (see P3 Rubrics). This will guide your group in creating a valid statement.

Design a system that identifies and sorts materials, while also being able to transport and deposit them into their respective bins. The container must be analyzed for contamination and verified if it is recyclable, to avoid incorrect waste being deposited. This allows for the recycling system to be sped up and its accuracy greatly improved using autonomation.

Milestone 2:

PROJECT THREE: MILESTONE 2 – COVER PAGE

Team Number: Mon-58

Please list full names and MacID's of all *present* Team Members.

Full Name:	MacID:
Aya Selman	selmaa1
Muhammad Saad Siddiqui	siddim68
Mila Ilijevska	ilijevm
Mohammad Muntazar Bhurwani	bhurwanm

MILESTONE 2 (STAGE 1) – SENSOR RESEARCH (COMPUTATION SUB-TEAM)

Team Number: Mon-58

You should have already completed this task individually *prior* to Design Studio 14.

1. Each team member is expected to research 3 types of sensors for characterizing bins
   → Refer to Table 3 of the Computation Sub-Team Objectives document
2. For each sensor:
   → Briefly describe how the sensor works
   → Indicate the attribute you would measure to characterize each bin (refer to Table 4 of the Computation Sub-Team Objectives document)

We are asking that you submit your work on both worksheets. It does seem redundant, but there are valid reasons for this:

- Each team member needs to submit their sensor research with the **Milestone Two Individual Worksheets** document so that it can be ***graded***
- Compiling your individual work into this **Milestone Two Team Worksheets** document allows you to readily access your team member's work
  - This will be especially helpful when completing ***Stage 3*** of the milestone

Team Number: Mon-58

Name: Aya Selman	MacID: selmaa1

Sensor Type	Description	Attribute(s)
Colour sensor	-type of photoelectric sensor which emits light from a transmitter and the detects the light reflected with a receiver [1] -two types: 1)illuminates object with broad wavelength light and differentiates the three types of colour in the receiver. [1] 2)illuminates object with red, green, and blue light independently [1] -the received light intensity of red, green, and blue are detected and ratio of light received is calculated [1] [1]"What is a Colour sensor?," KEYENCE. [Online]. Available: https://www.keyence.ca/ss/products/sensor/sensorbasics/color/info/. [Accessed: Jan. 25, 2021].	-each bin should have a different colour
LDR (light dependent resistor)	-when light falls upon the LDR, the resistance changes [2] -LDRs have a high resistance b/c there are few electrons that are free enough to move (majority of electrons are locked in crystal lattice) so when light falls on it, light photons are absorbed, and some energy is transferred to electrons; they break free from lattice and conduct electricity- results in lowering of resistance [2] [2] "Light Dependent Resistor LDR: Photoresistor," Electronic Notes [Online]. Available: https://www.electronics-notes.com/articles/electronic_components/resistors/light-dependent-resistor-ldr.php. [Accessed: Jan. 25, 2021].	-would focus on whether the bin is metallic or not, and the roughness of the metallic bin
Ultrasonic sensor	-measures the distance of a target object by emitting ultrasonic sound waves and converts reflected sound into electronic signal [3] -have two main components: the transmitter and receiver [3] -sensor measures time it takes between sound wave emission and return [3]	-place each bin a certain distance away

	[3]D. Jost, "What is an Ultrasonic Sensor?," FierceElectronics [Online]. Available: https://www.fierceelectronics.com/sensors/what-ultrasonic-sensor#:~:text=An%20ultrasonic%20sensor%20is%20an,sound%20that%20humans%20can%20hear). [Accessed: Jan. 25, 2021].	and depending on the distance it would determine which bin it is

Team Number: Mon-58

Name: Mila Ilijevska	MacID: ilijevm

Sensor Type	Description	Attribute (s)
Retro-reflective Photoelectric Sensor	This sensor detects objects. It has a light source with collimating lens and a photoelement with condensing lens.[4] The light source releases a beam of light which an object can pass through. A reflector reflects the light from the light source and reflects it to the photo element. When an object is in the path of the light beam, it prevents the light from being returned by the reflector thus detecting the presence of an object. [4] "Retro-reflective photoelectric sensor" by R. W. Fayfield. (September 20th, 2005). US6946643B1. Accessed on: Jan. 23, 2021. [Online]. Available: https://patents.google.com/patent/US6946643B1/en	This sensor can the detect the distance that an object is away from a sensor which can be used to detect a bins placeme nt.
Active Infrared (IR) Sensor	This sensor detects the presence of objects/ things near the sensor. It is composed of three main parts: an optical emitter and an optical detector, and a microcontroller.[5] The optical emitter is controlled by the microcontroller and it emits radiation. The optical detectors sensor if radiation is reflected off on an object and this is processed by the microcontroller. Once the object is not in close proximity to the sensors, the detectors are able to detect the decrease in radiation reflected. [5]"Active infrared presence sensor"by J. G. Brewington, J. L. Levine, D.S. Miller and M. A. Schappert. (March 23th, 2004.). US6710346B2. Accessed on: Jan. 23, 2021. [Online]. Available: https://patents.google.com/patent/US6710346B2/en	The sensor can act like a proximity sensor to detect where a bin is placed.
Hall Sensor	Hall sensors measure the change in voltage of devices placed in magnetic fields to sense the presence of an object.[6] If a device is the magnetic field there will be a noticeable change.	The sensor could detect

	[6] "Hall sensor", by ゲオルグ レーラー、ゲオルグ レーラー. (2019, Feb. 06). US6946643B1. Accessed on: Jan. 23, 2021. [Online]. Available: https://patents.google.com/patent/JP6466875B2/en?q=Hall+Sensor&oq=Hall+Sensor	the presence of a metallic bin.

MILESTONE 2 (STAGE 2) – CONCEPT SKETCHES (MODELLING SUB-TEAM)

Team Number: Mon-58

You should have already completed this task individually *prior* to Design Studio 14.

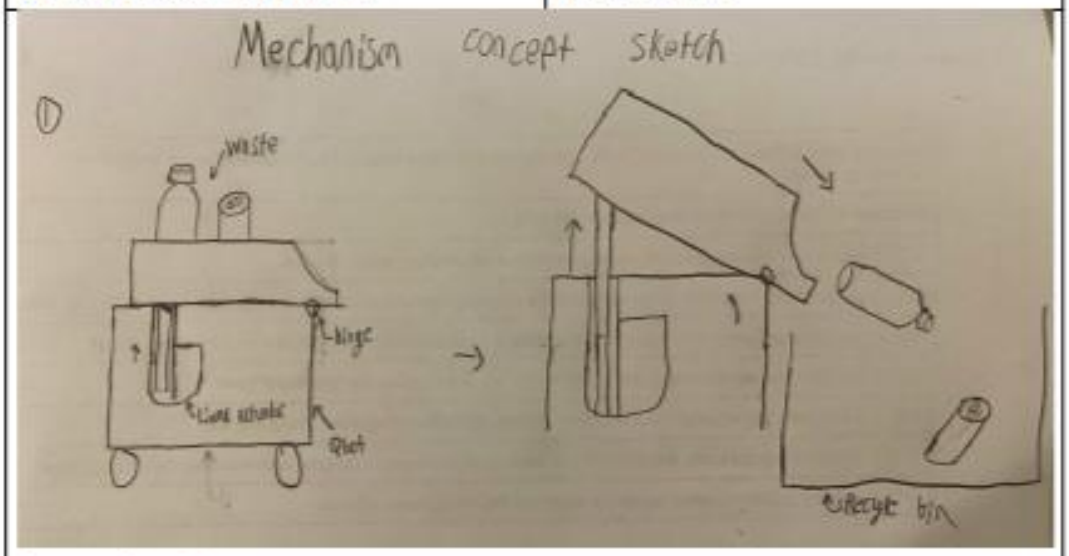
1. Copy-and-paste each sub-team member's refined sketch on the following pages (1 sketch per page)
   → Be sure to indicate each team member's Name and MacID

We are asking that you submit your work on both worksheets. It does seem redundant, but there are valid reasons for this:

- Each team member needs to submit their concept sketches with the **Milestone Two Individual Worksheets** document so that it can be ***graded***
- Compiling your individual work into this **Milestone Two Team Worksheets** document allows you to readily access your team member's work
  - This will be especially helpful when completing ***Stage 4*** of the milestone

Team Number: Mon-58

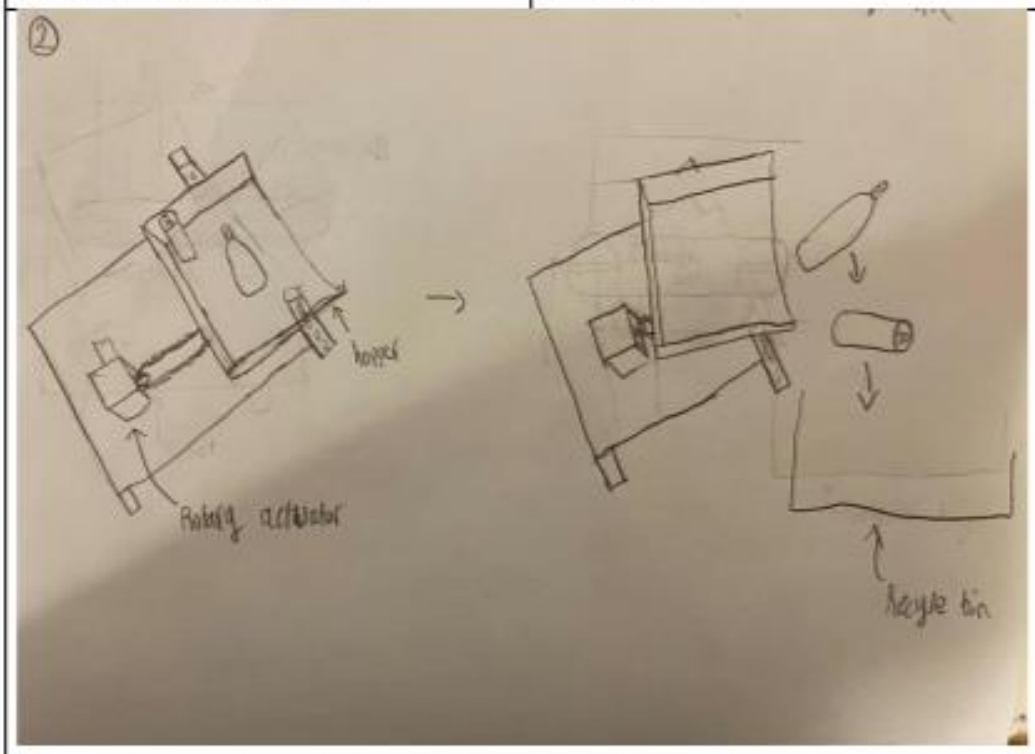
Name: Muhammad Saad Siddiqui	MacID: Siddim68



Concept Sketch 1

Team Number: Mon-58

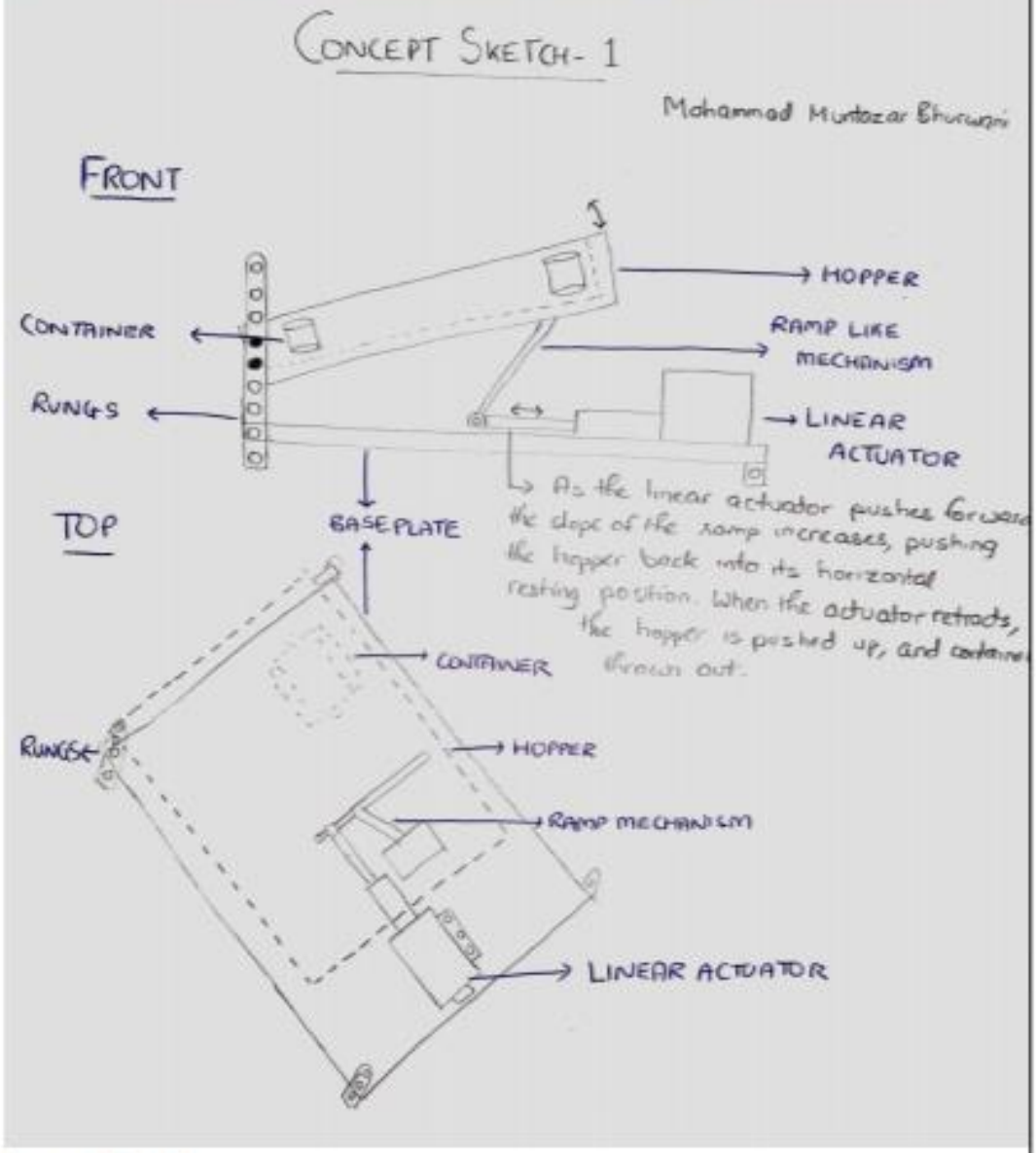
Name: Muhammad Saad Siddiqui	MacID: Siddim68



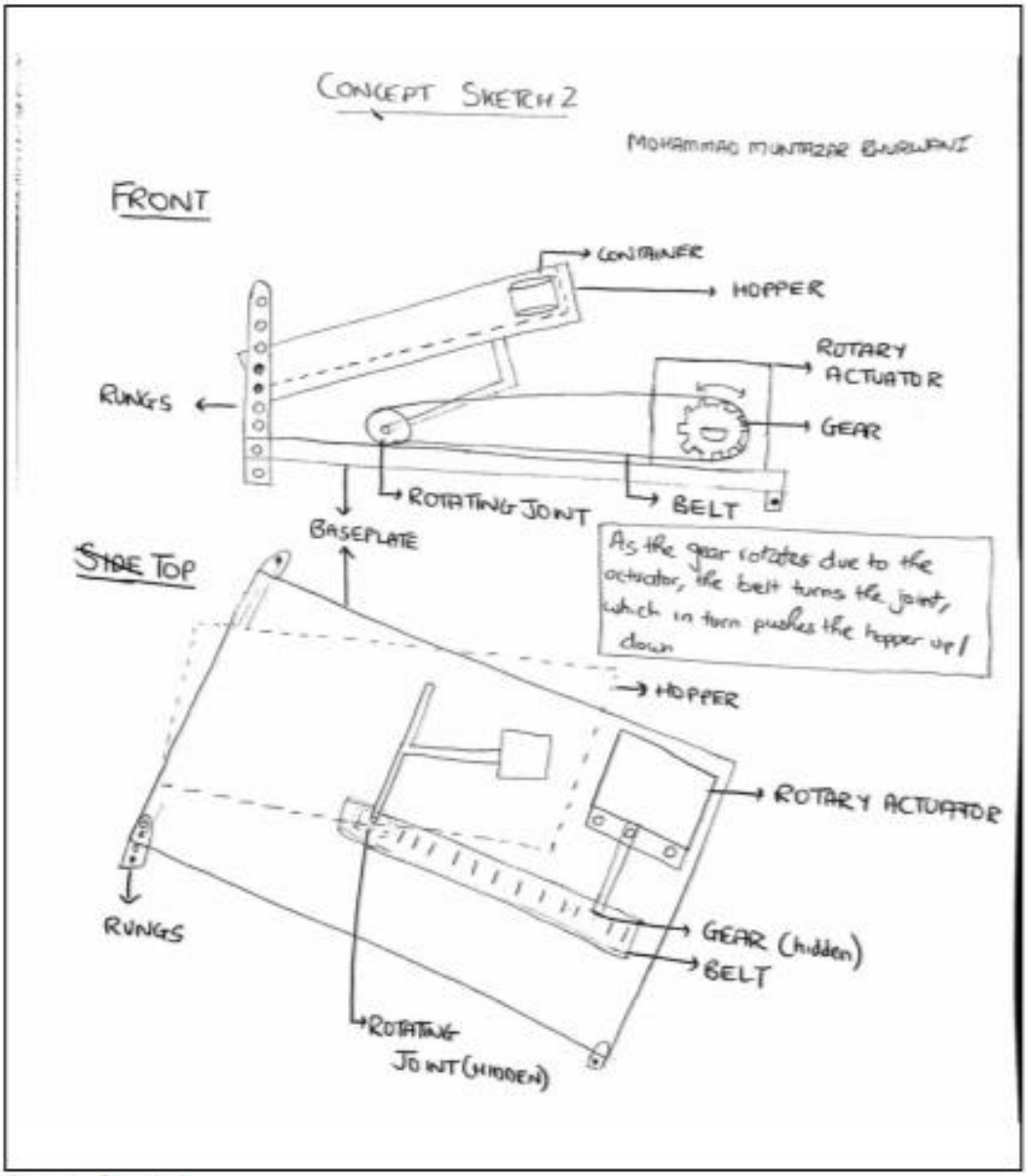
Concept Sketch 2

Team Number: Mon-58

Name: Mohammad Muntazar Bhurwani	MacID: bhurwanm



Concept Sketch 3



Concept Sketch 4

MILESTONE 2 (STAGE 3) – SENSOR CHARACTERIZATION (COMPUTATION SUB-TEAM)

Team Number: Mon-58

1. As a team, consolidate the results of your individual sensor research
   → Discuss your findings and appropriateness of each sensor for your application
   → Keep discussion brief, using point form

Sensor Type	Findings and Appropriateness for Application
Colour sensor	- Detects the colour of objects - casts red, green, and blue light onto objects and the received light intensity is detected and the ratio of light received is calculated - this sensor could be used to detect the colour of each bin and differentiate between them based on their colours
LDR (light dependent resistor)	- the resistance changes depending on how much light reaches the LDR - LDRs have a high resistance and the resistance lowers when light reaches the LDR - This sensor could differentiate the bins based on whether they are metallic and the roughness of the metallic bins
Ultrasonic sensor	- Detects the distance of an object. - Emits ultrasonic sound waves, measures time it takes between sound wave emission and return, and using the collected data it determines how far the bin is - This sensor could be used to detect the distance of a bin, and differentiate between them based on their respective distances away from the sensor
Retro-reflective Photoelectric Sensor	- Detects the presence of other objects. - It has a light source that releases a beam of light, which an object can pass through. - It has a reflection that can reflect light onto a photoelement. - If there is an object less light will be reflected. - Based on this, the sensor could be used to determine the presence of a bin and the distance it is away from a Q-bot.
Active Infrared (IR) Sensor	- It has an emitter that emits radiation that can be detected by a optical detector - If an object is present radiation will be reflected off it

	- This sensor could be used detect the presence of the bins.
Hall Sensor	- This sensor uses magnetic fields to detect the presence of an object. - An object would cause a difference in voltage in a magnetic field. - This could also be used to detect the presence and distance of a bin.

2. Identify one sensor to incorporate into your computer program

We chose the colour sensor to incorporate into our computer program. This would allow us to easily configure each bin to a different colour so that it would be simple to identify in a computer program.

3. Identify an attribute value for each bin

Bin ID	**Attribute Value**
Bin01: Metal Bin	Red: 1.0
Bin02: Paper Bin	Green: 1.0
Bin03: Plastic Bin	Blue: 1.0
Bin04: Garbage Bin	Red: 1.0 Blue: 1.0 The combination of the red and blue makes purple to have a fourth bin colour for identification.

MILESTONE 2 (STAGE 4) – DECISION MATRIX (MODELLING SUB-TEAM)

Team Number: Mon-58

1. As a team, establish a weighting factor for each criterion
   → Move row-by-row
   - If *Criteria 1* is preferred over *Criteria 2*, assign a 1. Otherwise, assign 0
   - If *Criteria 1* is preferred over *Criteria 3*, assign a 1. Otherwise, assign 0

   → Add additional rows/columns as needed

	Lightweight	*Simplicity*	*Effectiveness of motion*	Strength	**Score**
Lightweight	0	1	0	1	2
Simplicity	0	0	0	1	1
Effectiveness of motion	1	1	0	0	2
Strength	0	0	1	0	1

2. As a team, evaluate your concepts against each criterion using your weighting

	Weight	*Concept 1*		*Concept 2*		*Concept 3*		*Concept 4*	
		Rating	Weighted Rating	Rating	Weighted Rating	Rating	Weighted Rating	Rating	Weighted Rating
Lightweight	2	3.5	7	3	6	3.5	7	2.5	5
Simplicity	1	4	4	2	2	4	4	3.5	3.5
Effectiveness of motion	2	2	4	3	6	3	6	4	8
Strength	1	1	1	2.5	2.5	3	3	3	3
TOTAL		10.5	16	10.5	16.5	13.5	20	13	19.5

3. Discuss conclusions based on evaluation, including what concept you've chosen

Due to its lightweight and simple design, concept sketch 3 will be able to transport container more effectively than the other designs. Having a simple design allows us to and clean our mechanism with ease. Having a simple design allows us to dissemble and clean or tend to our mechanism with ease. When dealing with contaminated containers, spills can be a big problem and even interfere with complex parts. It also fulfills the requirements of the mechanism to be connected to the baseplate in 2 positions.

Another aspect of this design is that it can properly and securely hold the hopper in place reducing the chance of spilling containers. This design is very effective in terms of motion and allows us to take full advantage of the actuator. Compared to concept sketch number 4 which required gears and belts, Sketch 3's mechanism is much simpler. This is very important for a variety of reasons; with less moving parts we have less chance of error which increases our efficiency.

Compared to concept sketch 1, our mechanism is much sturdier and more reliable. In concept 1 the hopper is resting on the actuator itself and only that. This leads to more weight directly on the actuator and more chances of failure. **In conclusion, based on concept sketch 3 we will use a linear actuator with a ramp mechanism to transport our containers.**

Milestone 3:

PROJECT THREE: MILESTONE 3 – COVER PAGE

Team Number: Mon-58

Please list full names and MacID's of all *present* Team Members

Full Name:	MacID:
Mila Ilijevska	ilijevm
Aya Selman	selmaa1
Mohammad Muntazar Bhurwani	bhurwanm
Muhammad Saad Siddiqui	siddim68

MILESTONE 3 (STAGE 1A) – WORKFLOW PSEUDOCODE (COMPUTATION SUB-TEAM)

Team Number: Mon-58

You should have already completed this task individually *prior* to Design Studio 15.

1. Write out a pseudocode outlining the *high-level workflow* of your computer program on the following page
   → Only one team member is responsible for this task (not *both*)
   → Be sure to clearly indicate who each code belongs to

We are asking that you submit your work on both worksheets. It does seem redundant, but there are valid reasons for this:

- Each team member needs to submit their pseudocode with the **Milestone Three Individual Worksheets** document so that it can be ***graded***
- Compiling your individual work into this **Milestone Three Team Worksheets** document allows you to readily access your team member's work
  - This will be especially helpful when completing ***Stage 3*** of the milestone

Team Number: Mon-58

Name: Mila Ilijevska	MacID: ilijevm

Pseudocode

Start

While user has not prompted to stop

While all containers are going to the same bin **and** there are less than 3 containers placed on the Q-bot, **and** the total mass of all containers on the Q-bot is less than 90-grams.

Determine the mass of the container.

Set variable for the mass of the container.

Determine the material of the container.

Set variable for the material of the container.

Set variable for the appropriate destination bin (Bin01, Bin02, Bin03, Bin04), based on mass and material type.

Q-arm picks up and transfers container onto to Q-bot.

Q-arm returns to home position.

Q-bot transfers the containers to the recycling station following the line on the floor.

A sensor detects the bin type, and the Q-bot moves toward the correct bin based on data from sensors.

The Q-bot deposits the container into the bin.

Q-bot returns to home position.

End

MILESTONE 3 (STAGE 1B) – WORKFLOW FLOWCHART / STORYBOARD (COMPUTATION SUB-TEAM)

Team Number: Mon-58

You should have already completed this task individually *prior* to Design Studio 15.

1. Only one team member is responsible for this task (not *both*)
2. Copy-and-paste your flowchart or storyboard on the following page
   → Be sure to include your Team Number, Name and MacID
3. Take a photo of your flowchart / storyboard
4. Insert your photo as a Picture (Insert > Picture > This Device)

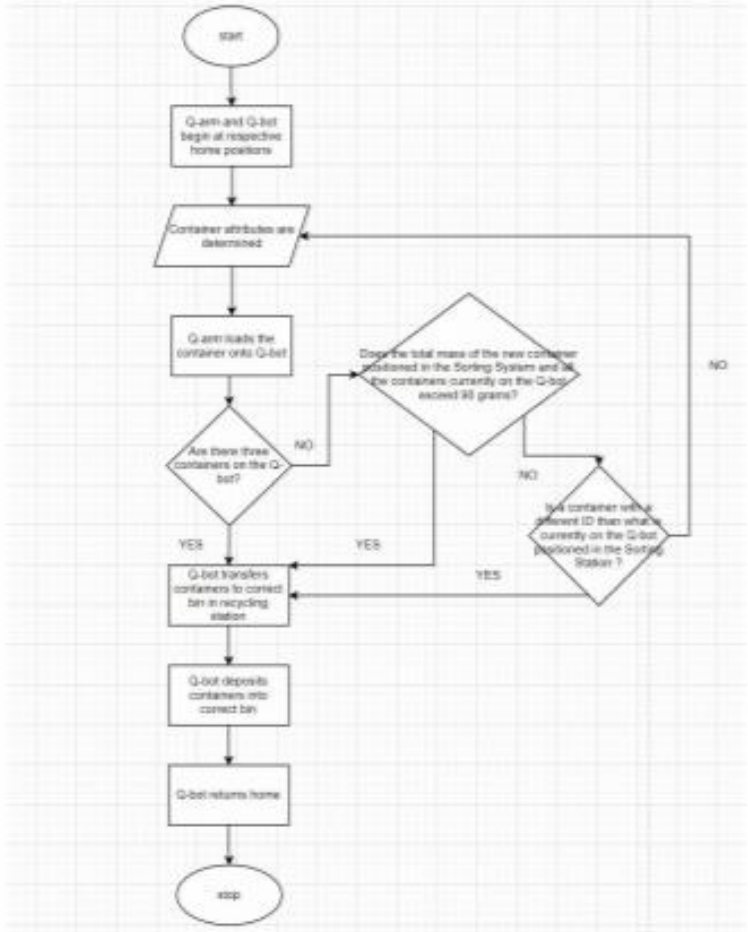
We are asking that you submit your work on both worksheets. It does seem redundant, but there are valid reasons for this:

- Each team member needs to submit their flowchart/storyboard screenshots with the **Milestone Three Individual Worksheets** document so that it can be ***graded***
- Compiling your individual work into this **Milestone Three Team Worksheets** document allows you to readily access your team member's work
  - This will be especially helpful when completing ***Stage 3*** of the milestone

Team Number: Mon-58

Name: Aya Selman	MacID: selmaa1

Insert screenshot(s) of your flowchart or storyboard.



MILESTONE 3 (STAGE 2) – DETAILED SKETCHES (MODELLING SUB-TEAM)

Team Number: Mon-58

You should have already completed this task individually *prior* to Design Studio 15.

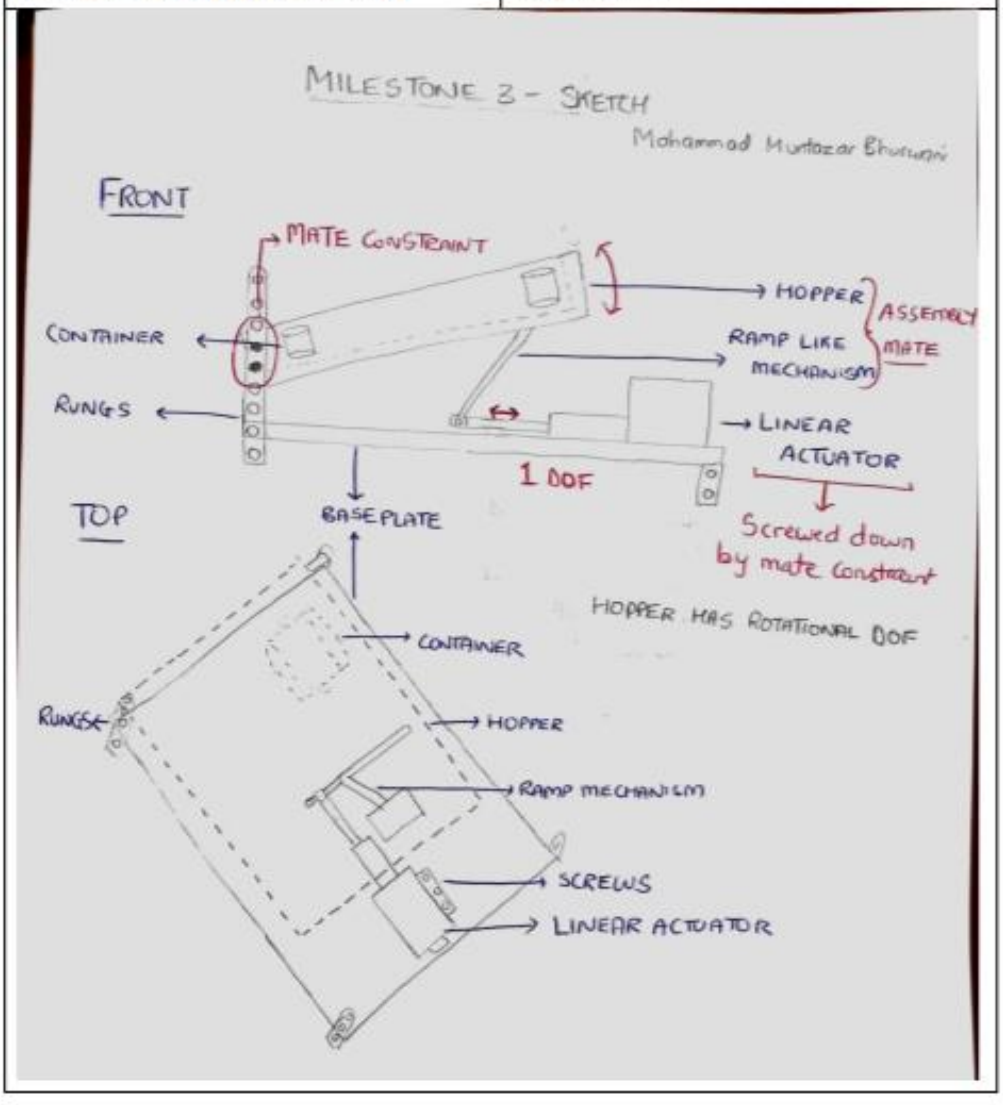
1. Copy-and-paste each sub-team member's detailed sketch on the following pages (1 sketch per page)
   → Be sure to indicate each team member's Name and MacID

We are asking that you submit your work on both worksheets. It does seem redundant, but there are valid reasons for this:

- Each team member needs to submit their detailed sketches with the **Milestone Three Individual Worksheets** document so that it can be ***graded***
- Compiling your individual work into this **Milestone Three Team Worksheets** document allows you to readily access your team member's work
  - This will be especially helpful when completing ***Stage 4*** of the milestone

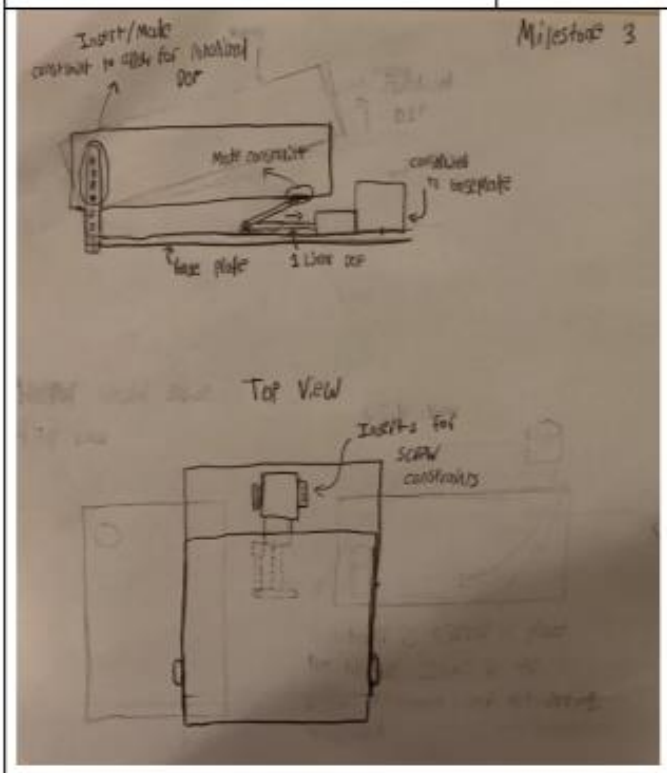
Team Number: Mon-58

Name: Mohammad Muntazar Bhurwani	MacID: bhurwanm



Team Number: Mon-58

Name: Muhammad Saad Siddiqui	MacID: Siddim68



MILESTONE 3 (STAGE 3) – PROGRAM TASK PLANNING (COMPUTATION SUB-TEAM)

Team Number: Mon-58

1. As a team, write out the pseudocode or create a flowchart for the indicated tasks in the space below.
   → If creating a flowchart, complete your flowchart on a separate sheet of paper, take a photo of your sketch and insert photo as a Picture (Insert > Picture > This Device)

Dispense Container

Start

Container mass is determined and assigned a variable

Container material is determined and assigned a variable

Container target bin is determined and assigned a variable

Container is positioned in sorting station for pick-up by Q-arm

Stop

Load Container

Start

Q-arm moves and positions gripper end-effector adjacent to container.

Q-arm picks up container by closing gripper around container.

Q-arm transfers the container to hopper mounted on the Q-bot.

Q-arm releases the container by opening the gripper

Q-arm moves back to home position

Stop

Transfer Container

Start

Colour sensor is activated using the built-in function within the P3 library.

Q-bot moves forward along the trajectory of a line on the floor.

Colour sensor determines which bin it needs to go to.

If the container is metal, the Q-bot goes to Bin01, which is red.

If the container is paper, the Q-bot goes to Bin02, which is green.

If the container is plastic, the Q-bot goes to Bin03, which is blue.

If the container is garbage, the Q-bot goes to Bin04, which is purple.

Q-bot follows the trajectory of the line corresponding to the correct bin and stops in front of the correct bin.

Sensor is deactivated using the built-in function within the P3 library.

Stop

Deposit Container

Start

Rotate Q-bot 90-degrees clockwise.

Camera on Q-bot measures the distance the Q-bot should travel.

Q-bot traves forward.

Q-bot stops once it is close to the bin.

Rotate Q-bot 90-degrees counterclockwise.

The hooper rotates until all the containers fall into the bin. (function in built-in library)

Rotate Q-bot 90-degrees counterclockwise.

Q-bot should travel forward, following line offset to the floor.

Once the line cannot be found, Q-bot rotates 90-degrees clockwise.

End

Return Home

Start

Q-bot moves forward along trajectory of the line on the floor.

Once the line stops, Q-bot stops moving.

Q-bot rotates 180-degreees.

End

MILESTONE 3 (STAGE 4) – PRELIMINARY MODELLING (MODELLING SUB-TEAM)

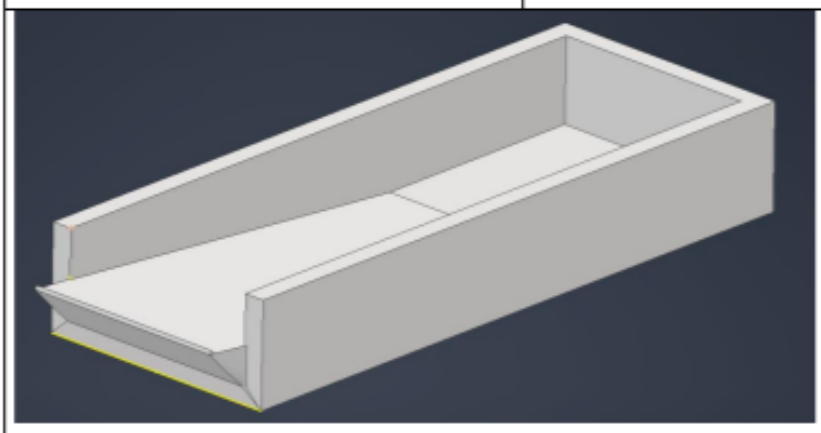
Team Number: Mon-58

1. As a team, create solid models of the various components of your device in Autodesk Inventor, based on the detailed sketches.
   → Take multiple screenshots of each solid model you create
   → Insert your photo(s) as a Picture (Insert > Picture > This Device)
   → **Do not include more than *two* solid modelling screenshots per page**

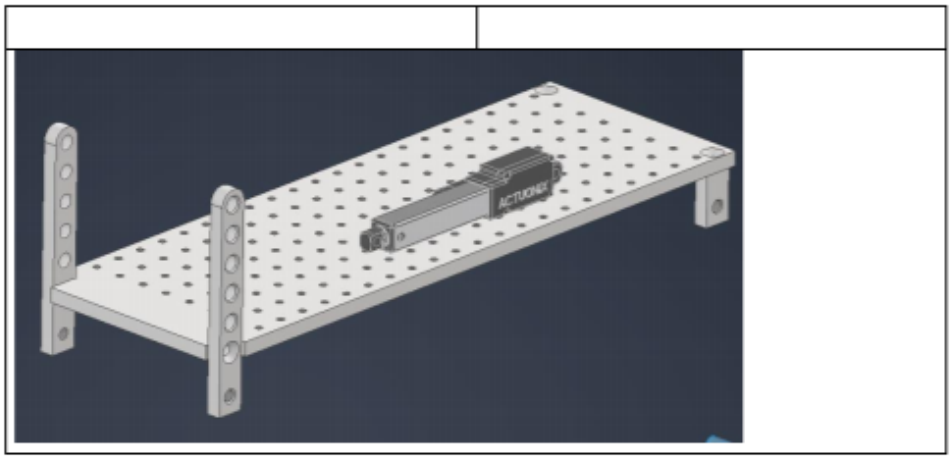
Return Home

Start
Q-bot moves forward along trajectory of the line on the floor.
Once the line stops, Q-bot stops moving.
Q-bot rotates 180-degreees.
End

Team Number: Mon-58



Team Number: Mon-58



Milestone 4:

PROJECT THREE: MILESTONE 4 – COVER PAGE

Team Number: Mon-58

Please list full names and MacID's of all *present* Team Members

Full Name:	MacID:
Mohammad Muntazar Bhurwani	bhurwanm
Aya Selman	selmaa1
Mila Ilijevska	ilijevm
Muhammad Saad Siddiqui	Siddim68

MILESTONE 4 (STAGE 3) – DESIGN REVIEW FEEDBACK (MODELLING SUB-TEAM)

Team Number: Mon-58

Use the space below to document mentor feedback for your design.

- Looks sorted – Mechanism seems simplistic
- Actuator is placed within the specified range
- Successfully holds all 3 containers

Use the space below to propose design refinements based on the feedback.

- Remove unnecessary DOF
- Add support for hopper at rest

MILESTONE 4 (STAGE 3) – DESIGN REVIEW FEEDBACK (COMPUTATION SUB-TEAM)

Team Number: Mon-58

Use the space below to document mentor feedback for your design.

- Possible issue: loop the other way (TA confirmed it did not matter)
- Try to play around with the parameters passed into deposit function

Use the space below to propose design refinements based on the feedback.

- Change the length of time that the bot moves forward in the deposit function to ensure it works properly.

List of Sources

Source Material Database:

[1] “What is a Colour sensor?,” KEYENCE. [Online]. Available: https://www.keyence.ca/ss/products/sensor/sensorbasics/color/info/. [Accessed: Jan. 25, 2021].

[2] “Light Dependent Resistor LDR: Photoresistor,” Electronic Notes [Online]. Available: https://www.electronics-notes.com/articles/electronic_components/resistors/light-dependent-resistor-ldr.php. [Accessed: Jan. 25, 2021].

[3] D. Jost, “What is an Ultrasonic Sensor?,” FierceElectronics [Online]. Available: https://www.fierceelectronics.com/sensors/what-ultrasonic-sensor#:~:text=An%20ultrasonic%20sensor%20is%20an,sound%20that%20humans%20can%20hear). [Accessed: Jan. 25, 2021].

[4] R. W. Fayfield, “US6946643B1 - Retro-reflective photoelectric sensor,” Google Patents. [Online]. Available: https://patents.google.com/patent/US6946643B1/en. [Accessed: Jan. 23, 2021].

[5] J. G. Brewington, J. L. Levine, D. S. Miller, and M. A. Schappert, “US6710346B2 - Active infrared presence sensor,” Google Patents. [Online]. Available: https://patents.google.com/patent/US6710346B2/en. [Accessed: Jan. 23, 2021].

[6] ゲオルグ レーラー and ゲオルグ レーラー, “JP6466875B2 - Hall sensor,” Google Patents. [Online]. Available: https://patents.google.com/patent/JP6466875B2/en?q=Hall%2BSensor&oq=Hall%2BSensor. [Accessed: 23-Jan-2021].

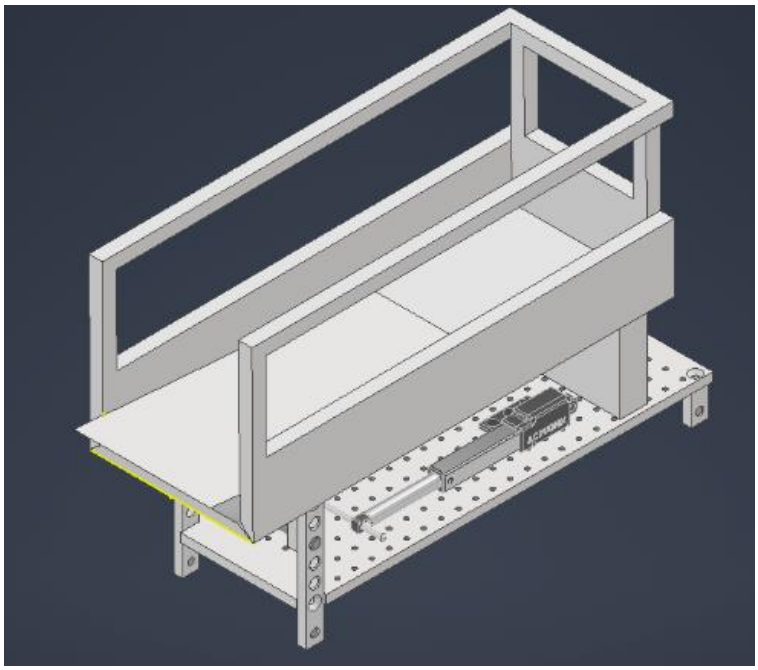
[7] 1P13 Staff, “P3 Project Module,” McMaster University Online Courses. [Online]. Available: https://avenue.cllmcmaster.ca/d2l/le/content/340370/viewContent/3021767/View. [Accessed: Jan. 2021].

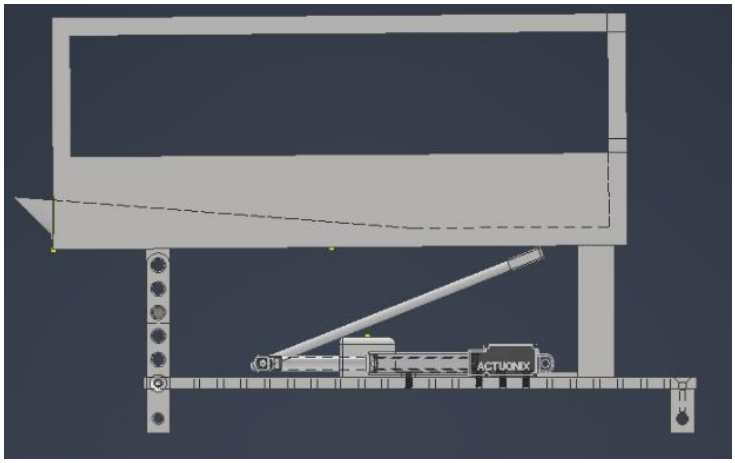
[8] 1P13 Staff, “P3 Python Library Documentation,” McMaster University Online Courses. [Online]. Available: https://avenue.cllmcmaster.ca/d2l/le/content/340370/viewContent/3012232/View. [Accessed: Feb. 2021].

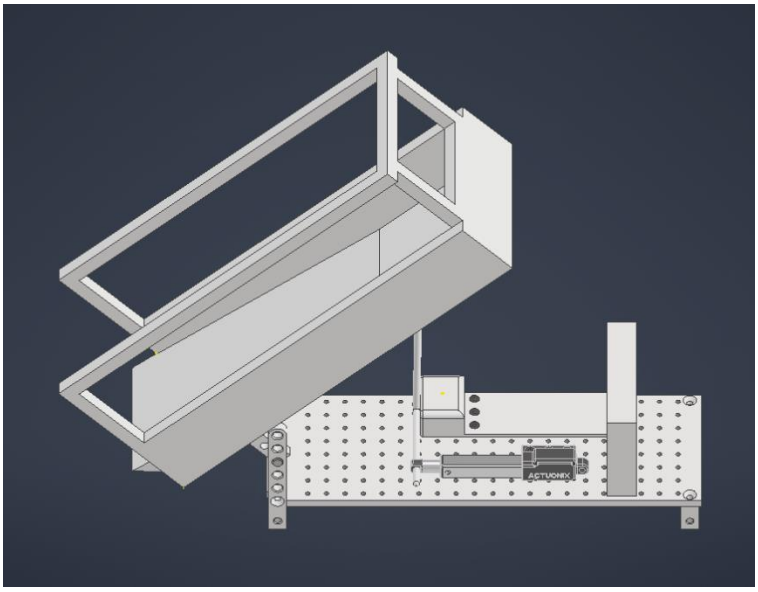
[9] 1P13 Staff, “P3 Q-Labs Environment Configuration,” McMaster University Online Courses. [Online]. Available: https://avenue.cllmcmaster.ca/d2l/le/content/340370/viewContent/3012233/View. [Accessed : Feb. 2021].

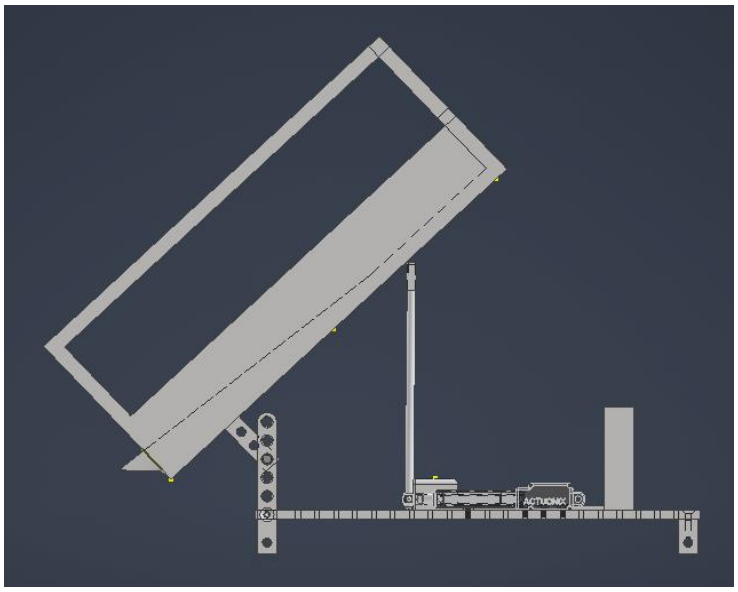
Appendix A – Modelling:

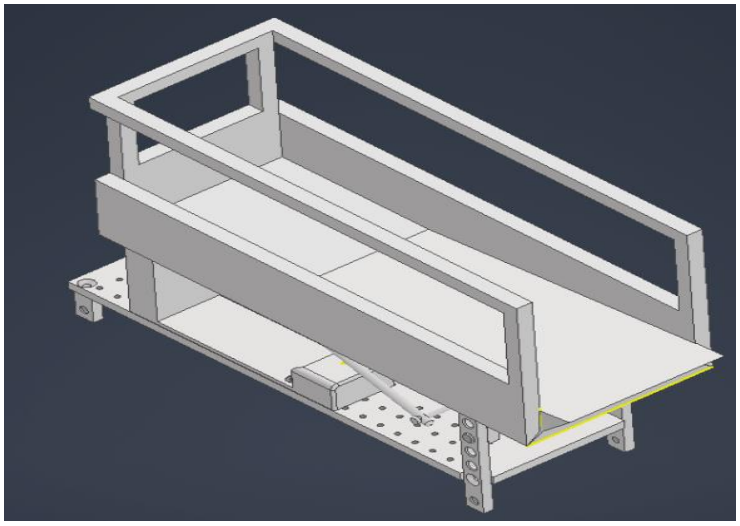
Screenshots of Solid Model

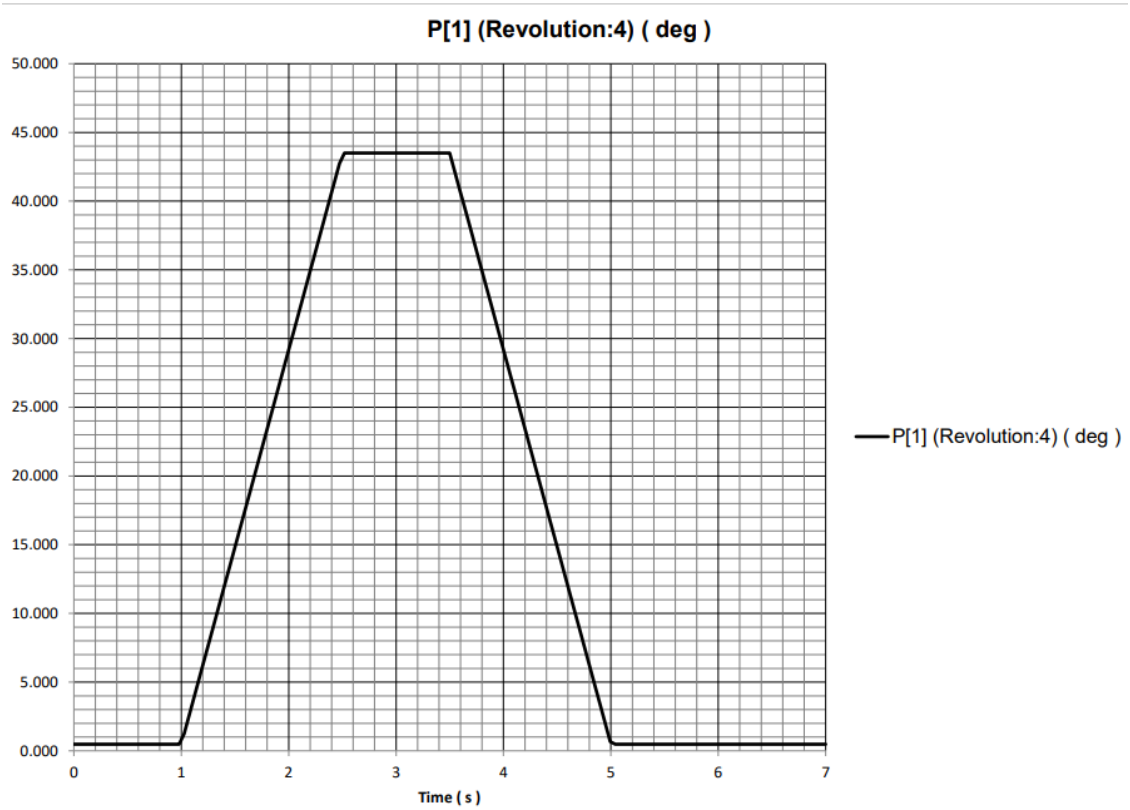


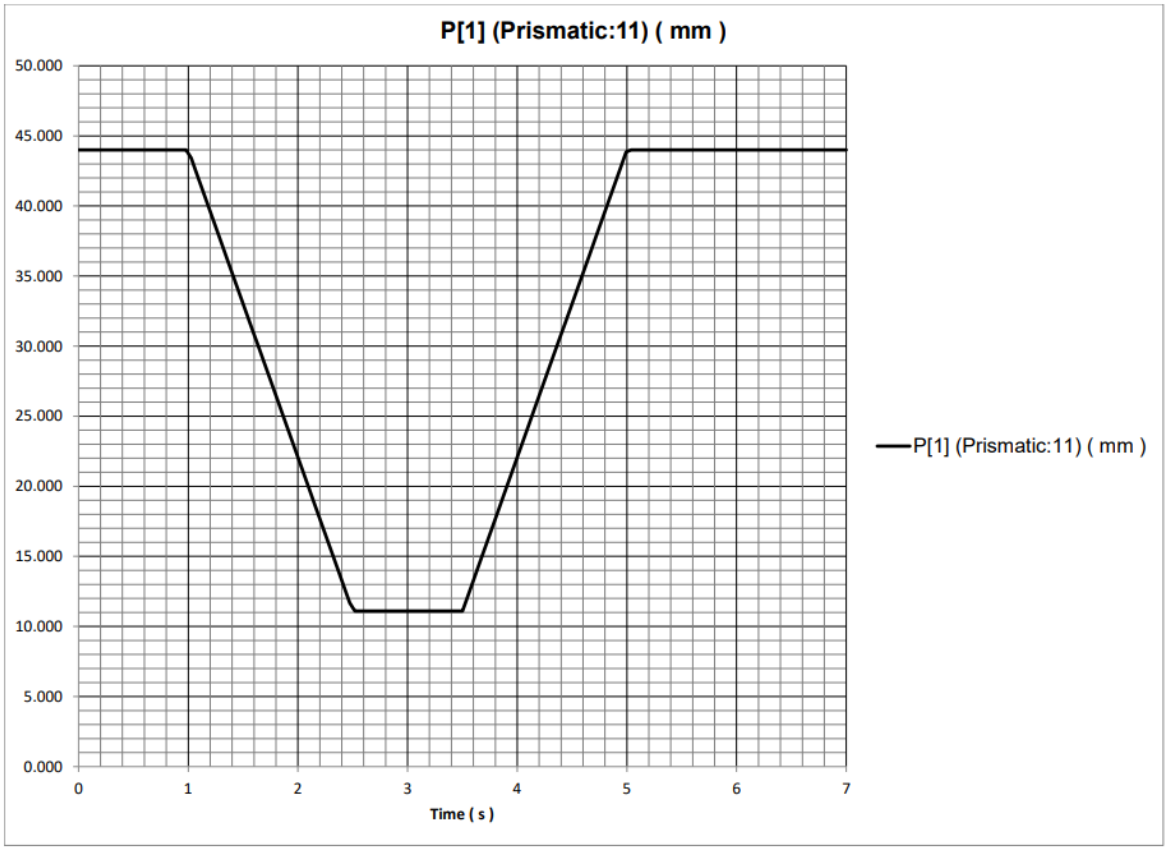


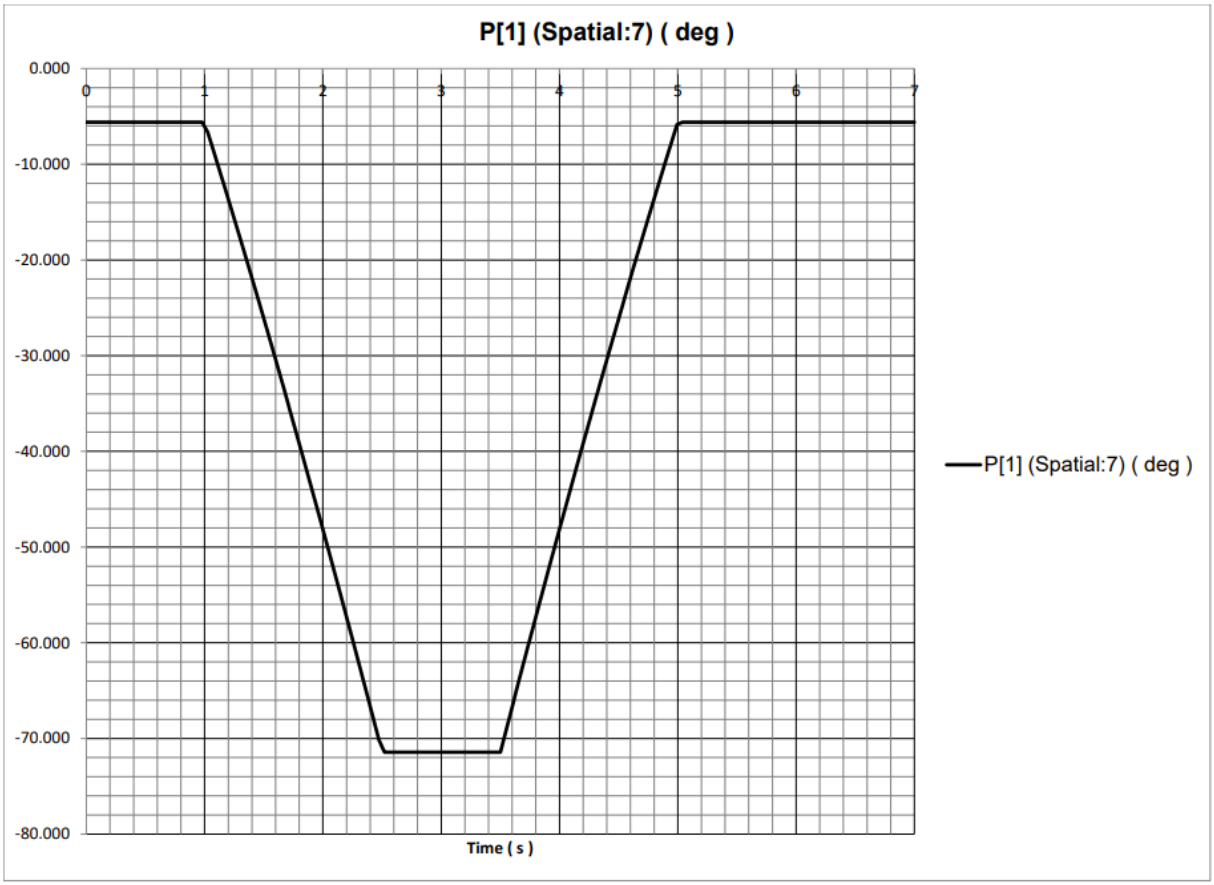






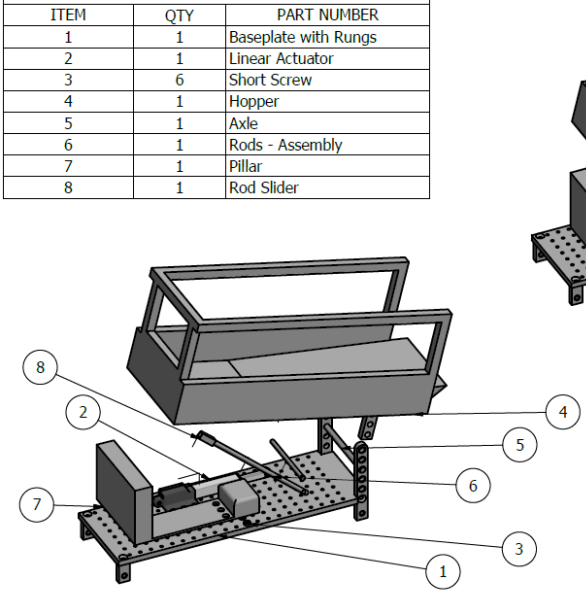

P[1] (Revolution:4) (deg)
50.000
45.000
40.000
35.000
30.000
25.000
20.000
15.000
10.000
5.000
0.000
0
1
2
3
4
5
6
7
Time (s)
P[1] (Revolution:4) (deg)

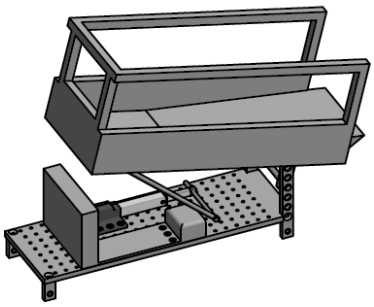

P[1] (Prismatic:11) (mm)
50.000
45.000
40.000
35.000
30.000
25.000
20.000
15.000
10.000
5.000
0.000
0
1
2
3
4
5
6
7
Time (s)
P[1] (Prismatic:11) (mm)


P[1] (Spatial:7) (deg)
0.000
-10.000
-20.000
-30.000
-40.000
-50.000
-60.000
-70.000
-80.000
0
1
2
3
4
5
6
7
Time (s)
P[1] (Spatial:7) (deg)

Fully Dimensioned Engineering Drawings:

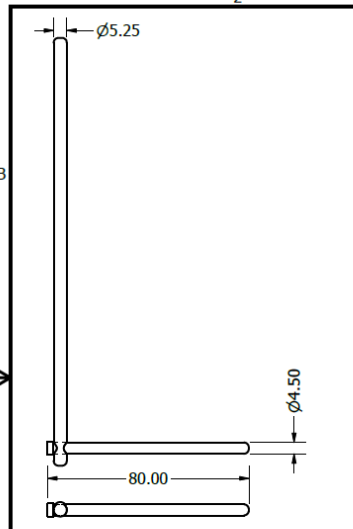
PARTS LIST		
ITEM	QTY	PART NUMBER
1	1	Baseplate with Rungs
2	1	Linear Actuator
3	6	Short Screw
4	1	Hopper
5	1	Axle
6	1	Rods - Assembly
7	1	Pillar
8	1	Rod Slider

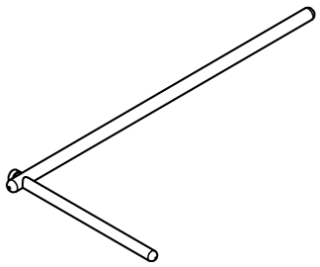


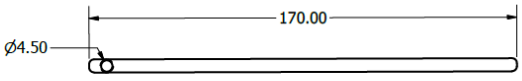


McMaster University — ENGINEER 1P13 Integrated Cornerstone Design Projects in Engineering

STUDENT NUMBER	LAB SECTION
MON-58	T15
Exploded View - Full Assembly	
DATE: 2/1/2021	SCALE: 1 / 4
ALL DIMENSIONS IN MILLIMETRES	
SHEET SIZE A	SHEET 1 OF 1

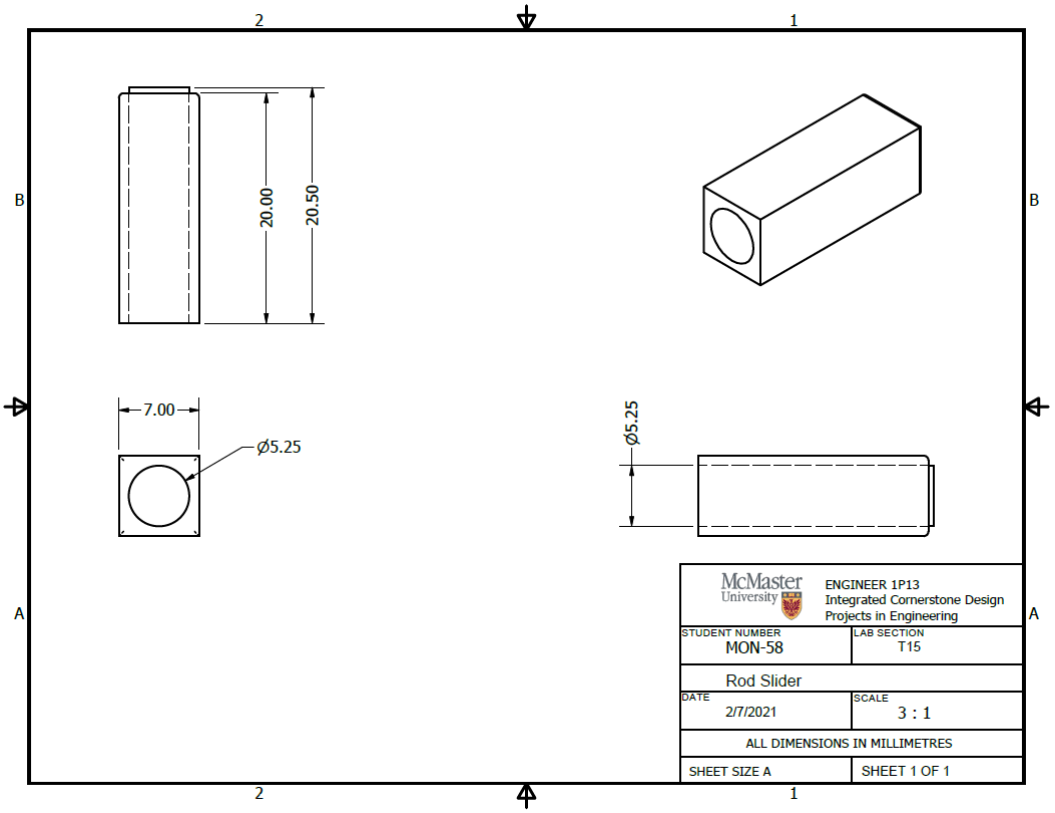


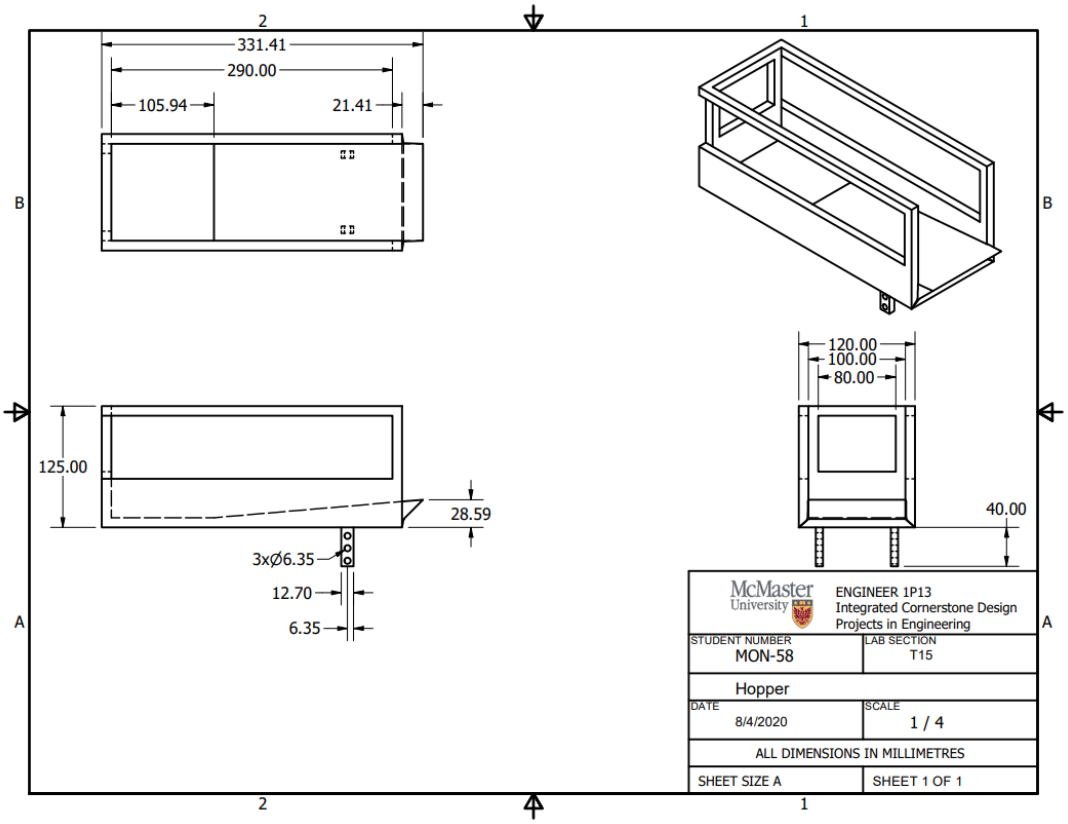


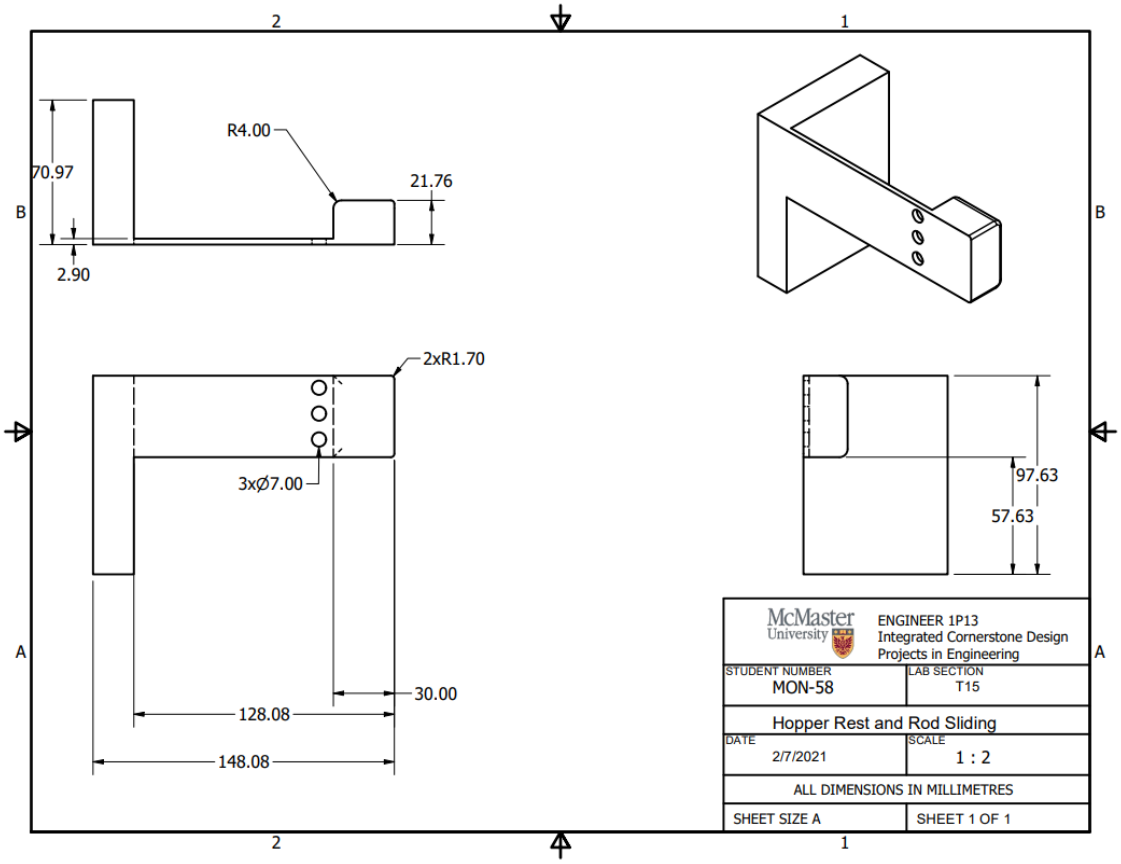


McMaster University — ENGINEER 1P13 Integrated Cornerstone Design Projects in Engineering

STUDENT NUMBER	LAB SECTION
MON-58	T15
Assembled Rods	
DATE: 2/7/2021	SCALE: 2/3
ALL DIMENSIONS IN MILLIMETRES	
SHEET SIZE A	SHEET 1 OF 1


20.00
20.50
7.00
Ø5.25
Ø5.25
McMaster University
ENGINEER 1P13
Integrated Cornerstone Design
Projects in Engineering
STUDENT NUMBER
MON-58
LAB SECTION
T15
Rod Slider
DATE
2/7/2021
SCALE
3 : 1
ALL DIMENSIONS IN MILLIMETRES
SHEET SIZE A
SHEET 1 OF 1


331.41
290.00
105.94
21.41
125.00
28.59
3xØ6.35
12.70
6.35
120.00
100.00
80.00
40.00
McMaster University
ENGINEER 1P13
Integrated Cornerstone Design
Projects in Engineering
STUDENT NUMBER
MON-58
LAB SECTION
T15
Hopper
DATE
8/4/2020
SCALE
1 / 4
ALL DIMENSIONS IN MILLIMETRES
SHEET SIZE A
SHEET 1 OF 1


R4.00
70.97
21.76
2.90
2xR1.70
3xØ7.00
30.00
128.08
148.08
97.63
57.63
McMaster University
ENGINEER 1P13
Integrated Cornerstone Design
Projects in Engineering
STUDENT NUMBER
MON-58
LAB SECTION
T15
Hopper Rest and Rod Sliding
DATE
2/7/2021
SCALE
1 : 2
ALL DIMENSIONS IN MILLIMETRES
SHEET SIZE A
SHEET 1 OF 1

Appendix B - Computation:

Computer Program:

```
import time
import random
import sys
sys.path.append('../')

from Common_Libraries.p3b_lib import *

import os
from Common_Libraries.repeating_timer_lib import repeating_timer

def update_sim():
    try:
        my_table.ping()
    except Exception as error_update_sim:
        print (error_update_sim)

### Constants
speed = 0.2 #Qbot's speed

### Initialize the QuanserSim Environment
my_table = servo_table()
arm = qarm()
arm.home()
bot = qbot(speed)

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

'''
Mon - 58 Coding Subteam
Mila Ilijevska (ilijevm)
Aya Selman (selmaa1)
```

```
import time
import random
import sys
sys.path.append('../')

from Common_Libraries.p3b_lib import *

import os
from Common_Libraries.repeating_timer_lib import repeating_timer

def update_sim():
    try:
        my_table.ping()
    except Exception as error_update_sim:
        print (error_update_sim)

### Constants
speed = 0.2 #Qbot's speed

### Initialize the QuanserSim Environment
my_table = servo_table()
arm = qarm()
arm.home()
bot = qbot(speed)

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

'''
Mon - 58 Coding Subteam
Mila Ilijevska (ilijevm)
Aya Selman (selmaa1)
def dispense_container(bin_type, container_ID, weight):
    '''
    Function: dispense_container()

    Purpose: This function saves a containers properties in perspective variables, updates the
    weight of all the containers, and dispenses a container.

    Input: bin_type - A string that holds the type of bin each container goes to.
           container_ID - Needed to determine the properties of a container
           weight - Represents the weight of container, needs to be updated

    Output: bin_type - Returns bin type destination of the new container.
            tweight - Returns weight of the new container.
    '''
    # Saves each container property into its own variable by using the index of each.
    material = my_table.container_properties(container_ID)[0]
    weight = my_table.container_properties(container_ID)[1]
    bin_type = my_table.container_properties(container_ID)[2]
    my_table.dispense_container()
    print("Material: ", material)
    print ("Bin type is: ", bin_type)
    print("The weight is ", weight)

    return bin_type, weight
```

```
def load_container(container_count):
    '''
    Function: load_container()

    Purpose: Based on set conditions, this function controls the Q-arms to up pick
    containers in the sorting station and load them on the Q-bot if their total weight
    is less than 90g, there are less than three containers on the Q-bot, and if the
    container matches all other bin destinations on the Q-bot.

    Input: container_count - represents the number of containers on the Q-bot

    Output:No variables returned. Motion described in purpose is executed within the funcion.
    '''

    # Represents the x,y,z coordinates for the 3 bottle drop off locations on the Q-bot
    position1 = [-0.1116, -0.4165, 0.4166]
    position2 = [0.0, -0.4048, 0.4166]
    position3 = [0.1048, -0.391, 0.4166]

    pick_up_location = [0.6726, 0.0, 0.285]

    # The following picks up the container
    arm.move_arm(pick_up_location[0], pick_up_location[1], pick_up_location[2])
    arm.control_gripper(45)
    arm.rotate_base(-55)
    arm.rotate_shoulder(-50)

    # The following determines which position the container should be loaded into.
    if (container_count == 1):
        arm.move_arm(position1[0], position1[1], position1[2])
    elif (container_count ==2):
        arm.move_arm(position2[0], position2[1], position2[2])
    elif (container_count==3):
        arm.move_arm(position3[0], position3[1], position3[2])

    # Places the container on the Q-bot
    arm.control_gripper(-20)
    time.sleep(3)
    arm.rotate_shoulder(-30)
    arm.home()
```

```
def transfer_container(bin_type):
    '''
    Function: dispense_container()

    Purpose: Function uses the threshold on the color sensor to transfer the containers on the
    Q-bot to their proper bin.

    Input: bin_type - Represents which bin the Q-bot needs to go to.

    Output: No ovariables return. This function transfers the containers to their proper
    destination bin.
    '''

    # The following assigns colours based on the bin type
    if (bin_type == 'Bin01'):
        color= 'Red'
    elif (bin_type == 'Bin02'):
        color= 'Green'
    elif (bin_type == 'Bin03'):
        color= 'Blue'
    else:
        color = 'Purple'

    # Depending on the bin type the Q-bot will move forward a certain amount
    # The color sensor measures the threshold values for the color associated with each bin.
    if (color == 'Red'):
        bot.activate_color_sensor(color)
        bot.forward_time(4.3)
        threshold = bot.read_red_color_sensor("Bin01", 0.6)
        bot.deactivate_color_sensor()
    elif (color == 'Green'):
        bot.activate_color_sensor(color)
        bot.forward_time(6.2)
        threshold = bot.read_green_color_sensor("Bin02", 0.6)
        bot.deactivate_color_sensor()
    elif (color == 'Blue'):
        bot.activate_color_sensor(color)
        bot.forward_time(8.1)
        threshold = bot.read_blue_color_sensor("Bin03", 0.6)
        bot.deactivate_color_sensor()
    else:
        bot.activate_color_sensor("Red")
        # The purple box uses  red color sesnor as it has red in it
        bot.forward_time(10)
        threshold = bot.read_red_color_sensor("Bin04", 0.6)
        bot.deactivate_color_sensor()

    #  To be infront the correct bin, the color sensor must read certain threshold
    if (threshold[1]> 4.5 or (threshold[1]<0.5)):
        #deposit_container()
        alternative_deposit_container()
```

```
def deposit_container():
    '''
    This function is currently NOT called as we did the bonus approach.

    Function: deposit_container()

    Purpose: This function moves the Q-bot close to the correct bin and angles
    the Q-bot in a way to safely deposit the containers and then uses the dump function
    to deposit the containers.

    Input: No variable parameters. In order for this function to run, containers
    must be loaded onto the Q-bot

    Output: Q-bot deposits the containers.

    '''
    # Follow line to bin
    bot.rotate(90)
    time.sleep(1)
    bot.forward_time(1)

    # The following makes the qbot adjacent to the bins
    bot.rotate(-92)
    time.sleep(1)
    bot.activate_actuator()

    bot.dump()
    time.sleep(1)
    bot.rotate(-92)
    bot.forward_time(1)
    bot.rotate(90)

    return_home()
```

```
def dispense_container(bin_type, container_ID, weight):
    '''
    Function: dispense_container()

    Purpose: This function saves a containers properties in perspective variables, updates the
    weight of all the containers, and dispenses a container.

    Input: bin_type - A string that holds the type of bin each container goes to.
           container_ID - Needed to determine the properties of a container
           weight - Represents the weight of container, needs to be updated

    Output: bin_type - Returns bin type destination of the new container.
            tweight - Returns weight of the new container.
    '''
    # Saves each container property into its own variable by using the index of each.
    material = my_table.container_properties(container_ID)[0]
    weight = my_table.container_properties(container_ID)[1]
    bin_type = my_table.container_properties(container_ID)[2]
    my_table.dispense_container()
    print("Material: ", material)
    print ("Bin type is: ", bin_type)
    print("The weight is ", weight)

    return bin_type, weight
```

```
def load_container(container_count):
    '''
    Function: load_container()

    Purpose: Based on set conditions, this function controls the Q-arms to up pick
    containers in the sorting station and load them on the Q-bot if their total weight
    is less than 90g, there are less than three containers on the Q-bot, and if the
    container matches all other bin destinations on the Q-bot.

    Input: container_count - represents the number of containers on the Q-bot

    Output:No variables returned. Motion described in purpose is executed within the funcion.
    '''

    # Represents the x,y,z coordinates for the 3 bottle drop off locations on the Q-bot
    position1 = [-0.1116, -0.4165, 0.4166]
    position2 = [0.0, -0.4048, 0.4166]
    position3 = [0.1048, -0.391, 0.4166]

    pick_up_location = [0.6726, 0.0, 0.285]

    # The following picks up the container
    arm.move_arm(pick_up_location[0], pick_up_location[1], pick_up_location[2])
    arm.control_gripper(45)
    arm.rotate_base(-55)
    arm.rotate_shoulder(-50)

    # The following determines which position the container should be loaded into.
    if (container_count == 1):
        arm.move_arm(position1[0], position1[1], position1[2])
    elif (container_count ==2):
        arm.move_arm(position2[0], position2[1], position2[2])
    elif (container_count==3):
        arm.move_arm(position3[0], position3[1], position3[2])

    # Places the container on the Q-bot
    arm.control_gripper(-20)
    time.sleep(3)
    arm.rotate_shoulder(-30)
    arm.home()
def transfer_container(bin_type):
    '''
    Function: dispense_container()

    Purpose: Function uses the threshold on the color sensor to transfer the containers on the
    Q-bot to their proper bin.

    Input: bin_type - Represents which bin the Q-bot needs to go to.

    Output: No ovariables return. This function transfers the containers to their proper
    destination bin.
    '''

    # The following assigns colours based on the bin type
    if (bin_type == 'Bin01'):
        color= 'Red'
    elif (bin_type == 'Bin02'):
        color= 'Green'
    elif (bin_type == 'Bin03'):
        color= 'Blue'
    else:
        color = 'Purple'

    # Depending on the bin type the Q-bot will move forward a certain amount
    # The color sensor measures the threshold values for the color associated with each bin.
    if (color == 'Red'):
        bot.activate_color_sensor(color)
        bot.forward_time(4.3)
        threshold = bot.read_red_color_sensor("Bin01", 0.6)
        bot.deactivate_color_sensor()
    elif (color == 'Green'):
        bot.activate_color_sensor(color)
        bot.forward_time(6.2)
        threshold = bot.read_green_color_sensor("Bin02", 0.6)
        bot.deactivate_color_sensor()
    elif (color == 'Blue'):
        bot.activate_color_sensor(color)
        bot.forward_time(8.1)
        threshold = bot.read_blue_color_sensor("Bin03", 0.6)
        bot.deactivate_color_sensor()
    else:
        bot.activate_color_sensor("Red")
        # The purple box uses  red color sesnor as it has red in it
        bot.forward_time(10)
        threshold = bot.read_red_color_sensor("Bin04", 0.6)
        bot.deactivate_color_sensor()

    #  To be infront the correct bin, the color sensor must read certain threshold
    if (threshold[1]> 4.5 or (threshold[1]<0.5)):
        #deposit_container()
        alternative_deposit_container()
```

```
def deposit_container():
    '''
    This function is currently NOT called as we did the bonus approach.

    Function: deposit_container()

    Purpose: This function moves the Q-bot close to the correct bin and angles
    the Q-bot in a way to safely deposit the containers and then uses the dump function
    to deposit the containers.

    Input: No variable parameters. In order for this function to run, containers
    must be loaded onto the Q-bot

    Output: Q-bot deposits the containers.

    '''
    # Follow line to bin
    bot.rotate(90)
    time.sleep(1)
    bot.forward_time(1)

    # The following makes the qbot adjacent to the bins
    bot.rotate(-92)
    time.sleep(1)
    bot.activate_actuator()

    bot.dump()
    time.sleep(1)
    bot.rotate(-92)
    bot.forward_time(1)
    bot.rotate(90)

    return_home()

def alternative_deposit_container():
    '''
    Function: alternative_deposit_container()

    Purpose: This function moves the Q-bot close to the correct bin and angles
    the Q-bot in a way to safely deposit the containers. It uses the actuator method
    to deposit the containers in the bin.

    Input: No variable parameters. In order for this function to run, containers
    must be loaded onto the Q-bot

    Output: Q-bot deposits the containers.

    Bonus approach.
    '''

    # The list times represents the times from the modeling teams stimulation.
    modelling_file = bot.process_file("p3_Stimulation.txt")
    # Each row is saved into two seperate lists.
    times = modelling_file[0]
    angles = modelling_file[1]

    bot.rotate(90)
    time.sleep(1)
    bot.forward_time(1)
    bot.rotate(-92)
    time.sleep(1)
    bot.activate_actuator()

    # For loop and if statements makes the actutor's movement match the modelling team's stimulation.
    for i in range(len(times)):
        if times[i]==0:
            time.sleep(1)
        elif times[i]== 1:
            bot.rotate_actuator(angles[i])
            time.sleep(1.5)
        elif times[i]== 2.5:
            time.sleep(1)
        elif times[i] == 3.5:
            bot.rotate_actuator(angles[i])
            time.sleep(1.5)
        elif times[i]==5:
            time.sleep(1)

    bot.deactivate_actuator()
    bot.rotate(-92)
    bot.forward_time(1)
    bot.rotate(90)

    return_home()
```

```
def return_home():
    '''
    Function: return_home():

    Purpose: This functions causes the Q-bot to move forwards and follow the trajectory of the
    yellow line on the follow and stop once it reaches the end of the line.

    Input: No input parameter variable.

    Output: No output parameter variable.

    '''
    # Set lost_line as 0 and the while loop will keep adjusting it
    lost_line = 0

    # Q-bot follows the line and the velocity will keep getting adjusted based on the follow_line function
    while lost_line <2:
        lost_line, velocity = bot.follow_line(speed)
        # this speed was best for the Qbot to go straight, then loop around.
        bot.forward_velocity(velocity)

    # Q-bot will move forward until it reaches home position.
    bot.stop()
    bot.forward_time(0.6)
    time.sleep(0.5)
    bot.rotate(188)

def main():
    '''
    Function: main():

    Purpose: This function is where the main processes are taken care of and various functions
    are called. This is where the program loops (depending on user inputs).

    Input: Main function. No input parameters

    Output: Main Function no output parameter. Functions are called based on conditionals

    '''
    # Initializes variables
    container_count =0
    total_weight =0
    bin_type = " "
    weight = 0

    while True:
        container_ID = random.randint(1, 6)
        container_count +=1
        print("Number of containers dispensed : ", container_count)

        bin_type, weight = dispense_container(bin_type, container_ID, weight)
        total_weight += weight
        if (total_weight >= 90):
            print("MAX Weight reached ")

        # When there's one container, its bin id is saved to be used for the conditional transfer.
        if (container_count == 1):
            old_bin_type = bin_type
            load_container(container_count)
        else:
            # The following only allows a container to be loaded if it satisfies the three conditions.
            if (old_bin_type == bin_type and container_count <= 3 and total_weight <=90):
                load_container(container_count)
            else:
                transfer_container(old_bin_type)

                # User input determines is the program will continue
                run_again = input("Please enter 1 to continue the program or anything else to end it: " )
                print("Input is ", run_again)

                if (run_again == '1'):
                    # Variables are reset to allow for proper
                    old_bin_type = bin_type # Makes the bin id match the bottle already on the table
                    container_count = 1
                    total_weight = weight
                    load_container(container_count)
                else:
                    break
        arm.home()
main()
##-------------------------------------------------------------------------------------
## STUDENT CODE ENDS
##-------------------------------------------------------------------------------------
update_thread = repeating_timer(2,update_sim)
```

```
def alternative_deposit_container():
    '''
    Function: alternative_deposit_container()

    Purpose: This function moves the Q-bot close to the correct bin and angles
    the Q-bot in a way to safely deposit the containers. It uses the actuator method
    to deposit the containers in the bin.

    Input: No variable parameters. In order for this function to run, containers
    must be loaded onto the Q-bot

    Output: Q-bot deposits the containers.

    Bonus approach.
    '''

    # The list times represents the times from the modeling teams stimulation.
    modelling_file = bot.process_file("p3_Stimulation.txt")
    # Each row is saved into two seperate lists.
    times = modelling_file[0]
    angles = modelling_file[1]

    bot.rotate(90)
    time.sleep(1)
    bot.forward_time(1)
    bot.rotate(-92)
    time.sleep(1)
    bot.activate_actuator()

    # For loop and if statements makes the actutor's movement match the modelling team's stimulation.
    for i in range(len(times)):
        if times[i]==0:
            time.sleep(1)
        elif times[i]== 1:
            bot.rotate_actuator(angles[i])
            time.sleep(1.5)
        elif times[i]== 2.5:
            time.sleep(1)
        elif times[i] == 3.5:
            bot.rotate_actuator(angles[i])
            time.sleep(1.5)
        elif times[i]==5:
            time.sleep(1)

    bot.deactivate_actuator()
    bot.rotate(-92)
    bot.forward_time(1)
    bot.rotate(90)

    return_home()
```

```
def return_home():
    '''
    Function: return_home():

    Purpose: This functions causes the Q-bot to move forwards and follow the trajectory of the
    yellow line on the follow and stop once it reaches the end of the line.

    Input: No input parameter variable.

    Output: No output parameter variable.

    '''
    # Set lost_line as 0 and the while loop will keep adjusting it
    lost_line = 0

    # Q-bot follows the line and the velocity will keep getting adjusted based on the follow_line function
    while lost_line <2:
        lost_line, velocity = bot.follow_line(speed)
        # this speed was best for the Qbot to go straight, then loop around.
        bot.forward_velocity(velocity)

    # Q-bot will move forward until it reaches home position.
    bot.stop()
    bot.forward_time(0.6)
    time.sleep(0.5)
    bot.rotate(188)

def main():
    '''
    Function: main():

    Purpose: This function is where the main processes are taken care of and various functions
    are called. This is where the program loops (depending on user inputs).

    Input: Main function. No input parameters

    Output: Main Function no output parameter. Functions are called based on conditionals

    '''
    # Initializes variables
    container_count =0
    total_weight =0
    bin_type = " "
    weight = 0
```

```
    while True:
        container_ID = random.randint(1, 6)
        container_count +=1
        print("Number of containers dispensed : ", container_count)

        bin_type, weight = dispense_container(bin_type, container_ID, weight)
        total_weight += weight
        if (total_weight >= 90):
            print("MAX Weight reached ")

        # When there's one container, its bin id is saved to be used for the conditional transfer.
        if (container_count == 1):
            old_bin_type = bin_type
            load_container(container_count)
        else:
            # The following only allows a container to be loaded if it satisfies the three conditions.
            if (old_bin_type == bin_type and container_count <= 3 and total_weight <=90):
                load_container(container_count)
            else:
                transfer_container(old_bin_type)

                # User input determines is the program will continue
                run_again = input("Please enter 1 to continue the program or anything else to end it: " )
                print("Input is ", run_again)

                if (run_again == '1'):
                    # Variables are reset to allow for proper
                    old_bin_type = bin_type # Makes the bin id match the bottle already on the table
                    container_count = 1
                    total_weight = weight
                    load_container(container_count)
                else:
                    break
        arm.home()
main()
##-------------------------------------------------------------------------------------------
## STUDENT CODE ENDS
##-------------------------------------------------------------------------------------------
update_thread = repeating_timer(2,update_sim)
```