# Project Two – Get a Grip Challenge:

## Design a System for Sterilizing Surgical Tools using Remote Sensing and Actuation

*ENGINEER 1P13 – Integrated Cornerstone Design Projects*

Tutorial 02

MON-30

Qisheng Wang (wangq157)

Charlotte Casey (caseyc6)

Gurleen Dhillon (dhillg25)

Martin Ivanov (ivanom4)

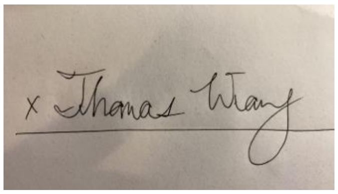
Submitted: December 9, 2020

## *Table of Contents*

## *Academic Integrity Statement*

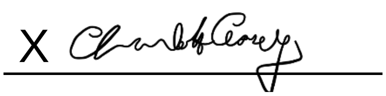
The student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University.

Qisheng Thomas Wang 400301954



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.

Charlotte Casey 400336247



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.

Gurleen Dhillon 400301955

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.

Martin Ivanov 400318848

## *Executive Summary*

In remote and rural regions across Canada, lowered accessibility to healthcare facilities requires increased travel, resulting in an overall negative impact on the health of local communities [1]. This indicates that those living in remote areas find it difficult to access basic healthcare – this demographic being disproportionately composed of Indigenous Canadians [2]. One solution to this issue is the development of technologies that allow healthcare professionals to perform treatments remotely. In this project, our goal is to develop a sterilization system for surgical tools that can operate remotely through muscle sensor data. This system is designed to transfer medical tools in a specialized container to a sterilization autoclave using a robotic arm with a gripper on one end.

The robotic arm, also known as the Q-arm, is programmed to use values produced by a muscle sensor emulator to pick up a container and drop it off at a specific autoclave location. Our solution is simulated using the Q-Labs environment with Python code and operated using constantly changing muscle sensor data generated by a simulation of readings produced by flexing one's arms. Different values from the muscle sensor correlate with different actions performed by the Q-arm, like opening and closing the Q-arm's gripper, picking up and dropping off a container, and opening or closing designated colour bins. Furthermore, the program possesses the ability to randomize the order in which containers are moved to their designated locations, thus ensuring that all possible sizes of containers and types of tools are compatible with our program. Additionally, the program has been designed so that there can be no more than one container per drop off autoclave to prevent overloading the sterilization process. Overall, the designed Python program is able to generate a sequence of events that allows the user to operate the sterilization of surgical tools from a remote location.

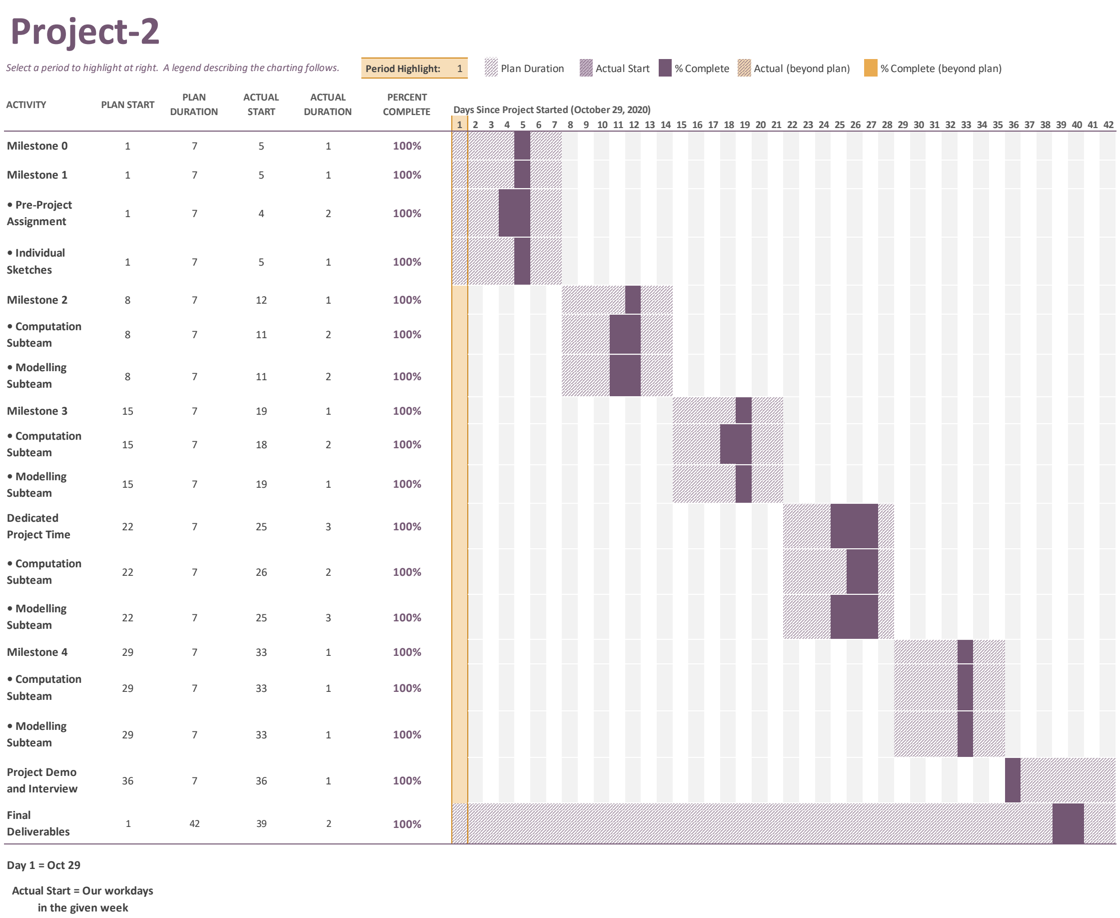
The container itself is designed to hold a medical tool securely while being manipulated by the Q-arm, facilitate the sterilization of a tool within the autoclave, and be as lightweight as possible. Additionally, the container needs to be able to fit within a designated footprint of the enclave. Our group was assigned a scalpel handle for our tool and a trapezoidal footprint. In Milestone 1, different means of picking up the container were found, based on the functions, constraints, and objectives we had determined previously. Over the course of Milestone 2 and 3, designs were modelled and reiterated using low-fidelity prototypes, Inventor 2021, and a Pugh Matrix evaluating the performance of our design versus a pre-existing solution. The final design for the container includes an extended rim under which the Q-arm can grab, two protrusions across the bed housing the tool to hold it securely, and an open lid and holes through the bottom and sides of the container that satisfy the design objectives of allowing the sterilization of the tool and using as little material as possible.

# Project Schedule

## Preliminary Gantt Chart


| ACTIVITY | PLAN START | PLAN DURATION | ACTUAL START | ACTUAL DURATION | PERCENT COMPLETE |
|---|---|---|---|---|---|
| Milestone 0 | 1 | 7 | 5 | 1 | 100% |
| Milestone 1 | 1 | 7 | 5 | 1 | 100% |
| Milestone 2 | 8 | 7 | 12 | | |
| Milestone 3 | 15 | 7 | 19 | | |
| Milestone 4 | 22 | 7 | 26 | | |
| Dedicated Project Time | 29 | 7 | 33 | | |
| Project Demo and Interview | 36 | 7 | 40 | | |
| Final Deliverables | 1 | 42 | 5 | | |

## Final Gantt Chart




| ACTIVITY | PLAN START | PLAN DURATION | ACTUAL START | ACTUAL DURATION | PERCENT COMPLETE |
|---|---|---|---|---|---|
| Milestone 0 | 1 | 7 | 5 | 1 | 100% |
| Milestone 1 | 1 | 7 | 5 | 1 | 100% |
| • Pre-Project Assignment | 1 | 7 | 4 | 2 | 100% |
| • Individual Sketches | 1 | 7 | 5 | 1 | 100% |
| Milestone 2 | 8 | 7 | 12 | 1 | 100% |
| • Computation Subteam | 8 | 7 | 11 | 2 | 100% |
| • Modelling Subteam | 8 | 7 | 11 | 2 | 100% |
| Milestone 3 | 15 | 7 | 19 | 1 | 100% |
| • Computation Subteam | 15 | 7 | 18 | 2 | 100% |
| • Modelling Subteam | 15 | 7 | 19 | 1 | 100% |
| Dedicated Project Time | 22 | 7 | 25 | 3 | 100% |
| • Computation Subteam | 22 | 7 | 26 | 2 | 100% |
| • Modelling Subteam | 22 | 7 | 25 | 3 | 100% |
| Milestone 4 | 29 | 7 | 33 | 1 | 100% |
| • Computation Subteam | 29 | 7 | 33 | 1 | 100% |
| • Modelling Subteam | 29 | 7 | 33 | 1 | 100% |
| Project Demo and Interview | 36 | 7 | 36 | 1 | 100% |
| Final Deliverables | 1 | 42 | 39 | 2 | 100% |

**Logbook of Additional Meetings and Discussions**

| Date | Meeting Purpose |
|---|---|
| Nov. 2 | - Finish sketches for milestone 1<br>- Finish Gantt chart<br>- Fix constraints, objectives, functions table |
| Nov. 9 | - Finish pseudo code<br>- Finish uploading prototype pictures and comparison chart |
| Nov. 16 | - Finish suggestions of design refinements<br>- Finish pseudocode for the remaining tasks |
| Nov. 22 | - Coding team finalized code |
| Nov. 29 | - Discuss code with TA<br>- Discuss final deliverables |
| Dec. 3 | - Meeting prep before interview |
| Dec. 6 | - Working on final deliverables |
| Dec. 7 | - Finish on final deliverables |

# *Scheduled Weekly Meetings*

### Weekly Design Studio Agenda's

| Date | Agenda |
|---|---|
| Nov. 9 | - Attendance<br>- What was accomplished last week<br>- What was completed before this weeks design studio<br>- Was last week's brainstorming useful<br>- What we will be doing this week |
| Nov. 16 | - Attendance<br>- What we were able to finish last week<br>- What do we need to touch up before moving forward<br>- What we got done before design studio (pre-design studio)<br>- What we plan on getting done today |
| Nov. 23 | - Attendance<br>- What we got done last week<br>- What each sub-team accomplished<br>- What we got done before design studio (pre-design studio)<br>- What we plan on getting done today |
| Nov. 30 | - Go with/without warning session<br>- Preparation and finalizing our |

### Weekly Design Studio Meetings Minute's

| Date | Meeting Minutes |
|---|---|
| Nov. 9 | - Everyone was here and doing well<br>- What we did last week<br>    o Assigned roles<br>    o Brainstormed how-to pick-up container<br>        ▪ Forklift, hook, handle, etc.<br>    o Decided all the functions, objectives, and constraints<br>- What we did before this week's design studio<br>    o Making sketches for prototype<br>    o Made flow chart for code<br>- Ideas for prototypes<br>    o We had a few good ideas from last weeks brainstorming to make a prototype<br>- This week we will be making pseudocode and prototype |
| Nov. 16 | - Everyone is here and doing well<br>- What we did last week<br>    o Gurleen and charlotte made prototypes based on sketches<br>    o Martin and Thomas compared pseudocodes<br>        ▪ Found pros and cons in them<br>        ▪ Made general feedback and found issues<br>- Changes from last week (pre-design studio) |

| | |
|---|---|
| | o Prototype<br>▪ Gurleen - Making slot and container features a reasonable size<br>▪ Charlotte - Add in features to add sterilization<br>o Coding<br>▪ Martin - coded location part (found xyz coordinates for one and slightly altered for new locations)<br>▪ Thomas - coded effector part<br>- What we are planning to do today<br>o Gurleen and Charlotte - doing evaluation based on criteria and suggest design refinements<br>o Thomas and Martin - document errors of preliminary python program and make pseudocode for rest of tasks |
| Nov. 23 | - Everyone here and doing well and was here last week<br>- What we did last week<br>o Gurleen and Charlotte used Autodesk to model their design and posted the pictures in the milestone 3 doc and compared to datum using their constraints and found a way to improve their designs<br>o Martin and Thomas identified location of drop off and created a function for the robotic arm to tell it where to move, wrote pseudocode for all the other tasks, and went over each others code<br>▪ Martin did drop off and Thomas did move-end effector<br>- What we are doing before this week<br>o Coding team got together to write the code and ran it multiple times<br>o Modeling team made some adjustments to the container models<br>- What we are doing this week<br>o Improve on what we have<br>▪ Modeling – finishing update for design and get started on g-code<br>▪ Coding – figuring our how to randomize spawn code and getting it to not repeat |

# *Design Studio Worksheets*

## Milestone 0

Project Leads:

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

| Role: | Team Member Name: | MacID |
|---|---|---|
| Manager | Thomas Wang | wangq157 |
| Administrator | Martin Ivanov | ivanom4 |
| Coordinator | Gurleen Dhillon | dhillg25 |
| Subject Matter Expert | Charlotte Casey | caseyc6 |

## Milestone 1

Name: Thomas Wang | MacID: wangq157

*Objectives*
- Should be easy for robot arm to pick up (grippy sides/handles)
- Should have maximized internal storage
- Should be easily sterilizable (no hard-to-reach crevasses where it may be hard to clean)
- Should be able to resist external changes in heat (thermal insulator)
- Should be able to withstand high pressures
- Should be able to identify the difference between different containers

Constraints
- Must have dimensions less than size of autoclave holder (max width of 80mm)
- Must be light weight enough to be picked up by the robot arm (max mass of 350 grams)
- Must have storage for tools

Functions
- Store contents
- Transportable container
- Place container in autoclave bin
- Hold tool securely
- Move equipment in an efficient manner that isn't error prone
- Sort objects based on colour scheme
- Prepare instruments for sterilization

Name: Gurleen Dhillon | MacID: dhillg25

*Objectives (attributes/behaviour design should have)*
- Should be easy for arm to pick up container and place is autoclave
- Should be connected to muscle sensor
- Should work efficiently
- Should not break easily

*Constraints (constraints)*
- *Size of container should be small enough for arm to pick up (max width is about 80 mm)*
- *Each tool/bin should be placed according to colour*
- *Code must be in python*
- *Mass must be less than 350 grams*

*Functions (functionality of design)*
- Arm should be able to pick up tool
- Arm should securely hold surgical tool in place
- Arm should be able to move container to location
- Arm should be able to place container in the correct spot/autoclave bin (on top for small tools, and inside for big tools)
- Should be able to open and close autoclave bin
- Should be able to identify correct bin based on colour and size
- Program should end when all items in place

Name: Charlotte Casey | MacID: caseyc6

*Objectives*
- Should be intuitive to use
- Should be easy to set up
- Should require minimal maintenance
- Should be cost effective

*Constraints*
- Must sanitize tools
- Must pick up and put down tools
- Must deposit tools into autoclave
- Container must be lightweight enough for the arm to pick it up (mass < 350 g)
- Container must be small enough to be manipulated by the robotic arm (width < 80 mm)

*Functions*
- Able to effectively apply sterilizing agent
- Able to securely hold instrument within the container
- Able to move equipment in a controlled manner
- Able to sort containers based on different attributes

Name: Martin Ivanov | MacID: ivanom4

*Objectives*
- Durable and long lasting
- Cost effective
- Size efficient
- Easy to use

*Constraints*
- Able to be picked up and manipulated by robot (mass < 350 grams), (width < 80mm)
- Able to fit in autoclave
- Able to have the instrument fit inside of the container
- Able to facilitate sterilization

*Functions*
- Easily pick up instrument
- Securely hold instrument in place during transfer
- Accurately place instrument into autoclave
- Complete the entire task in a time effective manner
- Prepare instrument for sterilization

| Objectives | Constraints | Functions |
|---|---|---|
| Simple overall design/Easy to set up | Small size of container | Easy to pick up instrument |
| Minimize amount of unused space within the container | Light weight of container | Securely hold instrument within the container |
| Require minimal maintenance | Shape of container fits in autoclave | Accurately sort objects based different attributes |
| Cost effective | Able to facilitate sterilization | Complete task in efficient and timely manner |
| | | Prepare instrument for sterilization |

2. What is the primary function of the entire system?

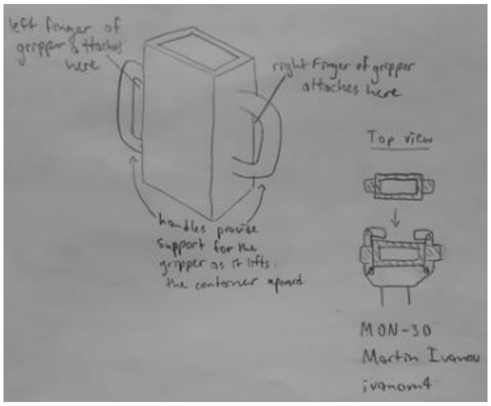
| |
|---|
| Securely picking up and placing the container |

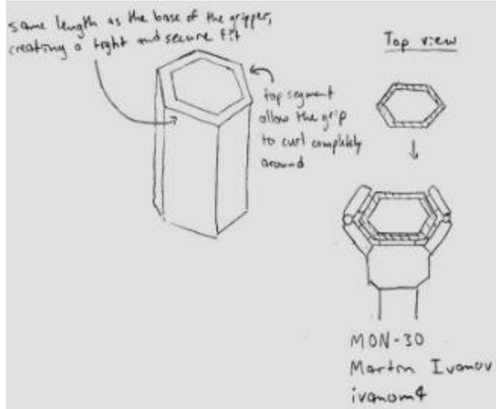
3. What are the secondary functions?

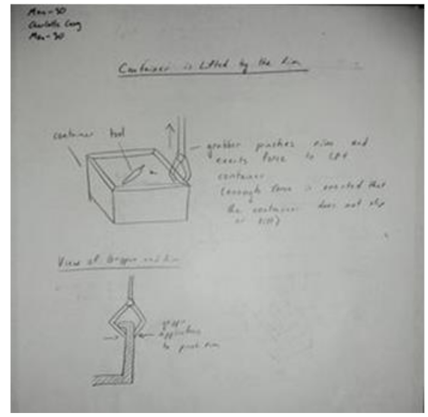
| |
|---|
| Sorting different objects based on different attributes |
| Arm moves the container in a secure and efficient trajectory to the autoclave |
| Instrument is securely placed within the container |
| Open the bigger autoclave bin storage section |

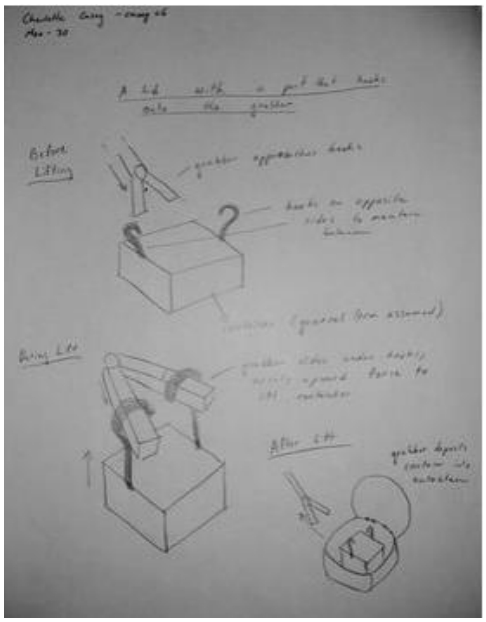
| Function | Means | | | | |
|---|---|---|---|---|---|
| **Pick up** | Pick up container around its exterior | Handle to hold | Picking up container by the rim | Forklift (container placed on different platform) | Hook |
| **Transfer tools** | Conveyer belt | Pulley | Catapult | Chute/ rail system | Slide |
| **Automatically Opening Storage System** | Handle | Motion sensor | Tripwire | Pull string | Lever/ button |

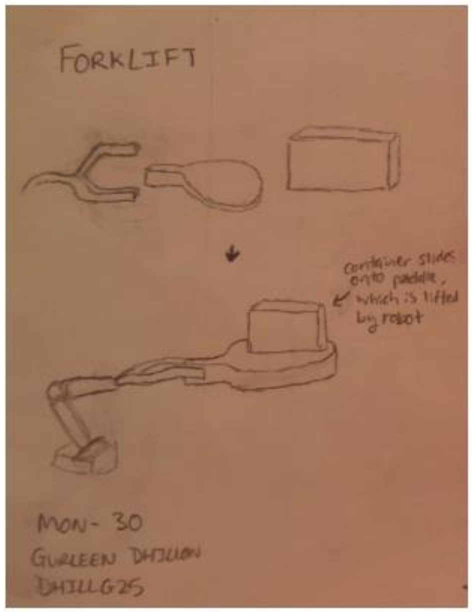
Legend: Thomas, Charlotte, Gurleen, Martin
(Handle: Interior handle vs Exterior handle)
(Pick up around exterior: textured surface vs shape with respect to the arm)
(Pick up by rim: as it sounds)
(Forklift: container slides onto platform which is picked up by the claw)
(Hook: into a hole in the container vs. A lid with a part that can get hooked)

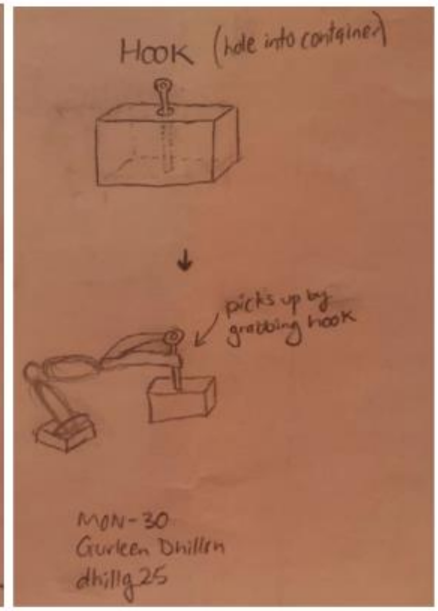

left finger of gripper attaches here
right finger of gripper attaches here
Top view
handles provide support for the gripper as it lifts the container upward
MON-30
Martin Ivanov
ivanom4

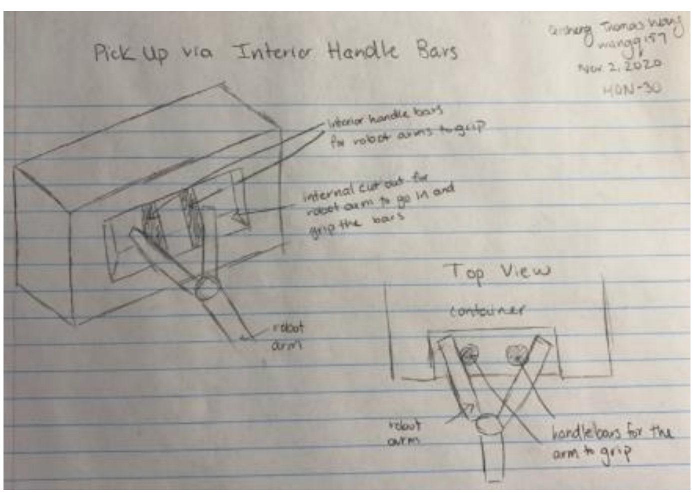

same length as the base of the gripper, creating a tight and secure fit
Top view
top segment allow the grip to curl completely around
MON-30
Martin Ivanov
ivanom4

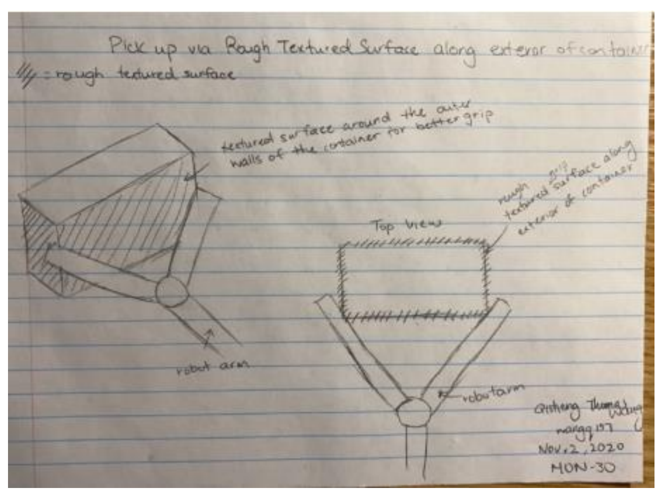




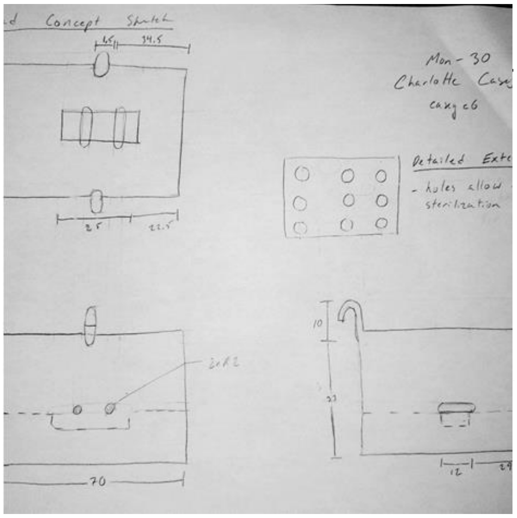

FORKLIFT
container slides onto paddle, which is lifted by robot
MON- 30
GURLEEN DHILLON
DHILLG25

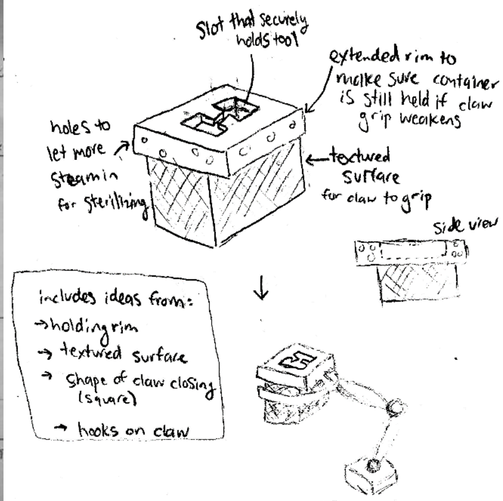

HOOK (hole into container)
picks up by grabbing hook
MON-30
Gurleen Dhillon
dhillg25

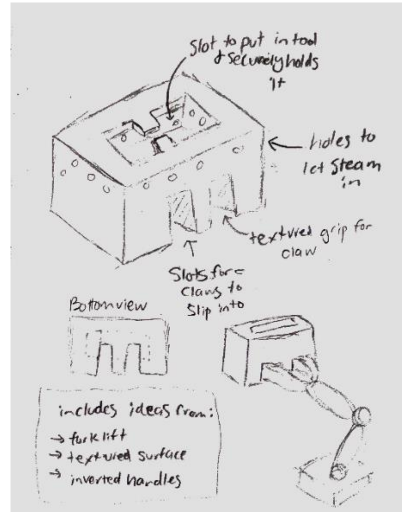

Pick Up via Interior Handle Bars
Nov 2, 2020
MON-30
interior handle bars for robot arms to grip
internal cut out for robot arm to go in and grip the bars
Top View
container
robot arm
handlebars for the arm to grip

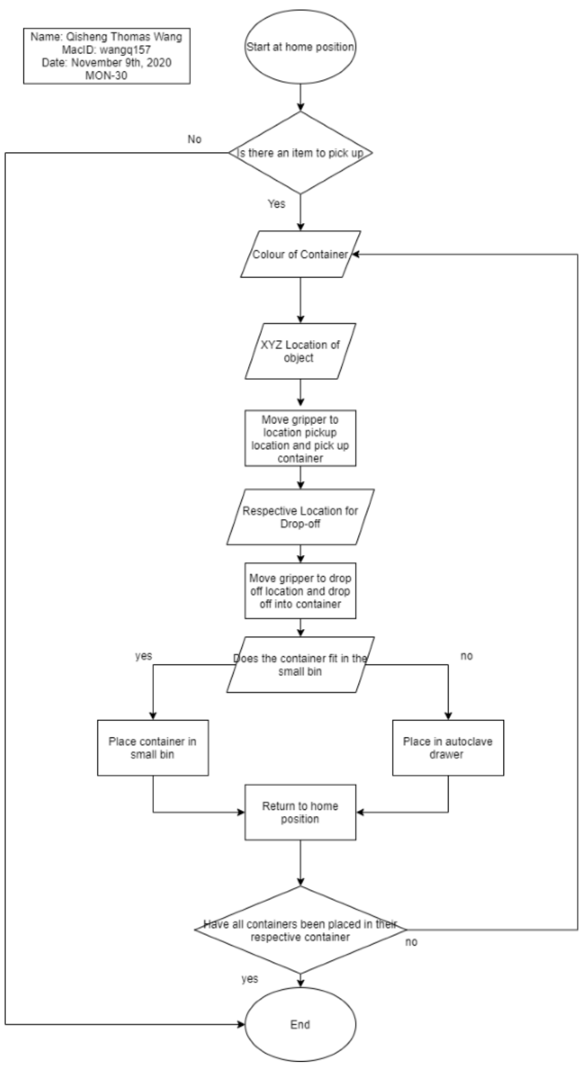

= rough textured surface
textured surface around the outer walls of the container for better grip
Top view
robot arm
Nov. 2, 2020
MON-30

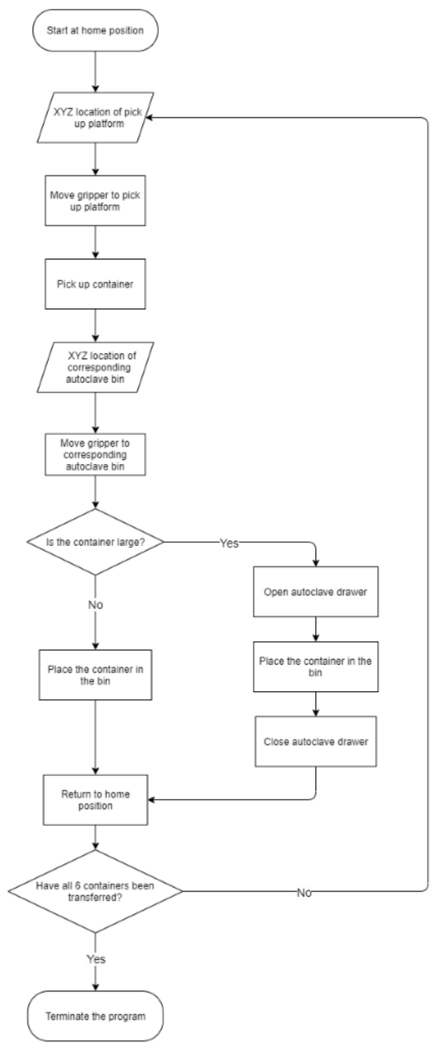
## Milestone 2




**Mon-30**
Martin Ivanov
ivanom4

| Attribute | Prototype1 | Prototype 2 |
|---|---|---|
| Similarities | - Rectangular/ cubic box shape<br>- Slot in box for containing the tools | |
| Differences | - Removable lid<br>- Textured surface (not depicted in model)<br>- Multiple methods of securing the container within the gripper (textured surface, rim) | - Hooks that extend from the top<br>- Exterior is largely featureless |
| Advantages | - Multiple methods of securing the container provide additional security<br>- Can be manipulated by the grabber from several angles | - Secures tool with bars and indent<br>- Can securely hold container (2 hooks), and can be held by 1 hook if the other breaks |
| Disadvantages | - May waste a lot of materials when building | - Hooks may be difficult to latch onto with the grabber<br>- Grabber must operate in a very specific motion<br>- Hooks may take up extra vertical space in the autoclave |
| Alignment with Constraints | Aligns with constraints designated for container | Aligns with constraints designated for the container |
| Reliability in picking up tool | Very reliable | Very reliable |
| Reliability in securing the tool | Secures tool very well | Secures tool well (hinged bars) |
| Extent to which it allows for sterilization | Holes allow sterilization (not depicted in prototype) | Holes allow sterilization |

Prototype 1:

*Prototype 2:*

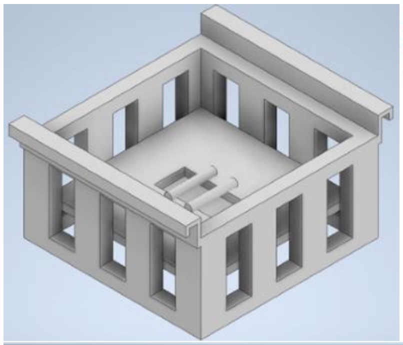
| *Similarities* | *Differences* |
|---|---|
| Both identified that required locations for pick up and drop off are inputs to the code | Checking if the next container should be picked up at the beginning vs the end |
| Start and ended in home position | Whether or not autoclave drawer was needed was quantified as "container being large or small" vs "whether it can fit in the small bin or not (can't fit in small bin means must place in autoclave drawer)" |
| Boolean operator throughout the workflow diagram (whether something is True/False - aka Yes/No) | One is looped so that the cycle repeats until 6 items have been sorted (for loop – for i in range 6) vs while there is item to pick up, run the code (while item is identified for pickup at location XYZ, execute the code) |
| Both achieve all functions/objectives of the program | Some steps are combined such as the movement and the pickup vs having steps all separate |

*Recommendations/General Feedback:*

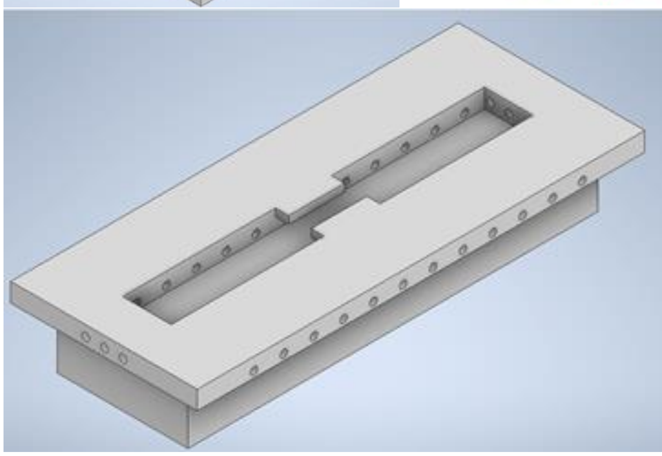
- *Using a loop is better because it avoids issues regarding the code running infinitely (set the range of I values to be the desired number). This also helps mitigate any issue of overflow in the bin or drawer.*
- *Using more objective way of identifying the object's size: does the object fit into the smaller bin Yes or No (whether it can fit into the small bin – thus being able to fit in the small bin indicates that the autoclave drawer isn't needed. If container can't fit in small bin, autoclave drawer must be used)*
- *Should be more explicit mention of the colour of the container*
- *Keep steps of the workflow concise but not too general*

1. Identify if there is a box to pick up
2. Classify the colour of the container
3. Analyze whether it satisfies the condition of fitting in the smaller bin (if it needs to be placed in the autoclave drawer)
4. Pick up the container with the gripping arm
5. Move arm to respective colour drop off location
6. Place in small bin if autoclave isn't needed, otherwise open autoclave drawer and place container inside then close the drawer
7. Return gripping arm back to home location
8. Repeat steps 1-7 until there is no more containers to pick up

**Milestone 3**



```python
#Name: Qisheng Thomas Wang
#MacID: wangq157
def move_end_effector(target_location):
    threshold = 0
    print("Flex muscles to tell arm what to do!")
    if arm.emg_left() > threshold and arm.emg_right() == arm.emg_left():
        arm.move_arm(0,4064,0,0.4826) #home location
    elif arm.emg_left() > threshold and arm.emg_right() == 0
        arm.move_arm(0.5304, 0.0, 0.0257) #pickup location
    elif arm.emg_right() > threshold and arm.emg_left() == 0:
        arm.move_arm(target_location[0],target_location[1],target_location[2])
    else:
        print("Please flex your muscles") #awaiting further instruction
```



```python
## Name: Martin Ivanov
## macID: ivanom4

def identify_location(container_id):
    if container_id == 1:
        target_location = [-0.6181, 0.225, 0.3855]
    elif container_id == 2:
        target_location = [0.0, -0.6578, 0.3855]
    elif container_id == 3:
        target_location = [0.0, 0.6578, 0.3855]
    elif container_id == 4:
        target_location = [-0.4269, 0.1554, 0.172]
    elif container_id == 5:
        target_location = [0.0, -0.4444, 0.156]
    elif container_id == 6:
        target_location = [0.0, 0.4443, 0.156]
    else:
        print("Invalid constainer ID. Target set to home position")
        target_location = [0.4064, 0, 0.4826]
    return target_location
```

| | Datum | Handles (Charlotte) | Textured sides (Gurleen) |
|---|---|---|---|
| Facilitate sterilization effectively | S | S | S |
| Facilitate manipulation by the q- arm gripper | S | S | + |
| Stabilize tool | S | S | S |
| Lightweight | S | S | - |
| Minimize maintenance (no hard to clean corners) | S | - | S |
| Maximize durability | S | S | S |
| Minimization of material usage | S | S | - |
| Total + | 0 | 0 | 1 |
| Total – | 0 | 1 | 2 |
| Total Score | 0 | -1 | -1 |

Handle design: Sterilization could be improved by adding holes to the interior walls of the basin meant to secure the tools, thereby allowing more steam to reach the tool. Additionally, handles could be added to the other sides to allow for the gripper to pick up the container from more angles. Corners can be rounded to minimize maintenance efforts.

Textured sides:

1. fix measurements to fit the 4mm constraint and footprint
2. make the texture block hollow to minimize material usage and allow sterilization from bottom
3. make textures visible and round corners in the cad model

| Identify Autoclave Bin Location Task | Team Member Name: Martin Ivanov |
|---|---|
| • *Add comments to identify colour and size of each container ID*<br>• *Included else for if invalid container ID is entered*<br>• *Identified all drop off bin locations*<br>• *Gave coordinates according to give ID's*<br>• *Potentially add more descriptive comments to tell reader which step is which (although it is already very concise and clear)* | |
| **Move End-Effector Task** | **Team Member Name**: Thomas Wang |
| • *Identify locations more effectively*<br>• *Threshold variable is declared inside the function*<br>• *Gave user instruction to make the program easier to use*<br>• *Potentially add more descriptive comments to make code easier to understand and interpret for readers*<br>• *Passed a variable into the function from external code* | |

**Control Gripper**

1. Input whether the arm needs to pick up or drop off a container.
2. If the arm is picking up a container, tighten the gripper in the positive direction to grab hold of the object.
3. Otherwise, if the arm is dropping off the container then loosen the gripper in the negative direction to release the object.
4. If the gripper is done dropping off an object, return the gripper to initial gripping position (non-tightened)

**Open Autoclave Bin Drawer**

1. Input container ID
2. Identify the object's size and colour
3. Output whether autoclave drawer should open and the colour on screen
4. If container is small, do not open autoclave drawer
5. Otherwise, container is large thus open the autoclave drawer with the corresponding colour
6. After object has been placed, close autoclave drawer

**Continue or Terminate**

1. Input cumulative number of containers that have been picked up
2. If number of repetitions is less than 6, continue
3. If number of repetitions is equal to 6, return to home position and terminate

**Milestone 4**

## MODELLING SUB-TEAM

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

- If all the measurements are less than 4mm it should be great
- Allows good space for sterilization
- Fits the footprint and secures the tool
- Fits the constraints
- Checks all the constraints
- Print time and weight are good

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

- No feedback proposing design refinements

## COMPUTATION SUB-TEAM

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

- We made sure each function is written separately
- Tweak the drop off locations
- Our end effector returns home after every cycle
- We used the arm muscle emulator values properly in order to control whether the function executes or not
- Our program works for both the small and large containers
- Our main function contains all other properly
- Our program is able to loop seamlessly between cycles

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

- Make sure drop off the container exactly in line with the bin itself (for small containers)
- Make sure there is no clipping between the containers and the bins

## *List of Sources*

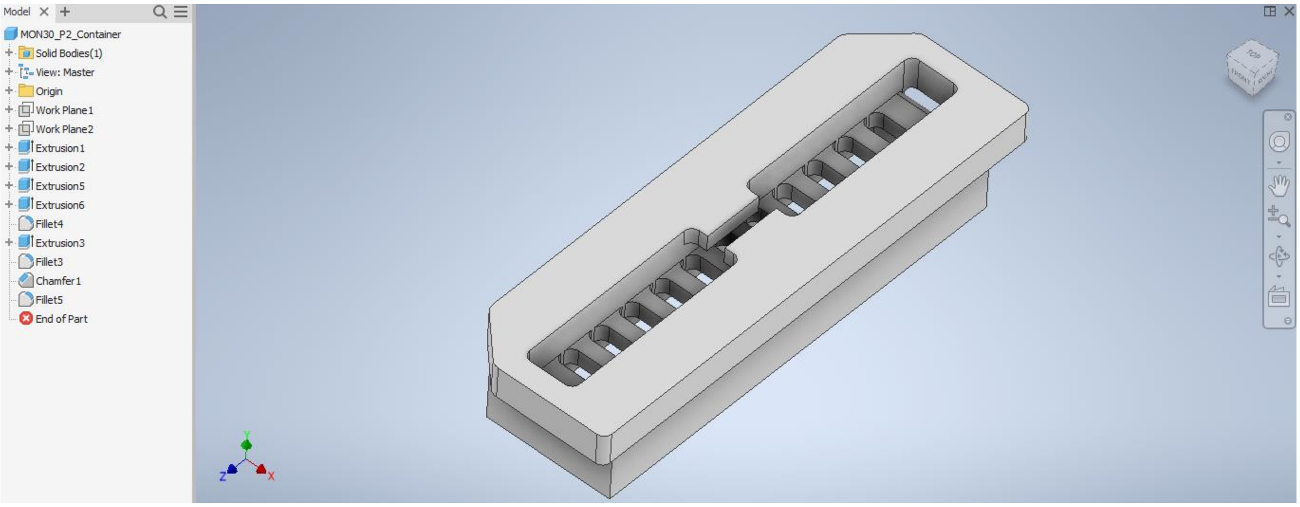
[1] L. Dotto, "Long-distance surgery" [Online]. *The Globe and Mail Canada*; 2004 Oct 2. Available: https://www.theglobeandmail.com/technology/science/long-distance-surgery/article4220774/. [Accessed December 6, 2020].

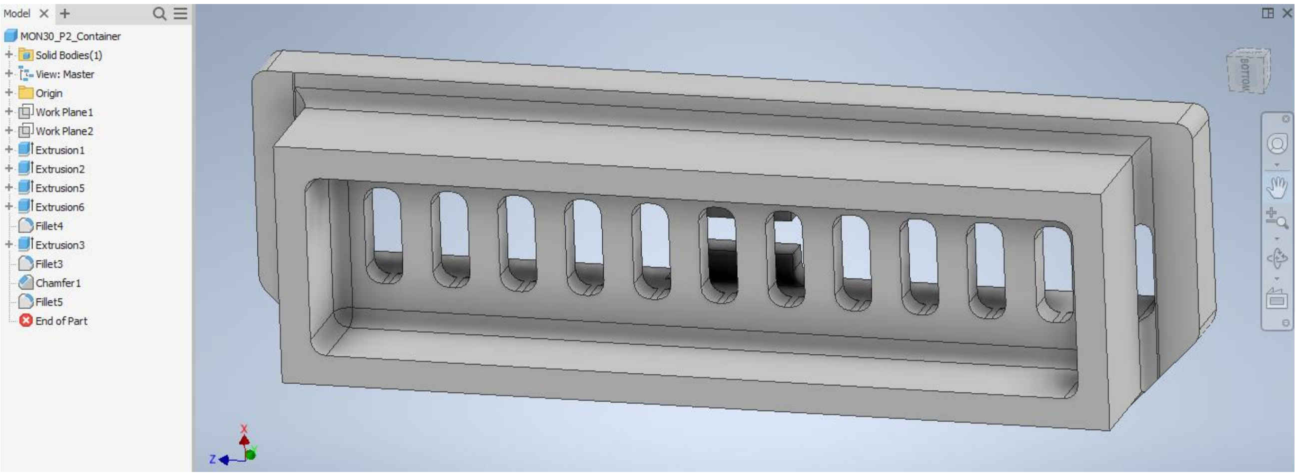
[2] National Collaborating Centre for Indigenous Health, "Social Determinants of Health: Access to Health Services as a Determinant of First Nations, Inuit and Metis Health". [Online]. Available: https://www.nccih.ca/docs/determinants/FS-AccessHealthServicesSDOH-2019-EN.pdf. [Accessed December 6, 2020].

[3] "Project 2: Get a Grip," course material for ENG 1P13A, Department of Engineering, McMaster University, Fall, 2020.

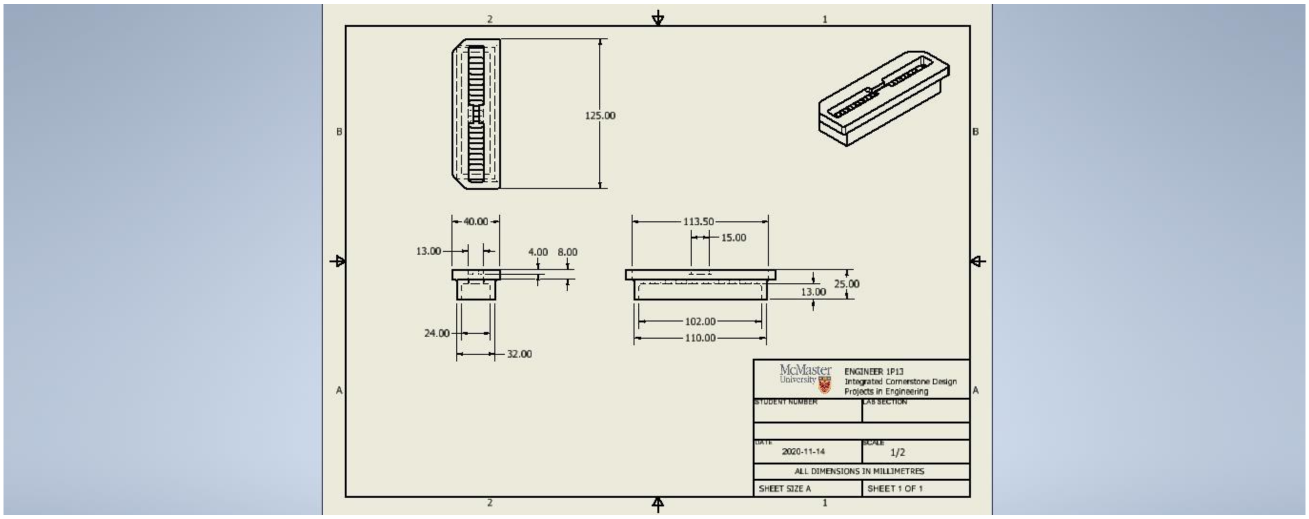
## *Appendices*

### CAD Model




### Engineering Drawing



## Computer Program

```
## ----------------------------------------------------------------------------------------------------------
## TEMPLATE
## Please DO NOT change the naming convention within this template. Some changes may
## lead to your program not functioning as intended.

import sys
sys.path.append('../')

from Common_Libraries.p2_lib import *

import os
from Common_Libraries.repeating_timer_lib import repeating_timer

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

arm = qarm()

update_thread = repeating_timer(2, update_sim)

import time
import random
## STUDENT CODE BEGINS
## ----------------------------------------------------------------------------------------------------------
## Example to rotate the base: arm.rotateBase(90)

# Mon-30
# Martin Ivanov - ivanom4
# Thomas Wang - wangq157
```

```
#Wrote by Martin
def identify_location(container_id): #provides location for each container drop-off
    if container_id == 1: #small red
        target_location = [-0.615,0.245,0.3855]
    elif container_id == 2: #small green
        target_location = [0,-0.6578,0.3855]
    elif container_id == 3: #small blue
        target_location = [0,0.6578,0.3855]
    elif container_id == 4: #big red
        target_location = [-0.44,0.16,0.17]
    elif container_id == 5: #big green
        target_location = [0.0, -0.4444,0.156]
    elif container_id == 6: #big blue
        target_location = [0.0, 0.4444,0.156]
    else: #home location
        target_location = [0.4064,0,0.4826]
    return target_location #return coordinates for target location
```

```python
#Wrote by Thomas
def move_end_effector(target_location, cyclepoint): #moves the robot arm
    threshold = 0
    while True:
        time.sleep(5)
        if arm.emg_left() > threshold and arm.emg_right() == 0: #flex left arm but not right to move arm
            if cyclepoint == "home":
                arm.move_arm(0.4064,0,0.4826) #home location
            elif cyclepoint == "pickup":
                arm.move_arm(0.5304, 0.0, 0.0257) #pickup location
            elif cyclepoint == "dropoff":
                arm.move_arm(target_location[0],target_location[1],target_location[2]) #target location
            break
        else:
            print("Awaiting input from user") #if there is invalid muscle sensor entry, await for correct input
```

```python
#wrote together
def gripper(cyclepoint): #Open and closes gripper
    threshold = 0
    while True:
        time.sleep(5)
        if arm.emg_right() > threshold and arm.emg_left() == 0: #flex right arm but not left to control the gripper
            if cyclepoint == "open_gripper": #open gripper at final cycle point to leave container in dropoff location
                arm.control_gripper(-30)
            elif cyclepoint == "close_gripper": #close gripper at beginning cycle point to pick up container
                arm.control_gripper(30)
            break
        else:
            print("Awaiting input from user") #if there is invalid muscle sensor entry, await for correct input
```

```python
#wrote together
def autoclave(container_id, cyclepoint): #open and closes autoclave drawer
    threshold = 0
    while True:
        time.sleep(5)
        if arm.emg_left() > threshold and arm.emg_right() == arm.emg_left():#flex both arms to equal amounts to open autoclave drawer
            if cyclepoint == "open_autoclave": #open autoclave drawer only if needed (when container ID is 4,5,6)
                if container_id == 4:
                    arm.open_red_autoclave(True)
                elif container_id == 5:
                    arm.open_green_autoclave(True)
                elif container_id == 6:
                    arm.open_blue_autoclave(True)
            elif cyclepoint == "close_autoclaves": #close autoclave drawer otherwise (when container ID is 1,2,3)
                arm.open_red_autoclave(False)
                arm.open_green_autoclave(False)
                arm.open_blue_autoclave(False)
            break
        else:
            print("Awaiting input from user") #if there is invalid muscle sensor entry, await for correct input
```

```python
#wrote together
def rng(): #randomizes the spawned container
    used = [] #create empty list
    x = random.randint(1,6) #randomly generate a number from 1-6 and that is the container ID
    while len(used) <= 5: #continue this process while there are less than 6 indexes in the list
        if not x in used: #if the random generated nubmer is not in the list, add to list
            used.append(x)
        x = random.randint(1,6) #otherwise, generate another number
    return used
```

```python
#wrote together
def main(container_id): #runs through one cycle of the program at a time, randomizing the generated container
    time.sleep(5)
    arm.spawn_cage(container_id) #spawn container
    target_location = identify_location(container_id) #identify where to dropoff
    move_end_effector(target_location, "pickup") #move arm to pickup
    gripper("close_gripper") #close gripper to pick up container
    move_end_effector(target_location, "home") #return to home
    if container_id >= 4: #open autoclave if needed
        autoclave(container_id, "open_autoclave")
    move_end_effector(target_location, "dropoff") #move to dropoff location
    gripper("open_gripper") #close gripper to leave container at dropoff location
    move_end_effector(target_location, "home") #return home
    if container_id >= 4: #close autoclave if it was open
        autoclave(container_id, "close_autoclaves")
```

```python
#wrote together
order = rng() #generate the order for which container spawns
for i in range(6): #loop through all 6 of the containers once
    main(order[i])
```