myCobot Labs

EGR 545: Robotics Systems 1

Professor Sangram Redkar

Made by Sai Srinivas Tatwik Meesala

Introduction

https://docs.elephantrobotics.com/docs/gitbook-en/

The original design of myCobot is to help friends who are interested in a 6-axis series robot to learn it from entry to master, creating unprecedented experience and teaching value.

With myCobot, you can learn that

- Hardware
    - Embedded Microcontroller Based on ESP32
        - Motor and Steering Gear
        - M5Stack Basic/Atom
- Software
        - Arduino
        - C++
        - Python
        - ROS, MoveIt
        - Communication Data

DATA SHEET OF **myCobot 280 M5 2023**

KEY CONCEPTS FOR THIS LAB

Different ways that robot arms can move: Move Linear and Move Joints.

Differentiate between absolute and relative coordinates.

Differentiate between teaching and recording points.

Starting up and connecting with myCobot.

How to utilize a robot arm to move through a group of points by holding the pen with an end effector and writing the word CIM.

Python Setup

Note: For installation, please make sure that your computer is 64-bit or 32-bit. Right-click my computer and select properties.

- Python official download address: https://www.python.org/downloads/
- Click the Downloads option to start downloading Python, click the Add Python 3.10 to PATH.
- and click Install Now to start installing Python.
- After installation of Python, make sure the environment variables are currently configured.

Pymycobot installation: Open a console terminal (shortcut Win + R, input cmd to the terminal) and input the following command:

```
pip install pymycobot --upgrade --user
```

Source code installation. Open a console terminal (use shortcut Win + R, input cmd to the terminal) and input the following command to install:

```
git clone https://github.com/elephantrobotics/pymycobot.git <your-path>
```

```
cd <your-path>/pymycobot-main
```

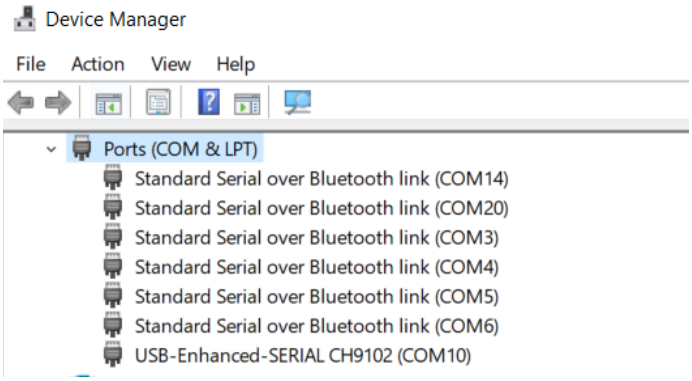
Run one of the following cmd• according to your version of python

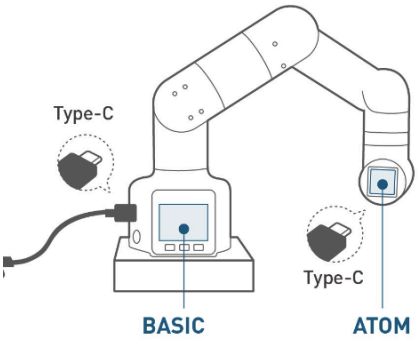
```
python setup.py install
```

Preparations

To find the COM port of the robot:

Open the **Device Manager** and expand the ports and find the COM port number





(Mac: ls /dev/tty.* | grep usb)

(Linux: ls /dev/tty*)

Download:

https://github.com/elephantrobotics/myCobot/releases/download/0/myCobot._windows_x64.zip

And follow the instructions after exactly 1 minute:

https://www.youtube.com/watch?v=VKd8b989M8g

Instead of MainControl, download Transponder on the **Basic** (while USB is connected to the display)

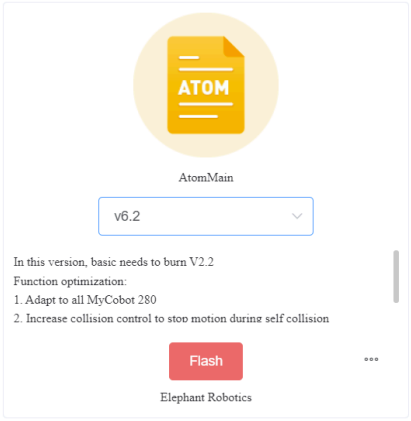
After Transponder onto **Basic** is downloaded.

Use myStudio to flash **Atom**, myStudio address:

https://www.elephantrobotics.com/en/downloads/

Download and flash

- ☑ Atom v6.2



Make sure that Atom is flashed into the top Atom, Transponder is flashed into the base Basic.

Using Python

After the above preparations are completed, the manipulation of the robotic arm can be implemented. First, open the installed PyCharm/VS Cose (if 1 does not have a self-querying Python editor available), create a new Python file, and our library by entering the following code:

```
from pymycobot import MyCobot
```

Go through the following link for all python commands:

https://docs.elephantrobotics.com/docs/gitbook-en/7-ApplicationBasePython/7.8_API.html

You may use for COM port:

```
  Example:
     mycobot-M5:
         linux:
            mc = MyCobot("/dev/ttyUSB0", 115200)
        or mc = MyCobot("/dev/ttyAMA0", 115200)
         windows:
            mc = MyCobot("COM3", 115200)
     mycobot-raspi:
         mc = MyCobot(PI_PORT, PI_BAUD)
```

First set all motors to zero position and now Calibrate it:

```
from pymycobot import MyCobot
import time

mc = MyCobot("COM7",115200)

for i in range(1,7):
    mc.set_servo_calibration(i)
    time.sleep(0.5)

mc.release_all_servos()
```

Try running the following code

```
from pymycobot.mycobot import MyCobot
import pymycobot
from pymycobot import PI_PORT, PI_BAUD
import time
import os
import sys
from pymycobot.mycobot import MyCobot
from pymycobot.genre import Angle, Coord

mc = MyCobot("COMXX", 115200)
mc.send_angles([0, 50, 0, 0, 0, 0], 50)
i = 7
while i > 0:
    mc.set_color(0,0,255) #blue light on
    time.sleep(2)    #wait for 2 seconds
    mc.set_color(255,0,0) #red light on
    time.sleep(2)    #wait for 2 seconds
    mc.set_color(0,255,0) #green light on
    time.sleep(2)    #wait for 2 seconds
    i -= 1

mc.release_all_servos()
```

Procedure

Typical Start-Up Procedure

- Attach the grabber as an end effector to the myCobot
- After following the Python installation guide, open VSCode
- Release all the servos of the Cobot (python)
- After using get_coords(), move the robot around manually and get the coordinates (get_coords()) of the end effector and also get the angle of each joint by get_angles().

Are the axes defined as you expected? Explain:

☑ At the end of each code remember to release all the servos

Prerequisite

Write a new Python code:

First, set all positions of the motors to 0 degrees. (Make adjustments so that the robot doesn't collide with the camera module)

After some delay, open the gripper halfway.

After some delay and releasing the motors, then start an infinite loop to extract and print the angles of the motors of the cobot. (Hint: Add some delay of 0.5 in the loop after printing angles/coordinates)

Write the code below/in Lab notes:

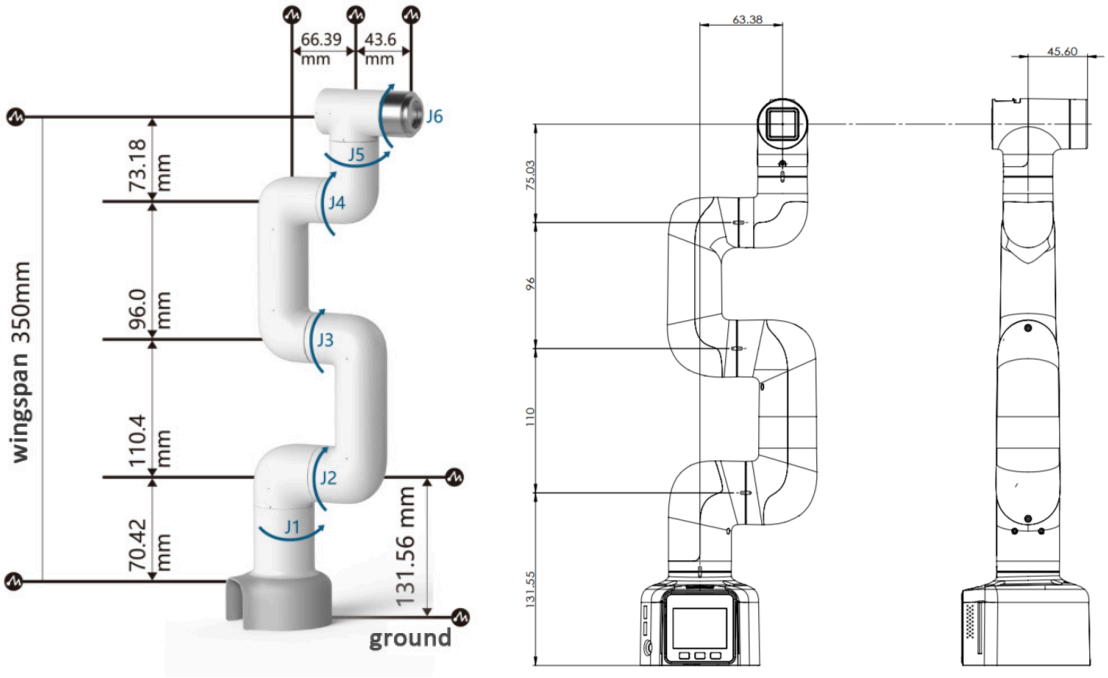
Cheats for Python:

(get_coords, get_angles, set_gripper_value, send_coords, send_angle, release_all_servos)

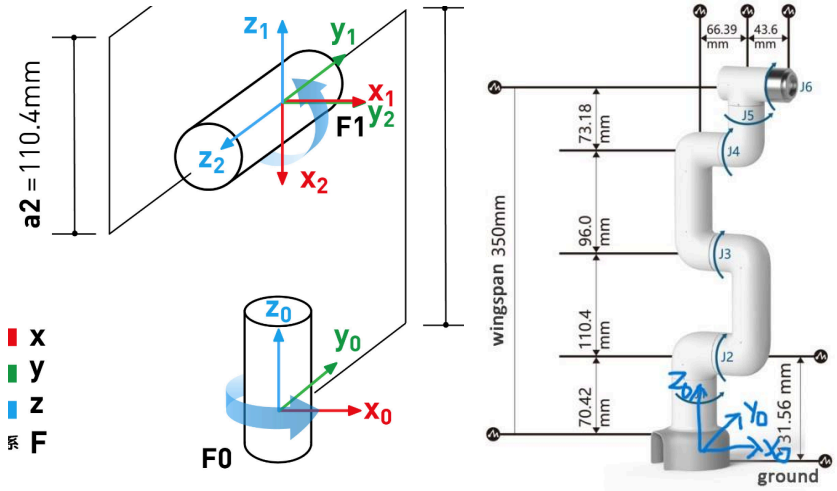
```
print("::get_angles() ==> degrees: {}\n".format(mc.get_angles()))
```

Lab 1

Perform homogenous transformation for the robot. And at last, Measure the length of the robot joints and verify them. (Refer to the datasheet for accurate measurements)



Make sure that your drawn zeroth frame matches the of the robot's coordinates.



(Show your work (along with a kinematic diagram) in the lab manual, you may use **Matlab**)

You may verify your Matlab by imputing angles:

1. Theta(1 to 6) = [0,0,0,0,0,0]

   Coordinates [px, py, pz]: [45.5,-62.7, 413]

2. Theta(1 to 6) = [0,90,0,0,0,0]

   Coordinates: [-282.8, -65.2, 168.9]

3. Theta(1 to 6) = [0,0,90,0,0,0]

   Coordinates: [-170, -64.2, 288.1]

4. Theta(1 to 6) = [0,0,0,90,0,0]

   coordinates: [-77, -64.1, 382..0]

Enter the following joint angles and calculate the position and orientation of the robot end-effect (e.e):

- Joint 1: (call grader for values)
- Joint 2:
- Joint 3:
- Joint 4:
- Joint 5:
- Joint 6:

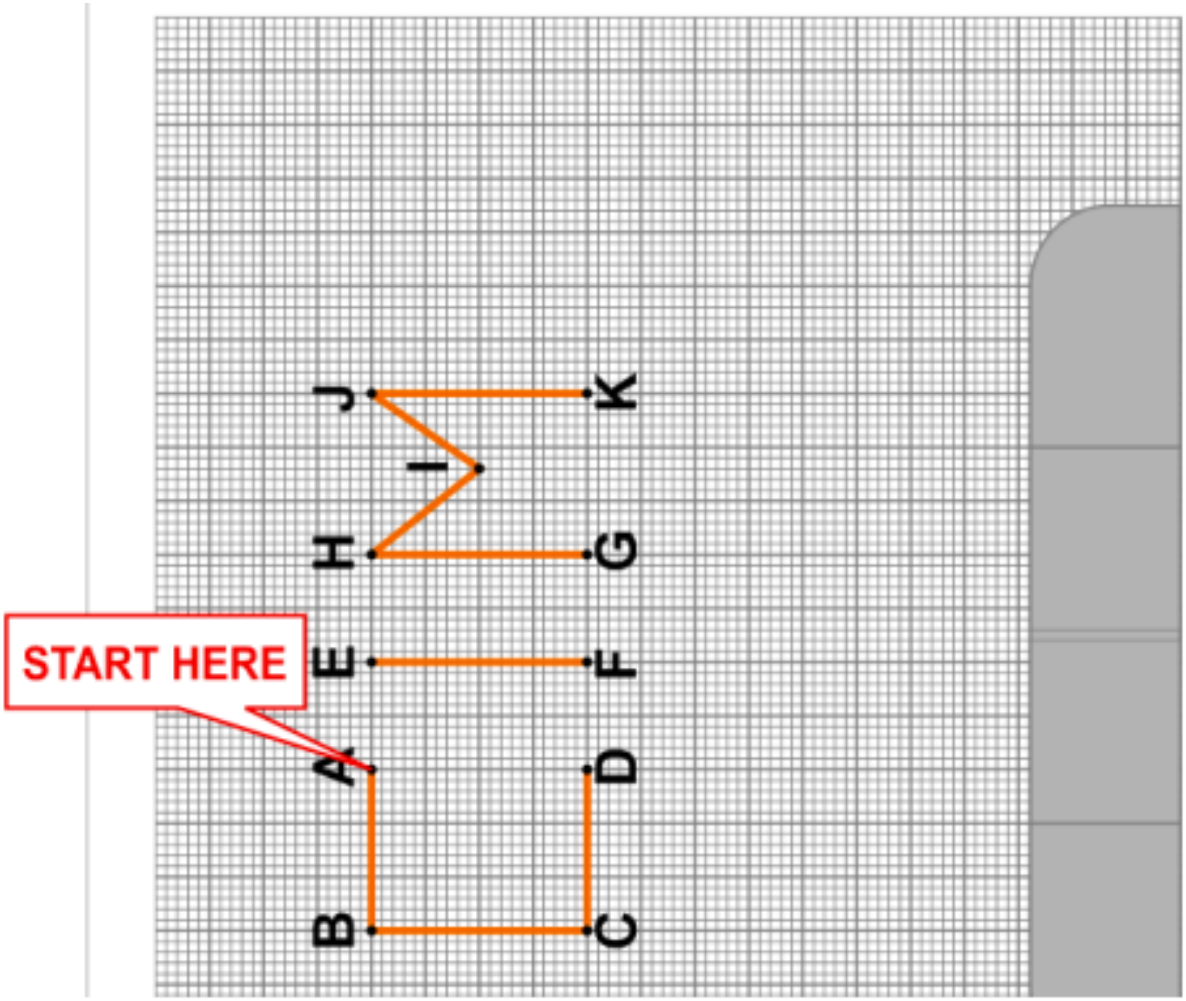
Position and orientation of your calculations:

Now compare your results with the robot.

1. Now program the robot with the joint angles and check if your results are right (by using get_coords or get_angles, set_gripper_value, or send_angle)

Lab 1.2

1. Handwrite the word “CIM” on the template below.



2. Release all the servos, start at point A on your diagram, and move the robot from point to point to re-write the word using straight lines. Be sure to pick up the pen between letters and note down the robot to a position away from the paper when finished.

What do you think is the best way to write the letters CIM (order of movement)? Did you take time into consideration? How?

- Using get_coords that returns: list: A float list of coord - [x, y, z, rx, ry, rz]. Fill out the chart below, then complete the program to write the letters “C I M” Leave the height of the pen at some number that is above the paper, and we can adjust that later. (note you can also use focus_servo to power on the designated servo, use set_gripper_value to make the robot hold the pen)

Points	px	py	pz	rx	ry	rz
A						
B						
C						
D						
E						
F						
G						
H						
I						
J						
K						

Now by using send_coords and delays write down the letters. Be sure to pick up the pen between letters.

After successfully writing the letters, try running at different speeds.

```python
from pymycobot.mycobot import MyCobot
import pymycobot
from pymycobot import PI_PORT, PI_BAUD
import time
import os
import sys
from pymycobot.genre import Angle, Coord


mc = MyCobot("COMXX", 115200)

def cordsGenrator():
    yield [px, py, pz, rx, ry, rz]
    yield []
    yield []

speed = 30

for cords in cordsGenrator():
    # mc.send_coords(cords, speed)
    mc.sync_send_coords(cords, speed, 1, timeout=2)
    time.sleep(2)
```

Conclusion

2. Why are RELATIVE COORDINATES necessary in robotics? Explain.

3. Does the speed of motors influence the ACCURACY of your robot?

4. What would be the effect on the robot's accuracy at higher speeds if the mass of the pen was greatly increased?

5. After completing this activity, how would you define the difference between a robot's accuracy versus its repeatability?

6. How would you calculate point I in this activity using mathematics if it were not given?

Optional Page (otherwise move to Lab 1.3)

Finished early? Try some of the actions below. When finished, show your instructor and have them initial on the line.

1. Teach the robot to write your name.

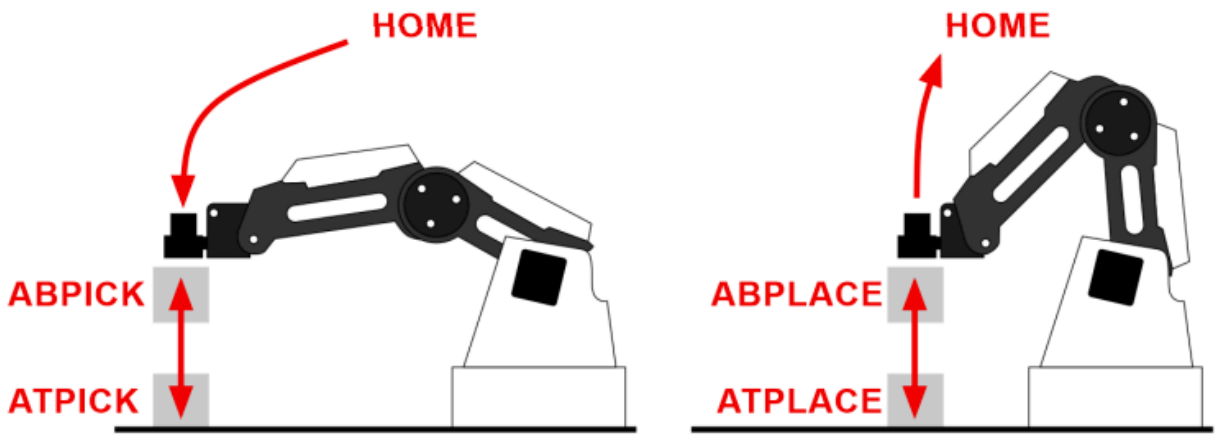
2. Teach the robot to make a barcode with a pen.

Teach the robot to write arcs

Lab 1.3

Robotic arms are excellent for performing pick and place operations such as placing small electronic components on circuit boards, as well as large boxes on pallets. A pick-and-place operation will require at least 5 points

1. Home
2. Above the pick point
3. At the pick-point
4. Above the pick point
5. Above the place point
6. At the drop point
7. Above the place point
8. Home



As a rule, always go to a position above the pick or place point first so that the robot can accurately and repeatedly place the object straight down in a linear motion, with no friction or interference.

KEY VOCABULARY

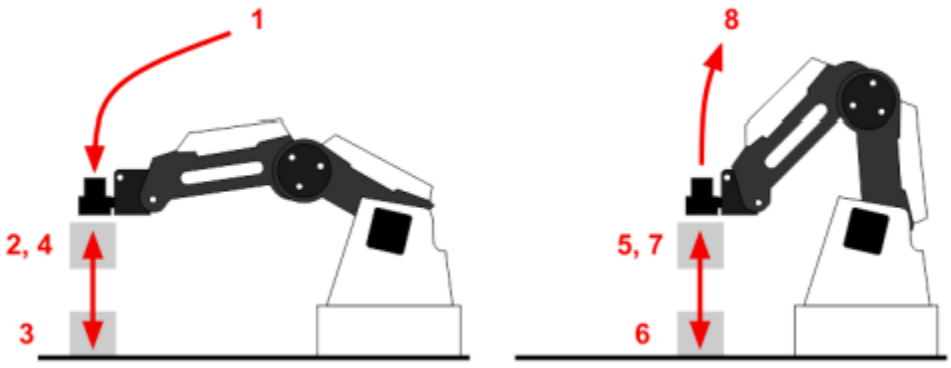
- Relative positions
- Palletizing
- Pick and Place
- Home

- Linear movements
- Joint movements
- End effector
- End of Arm Tooling (EoAT)

PROCEDURE

- After the gripper is attached (you may use a suction pump or claw grabber). Place one of the wooden cups in front of the robot. (You may also use suction, refer to lab 3)
- Now by using get_coords or get_angles, set_gripper_value, send_coords or send_angle try to pick up the block and place it further from the robot.
- Use the following steps as a guide:

1. Home
2. Above Pick
3. At Pick
4. Above Pick
5. Above Place
6. At Place
7. Above Place
8. Home



- Name the positions in the name column of the program

(Be sure to name the positions something relevant so that others will be able to tell what the positions are. Example: A point named AbPick means the point above the place where it is picked up.)

- Be sure to open and close the gripper using set_gripper_value and set_gripper_state.

Run your program and see what happens. Did it work the first time? If not, what did you have to change to make it work?

CONCLUSION

How can you get the suction to turn on in time to pick up the part, or get it to shut off in time to drop it off correctly? Explain below after you have tried it in the program.

What happens if you replace the wooden cube with any other object? Run it and see. Describe below what happens

LAB 1.4

A pick-and-place style of moving objects around is a staple of industrial robots. Another reason to use robots in industry and automation is because of the danger to humans. Robots can work in adverse environments that are dangerous to humans; especially when dealing with chemicals and other toxic substances.

A pick-and-place style of moving objects around is the staple of industrial robots. Another reason to use robots in industry and automation is because of the danger to humans. Robots can work in adverse environments that are dangerous to humans; especially when dealing with chemicals and other toxic substances.

Use the following documentation to understand how to suction pump works:

https://docs.elephantrobotics.com/docs/gitbook-en/2-serialproduct/2.7-accessories/2.7.2%20pump/2.7.2.1-pump.html

After attaching the suction pump to Cobot try to run it using the code in the given link.

Now write a program to perform the following action:

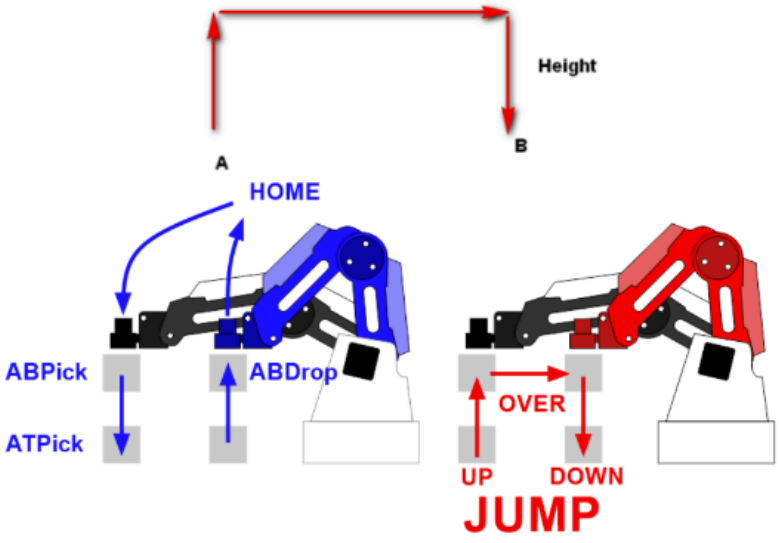
1. Go home
2. Pick up the object from the pallet
3. Place it a position 1 for 2 seconds and drop it
4. Pick up the object and place it in Position 2 for 2 seconds and drop it.
5. Pick up the object place it in another Pallet for 2 seconds and drop it
6. Pick up the object and move the object back to its original position.

Dropping action means placing the object on the ground and turning off the suction and hovering the arm above the position. Be sure to turn on/off the suction cup when necessary. (use delays appropriately)

Recording the positions/coordinates (it is not always very accurate) be sure to make adjustments accordingly. accurate. When you look at all the Z values of all the above pallets and positions 1 and 2, they are probably very different. Raise the robot at the same height when hovering above the pallet and positions 1 and 2.

This is called TEACHING. When fixing points like this it is also called TOUCHING UP points. Go and TOUCH UP all the z values to make the dipping operation happen at the same heights above the positions.

You can use this method to touch up the x and y values of the tanks. Be sure to use homogeneous transformation matrix calculations to send accurate coordinates.



How many lines of code did it take to write the program using what you have learned before?

How many points did you have to teach it?

Lab: Color Recognition

Dual-boot your computer with Ubuntu (20.04 or 22.04).

Before coming to the lab, read and implement until "3 Visualization Software": link

(Quick peek.)

Demonstrate the CoBot's capabilities by following the instructions provided in the documentation, specifically focusing on the "Color Recognition" section. As part of this demonstration, you will be utilizing color recognition (Anything method of your liking). In addition to running the code online, you will have to apply your own implementation using color data. This assignment is expected to meet the standards appropriate for a graduate-level class.

link

Links to an external site.

Upload all files. Upload a report and link to the video. The video must be professional and should be available publically. The report should contain details of what are you demonstrating (Problem statement/Mission), what you learned, your reflection on the lab, and the SERIAL NUMBER OF THE ROBOT that you used during the lab.

The report and video must professionally demonstrate your technical communication skills in IEEE/IJRR format with different title/cover page, page numbers, conclusion, and etc.

http://iaescore.com/gfa/ijra.docx to download the template.

You are expected to thoroughly understand the provided documentation and independently complete the assigned tasks.

Demonstrate the lab in front of the teaching staff, otherwise, you won't get graded for your submission.

For Mac only:

Remove:

```
if platform.system() == "Windows":
        cap_num = 1
        cap = cv2.VideoCapture(cap_num, cv2.CAP_V4L)

        if not cap.isOpened():
            cap.open(1)
    elif platform.system() == "Linux":
        cap_num = 0
        cap = cv2.VideoCapture(cap_num, cv2.CAP_V4L)
        if not cap.isOpened():
            cap.open()
```

And replace with:

```
cap = cv2.VideoCapture(cap_num)
```

Remove “self.plist”

And add:

```
self.mc = MyCobot("port", 115200)
```

Lab: YOLO integration

Use Python for programming.

Upload all *.py files

link

As part of this demonstration, you will be utilizing YOLO (Anything method of your liking). In addition to running the code online, you will have to apply your own implementation using Real-Time Object Detection.

Upload all files. Upload a report and link to the video. The video must be professional and should be available publically. The report should contain details of what are you demonstrating (Problem statement/Mission), what you learned, your reflection on the lab, and the SERIAL NUMBER OF THE ROBOT that you used during the lab.

The report and video must professionally demonstrate your technical communication skills in IEEE/IJRR format with different title/cover page, page numbers, conclusions, and etc.

http://iaescore.com/gfa/ijra.docx to download the template.

You are expected to thoroughly understand the provided documentation and independently complete the assigned tasks.

Demonstrate the lab in front of the teaching staff, otherwise, you won't get graded for your submission.

Lab: ROS integration

Use Python or ROS for programming.

Upload a 2-page report and link to the video.

Upload all *.py files

link

Link

Make sure that Atom is flashed into the top Atom and Transponder or minirobot is flashed into the base Basic.