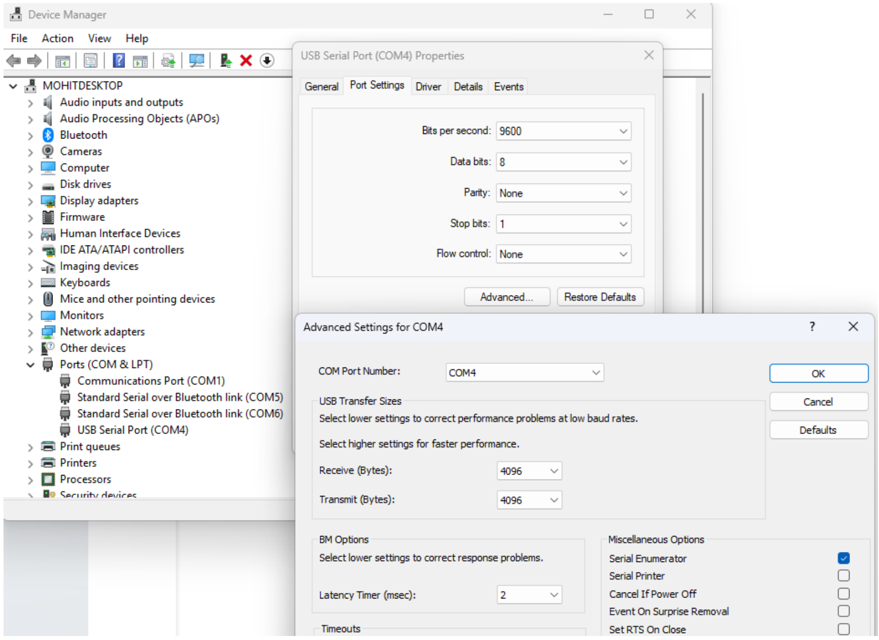
# Lab Procedure for Python

# Differential Kinematics: Visual Servoing

## Setup

1. It is recommended that you review Lab 4 - Application Guide before starting this lab.
2. Hardware Preparation:
    a. Ensure that the QArm Mini is securely attached to the base.
    b. Verify that the manipulator is in the rest position.
    c. Confirm that the QArm Mini is connected to the PC and turn it ON (the light in the switch should be red).
    d. Check and update the latency setting as shown in Figure 1:
        i. Navigate to Device Manager > Ports
        ii. Select the appropriate device - USB Serial Port (COMx) Make a note of the COM port Number.
        iii. Go to Port Settings > Advanced > Latency
        iv. Set the latency to 2 ms



Bits per second: 9600

Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

COM Port Number: COM4

USB Transfer Sizes

Select lower settings to correct performance problems at low baud rates.

Select higher settings for faster performance.

Receive (Bytes): 4096

Transmit (Bytes): 4096

BM Options

Select lower settings to correct response problems.

Latency Timer (msec): 2

Timeouts

Miscellaneous Options

Serial Enumerator ☑

Serial Printer □

Cancel If Power Off □

Event On Surprise Removal □

Set RTS On Close □

Figure 1. Latency Settings

## Visual Servoing

1. Launch **Visual Studio Code** and browse to the lab directory via the **File > Open Folder** menu. Once in the correct directory, open visual_servoing.py.

    a. Update the **id** parameter in line 8 to match the COM port you noted during setup.

2. In this lab, you will apply concepts from the previous Image Processing lab and implement visual servoing to enable the QArm Mini to track and follow an object of interest using live camera feedback. At the top of the script, you'll see a `mode` variable, that controls how the arm behaves:

    a. **Mode 1 – Joint Space Control:** In this mode, the camera detects the object's position in the image, and the base and elbow joints are adjusted accordingly. The offset from the center of the image is treated as an error, which is used to calculate joint velocities. These velocities are then integrated over time to update the joint positions.

    b. **Mode 2 – Differential Kinematics Control:** This mode uses all four joints of the QArm Mini. Instead of directly commanding joint angles, we first determine how the end-effector should move in space (its velocity in 3D). This desired velocity is transformed into joint velocities using the inverse Jacobian matrix.

    Just like in Mode 1, these velocities are integrated over time to compute updated joint positions. The advantage of this method is that the end-effector moves within a consistent frame of reference, its motion relative to the object and the camera remains stable.

3. Find any solid-colored object (e.g., a blue phone, a green marker, a purple notebook etc). You will use it later in this lab.

4. In Section C – Image Processing (lines 40 to 50), the image is processed using the same techniques introduced in the previous lab. If you are using the same object as before, update the `colorThresh` upper and lower bounds accordingly. If you're working with a different object, uncomment line 52 and display the appropriate camera feed to help fine-tune the thresholding values. Confirm that the object is being accurately tracked, as shown in Figure 2.

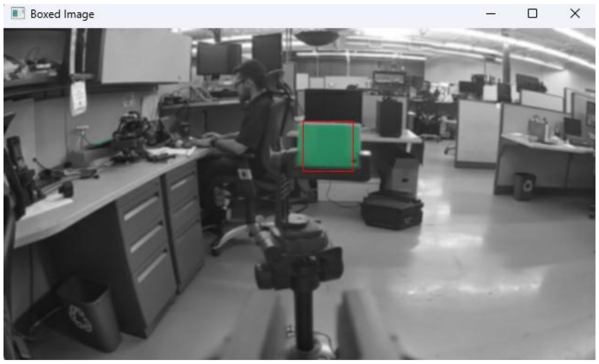


Figure 2. Successfully Tracking a Green Object

5. Ensure that the variable mode on line 31 is set to 1 (i.e., `mode = 1`)

6. Ensure the space around the QArm Mini is clear of any objects.

7. Run the script using the ▷ button in the top-right corner.

8. Slowly move the object in different direction (left, right, up, and down) and observe the manipulator following these movements.

   Note:

   a. If the object is no longer visible in the camera frame or cannot be clearly detected, the manipulator will return to its home position by default

9. 9. Verify that the QArm Mini is able to track the object using only the elbow and base joints. Try slowly moving the object toward the top or bottom of the camera frame—observe how the arm responds. What limitations do you notice in the movement or tracking accuracy?

10. Terminate the script by pressing CTRL-C until you see the message: **Received user terminate command**.

11. On Line 31, update the variable `mode = 2`

12. Rerun the script using the ▷ button in the top-right corner.

13. Observe and compare the manipulator's behavior in All Joints mode versus 2 Joints mode. How does the tracking motion change?

14. Hold the object in a stationary position. How do you expect the manipulator to behave? And is the behaviour as expected?

15. Upon completing the experiment, terminate the script by pressing CTRL-C until you see the message: **Received user terminate command**.

16. Turn off the QArm Mini and, gently bring it back to its resting position.