M. Huraira Anwer - 25100314 Sadaan Tahir - 25100224

EE 3002 L1 (Junior Design Studio - Robotics)

Spring 2025 - LAB 6

Jetson Nano

The NVIDIA Jetson Nano Developer Kit is a small AI computer for makers, learners, and developers. It is a powerful computer that lets you run multiple neural networks in parallel for applications like image classification, object detection, segmentation, and speech processing. All in an easy-to-use platform that runs in as little as 5 watts. The specifications are given below:

GPU	128 Core Maxwell 0.5 TFLOPs (FP16)
CPU	4 x ARM A57 @ 1.43 GHz
Memory	4GB 64-bit DDR4 25.6 GB/s
Storage	microSD
WiFi/ Bluetooth	Requires External Chip
Mechanic al	69.6 mm x 45 mm 260-pin edge connector, no TTP

In this lab, we will work mainly on obstacle avoidance using different sensors and algorithms.

NOT A TASK : Robomaster Side Sensors

There is an IR sensor on the side of the Robomaster. To understand how to read values from the side sensor, use **02_tof_data** in **Robomaster_sdk/examples/14_sensor** folder.

Task 1: Potential Function [25 MARKS]

In this task, you will implement a **Python script** that implements go to goal (on the Robomaster) using obstacle avoidance (in this case, the positions of the obstacles will be known- you will hardcode them). Consider the following pseudocode for **Potential function** logic, you may or may not need to change/tweak this a little. (You don't need Jetson /VM to run this - you may run it on your normal OS where **Robomaster SDK** and **Python 3.6-3.8** is already set up.)

$$\text{current\_location} = [0, 0]$$

$$\text{goal\_location} = [gx, gy]$$

$$\text{current\_angle} = \theta$$

$$dx = \text{distance\_tolerance}$$

$$\text{distance} = \text{euclidean\_distance}(\text{current\_location} - \text{goal\_location})$$

$$d_t = \text{distance\_tolerance\_obstacle}$$

$$\text{obs\_x} = [obs\_1\_x, obs\_2\_x, \ldots]$$

$$\text{obs\_y} = [obs\_1\_y, obs\_2\_y, \ldots]$$

$$R_{\text{inv}} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$$

$$\text{potential\_field}(error) = -K_{\text{gains}} \times error$$

$$\text{while (distance} > \text{distance\_tolerance)}$$

$$\text{error\_goal} = \text{current\_location} - \text{goal\_location}$$

$$x\_obs = 0$$

$$y\_obs = 0$$

$$\text{for each obstacle:}$$

$$d\_obs = \text{euclidean\_distance}(\text{current\_location}, \text{obstacle\_x\_y})$$

$$\text{if } (d\_obs - d_t < 0):$$

$$\text{repulsive\_field} = \frac{1}{d\_obs^{1.5}} \quad (\#\text{you can use any power between } 1 - 2)$$

$$x\_obs = x\_obs + \text{repulsive\_field} \times (\text{current\_x} - \text{obs\_x})$$

$$y\_obs = y\_obs + \text{repulsive\_field} \times (\text{current\_y} - \text{obs\_y})$$

$$\text{end for loop}$$

$$\text{error\_obstacle} = [x\_obs, y\_obs]$$

$$\text{error} = \text{error\_goal} - \text{error\_obstacle}$$

$$u = \text{potential\_field}(error)$$

$$\begin{pmatrix} \text{linear\_velocity} \\ \text{angular\_velocity} \end{pmatrix} = R_{\text{inv}} \times u$$

$$\text{robot(linear\_velocity, angular\_velocity)}$$

Task 2: Potential Function with unknown obstacles (using LIDAR) [25 MARKS]

Jetson Nano Wireless / Remote connection setup:

To establish a wireless connection with jetson nano, follow the following steps. Mostly you just need terminal access, but you can set up for both terminal and GUI (Display setup).

For Both make sure both Jetson Nano and your desktop/laptop are connected to your phone's Hotspot.

For Display:

Setup on Nano

Wifi Connection:

- Connect Jetson to your phone's Hotspot and note Jetson IP address.
- From Your Phone's Hotspot Setting get Jetson IP address , or in jetson nano terminal do "sudo ifconfig" and note ip address under wlan0.

This is one time setup.

- Connect Display to Jetson using HDMI converter.
- sudo apt-get update
- sudo apt-get install x11vnc
- x11vnc -storepasswd (Use **"jetson"** as password)
- sudo nano /etc/systemd/system/x11vnc.service
    - Pase Following in terminal

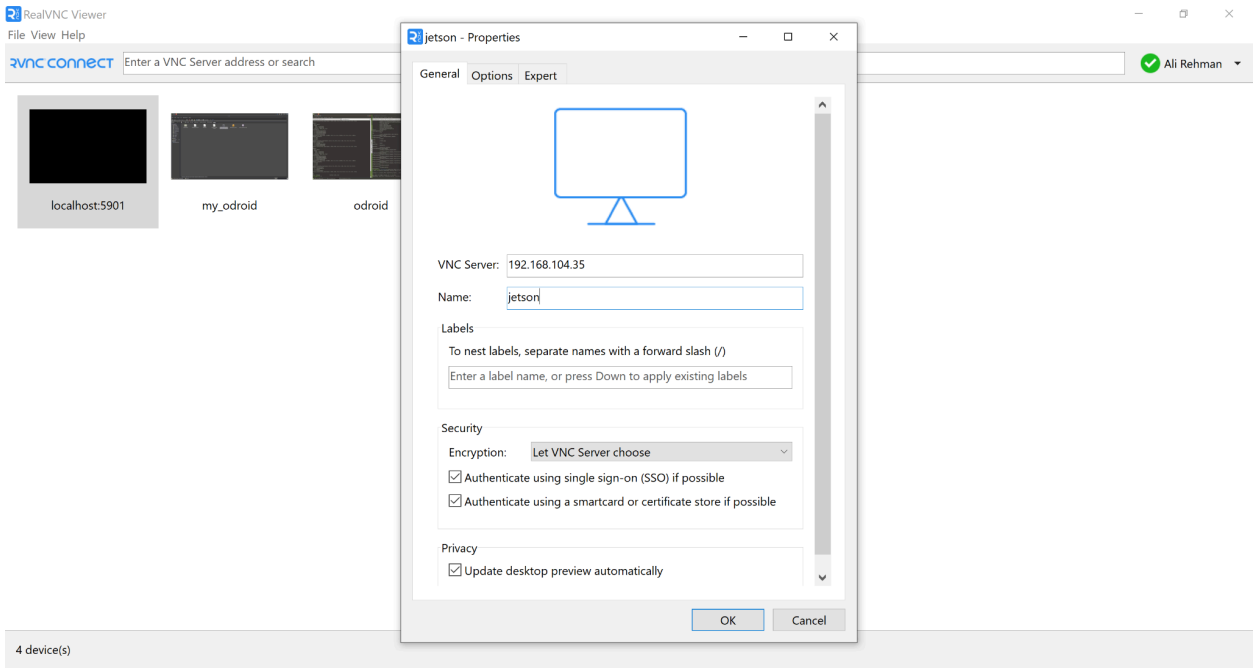
```
[Unit]
Description=Start X11VNC at startup.
After=multi-user.target

[Service]
Type=simple
ExecStart=/usr/bin/x11vnc -auth guess -forever -loop -noxdamage
-repeat -rfbauth /home/jetson/.vnc/passwd -rfbport 5900 -shared

[Install]
WantedBy=multi-user.target
```

- sudo systemctl daemon-reload
- sudo systemctl enable x11vnc.service
- sudo systemctl start x11vnc.service
- Restart device

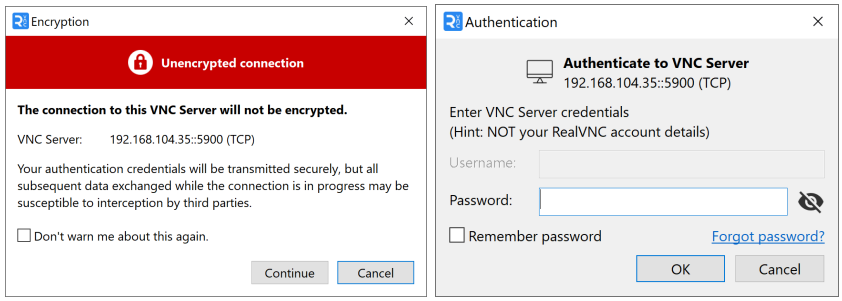
- Use following Command with HDMI Display connected whenever you need to create a new connection.
  x11vnc -usepw -display :0

Setup on Your Laptop;

- Download Real VNC Viewer
  https://www.realvnc.com/en/connect/download/viewer/windows/
- Open Real VNC Viewer
- In Real VNC viewer, create new connection by File -> New Connection.
- Enter Ip address and jetson in name field , click ok to save connection.

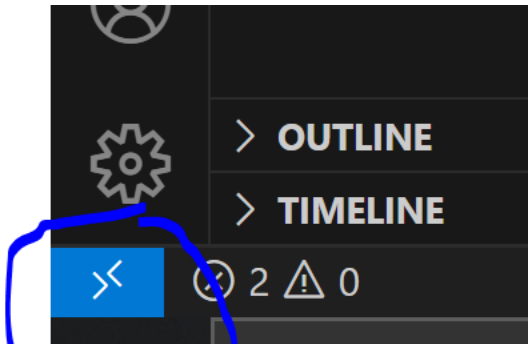


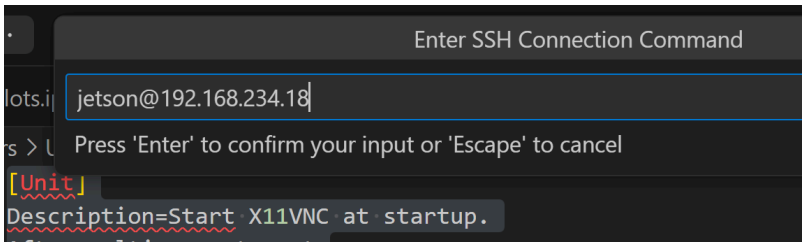
- Then double-click on connection, click continue and enter password. This will create wireless Display Connection with Jetson Nano.



For Just Terminal

- Launch VS Code.
- In Bottom Left Corner Find the and click on Following Icon:

- 

- Then Click on Connect to Host.
- Then Add New SSH Host.
- Enter your **jetson@ip-address** in field (replace ip-address with actual ip address of jetson nano, from your hotspot.
- 

- Save to config file.
- Select Linux if asked for device type.
- Then wait for Setup.
- Now you can open a jetson terminal in VS Code by going to terminal-> New terminal
- You can also edit files of jetson nano in your VS Code and run them.

Task

In this task, you will build on the previous **Python script**, but this time, you will be going towards the goal and using obstacles from the **Lidar 'scan'** topic (not hardcoding them yourself) making use of the **Lidars** and **Jetsons** that will be provided to you.

1) Connecting the Rp Lidar to Jetson
    a) Connect them with the cables given in the Lidar box
    b) Enter ls /dev/tty* in a terminal, it should be printing a lot of values including something like **/dev/ttyUSB0 ,** now give connecting permissions to Lidar with the following command: sudo chmod 666 /dev/ttyUSB0
    c) Create a new workspace in Jetson and clone the following package in src directory:
        git clone https://github.com/Slamtec/rplidar_ros
    d) Do catkin_make and source devel/setup.bash
    e) Launch the Rplidar with this command:
        roslaunch rplidar_ros rplidar_a2m12.launch
    f) Verify it by checking the data from rostopic echo /scan

Whenever you are working remotely, with Jetson connected to Robomaster, always change connection type to ep_robot.initialize(conn_type="rndis").

Submission Requirements:

1. Include a **zip file** containing the following items:
   1.1. **All the python scripts implemented**.
   1.2. **Videos** showcasing working python codes **for Task 2.**
2. Submit a **LAB report** in PDF format (please do not submit word files). This report should provide explanations for all tasks performed along with screenshots, whenever suitable.