

Topic#3 Lab3 實驗結報

姓名：蔡佩蓉 學號：109511286

一、實驗目的

本次實驗旨在利用 ROS（機器人操作系統）與 Android Studio 結合，開發一個 IoT 應用，以控制 Turtlebot3 機器人。具體目標包括：

- 使用 ROS 系統控制 Turtlebot3 機器人在 Oracle VM 中運動（增添左右移動功能）。
- 瞭解和應用 Socket 通信協議。
- 結合上述兩點通過 App 控制機器人的移動。

二、實驗過程 (Code + 說明)

Q1: 修改 Turtlebot 的 keyboard 控制程式，加上左右移動

在程式碼中，透過修改 turtlebot3_teleop_key.py 鍵盤輸入的部分，實現 Turtlebot 的左右移動。在 while 迴圈中，增加對'a'和'd'鍵的判斷，分別增減目標角速度，實現左右移動的效果。

```
elif key == 'a' :  
    # add turn left action  
    target_angular_vel = checkAngularLimitVelocity(target_angular_vel + ANG_VEL_STEP_SIZE)  
    status = status + 1  
    print(vels(target_linear_vel, target_angular_vel))  
  
elif key == 'd' :  
    # add turn right action  
    target_angular_vel = checkAngularLimitVelocity(target_angular_vel - ANG_VEL_STEP_SIZE)  
    status = status + 1  
    print(vels(target_linear_vel, target_angular_vel))
```

Q2: 以 APP 控制 Turtlebot 前後移動

透過 Android Studio 開發 App，建立 Socket 通信連接至 Turtlebot，實現前後移動的控制。在 App 中設計相應的按鈕，並透過 Socket 將指令發送至 Turtlebot，改變目標線速度，達成前後移動效果。

修改 turtlebot3_app.py 中的 HOST (IP) 和 PORT:

```
#socket  
HOST = '192.168.185.42'  
PORT = 8001
```

修改 Button APP 中 (SendData.java) 的 HOST (IP) 和 PORT:

```
Socket socket = new Socket( host: "192.168.185.42", port: 8001);
```

Q3: 在 APP 中加上”左,右”，使 Turtlebot 可以全方位移動
進一步在 App 中新增左右按鈕，透過 Socket 通信將相應指令發送至 Turtlebot，實現全方位移動。修改 Turtlebot 的控制程式以支援左右移動，使 Turtlebot 能夠在所有方向上靈活運動。

MainActivity.java 中增加的部分:

```
Button button_left = (Button)findViewById(R.id.left);  
Button button_right = (Button)findViewById(R.id.right);
```

```
button_left.setOnClickListener(new Button.OnClickListener(){  
    //add turn left action  
    @Override  
  
    public void onClick(View v) {  
        System.out.println("left");  
        new SendData().execute( ...params: "a");  
    }  
  
});  
  
button_right.setOnClickListener(new Button.OnClickListener(){  
    //add turn right action  
    @Override  
  
    public void onClick(View v) {  
        System.out.println("right");  
        new SendData().execute( ...params: "d");  
    }  
  
});
```

activity_main.xml 中增加的 left 和 right buttons:

```
<Button
    android:id="@+id/left"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_marginStart="25dp"
    android:layout_marginLeft="25dp"
    android:layout_marginTop="275dp"
    android:layout_marginEnd="286dp"
    android:layout_marginRight="286dp"
    android:text="LEFT" />

<Button
    android:id="@+id/right"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_marginStart="275dp"
    android:layout_marginLeft="275dp"
    android:layout_marginTop="275dp"
    android:layout_marginEnd="36dp"
    android:layout_marginRight="36dp"
    android:text="RIGHT" />
```

三、問題及解法

問題 1: Socket 通信問題

解法：確保 Android App 和 Turtlebot 之間的 Socket 通信正常運作。需要檢查 IP 地址、端口號等設定。

*注意: ‘.bashrc’ 中的 export ROS_MASTER_URI 和 export ROS_HOSTNAME 的 VM 的 IP 需與 (e.g. Q2) Socket 中的 Host (IP) 一致。

四、心得

透過本次實驗，我們深入了解了 ROS 和 Android Studio 的整合應用，並成功實現了對 Turtlebot 的控制。通過 Socket 通信，我們實現了在 App 中控制 Turtlebot 的前後移動和全方位移動的功能。這次實驗不僅提升了我們對 IoT 應用的理解，也加深了我們對 ROS 和 Android Studio 的應用能力。這次實驗也讓我更加熟悉了程式開發過程中常見的問題和解決方法，提高了我的解決問題能力和程式設計技能。