

1. 實驗目的

學習如何建立 ROS 環境並藉由 socket 來接收傳遞來自 android studio 的 message。

2. 實驗過程 (Code + 說明)

Q1

```
try:
    print(msg)
    while not rospy.is_shutdown():
        key = getKey()
        if key == 'w' :
            target_linear_vel = checkLinearLimitVelocity(target_linear_vel + LIN_VEL_STEP_SIZE)
            status = status + 1
            print(vels(target_linear_vel,target_angular_vel))
        elif key == 'x' :
            target_linear_vel = checkLinearLimitVelocity(target_linear_vel - LIN_VEL_STEP_SIZE)
            status = status + 1
            print(vels(target_linear_vel,target_angular_vel))
        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))
        elif key == ' ' or key == 's' :
            target_linear_vel = 0.0
            control_linear_vel = 0.0
            target_angular_vel = 0.0
            control_angular_vel = 0.0
            print(vels(target_linear_vel, target_angular_vel))
        else:
            if (key == '\x03'):
                break
```

照講義給的範例 code 修改，如上。而其中 roscore 指令是為了 ROS 環境，扮演的角色是 Master。在啟動任何 ROS 程式前都要先啟動，因為要先有 Master 才能管理接下來要啟動的 node。而 roslaunch 指令要搭配 launch file 一起使用，只要把所有要啟動的 node 與指定參數都寫進 launch file 中，當要啟動這些 node 時就只需 roslaunch 這個 launch file 就好了。

Q2 & Q3

```
protected Void doInBackground(String... params){
    try{
        try{
            Socket socket = new Socket( host: "172.20.10.2", port: 5050);
            PrintWriter outToServer= new PrintWriter(
                new OutputStreamWriter(
                    socket.getOutputStream()));
            outToServer.print(params[0]);
            outToServer.flush();
        }catch (IOException e){
            e.printStackTrace();
        }
    }catch (Exception e)
    {
        this.exception = e;
        return null;
    }
    return null;
}
```

修改 VM 的 IP 與 port 和 android studio 的 SendData.java 內的 IP 和 port 一樣，再藉由 socket 來接收來自 android studio 的 message。Q3 還要額外新增兩個 button，這部分將上面有的 code 貼一貼改一改參數即可。

3. 問題與解法

這次實驗算是滿簡單的，只要照講義做就不會有什麼大問題，我自己基本上除了筆電很卡很盪以外沒遇到其他問題。

4. 心得

ROS 的核心是 Master。會有個 list 紀錄有哪些 node，並透過這個 list 讓各個 node 間互相溝通。而 Node 是一個執行中的程式，每個 node 間可以透過 topic 或 service 等方式來溝通，node 也可以組成不同的 package。Topic 是 node 之間溝通的重要機制，一個 topic 可以有多個 node 來拿資料。Message 則是 topic 不可或缺的指示，是 node 之間透過 topic 傳遞資料時的資料格式。

我自己覺得這次實驗不難，照講義做，把程式碼塗塗改改就能完成了。但是有點花時間，因為檔案偏大，下載就先花了不少時間了，再來是不論跑 android studio 的 device 或是 VM 裡的模擬器都會讓我的筆電超級盪，這點就只能慢慢等了。