

第四章 多點導航教學

本章將說明 KUBOT 的多點導航實現，可以先熟悉第三張的建圖、導航兩大功能。

在進入本章節前，請先熟悉 **gmapping** 與 **navigation** 兩大功能。

事前準備：

1. 充好電的 KUBOT 小車一台
2. Linux 環境電腦一台

5-1 多點導航

Step.1 遠端 KUBOT 小車請參考第三章，進入 SSH。

Step.2 SSH 的終端機啟動導航+IMU 姿態修正+CAM：

```
kubot_one
```

並可以在遠端電腦的終端機啟動 RVIZ(需使用遠端電腦的終端機)

```
kubot_view
```

並可以看到視訊鏡頭(需使用遠端電腦的終端機)

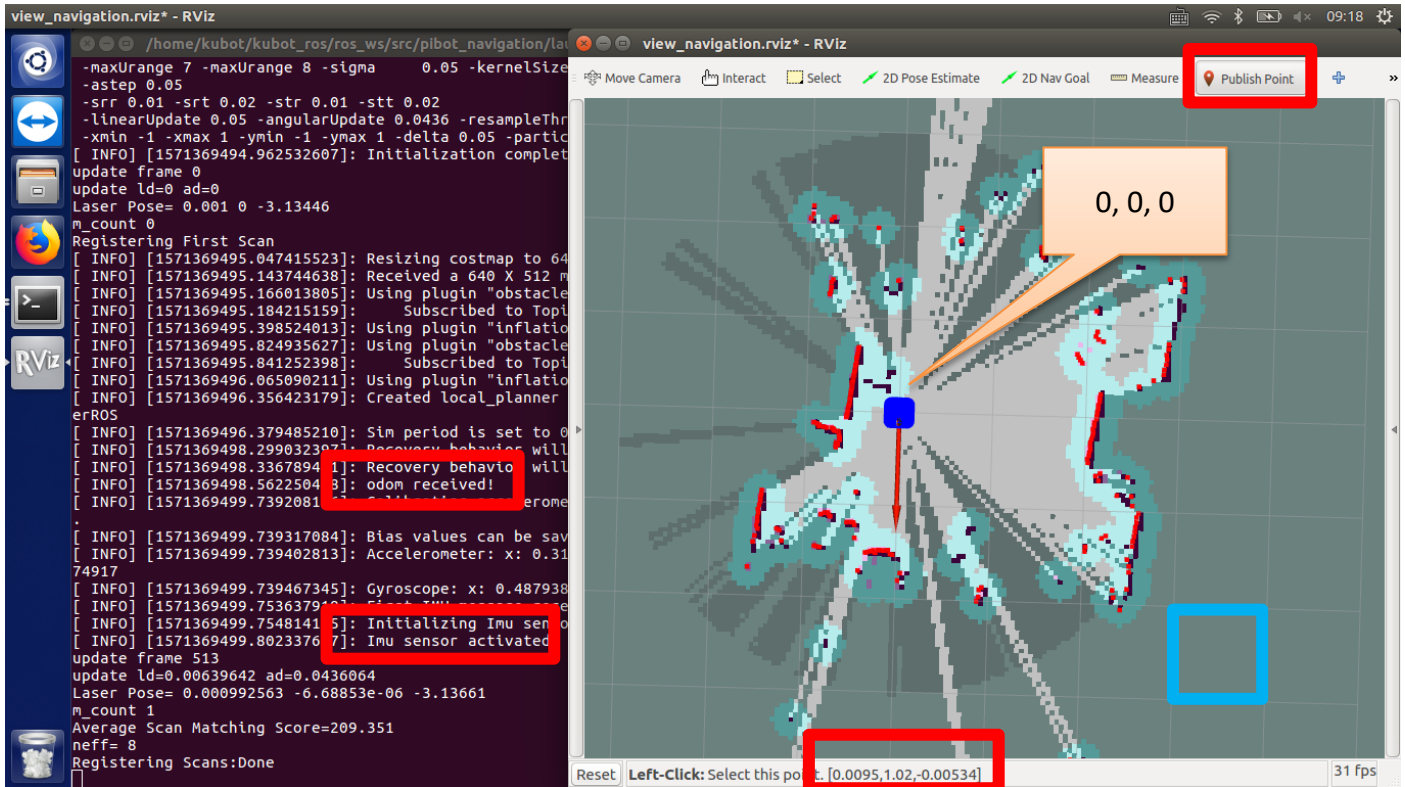
```
roslaunch image_view image_view image :=:/usb_cam/image_raw
```

此時也可以透過平板使用 map_nav APP 觀看



KUBOT

用戶手冊



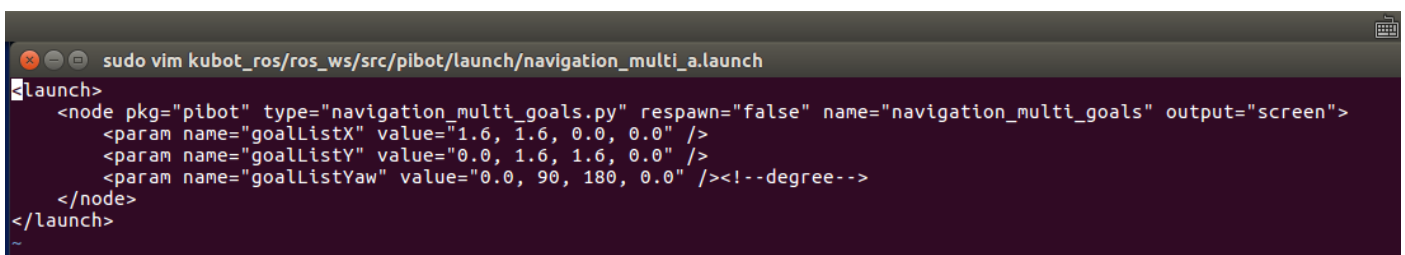
可以看到這支程式若順利執行，會出現 odom received 以及 Imu sensor activated，代表 KUBOT 底層控制啟動正常、並啟動 IMU 融合里程計 ekf 的姿態校正，若執行的過程中均沒有紅色錯誤，則代表鏡頭也成功啟動。在平板上也可以看到完整的畫面。

建圖的起點就是世界座標原點，所以如果有保存地圖後再次開啟，需要將 KUBOT 放回建圖時的位置！在 RViz 上可以透過 publish point，並用滑鼠移動，於下方狀態列看到 X、Y 座標。建圖時箭頭指的方向是 YAW 0°。

如藍色方框，其單位為 1mX1m，因此可以知道座標與實際尺寸單位為公尺。
將期望導航的座標點記錄下來，座標值(X, Y, Yaw)。接著進入設定座標：

kubot_set_a

可以看到下面畫面：



可以看到可以輸入座標的地方。該範例是行走至 A(1.6, 0.0, 0.0)、B(1.6, 1.6, 90).....以此類推

可以自行更換座標點。筆者建議第一點設為(0, 0, 0)

修改完成後輸入:wq 儲存並退出

接著並可以實現多點導航：

kubot_nav_a