

通訊網路實驗

IoT應用

ROS

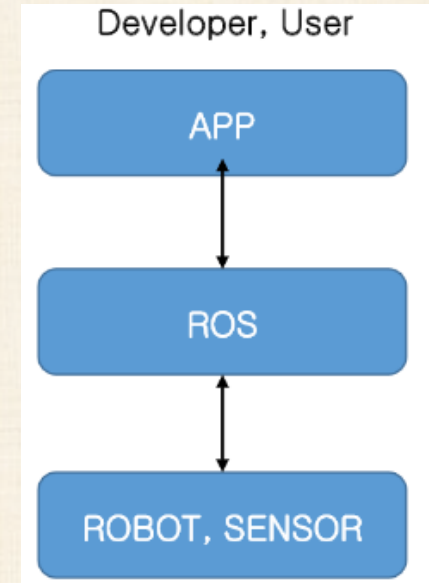
Android Studio

Dept. of Electrical and Computer Engineering (ECE)

National Yang Ming Chiao Tung University

課程大綱

- 1.認識、使用Socket
- 2.ROS系統、Turtlebot3
- 下載 turtlebot2023 、 Button檔案
 - [Google drive](#)
 - [OneDrive](#)



Demo項目

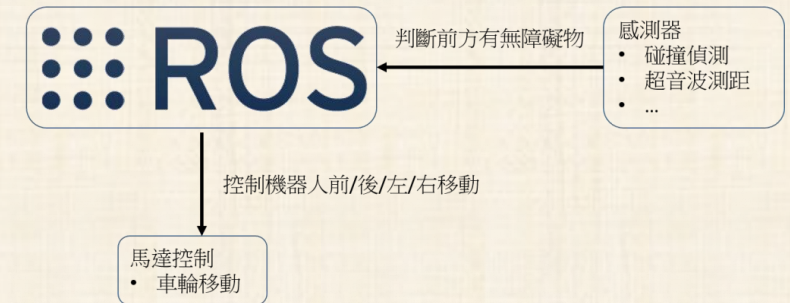
- Q1 :修改Turtlebot 的keyboard控制程式，加上左右移動
- Q2 :以APP控制Turtlebot前後移動
- Q3 : 在APP中加上”左,右”，使Turtlebot可以全方位移動
- 結報附程式中有修改部分即可

主要的機器人OS



ROS簡介(1)

- Robot Operating System，用傳統的作業系統(eg. Ubuntu, Windows)處理系統管理、人機介面等，提供多種功能，包含：控制devices、在process間傳遞訊息、管理封包等，都是機器人應用程式的基本功能
- Open-source，可支援不同devices間的溝通
- Goal:建立自己的生態系，讓大家都可以參與機器人軟體的開發
- 簡單來說，ROS結合Sensor、APP、Robot



ROS 簡介(2)－基本架構

□ Node

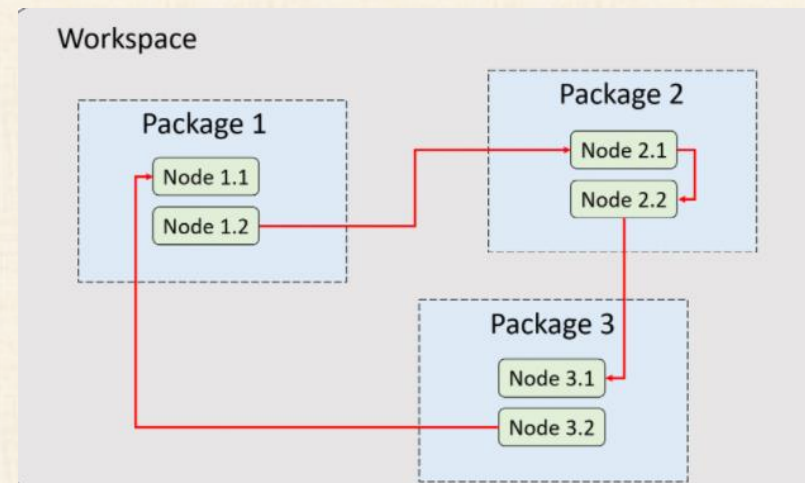
Node是最小的執行處理器，可視為一個program。

在ROS中，是由許多nodes構成的，每個Node都可以跟別的Node透過Topic、Service或其他方式來溝通、獲取資料。

Node可以組合成不同的package(功能包)

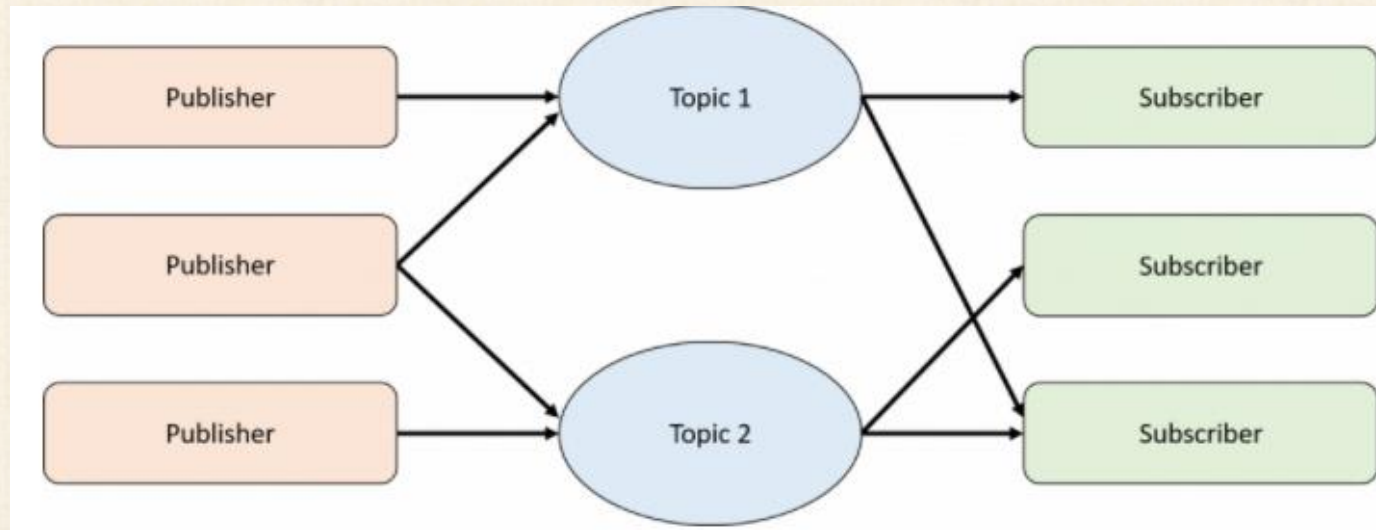
□ Master

負責Node與Node之間的溝通



ROS 常用名詞(1)

□ Publisher & Subscriber



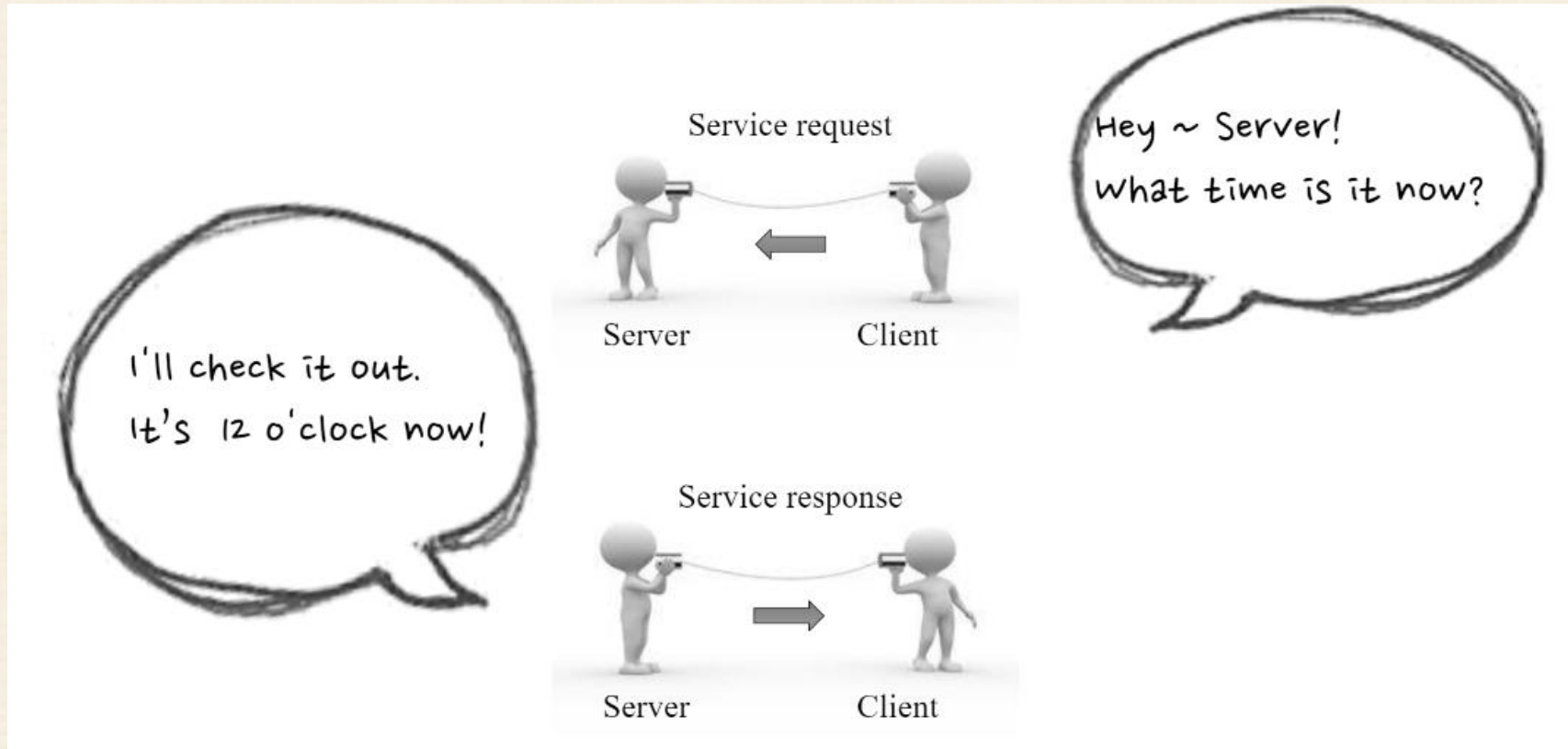
□ Topic : Publisher & Subscriber之間的橋樑

- Publisher 和 Subscriber可以是一對一、多對一、一對多傳輸messages

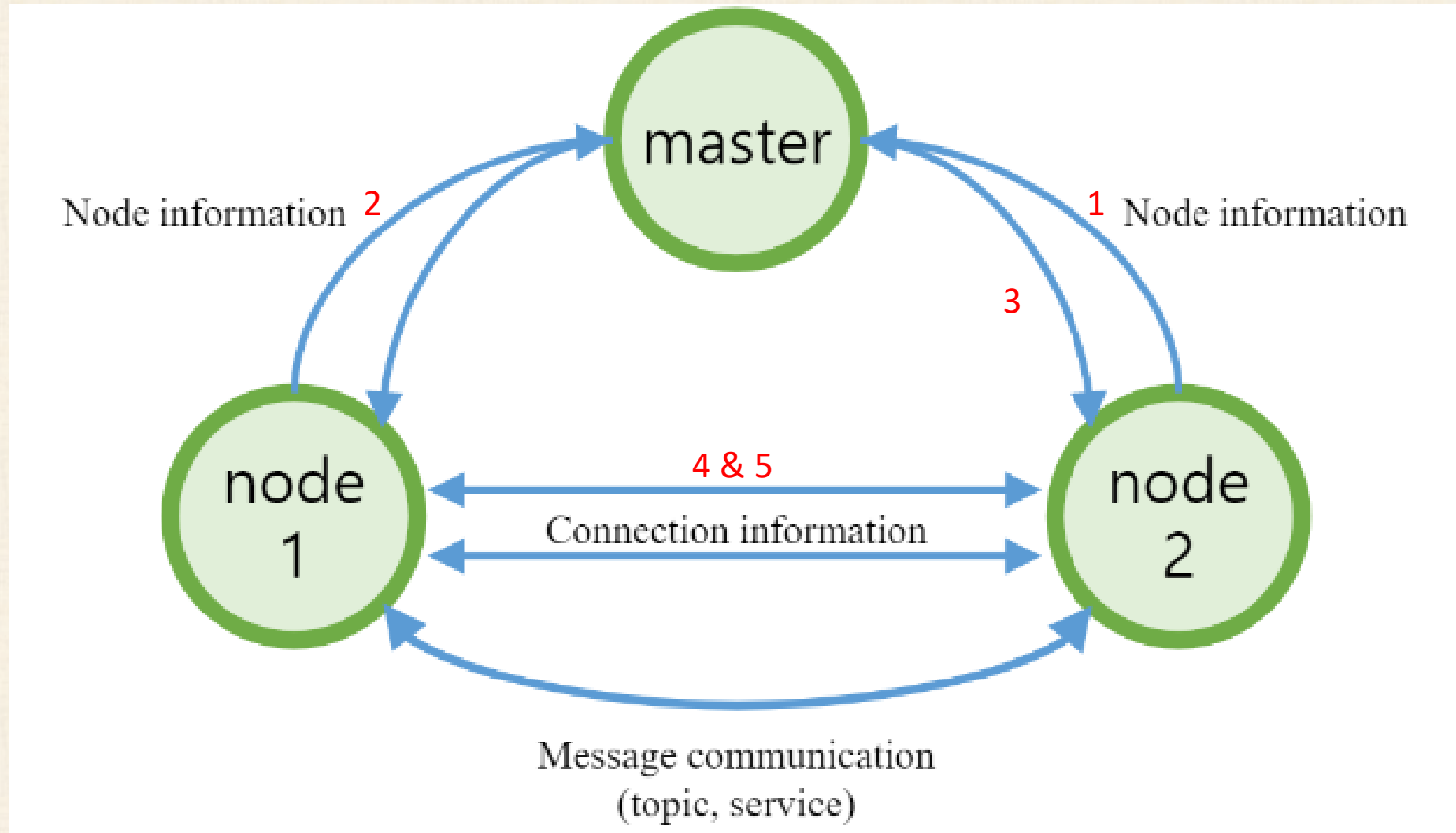
□ Message : Node之間透過topic傳遞資料時的資料格式

ROS 常用名詞(2)

- Service request/response
- Service server & Service client



Message Communication among Nodes



Turtlebot 3

□ TurtleBot3 Providers



WORLD'S MOST POPULAR ROS PLATFORM

TurtleBot is the world's most popular open source robot for education and research.



AFFORDABLE COST

TurtleBot is the most affordable platform for educations and prototype research & developments.



SMALL SIZE

Imagine the TurtleBot in your backpack and bring it anywhere.



EXTENSIBILITY

Extend ideas beyond imagination with various SBC, sensor, motor and flexible structure.



MODULAR ACTUATOR

Easy to assemble, maintain, replace and reconfigure.



OPEN SOURCE SOFTWARE

Variety of open source software for the user. You can modify downloaded source code and share it with your friends.



OPEN SOURCE HARDWARE

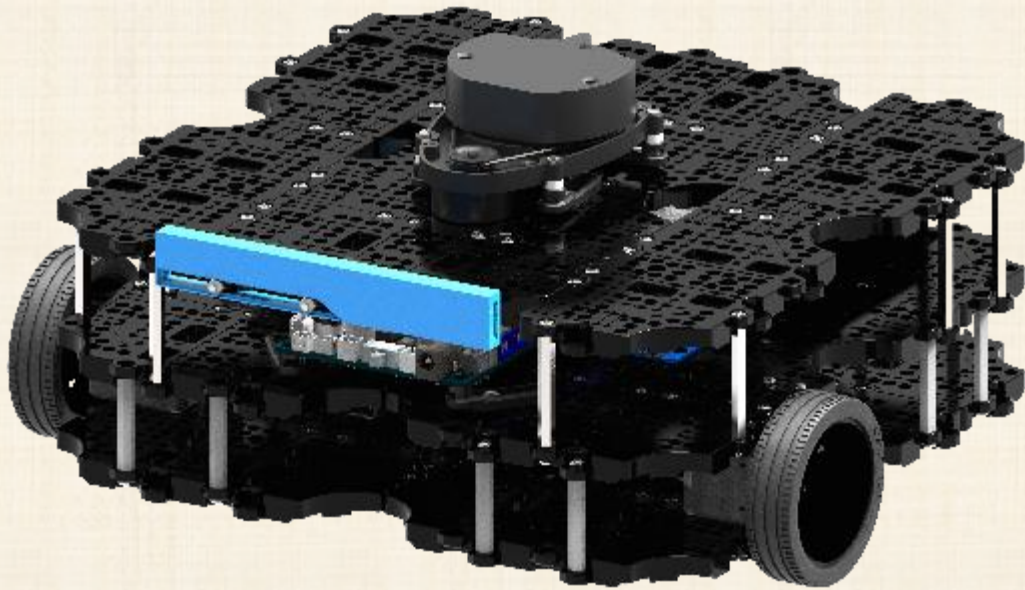
Schematics, PCB Gerber, BOM and 3D CAD data are fully opened to the user.



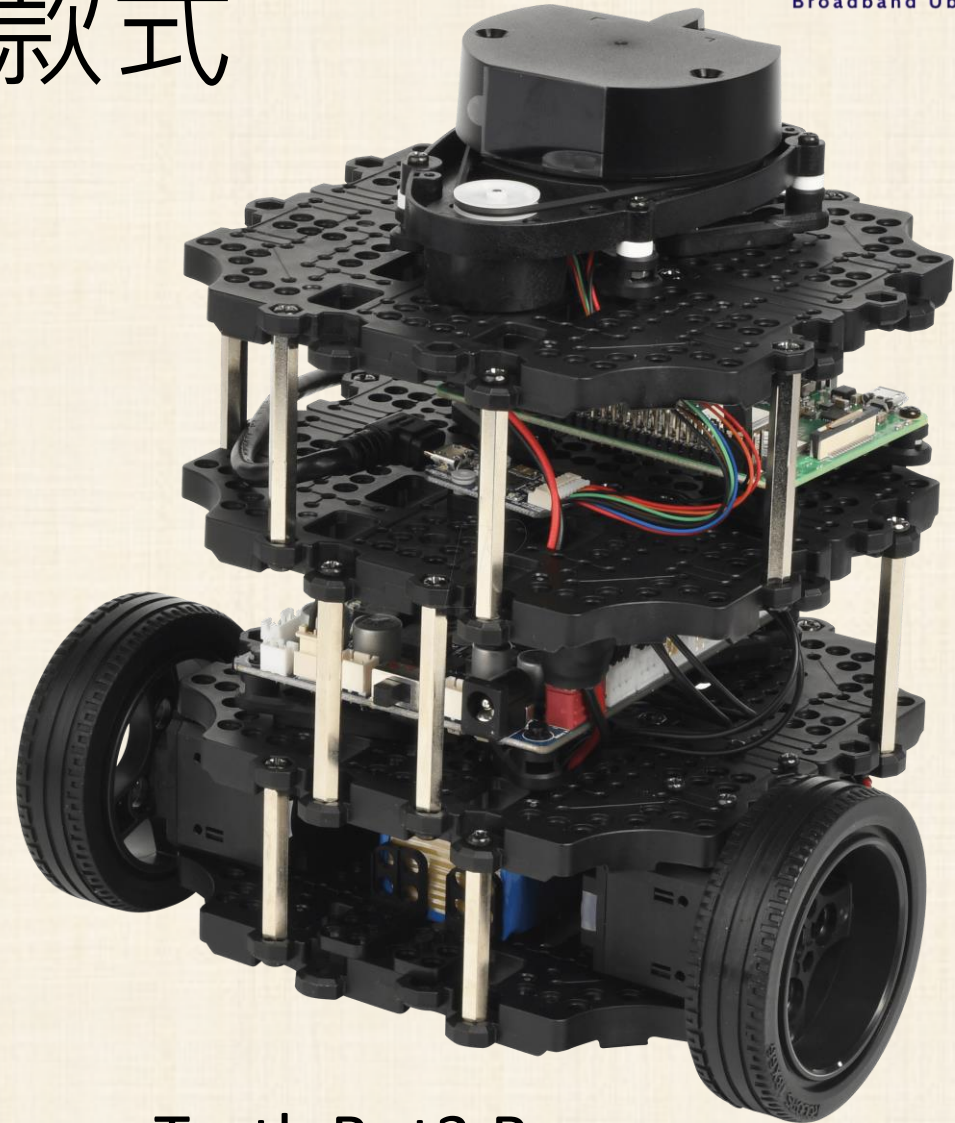
STRONG SENSOR LINEUPS

8MP Camera, Enhanced 360° LiDAR, 9-Axis Inertial Measurement Unit and precise encoder for your robot.

不同款式



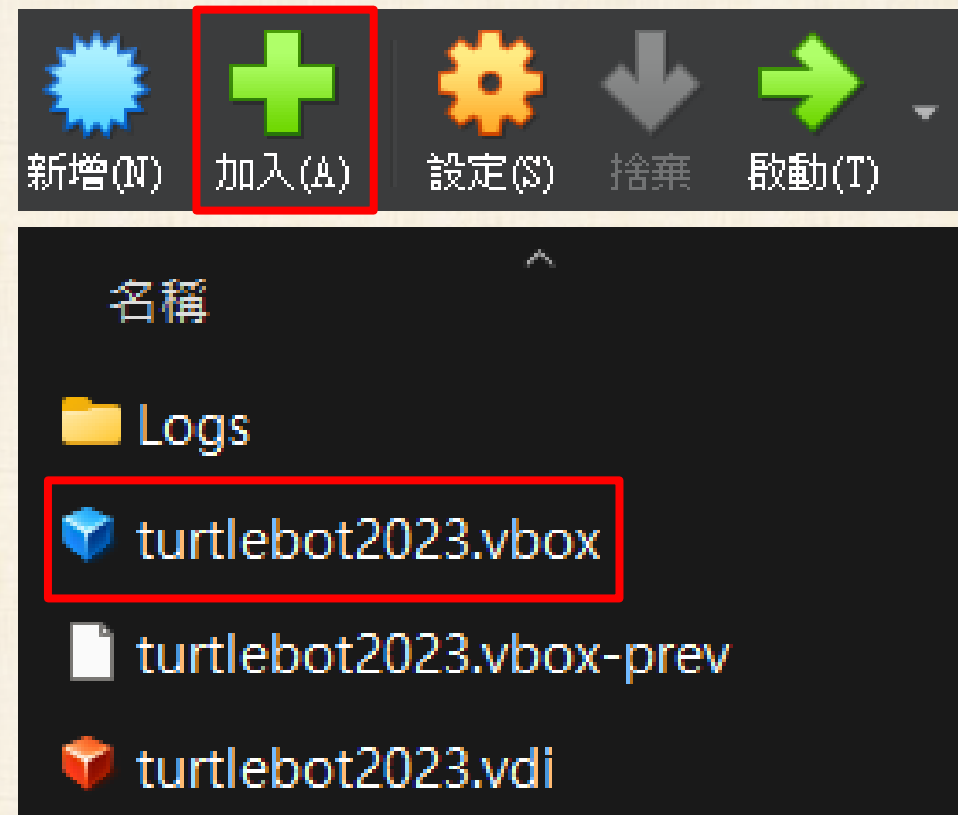
TurtleBot3 Waffle



TurtleBot3 Burger

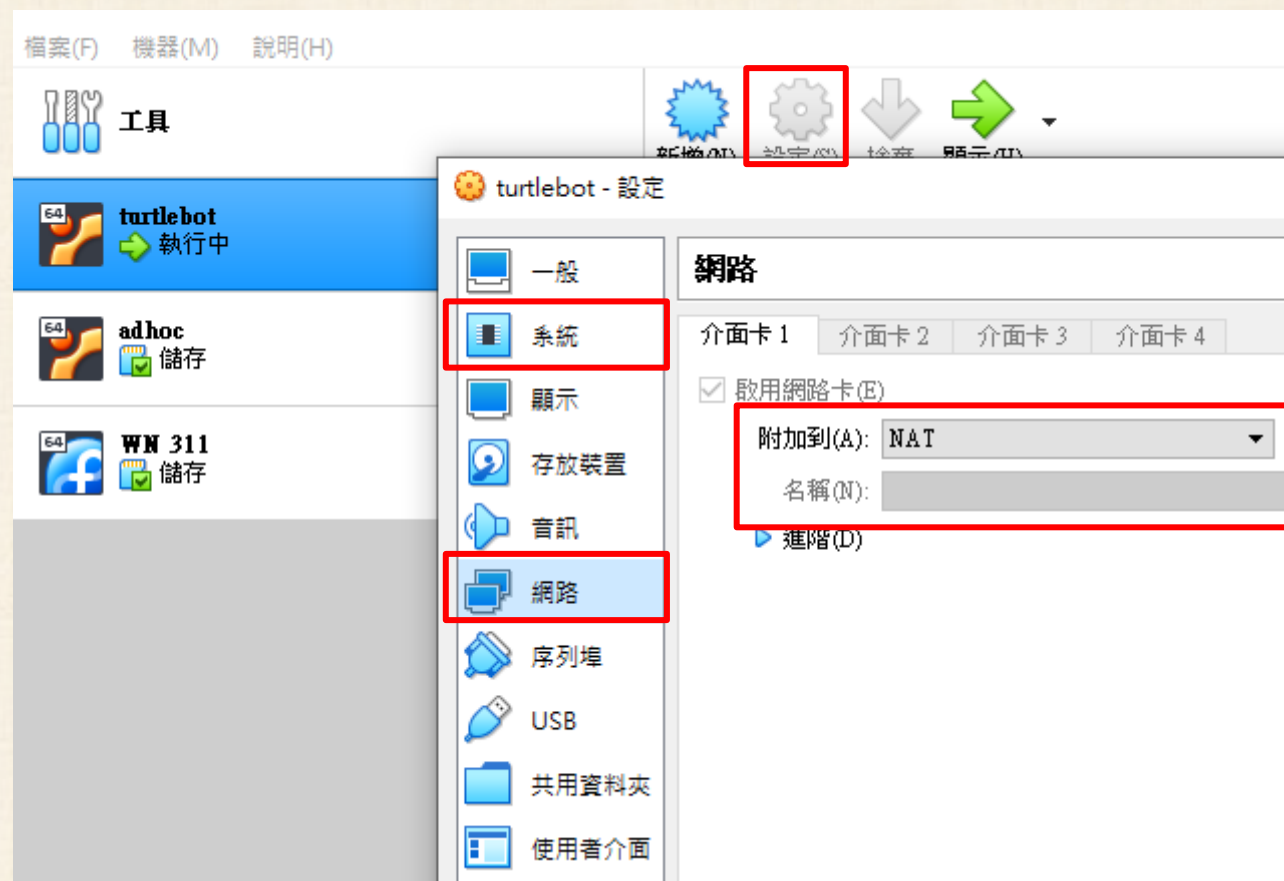
啟用虛擬機

- ✗ 下載turtlebot2023，並解壓縮
- ✗ 開啟VM，點擊加入
- ✗ 選擇turtlebot2023.vbox，開啟



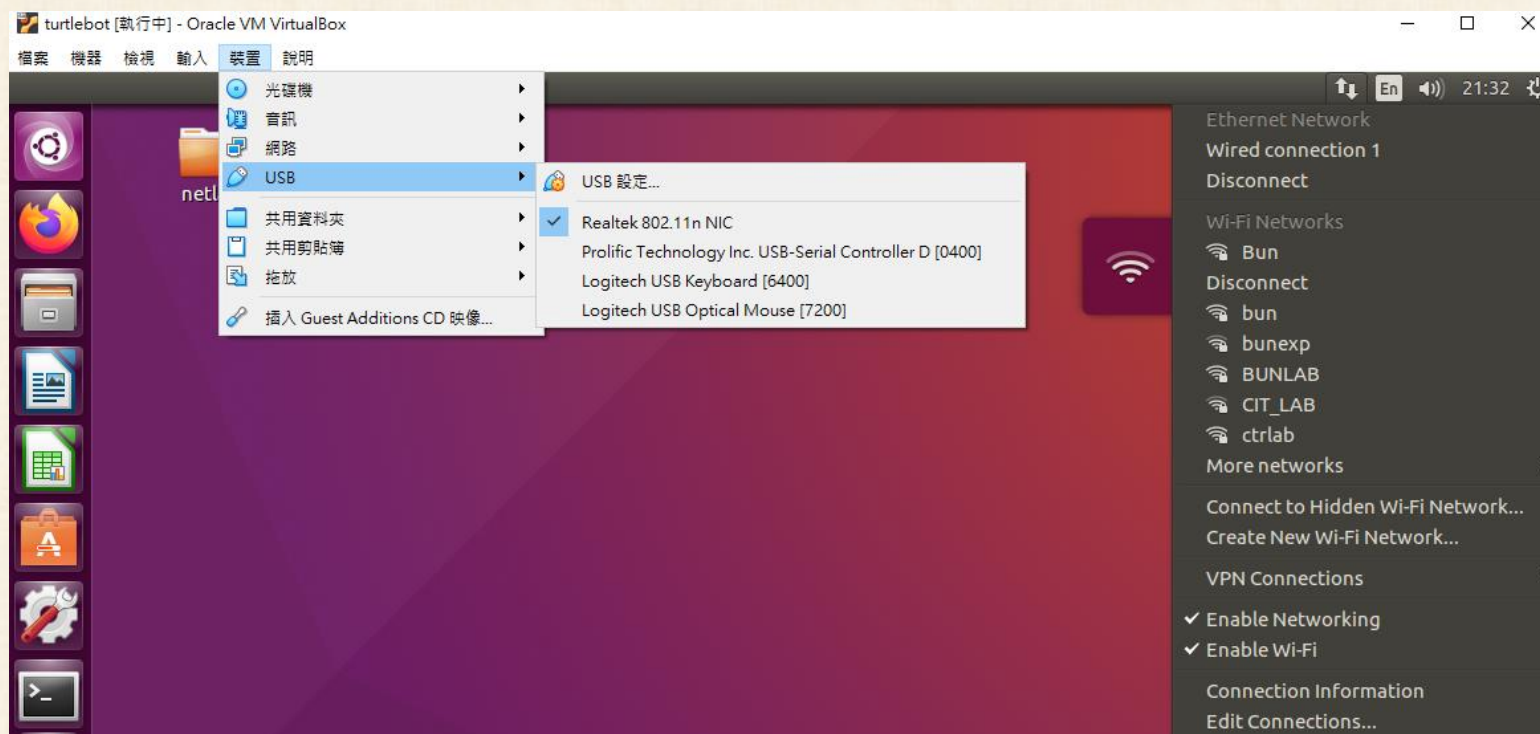
設定Oracle VM

- ✗ 選取turtlebot2023 > 設定 > 系統 > 處理器> 確認處理器(CPU)至綠色範圍即可
- ✗ 網路 > 介面卡設
 - ✗ 學校電腦：NAT
 - ✗ 自己筆電：橋接介面卡
- 儲存並啟動
- 密碼：turtlebot



學校電腦-USB網卡

- 登入後，裝置 > USB > 勾選插入的網卡型號
- 就可以選取Wi-Fi連線



筆電-橋接介面卡

- 筆電連上Wifi
- VM > 網路 > 介面卡 > 橋接介面卡 > 選筆電無線網卡



UUID error (遇到再看)

- ❑ `$ cd C:\Program Files\Oracle\VirtualBox`
- ❑ `$ VBoxManage internalcommands sethduuid XXX`
 - ❑ XXX為.vdi檔讀路徑

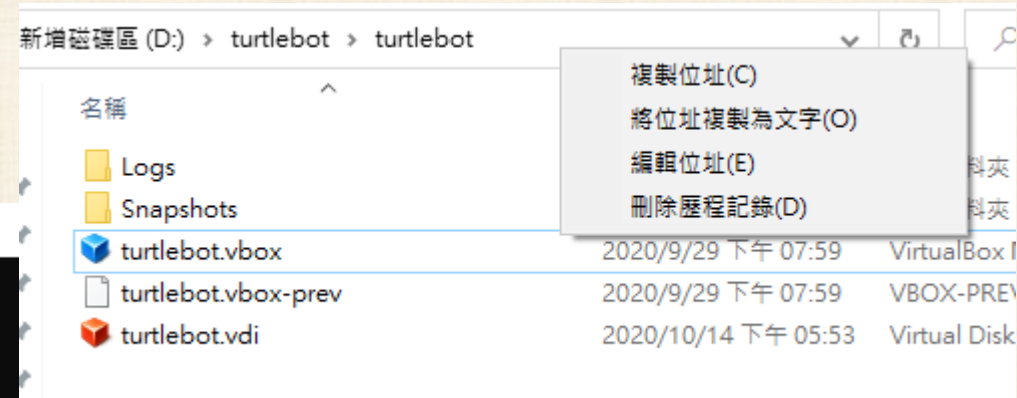
C:\Windows\system32\cmd.exe

Microsoft Windows [版本 10.0.19041.508]
(c) 2020 Microsoft Corporation。著作權所有，並保留一切權利。

C:\Users\Donny>cd "C:\Program Files\Oracle\VirtualBox"

C:\Program Files\Oracle\VirtualBox>VBoxManage internalcommands sethduuid D:\turtlebot\turtlebot\turtlebot.vdi
VBoxManage.exe: error: Cannot set a new UUID: VERR_VD_IMAGE_READ_ONLY

C:\Program Files\Oracle\VirtualBox>



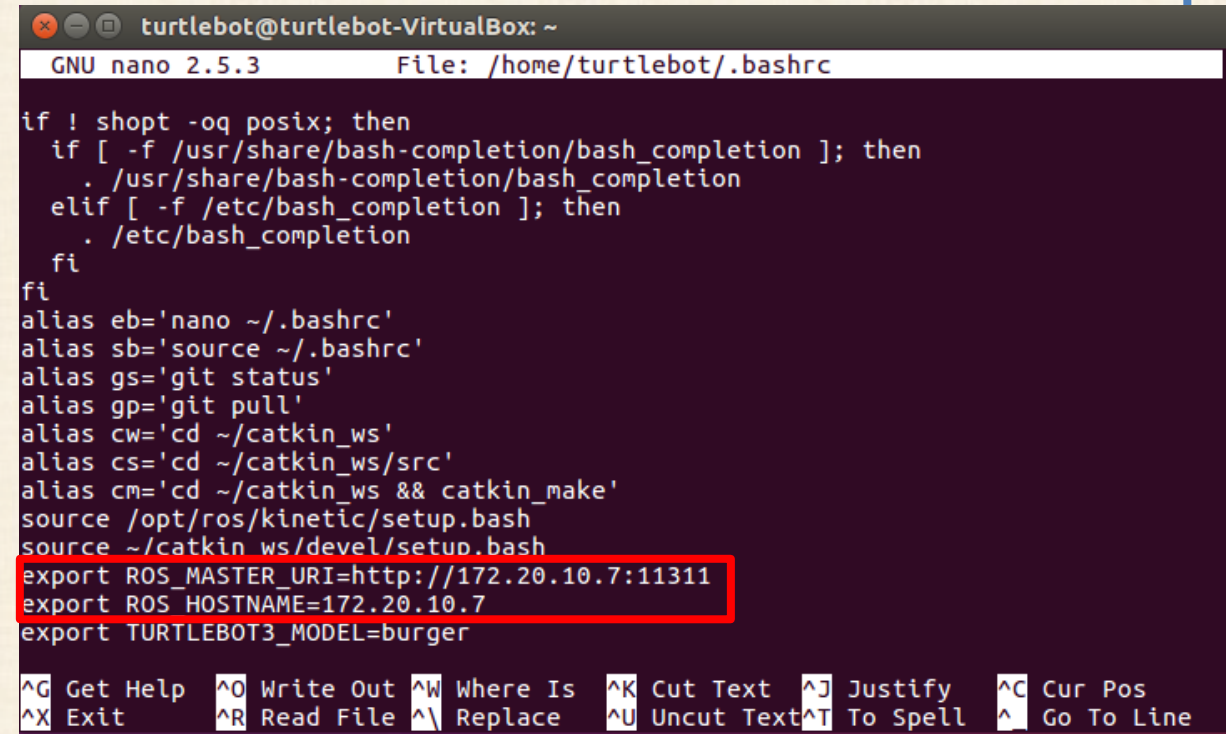
測試環境

- 1. Master
- 2. Node
- 3. Turtlebot模擬器(Gazabo)

改Turtlebot 中的 Master

- 1. \$ nano ~/.bashrc
- 2. 移至文字檔最底部
- 3. MASTER_URI改為VM的IP:11311
- 4. HOSTNAME改為VM的IP
- 5. Ctrl + X > Y存檔
- 6. \$ source ~/.bashrc

Hint: \$ ifconfig



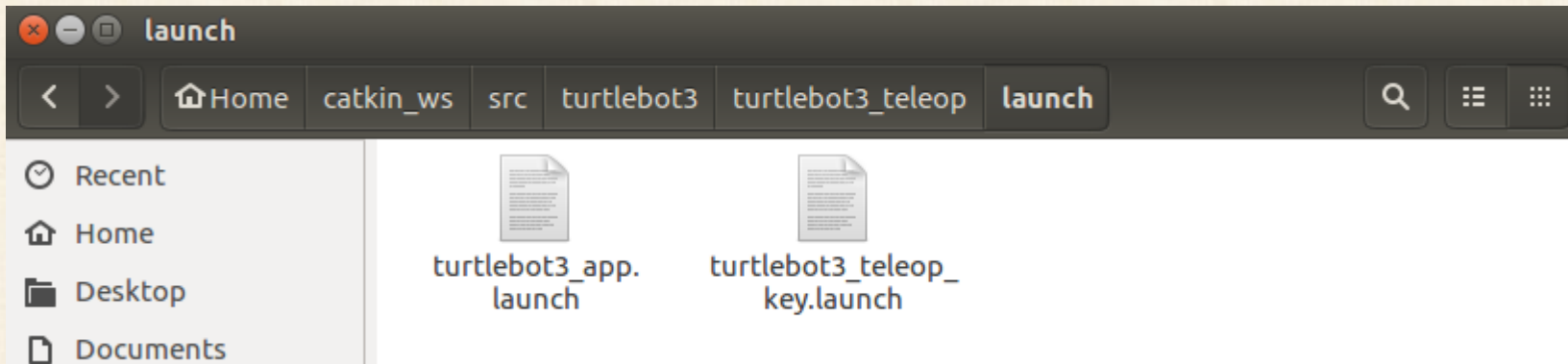
```
turtlebot@turtlebot-VirtualBox: ~  
GNU nano 2.5.3 File: /home/turtlebot/.bashrc  
  
if ! shopt -oq posix; then  
  if [ -f /usr/share/bash-completion/bash_completion ]; then  
    . /usr/share/bash-completion/bash_completion  
  elif [ -f /etc/bash_completion ]; then  
    . /etc/bash_completion  
  fi  
fi  
alias eb='nano ~/.bashrc'  
alias sb='source ~/.bashrc'  
alias gs='git status'  
alias gp='git pull'  
alias cw='cd ~/catkin_ws'  
alias cs='cd ~/catkin_ws/src'  
alias cm='cd ~/catkin_ws && catkin_make'  
source /opt/ros/kinetic/setup.bash  
source ~/catkin_ws/devel/setup.bash  
export ROS_MASTER_URI=http://172.20.10.7:11311  
export ROS_HOSTNAME=172.20.10.7  
export TURTLEBOT3_MODEL=burger  
  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```


Node, launch 關係

- nodes 為執行程式所在資料夾



- launch 會呼叫node裡的程式



Node, launch 關係

□ turtlebot3_teleop_key.launch

```
<launch>
  <arg name="model" default="$(env TURTLEBOT3_MODEL)" doc="model type [burger, waffle, waffle_pi]"/>
  <param name="model" value="$(arg model)"/>

  <!-- turtlebot3_teleop_key already has its own built in velocity smoother -->
  <node pkg="turtlebot3_teleop" type="turtlebot3_teleop_key" name="turtlebot3_teleop_keyboard" output="screen">
  </node>
</launch>
```

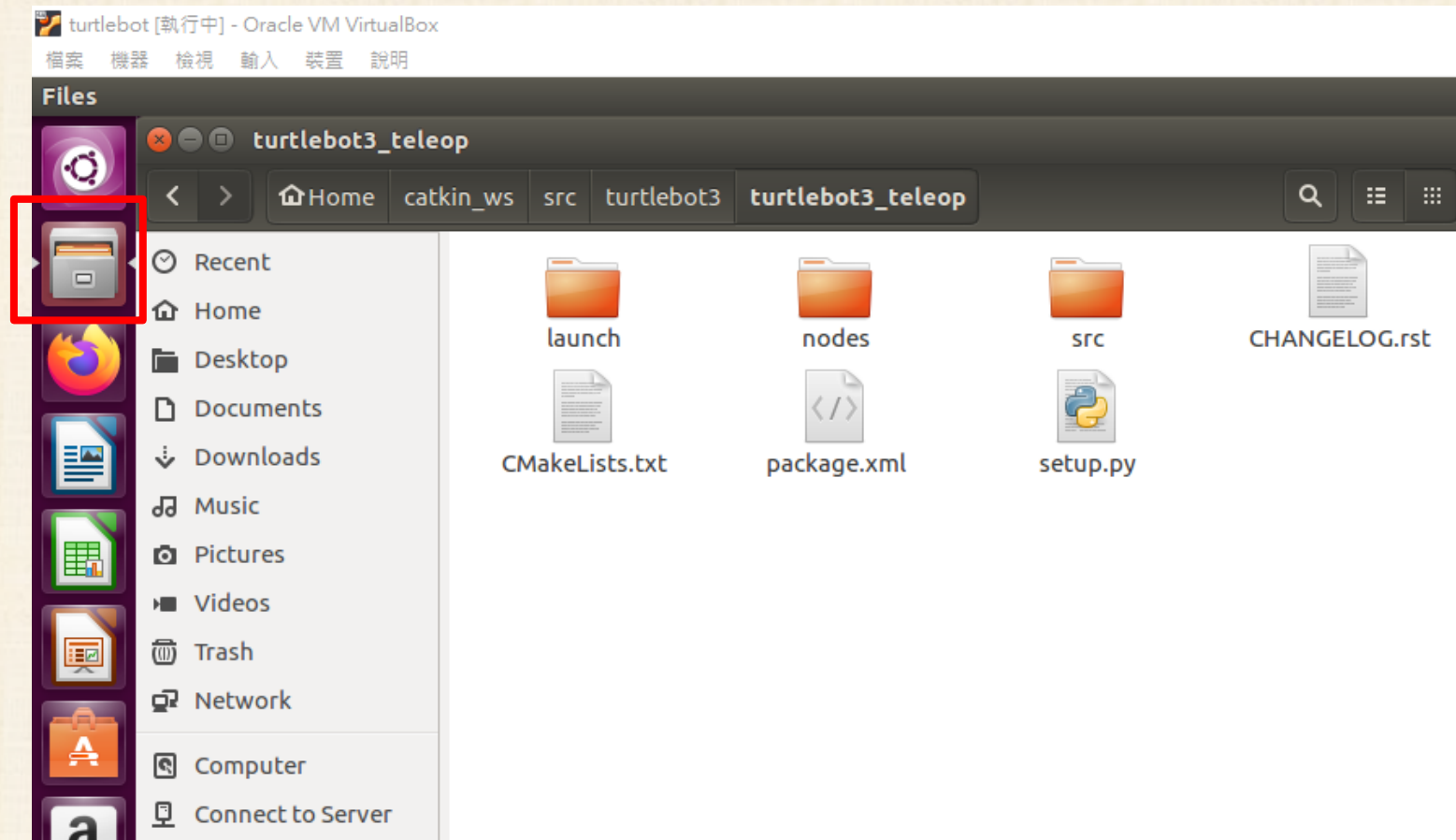
□ turtlebot3_app.launch

```
<launch>
  <arg name="model" default="$(env TURTLEBOT3_MODEL)" doc="model type [burger, waffle, waffle_pi]"/>
  <param name="model" value="$(arg model)"/>

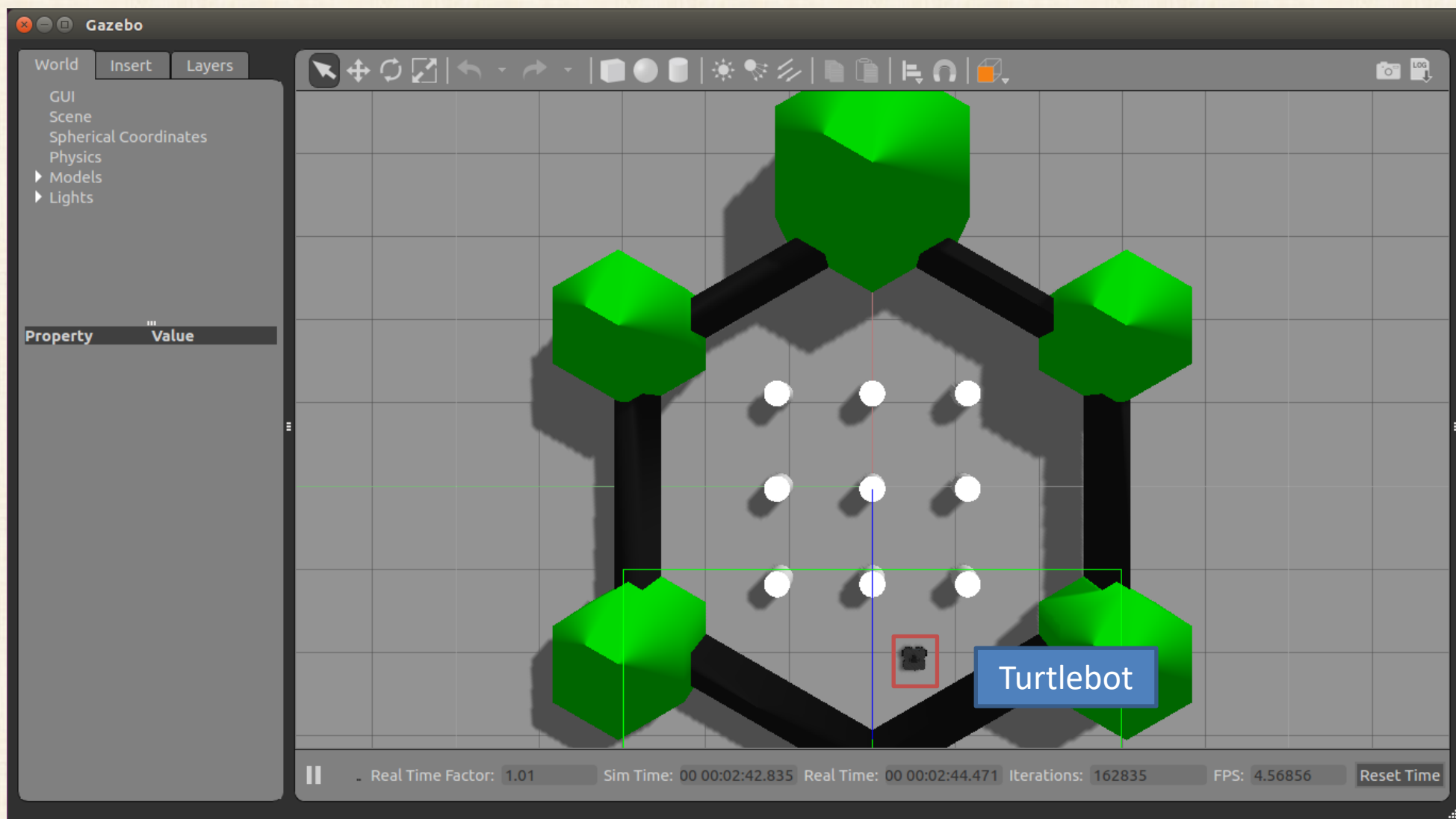
  <!-- turtlebot3_teleop_key already has its own built in velocity smoother -->
  <node pkg="turtlebot3_teleop" type="turtlebot3_app" name="turtlebot3_teleop_keyboard" output="screen">
  </node>
</launch>
```

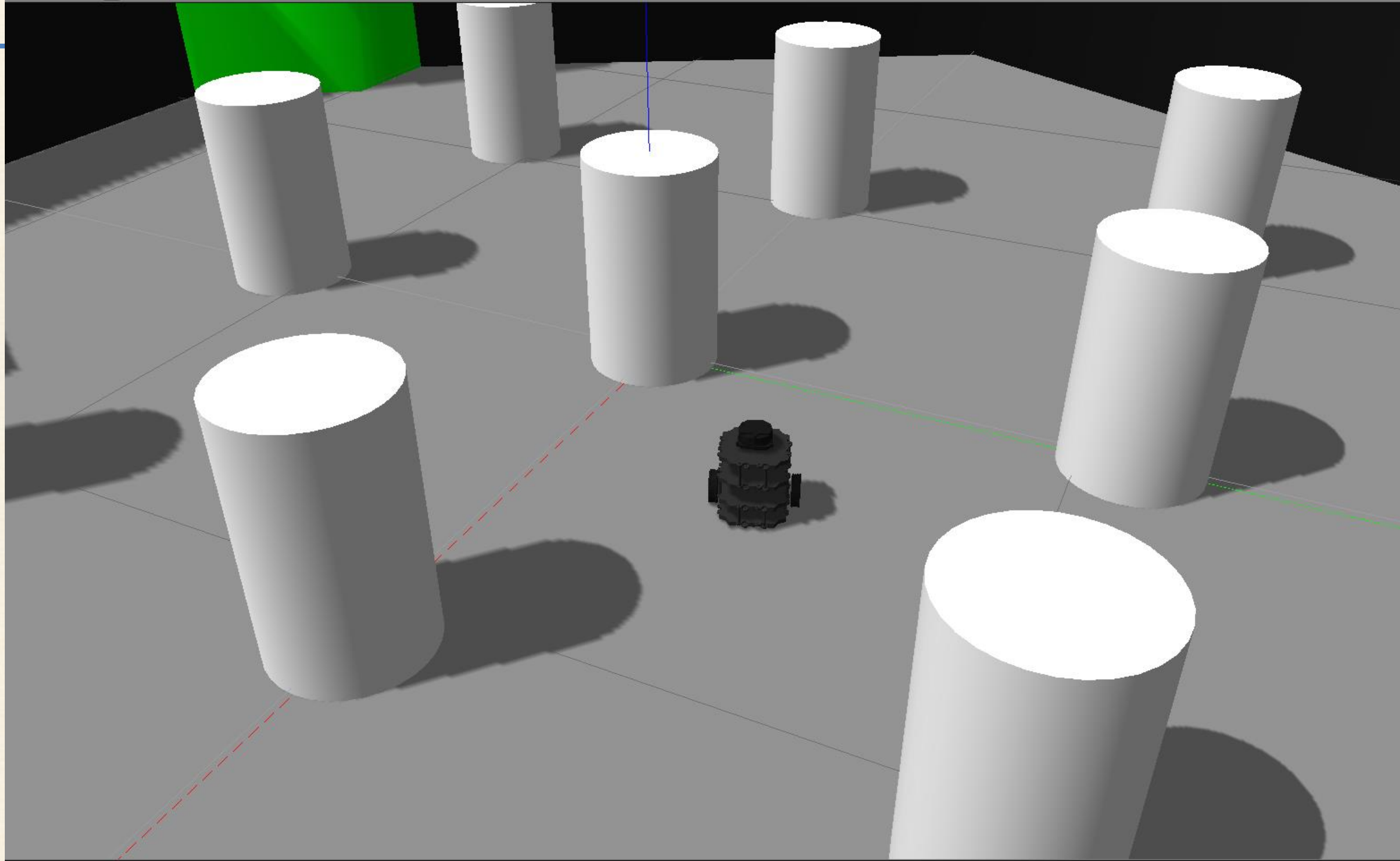
檔案存放位置

- Home > catkin_ws > src > turtlebot3 > turtlebot3_teleop



Turtlebot模擬器





鍵盤控制Turtlebot

```
Control Your TurtleBot3!
```

```
-----
```

```
Moving around:
```

```
      w
a     s     d
      x
```

```
w/x : increase/decrease linear velocity (Burger : ~ 0.22, Waffle an  
d Waffle Pi : ~ 0.26)
```

```
a/d : increase/decrease angular velocity (Burger : ~ 2.84, Waffle a  
nd Waffle Pi : ~ 1.82)
```

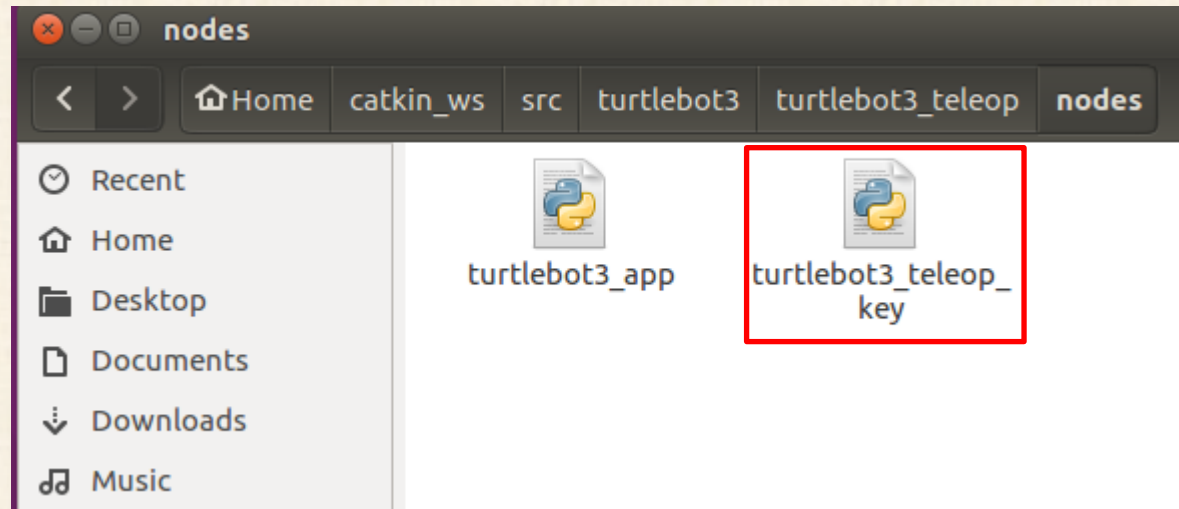
```
space key, s : force stop
```

```
CTRL-C to quit
```

```
currently:      linear vel 0.01   angular vel 0.0   W  
currently:      linear vel 0.02   angular vel 0.0   W  
currently:      linear vel 0.03   angular vel 0.0   W  
currently:      linear vel 0.04   angular vel 0.0   W  
currently:      linear vel 0.0    angular vel 0.0   S
```

完成程式碼-1

- 修改turtlebot3_teleop_key.py
 - 觀察上下控制寫法
 - 加入左右控制



完成程式碼-2

□ Hint

觀察前、後移動的部分以及這些變數、函式名稱

```
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))
```

```
114 def checkAngularLimitVelocity(vel):
115     if turtlebot3_model == "burger":
116         vel = constrain(vel, -BURGER_MAX_ANG_VEL, BURGER_MAX_ANG_VEL)
117     elif turtlebot3_model == "waffle" or turtlebot3_model == "waffle_pi":
118         vel = constrain(vel, -WAFFLE_MAX_ANG_VEL, WAFFLE_MAX_ANG_VEL)
119     else:
120         vel = constrain(vel, -BURGER_MAX_ANG_VEL, BURGER_MAX_ANG_VEL)
```

```
134 target_linear_vel = 0.0
135 target_angular_vel = 0.0
136 control_linear_vel = 0.0
137 control_angular_vel = 0.0
```

```
44 LIN_VEL_STEP_SIZE = 0.01
45 ANG_VEL_STEP_SIZE = 0.1
```

注意事項

- 程式中有挖空，不能直接執行
- 按一下按鍵, 等於在該方向 **+1 單位速度**

```
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

    elif key == 'd' :
        # add turn right action
```

執行程式

□ 共需開啟**3個**終端機，分別執行以下指令：

1. ROS Master：

- `$ roscore`

2. Open simulator：

- `$ cd catkin_ws`

- `$ export TURTLEBOT3_MODEL=burger`

- `$ roslaunch turtlebot3_gazebo turtlebot3_world.launch`

3. 執行程式：

`$ roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch`


```

roscore http://172.20.10.7:11311/
turtlebot@turtlebot-VirtualBox:~$ roscore
... logging to /home/turtlebot/.ros/log/66feb2ca-1f0a-11ed-ad72-080
027f397bc/roslaunch-turtlebot-VirtualBox-4211.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://172.20.10.7:43641/
ros_comm version 1.12.14

SUMMARY
=====

PARAMETERS
* /rostdistro: kinetic
* /rosversion: 1.12.14

NODES
auto-starting new master
process[master]: started with pid [4221]
ROS_MASTER_URI=http://172.20.10.7:11311/

```

1

```

/home/turtlebot/catkin_ws/src/turtlebot3_simulations/turtlebot3_gazebo/launch/turtle
turtlebot@turtlebot-VirtualBox:~/catkin_ws$ export TURTLEBOT3_MODEL=burger
turtlebot@turtlebot-VirtualBox:~/catkin_ws$ roslaunch turtlebot3_gazebo turtlebo
t3_world.launch
... logging to /home/turtlebot/.ros/log/ae405926-1f0b-11ed-ad72-080027f397bc/ros
launch-turtlebot-VirtualBox-4844.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://172.20.10.7:44555/

SUMMARY
=====

PARAMETERS
* /robot_description: <?xml version="1...
* /rostdistro: kinetic
* /rosversion: 1.12.14
* /use_sim_time: True

NODES
/
gazebo (gazebo_ros/gzserver)
gazebo_gui (gazebo_ros/gzclient)

```

2

```

/home/turtlebot/catkin_ws/src/turtlebot3/turtlebot3_teleop/launch/turtlebot3_teleop
turtlebot@turtlebot-VirtualBox:~$ roslaunch turtlebot3_teleop turtlebot3_teleop_
key.launch
... logging to /home/turtlebot/.ros/log/66feb2ca-1f0a-11ed-ad72-080027f397bc/ros
launch-turtlebot-VirtualBox-4721.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://172.20.10.7:34763/

SUMMARY
=====

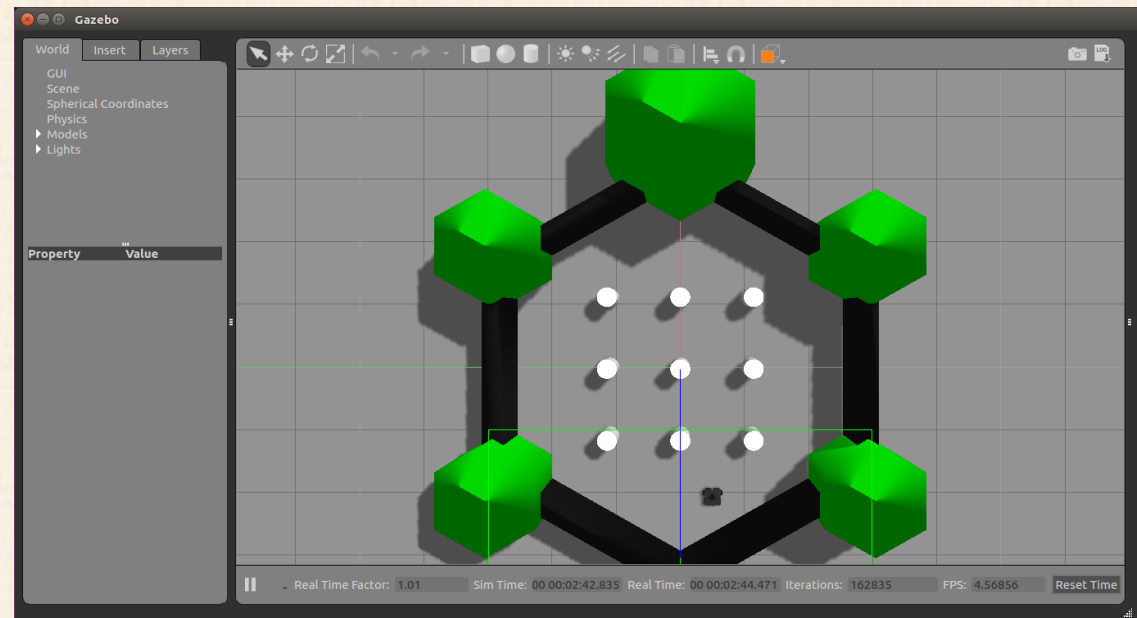
PARAMETERS
* /model: burger
* /rostdistro: kinetic
* /rosversion: 1.12.14

NODES
/
turtlebot3_teleop_keyboard (turtlebot3_teleop/turtlebot3_teleop_key)

ROS_MASTER_URI=http://172.20.10.7:11311

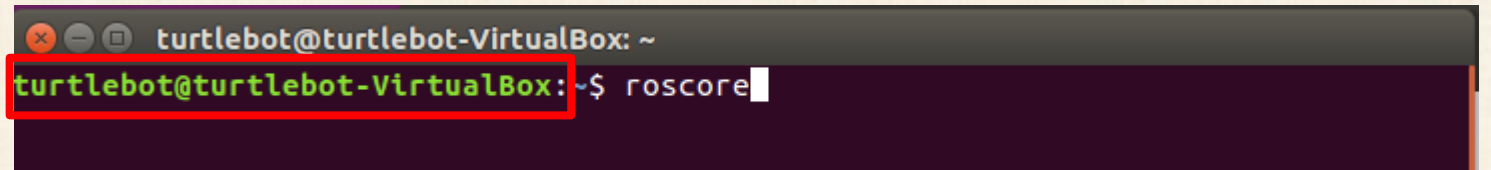
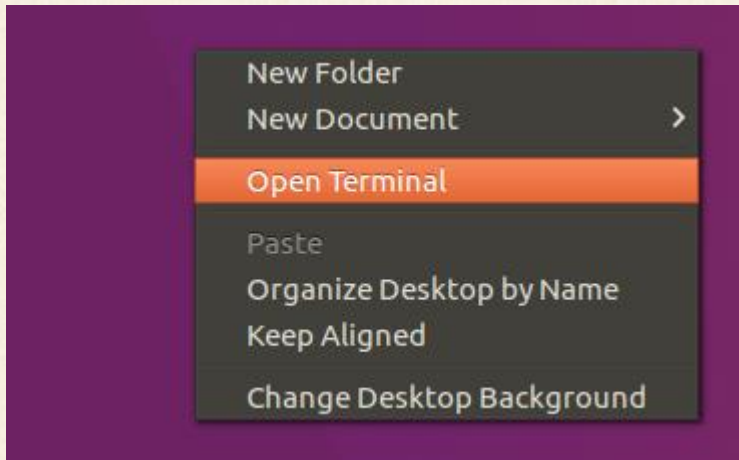
```

3



Step 1 (ROS Master)

- 在桌面右鍵 > Open Terminal > 輸入\$ roscore



Step 2 (Open simulator)

□ 桌面右鍵 > Open Terminal > 輸入

```
$ cd catkin_ws
```

```
$ export TURTLEBOT3_MODEL=burger
```

```
$ roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

```
turtlebot@turtlebot-VirtualBox:~$ cd catkin_ws/  
turtlebot@turtlebot-VirtualBox:~/catkin_ws$ export TURTLEBOT3_MODEL=burger  
turtlebot@turtlebot-VirtualBox:~/catkin_ws$ roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

Step 3 (執行程式)

□ 桌面右鍵 > Open Terminal

□ 輸入

```
$ roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

```
/home/turtlebot/catkin_ws/src/turtlebot3/turtlebot3_teleop/launch/turtlebot3_teleop_
turtlebot@turtlebot-VirtualBox:~$ roslaunch turtlebot3_teleop turtlebot3_teleop_
key.launch
... logging to /home/turtlebot/.ros/log/66feb2ca-1f0a-11ed-ad72-080027f397bc/ros
launch-turtlebot-VirtualBox-4721.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://172.20.10.7:34763/

SUMMARY
=====
```


Q1

- 1.修改turtlebot3_teleop_key.py，加上左右移動的程式碼
- 2.依照上Step 1~3測試程式可否運作

```
while not rospy.is_shutdown():
    key = getKey()
    if key == 'w' :
        target_linear_vel = checkLine
        status = status + 1
        print(vels(target_linear_vel,
    elif key == 'x' :
        target_linear_vel = checkLine
        status = status + 1
        print(vels(target_linear_vel,
    elif key == 'a' :
        # add turn left action

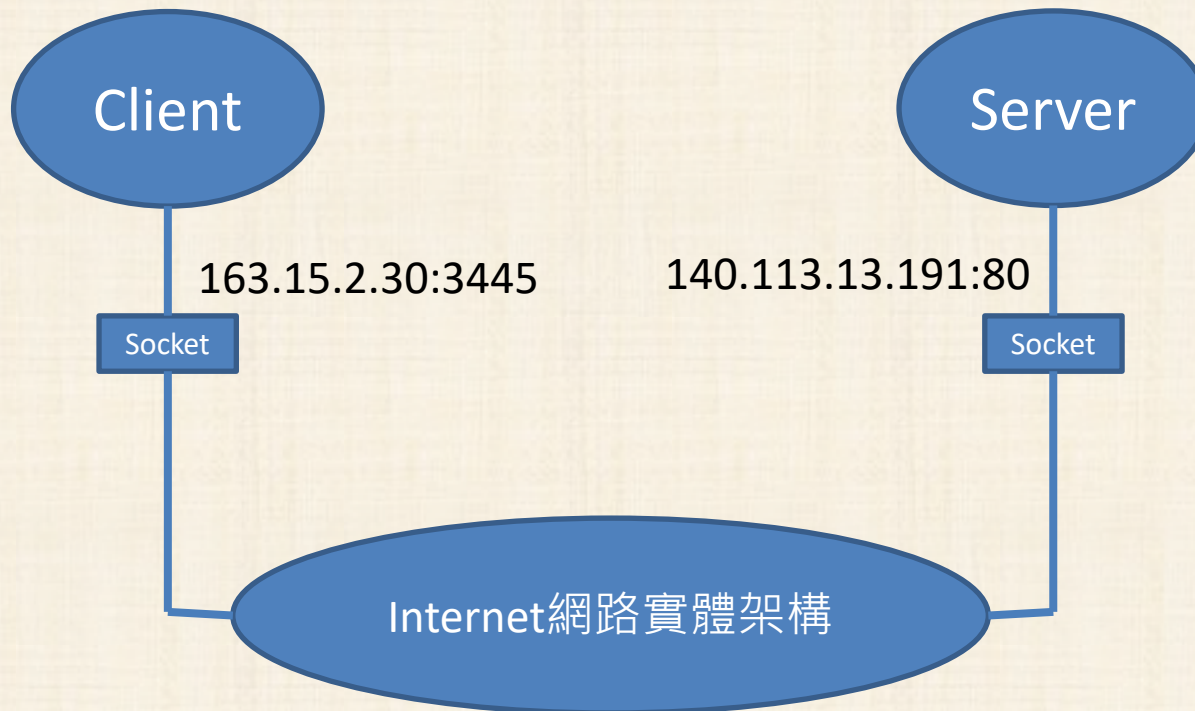
    elif key == 'd' :
        # add turn right action
```

Socket

- Socket 就是一個網路上的通訊端點，使用者或應用程式只要連接到 Socket 便可以和網路上任何一個通訊端點連線，任何一個 Socket 都給予一個特殊號碼（IP number + port number）
- Datagram sockets (connectionless) :
 - 利用UDP封包傳送，因此接收端socket 可能會收到次序錯誤的資料，並且部分資料也有可能遺失，優點是傳輸延遲低。
- Stream sockets (connection-oriented) :
 - 利用TCP封包來傳送，因此接收端Socket 可以收到順序無誤、無重複，並且正確的資料。好處是比上面那種方式可靠且有序的。

Socket

- 使用(IP, PORT) Socket連線的終端必須要**同網域**底下
連到同樣的wifi
- Port可以使用1024~65535



Socket程式-Server

```
#socket
HOST = '192.168.50.156'
PORT = 8001
try:
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
except socket.error, msg:
    sys.stderr.write("[ERROR] %s\n" % msg[1])
    sys.exit(1)

try:
    sock.bind((HOST,PORT))
    sock.listen(5)
except socket.error, msg:
    sys.stderr.write("[ERROR] %s\n" % msg[1])
    exit(1)
```

建立TCP Socket

綁定Socket的IP&PORT
最大連線數

```
conn,addr=sock.accept() 等待連線成功
print(addr)
msg1=conn.recv(1024)      接收內容
print(msg1)
```




Socket程式-Client


```
Socket socket = new Socket( host: "192.168.50.156", port: 8001);  
PrintWriter outToServer= new PrintWriter(  
    new OutputStreamWriter(  
        socket.getOutputStream()));  
outToServer.print(params[0]);  
outToServer.flush();
```

目標位置




Android Studio

Android Studio Setup Wizard

 **Welcome**
Android Studio

Welcome! This wizard will set up your development environment for Android Studio.
Additionally, the wizard will help port existing Android apps into Android Studio
or create a new Android application project.




Previ...

Next

Cancel

Finish

Android Studio Setup Wizard

 **Install Type**

Choose the type of setup you want for Android Studio:

☒ Standard

Android Studio will be installed with the most common settings and options.
Recommended for most users.

☐ Custom

You can customize installation settings and components installed.

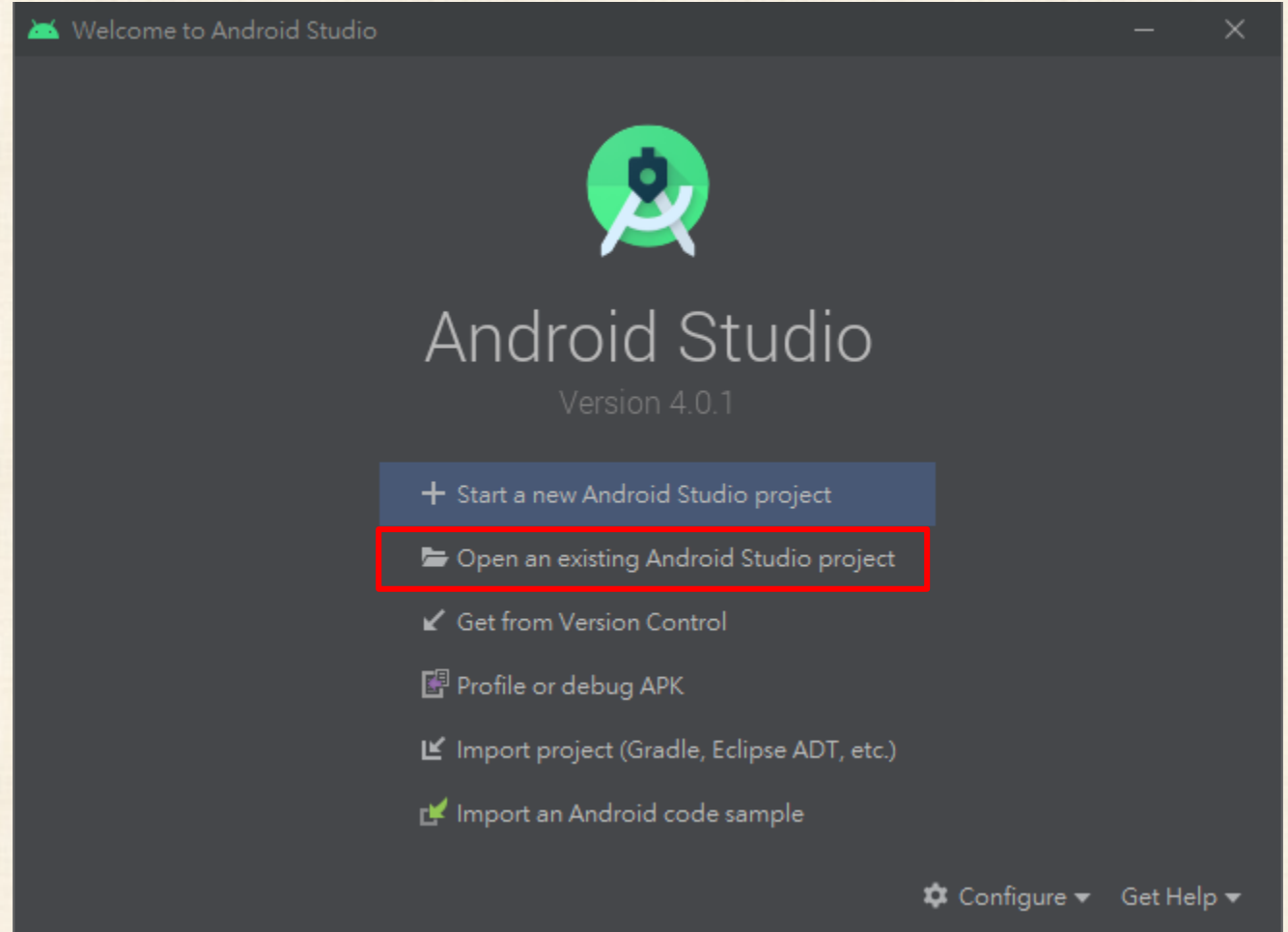
Previ...

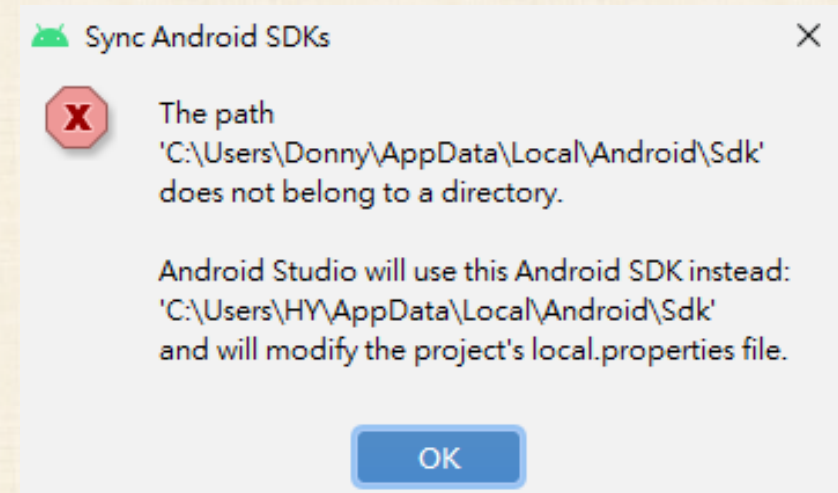
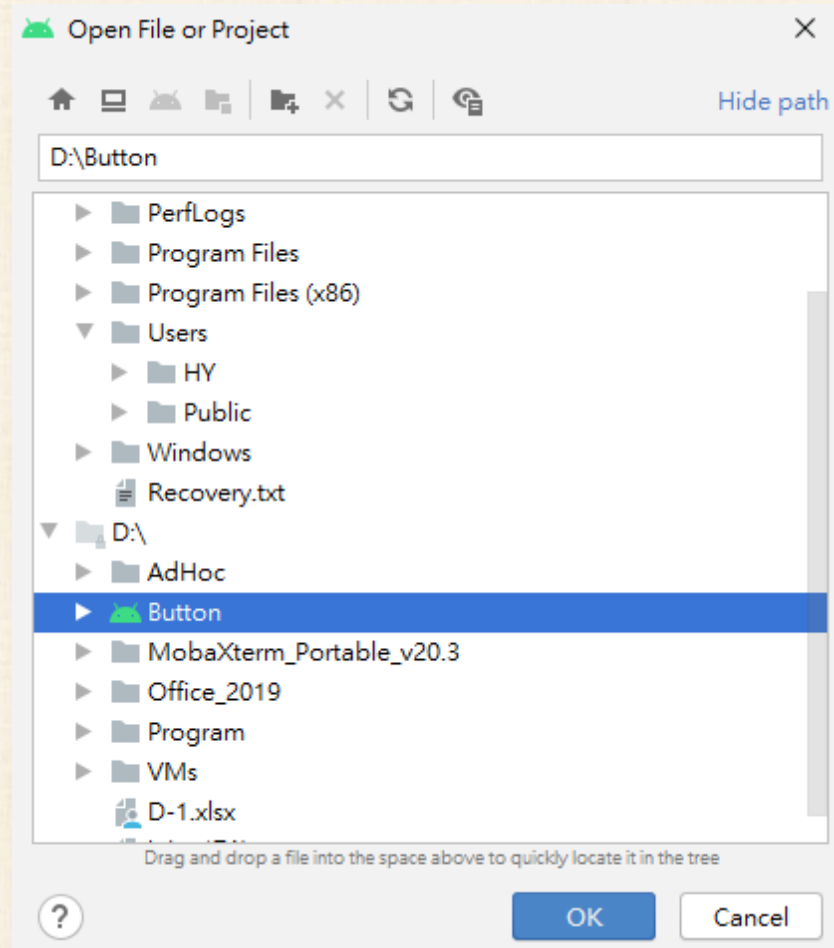
Next

Cancel

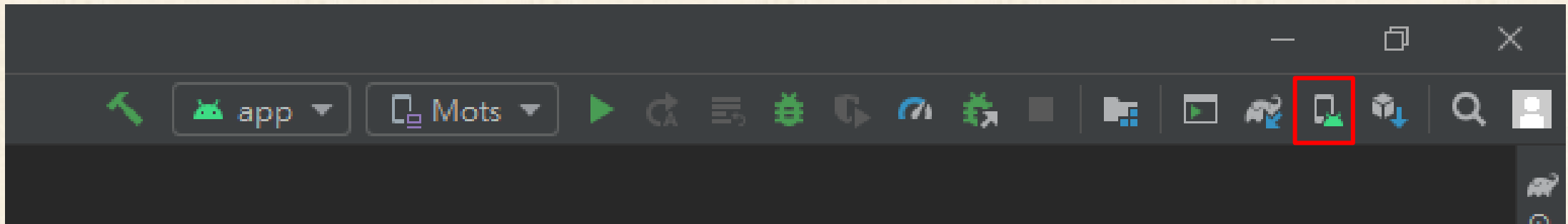
Finish

- 記得先下載Button app





□ 打開Device Manager



□ Create Device



Select Hardware

Choose a device definition

| Category | Name ▾ | Play Store | Size | Resolution | Density |
|----------|------------|------------|-------|------------|---------|
| Phone | Pixel 3a | | 5.6" | 1080x22... | 440dpi |
| | Pixel 3 XL | | 6.3" | 1440x29... | 560dpi |
| | Pixel 3 | | 5.46" | 1080x21... | 440dpi |
| | Pixel 2 XL | | 5.99" | 1440x28... | 560dpi |
| | Pixel 2 | | 5.0" | 1080x19... | 420dpi |
| | Pixel | | 5.0" | 1080x19... | 420dpi |
| | Nexus S | | 4.0" | 480x800 | hdpi |

New Hardware Profile

Import Hardware Profiles



Clone Device...

Pixel 2



Size: large
Ratio: long
Density: 420dpi



Previous

Next

Cancel

Finish

Select a system image

Recommended x86 Images Other Images

| Release Name | API Level ▼ | ABI | Target |
|---------------------------------|-------------|-----|-----------------------------|
| R Download | 30 | x86 | Android 10.0+ (Google Play) |
| Q Download | 29 | x86 | Android 10.0 (Google Play) |
| Pie | 28 | x86 | Android 9.0 (Google Play) |
| Oreo Download | 27 | x86 | Android 8.1 (Google Play) |
| Oreo Download | 26 | x86 | Android 8.0 (Google Play) |
| Nougat Download | 25 | x86 | Android 7.1.1 (Google Play) |
| Nougat Download | 24 | x86 | Android 7.0 (Google Play) |

若沒有下載要先按下載

可換其他網路下載
下載完再調整Wi-fi

Pie



API Level

28

Android

9.0

Google Inc.

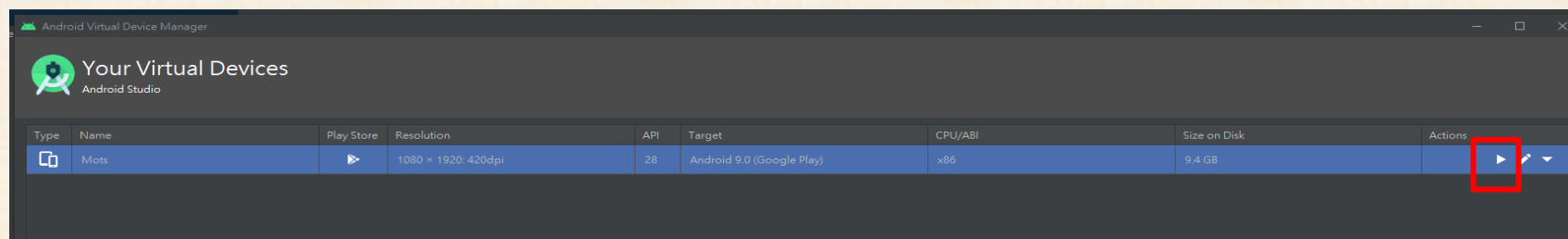
System Image

x86

We recommend these Google Play images because this device is compatible with Google Play.

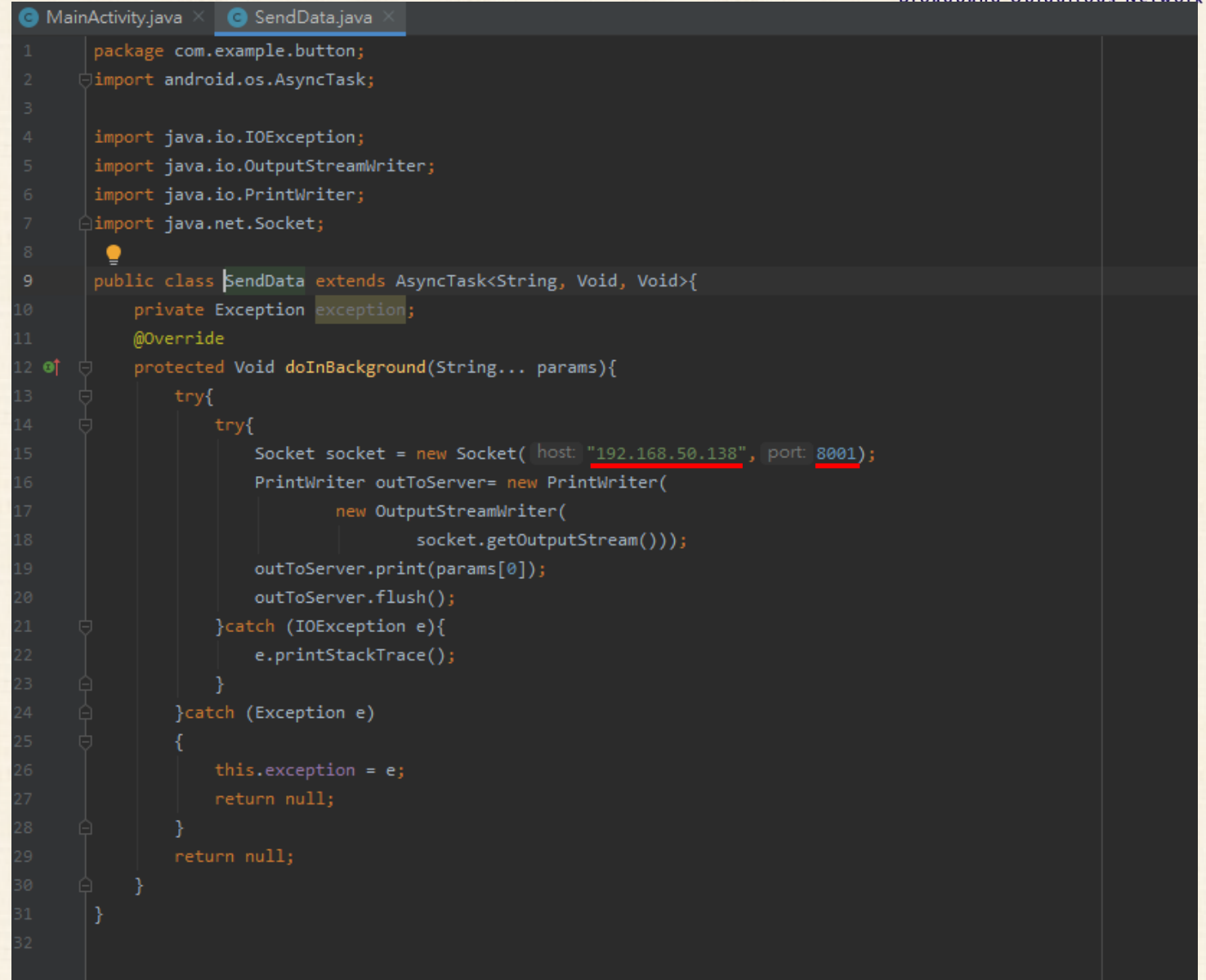
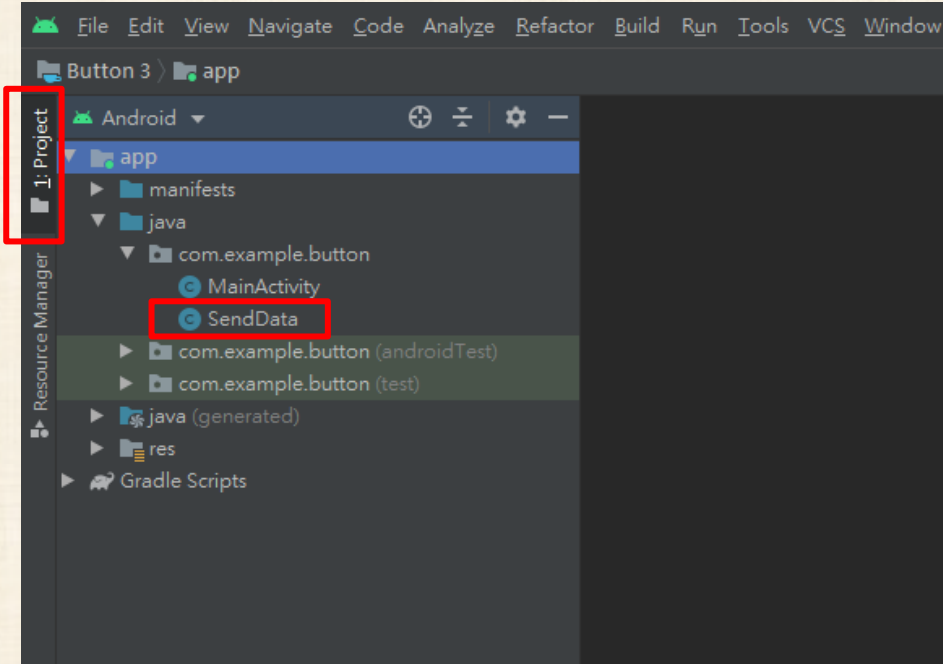
Questions on API level?

[See the API level distribution sheet](#)



- ❑ 如果遇到需要權限、密碼，打X or 略過
- ❑ 路徑資料夾不能有中文
 - ❑ 會出現gardle preject sync failed.

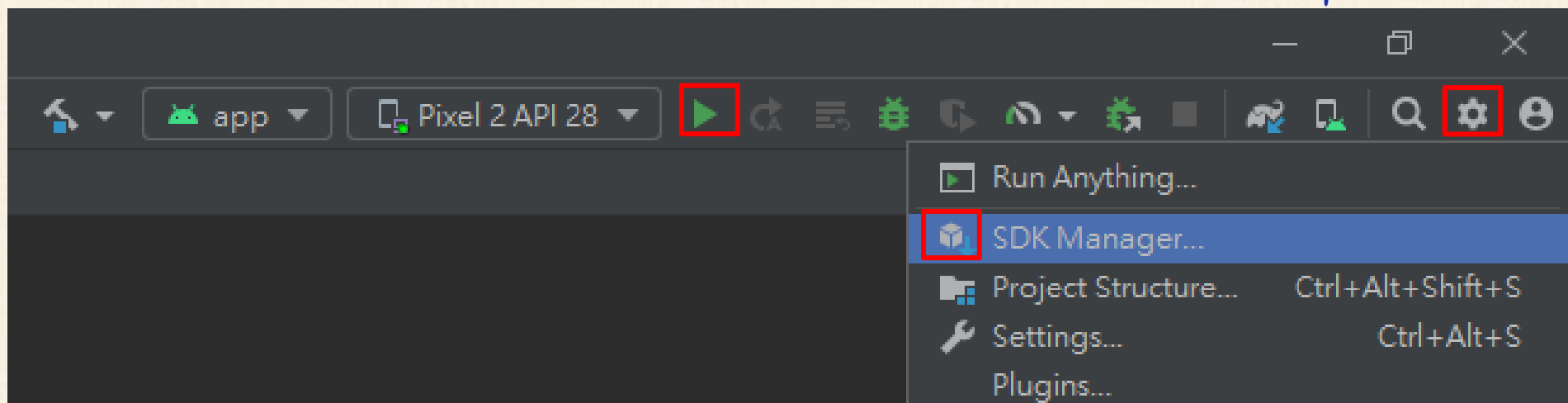




□ 改成控制PC(VM)的IP



開啟App



- 點箭頭把App燒入模擬器
- 如果箭頭不能點，點立方體，下載API level 28 SDK, 再試一次

System Settings
Passwords
HTTP Proxy
Data Sharing
Date Formats
Updates
Android SDK
Memory Settings
Notifications
Quick Lists
Path Variables
vmap

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

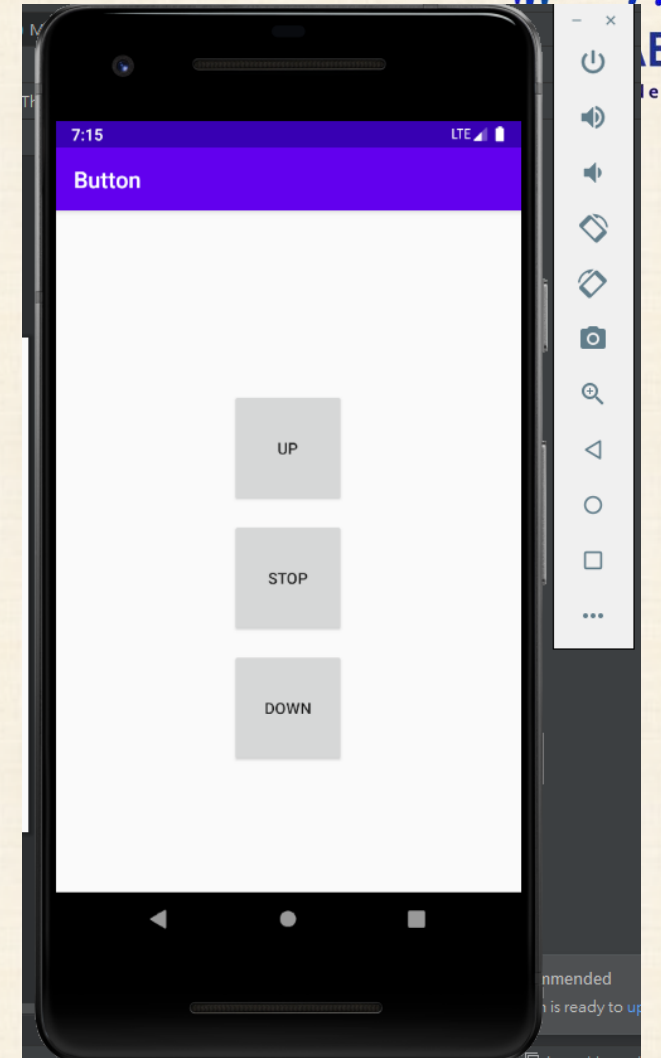
| | Name | API Level | Revision | Status |
|---|---|-----------|----------|---------------|
| × | <input type="checkbox"/> Android 11.0 (R) | 30 | 3 | Installed |
| | <input type="checkbox"/> Android 10.0 (Q) | 29 | 5 | Not installed |
| ± | <input checked="" type="checkbox"/> Android 9.0 (Pie) | 28 | 6 | Not installed |
| | <input type="checkbox"/> Android 8.1 (Oreo) | 27 | 3 | Not installed |
| | <input type="checkbox"/> Android 8.0 (Oreo) | 26 | 2 | Not installed |
| | <input type="checkbox"/> Android 7.1.1 (Nougat) | 25 | 3 | Not installed |
| | <input type="checkbox"/> Android 7.0 (Nougat) | 24 | 2 | Not installed |
| | <input type="checkbox"/> Android 6.0 (Marshmallow) | 23 | 3 | Not installed |
| | <input type="checkbox"/> Android 5.1 (Lollipop) | 22 | 2 | Not installed |
| | <input type="checkbox"/> Android 5.0 (Lollipop) | 21 | 2 | Not installed |

Debug

- 如果遇到 unable to delete directory
 - 關掉android studio
 - 到Button在的資料夾下，刪除build資料夾
- Installing missing SDK package

Q2

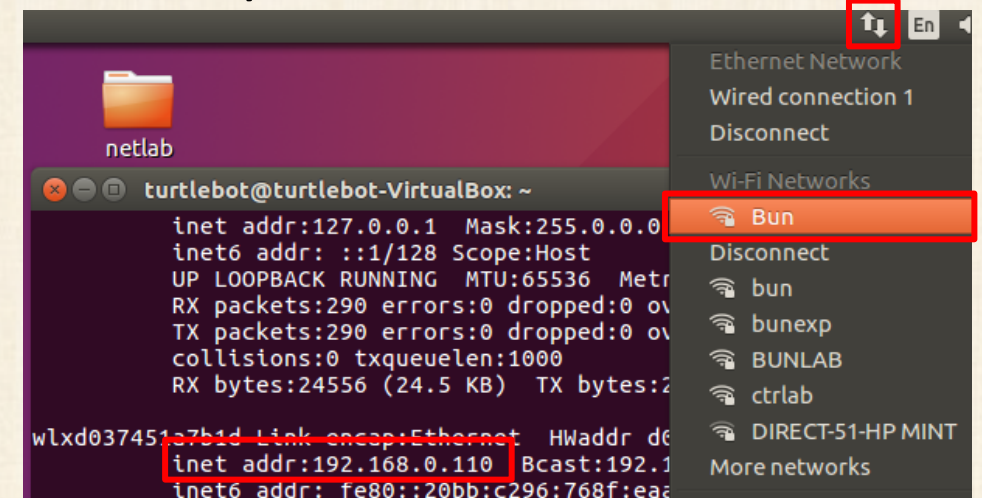
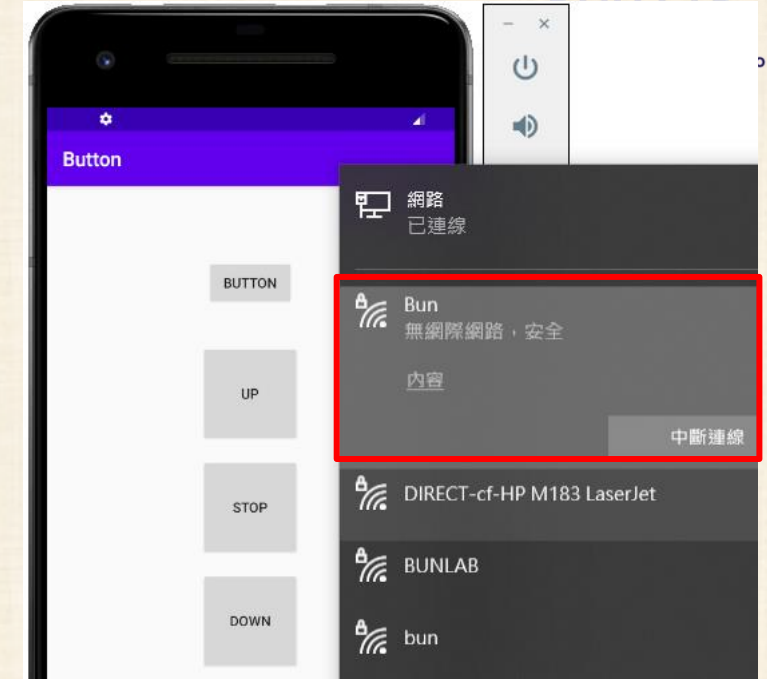
- GOAL : 以APP控制Turtlebot前後移動
- 修改turtlebot3_app.py中的IP、PORT
- 修改Button APP中的IP、PORT
- 執行另一個launch檔
 - `$ roslaunch turtlebot3_teleop turtlebot3_app.launch`





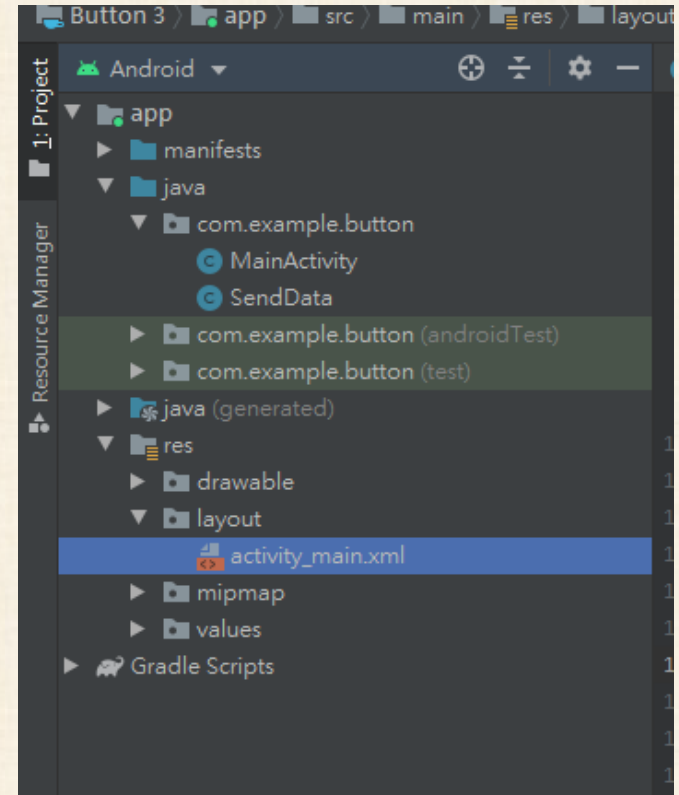
Debug

- address already in use
 - 更換 socket 的 PORT
- 舉手問助教前，請先確認以下
 - 1. VM 已連至 Bun
 - 2. 開啟Andriod APP的系統(筆電 or 外部系統), 已連至 Bun
 - 3. 是否已照投影片 17 頁設定
- Andriod Studio



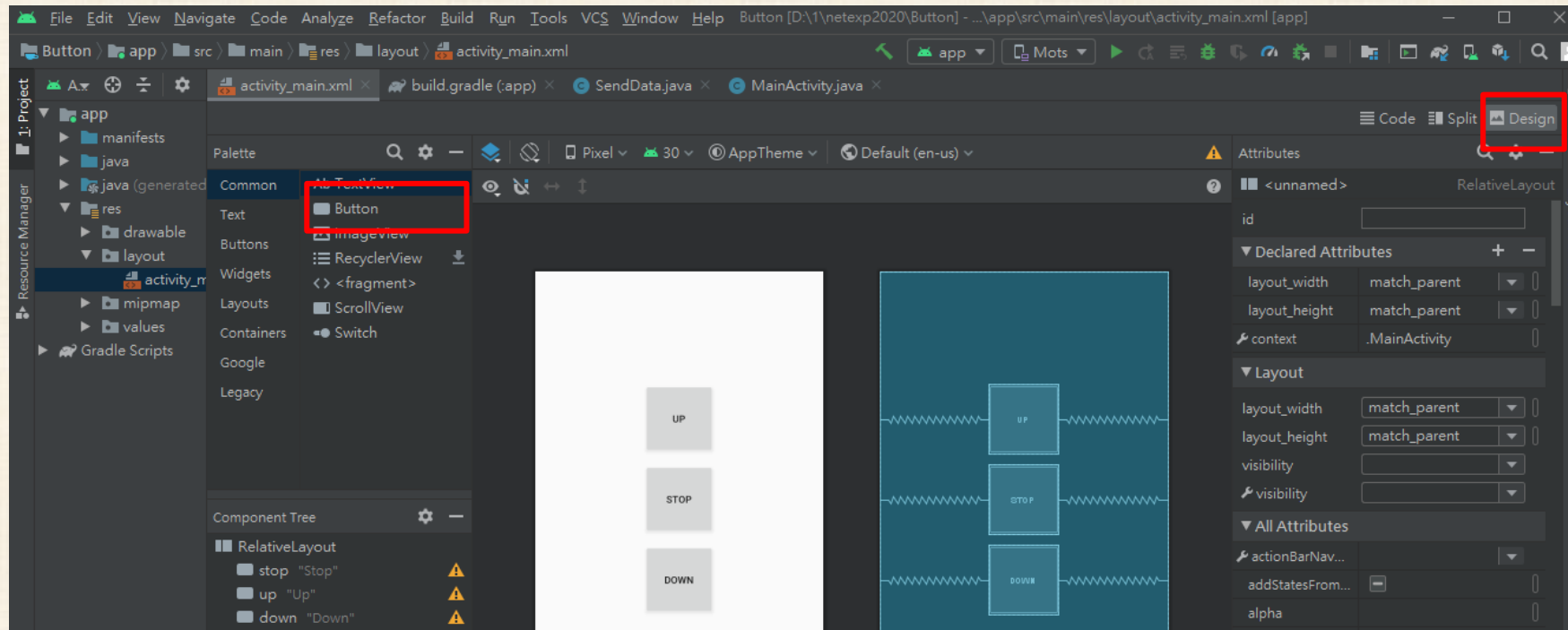
Q3

- 綜合Q1, Q2，在APP中加上“左, 右”按鈕，並在turtlebot_app.py加對應的移動程式，使Turtlebot可以全方位移動
- android studio layout 的.xml
 - 可以新增按鈕



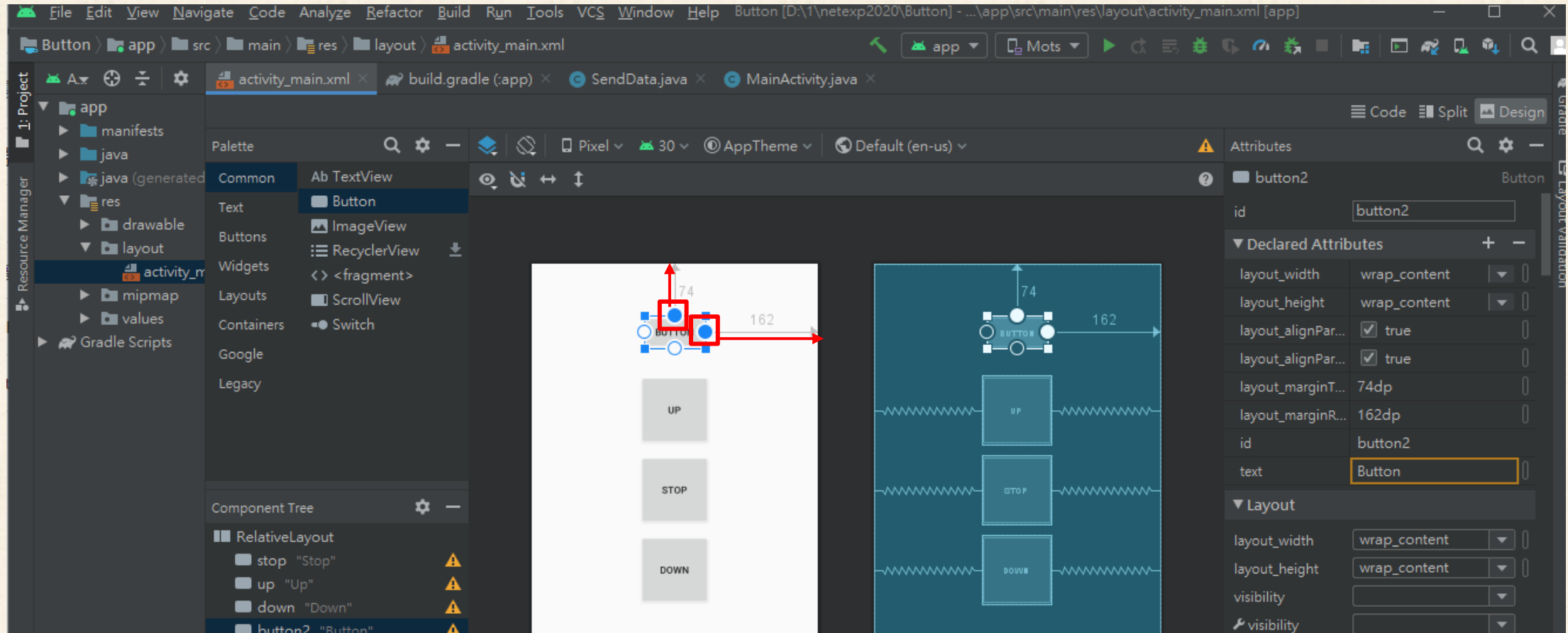
新增按鈕-1

- res > layout > activity_main.xml
- Design中可以按住拖動Button至頁面中，就會新增一個按鈕



新增按鈕-2

- 拖拉新按鈕x、y方向上端點至邊界，即可固定位置



新增按鈕-3

- 在 MainActivity.java 以及 activity_main.xml中增加對應動作 (觀察上、下按鈕)