

通訊網路實驗

Turtlebot3 應用

Dept. of Electrical and Computer Engineering (ECE)
National Yang Ming Chiao Tung University

Demo項目

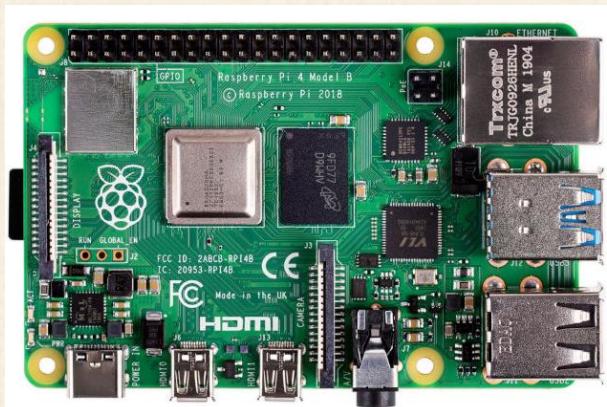
- 綜合之前上課內容，做出防碰撞機制
- 可用遠端Client操控機器人
- 雲端連結(下載turtlebot3_lab5)
 - [Google drive](#)
 - [OneDrive](#)

架構

Client

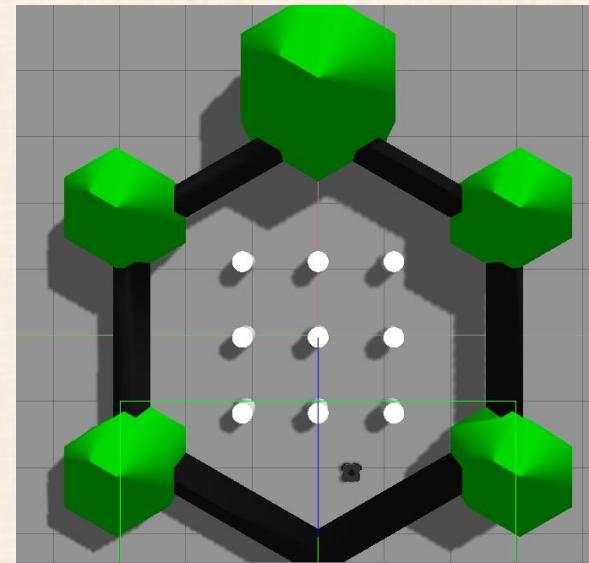


+



樹梅派控制機器人
同時超音波感測器監測距離

Server



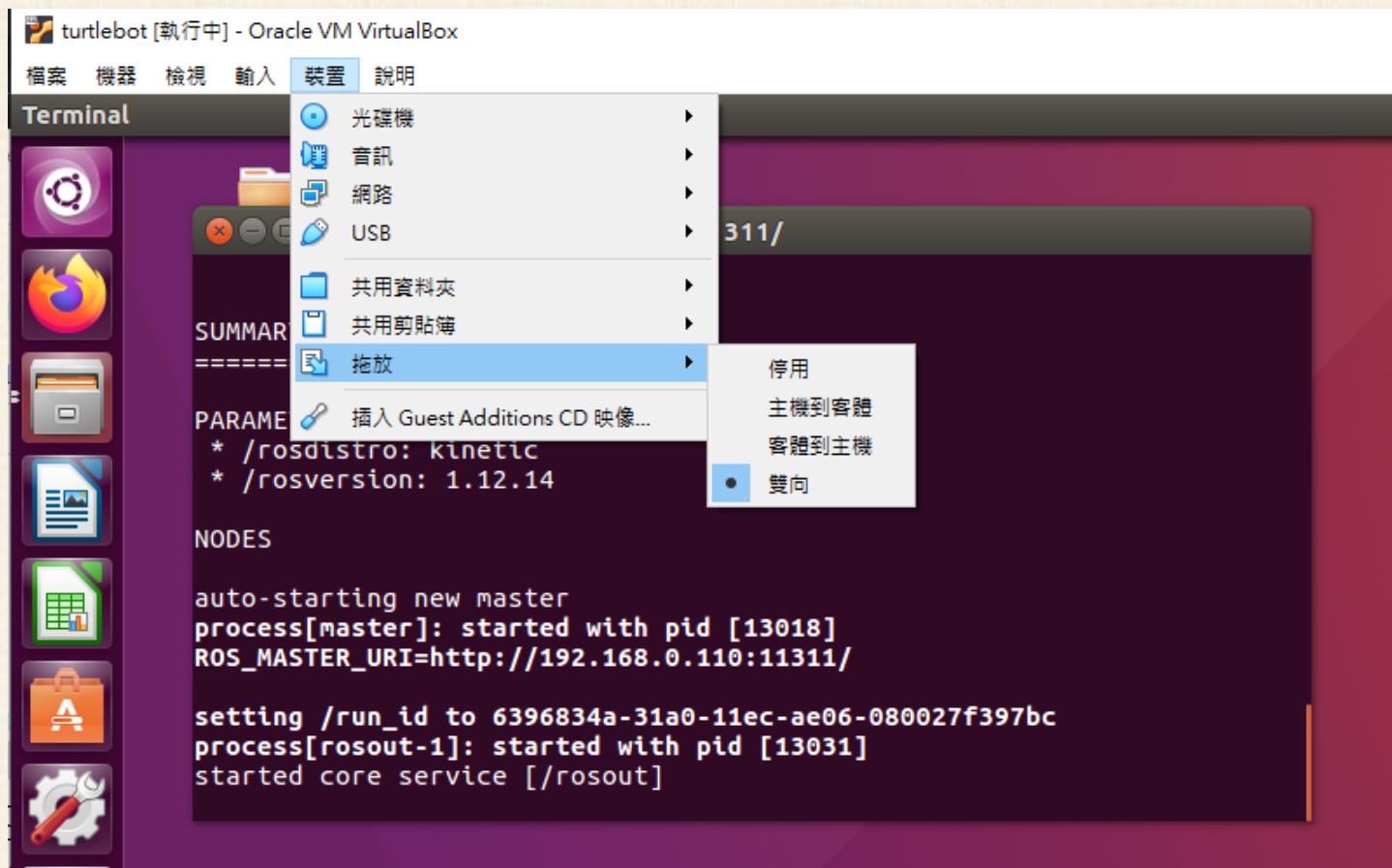
GAZEBO模擬器
(turtlebot)

socket通訊



VM 拖放功能

- 開啟此功能，可直接將檔案拖放至 node 資料夾



Q1

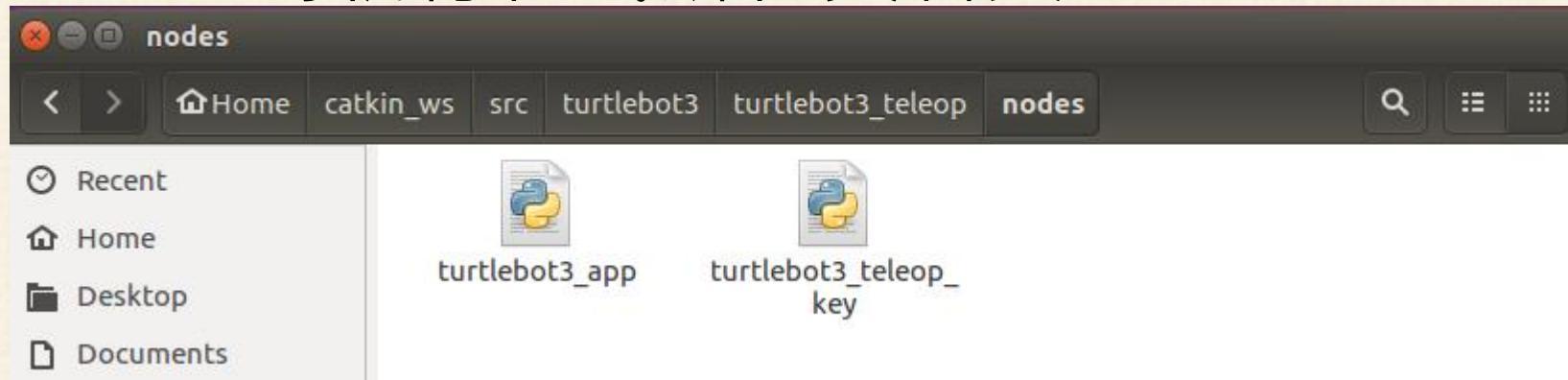
- 可兩人一組，共同設計
- 修改 `turtlebot3_lab5` 加入 Lab1 Q3的超音波code
- Server端設在VM(turtlebot也在VM)
- Client端 + 超音波感測器, 設在樹梅派
- 由Client端輸入 `w, a, s, d, x` 來控制機器人移動
同時超音波感測器持續監測距離
當距離小於10cm時(用手去擋)，傳送指令讓機器人停止

Server端

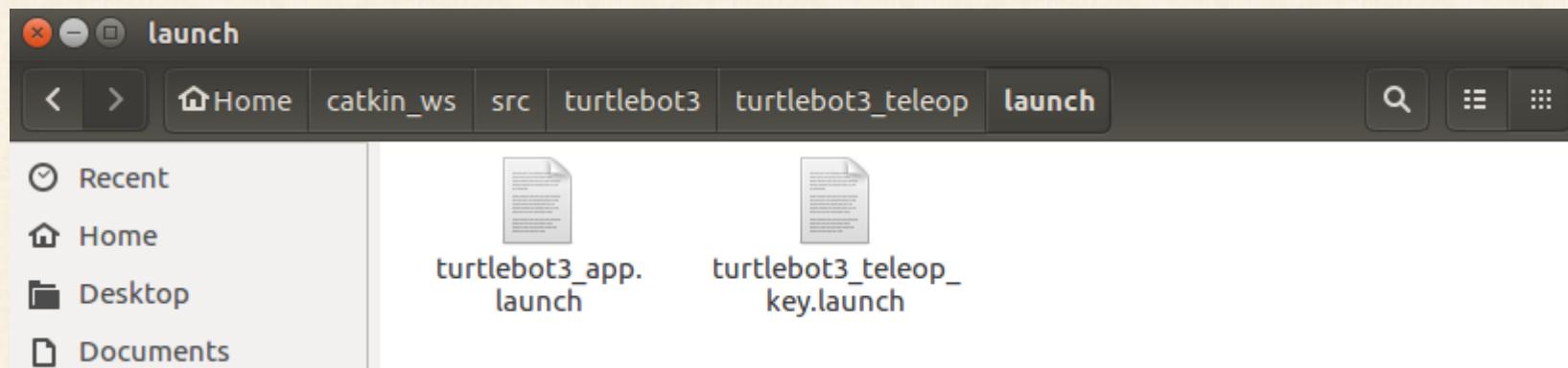
- 建立TCP Socket Server、接收資料、控制turtlebot
- 修改turtlebot3_lab5
 - (1)建立TCP Socket Server連線
 - (2)觀察 turtlebot3_lab5，在適當的位置，讓Server接收資料

Review-Node, launch 關係

- nodes 為執行程式所在資料夾

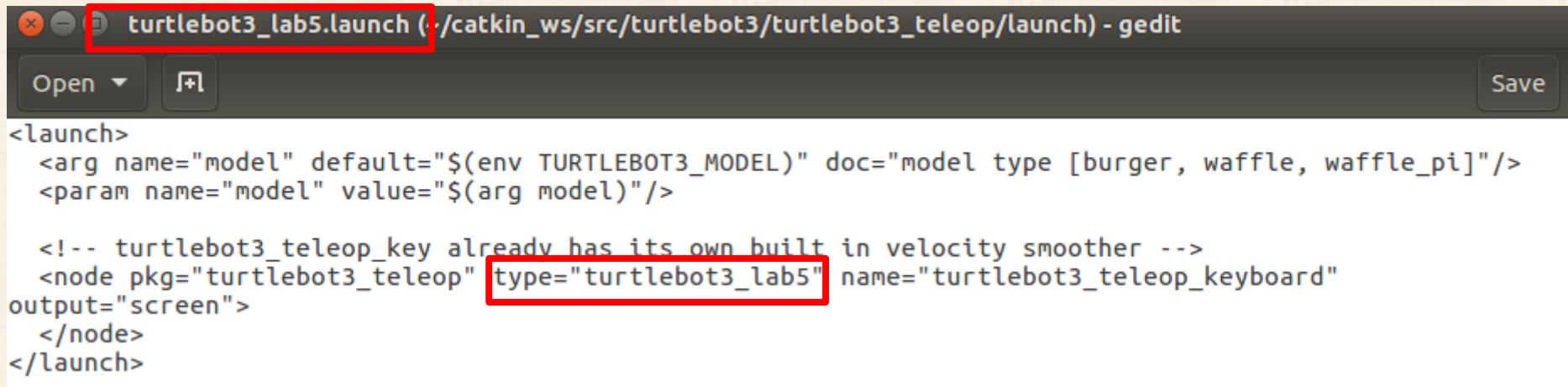


- launch 會呼叫node裡的程式



Node, launch

- 至雲端硬碟下載turtlebot3_lab5，放到node資料夾
- 複製一個新的launch檔，重新命名，更改圖中標示內容



```
<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_lab5" name="turtlebot3_teleop_keyboard"
output="screen">
  </node>
</launch>
```



修改執行權限

- cd 至 node 資料夾
- ls 查看資料夾內檔案
- 輸入 :`$ chmod 777 turtlebot3_lab5` 777：可讀可寫可執行 可參考[wiki](#)

```
turtlebot@turtlebot-VirtualBox: ~/catkin_ws/src/turtlebot3/turtlebot3_teleop/nodes
turtlebot@turtlebot-VirtualBox:~/catkin_ws/src/turtlebot3/turtlebot3_teleop/nodes$ ls
turtlebot3_app  turtlebot3_bot  turtlebot3_teleop_key
turtlebot@turtlebot-VirtualBox:~/catkin_ws/src/turtlebot3/turtlebot3_teleop/nodes$ █
```

```
turtlebot@turtlebot-VirtualBox: ~/catkin_ws/src/turtlebot3/turtlebot3_teleop/nodes
turtlebot@turtlebot-VirtualBox:~/catkin_ws/src/turtlebot3/turtlebot3_teleop/nodes$ ls
turtlebot@turtlebot-VirtualBox:~/catkin_ws/src/turtlebot3/turtlebot3_teleop/nodes$ chmod 777 turtlebot3_bot
turtlebot@turtlebot-VirtualBox:~/catkin_ws/src/turtlebot3/turtlebot3_teleop/nodes$ ls
turtlebot3_app  turtlebot3_bot  turtlebot3_teleop_key
turtlebot@turtlebot-VirtualBox:~/catkin_ws/src/turtlebot3/turtlebot3_teleop/nodes$ █
```

TCP Socket Server (1)

- 建立連線
 - `sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)`
 - `sock.bind((HOST,PORT))`
 - `sock.listen(5)` # 最多可接受多少socket連接
- 接受連線
 - `conn,addr=sock.accept()` # 接受連接，並回傳對象與IP
 - `conn.settimeout(0.5)` # 如timeout結束未收到data，會執行上一步驟
- HOST 為 Server 的 IP
- PORT 可使用1024~65535
- `sock`可自行命名，建立連線只需執行一次即可 (不用放到迴圈中)

TCP Socket Server (2)

- 設定timeout下, 接收資料

```
try:  
    # Receive data normally  
    msg=conn.recv(1024)  
    print(addr)  
    print(msg)  
    key = msg  
except socket.timeout:  
    # Timeout occurred, do something  
    continue
```

- 此處conn並非socket name，而是建立連線的對象

Client端

- 將Lab 1 Q3 的超音波程式加上 socket client內容 (要求連線、傳送資料)
加上 timeout input 來輸入指令
 - 如何持續輸入?
 - 如何持續監測距離?
 - 距離太短時，如何強制機器人停止?
- Hint : 可觀察Lab 4程式的Client socket 部分

TCP Socket Client

- 建立連線
 - `client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)`
 - `client.connect((HOST, PORT))`
- 傳送資料
 - `client.sendall(data)`

encode() : 把 str 轉換成 bytes.
decode() : 把 bytes 轉換成 str.
- HOST 為 Server IP
- PORT 需與Server 建立的 socket相同

Input timeout-1

- 為了使程式不會被input卡住，需使用有timeout的input方式
- 可在不中斷程式下，輸入值
- 用在超音波code(**Client 端**)中

```
1 import sys, select
2
3 while(1):
4     i, o, e = select.select([sys.stdin], [], [], 3)
5     if(i):
6         a = sys.stdin.readline().strip()
7         print(a, type(a))
8     else:
9         print ("Nothing")
10
```

```
turtlebot@turtlebot-VirtualBox:~$ ('s', <type 'str'>)
1 ('1', <type 'str'>)
Nothing
Nothing
```

Input timeout-2



```
1 import sys, select
2
3 while(1):
4     i, o, e = select.select([sys.stdin], [], [], 3)
5     if(i):
6         a = sys.stdin.readline().strip()
7         print(a, type(a))
8
9     else:
10        print ("Nothing")
```

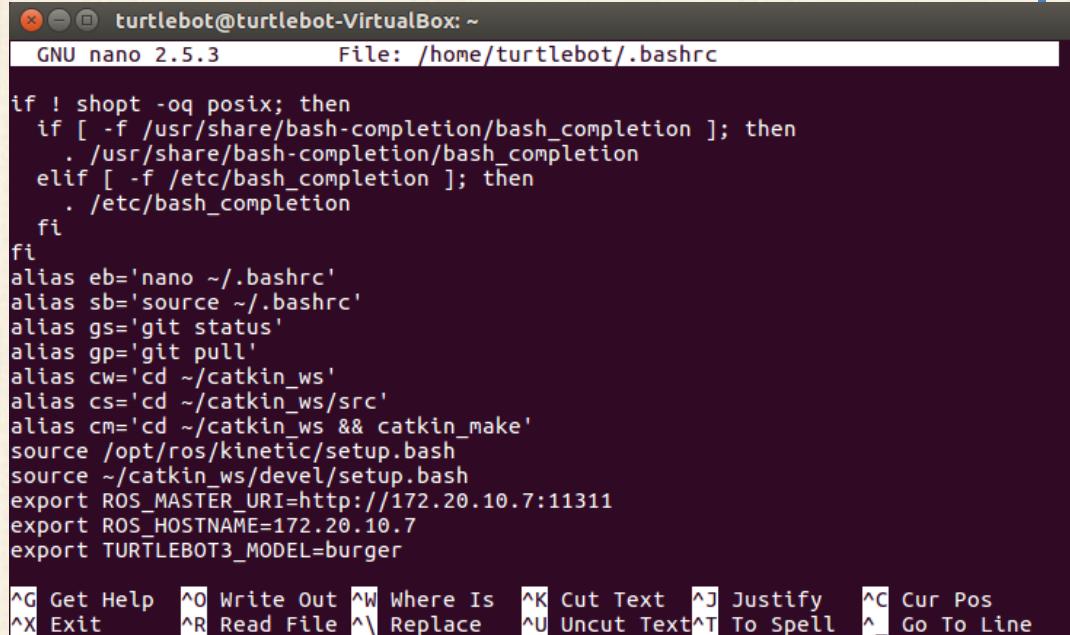
```
turtlebot@turtlebot-Virtu
s
('s', <type 'str'>)
1
('1', <type 'str'>
Nothing
Nothing
```

小叮嚀

- 偵測距離 or LED亮暗 的時間間隔 (`time.sleep(3)`)，請斟酌選擇，不要設太短，可以先設大一點，再慢慢調小測試
- 記得先啟動server，再啟動client
- address already in use : 更換 Server 跟 Client 的 PORT
- 終止程式指令 : `Ctrl + C`
如 `Ctrl + C` 無法終止，將Terminal 打叉關閉

Review - 改Turtlebot中的Master

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



```
turtlebot@turtlebot-VirtualBox: ~
GNU nano 2.5.3          File: /home/turtlebot/.bashrc

if ! shopt -q 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 ^L Go To Line
```

Review - Open simulator

- 桌面右鍵 > Open Terminal > 輸入

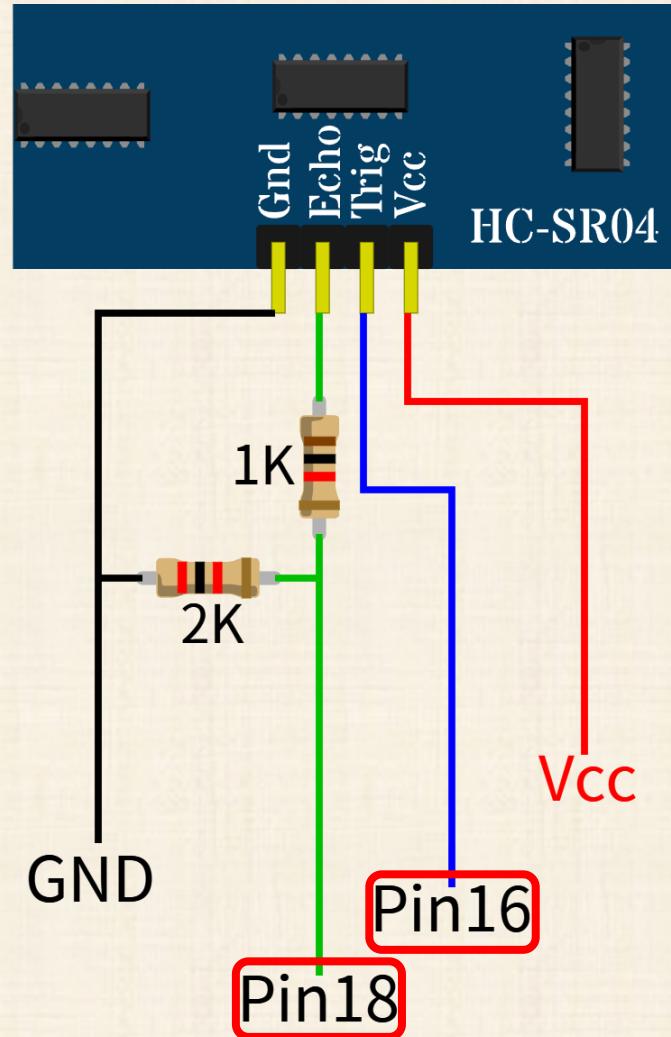
```
$ cd catkin_ws
```

```
$ 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
```

Review - 電路圖



1K:棕黑紅



2K:紅黑紅



Review - GPIO on Raspberry Pi

□ Z 字型的腳位編號

