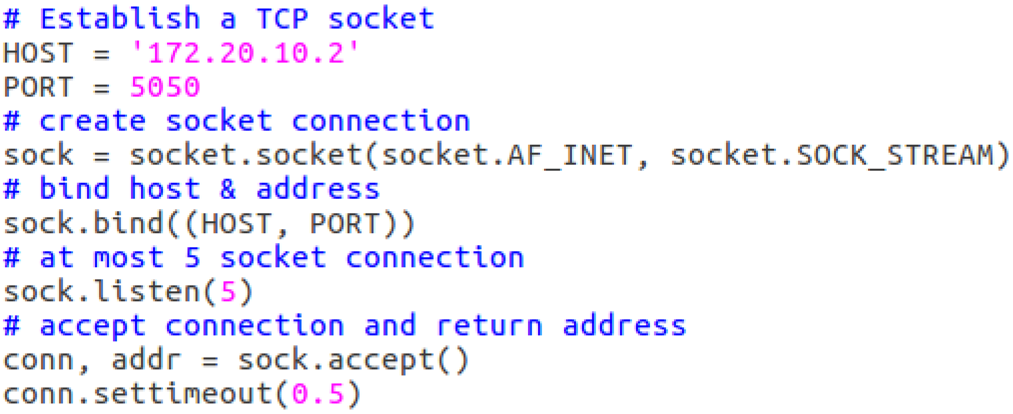
1. 實驗目的

學習如何用遠端 client 用 TCP socket 來連線，操作機器人

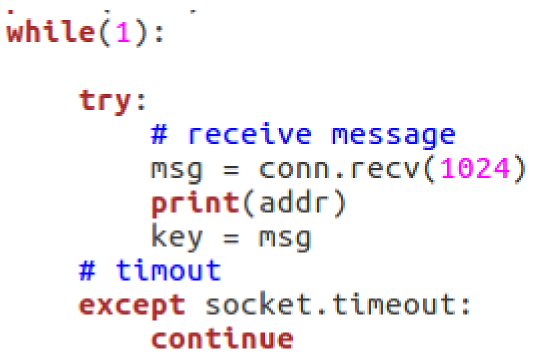
2. 實驗過程 (Code + 說明)

將 server 端設在 VM，client 端為樹莓派加上超音波感測器，用 TCP socket 將兩端建立連線，並透過 client 端來控制機器人的移動，用超音波感測器持續測量距離，控制機器人接收的指令。

Server 端



```
# Establish a TCP socket
HOST = '172.20.10.2'
PORT = 5050
# create socket connection
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# bind host & address
sock.bind((HOST, PORT))
# at most 5 socket connection
sock.listen(5)
# accept connection and return address
conn, addr = sock.accept()
conn.settimeout(0.5)
```

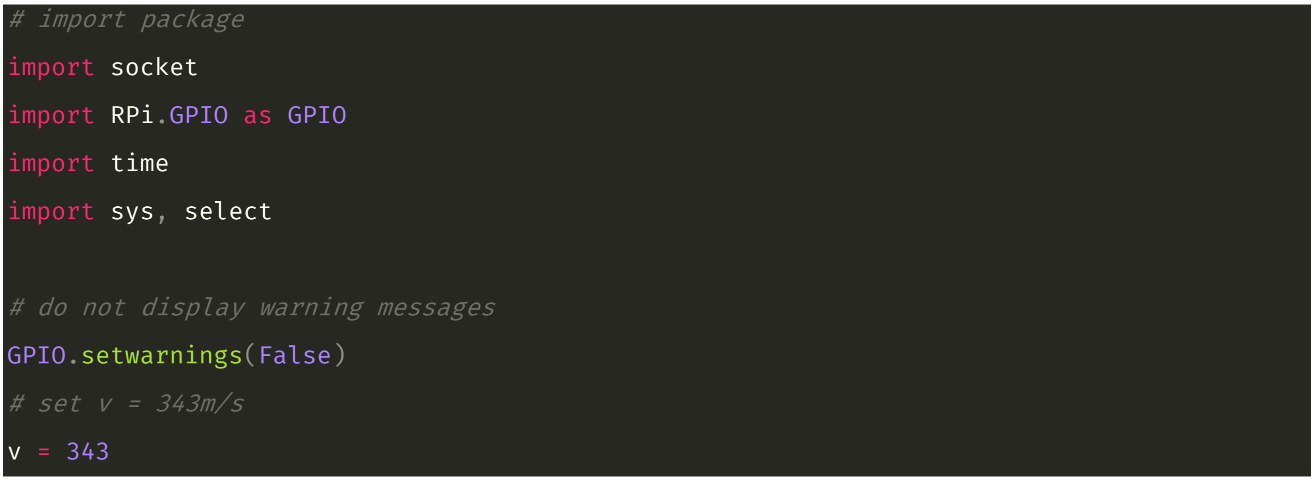


```
while(1):

    try:
        # receive message
        msg = conn.recv(1024)
        print(addr)
        key = msg
    # timout
    except socket.timeout:
        continue
```

其中 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)的 socket.AF_INET 為在伺服器之間串接，此處為 server 端和 client 端串接，socket.SOCK_STREAM 為使用 TCP 的方式提供可靠、雙向的通信頻道。

Client 端



```
# import package
import socket
import RPi.GPIO as GPIO
import time
import sys, select


# do not display warning messages
GPIO.setwarnings(False)
# set v = 343m/s
v = 343
```

```
# set pin
TRIG = 16
E = 18
LED_PIN = 12

HOST = "172.20.10.2"
PORT = 5050
client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.connect((HOST, PORT))

print('1')
# numbered in order according to GPIO pin
GPIO.setmode(GPIO.BOARD)
# set TRIG to GPIO.OUT
GPIO.setup(TRIG, GPIO.OUT)
# set E to GPIO.IN
GPIO.setup(E, GPIO.IN)
# set TRIG to a low logic level
GPIO.setup(TRIG, GPIO.LOW)
# set LED_PIN to GPIO.OUT
GPIO.setup(LED_PIN, GPIO.OUT)

def measure():
  # trigger a low logic level
  GPIO.output(TRIG, GPIO.HIGH)
  time.sleep(0.00001)
  GPIO.output(TRIG, GPIO.LOW)
  pulse_start = 0
  pulse_end = 0
  while GPIO.input(E) == GPIO.LOW:
    # start time
    pulse_start = time.time()
  while GPIO.input(E) == GPIO.HIGH:
    # time received from the echo
    pulse_end = time.time()
  # duration from emission to receiving the echo
  t = pulse_end - pulse_start
  # distance = time * velocity
  d = t * v
  # one-way distance
  d = d / 2
  # m to cm
  d = d * 100
  return d

while(1):
  d = measure()
  # output distance(cm)
  print d
  i, o, e = select.select([sys.stdin], [], [], 3)

  # if get message
  if(i):
    # read command
    key = sys.stdin.readline().strip()
    print(key, type(key))
    # if distance < 10cm
    if(d < 10):
      # send s, which means stop
      client.sendall('s')
    else:
      # else send command
      client.sendall(key)
  else:
    key = ''
    print("Nothing")

  # LED exhibits different reactions based on varying distances
  if(d < 10):
    print ("on")
    client.sendall('s')
    GPIO.output(LED_PIN, GPIO.HIGH)
  elif(d < 20):
    print("shine")
    GPIO.output(LED_PIN, GPIO.HIGH)
    time.sleep(0.1)
    GPIO.output(LED_PIN, GPIO.LOW)
    time.sleep(0.1)
  else:
    print("off")
    GPIO.output(LED_PIN, GPIO.LOW)
  time.sleep(1)

GPIO.cleanup()
```

i, o, e = select.select([sys.stdin], [], [], 3)，代表超時秒數設為 3，如果 3 秒內輸入列表裡的物件都沒有動靜就繼續往下執行。

3. 問題與解法

做到最後發現指令都有收到，機器人也會正確動，但 LED 燈卻怎樣都不會亮，最後發現是我自己接錯電路了

4. 心得

這次實驗也不難，把第一個 lab 的 q3 的 code 稍微修改一下就能完成了。