靜宜大學畢業專題　物聯網應用



指導老師：羅峻旗

組員：資工四Ａ　洪祐原

　　　資工四Ｂ　林晉玄

　　　資工四Ｂ　蔡冠玗

資工四B 鄭祐鈞

參展日期：108/12/05

參展地點：主顧樓大廳

目錄

壹． 前言

貳． 系統功能與特色介紹

1. 系統功能
2. 使用對象
3. 使用環境
4. 語音辨識APP 控制使用流程
5. 開發工具軟體及零件
6. 開發工具軟體
7. 垃圾桶開發所需零件及材料

肆． 系統畫面

1. 自走車樣貌
2. APP使用畫面
3. 系統程式碼
4. 結論與未來展望
   * 1. 結論
     2. 未來展望
5. 參考資料
6. 前言

　　在現代這個科技進步飛快的時代，自己家中也似乎跟著這個熱潮科技化了起來，各種智慧家電陸續推出，從燈泡、冷氣、到電鍋、空氣清淨機、智慧音箱等，各種家電都能智慧控制，更符合人們的使用習慣，能夠自動或是統一控制家電們，讓人們生活更加的便利。

　　而我們也想跟隨時代腳步，一邊學著物聯網的技術以便在未來職場使用、一邊創造屬於我們的行動物聯網。我們選擇「垃圾桶」為主題做發想。

1. 系統功能及特色介紹

**系統功能：**

* 1. 手機能連線垃圾車進行操控。
  2. 透過語音辨識控制垃圾車行動及開機。
  3. 辨識特定的圖片，做出相對應的行動。看到pizza將會前進，看到stop\_sign將會停止。

**系統特色：**

* 1. 有別於一般垃圾桶，智能垃圾桶將會給予回饋。
  2. 使用者可以透過語音及影像辨識的方式，控制垃圾車行動
  3. 由於raspberry pi 3b+ 本身硬體限制在跑YOLO物件辨識時需要半分鐘至一分鐘的時間，透過Darknet with NNPACK使其優化，使用四顆CPU同時進行辨識，讓辨識時間縮短為兩秒內。

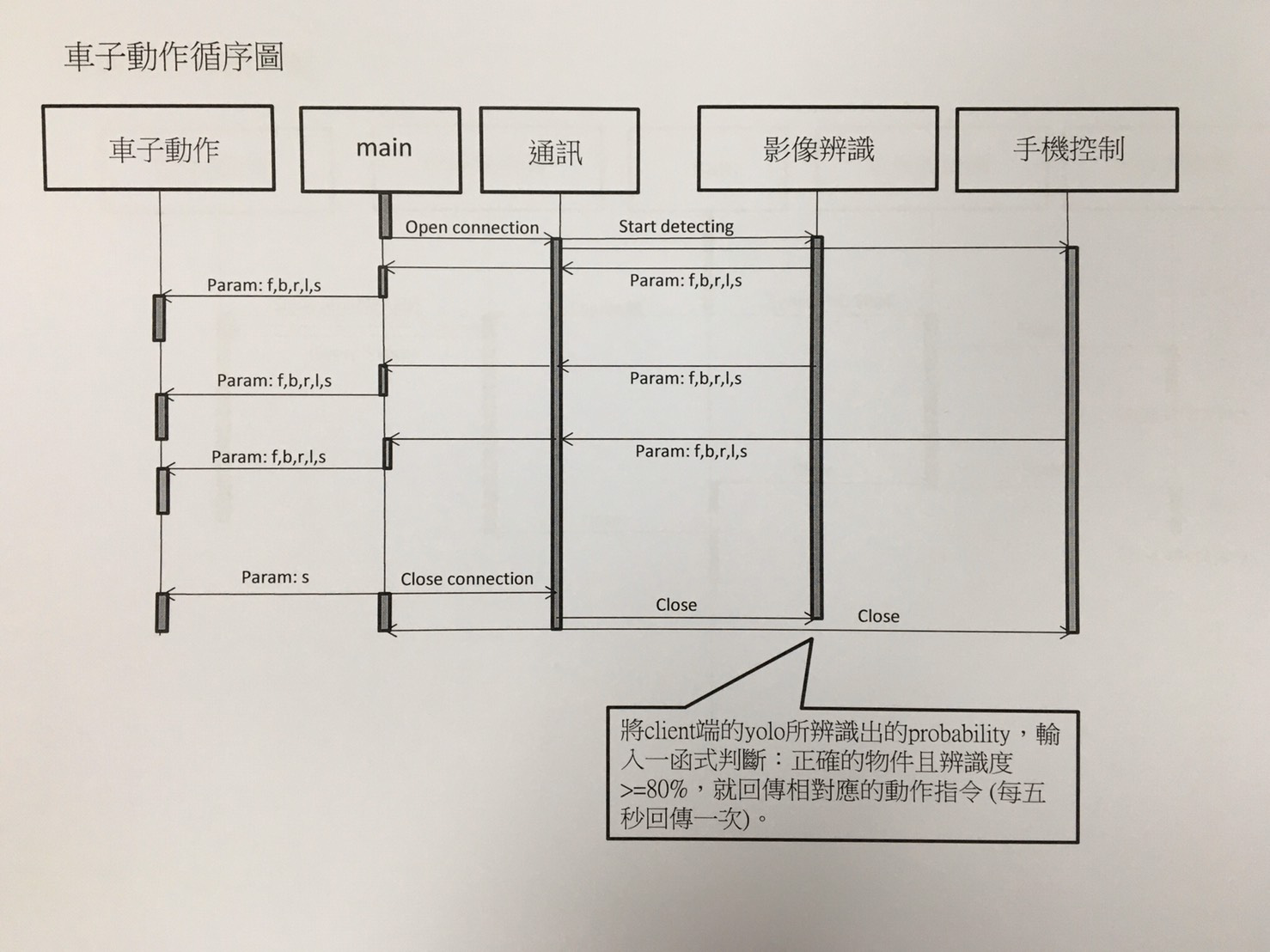
**使用對象：**

一般家庭或是上班族。

**使用環境：**

坪數約10坪內且地板平坦的空間

**系統流程：**



1. 開發工具軟體及零件
2. **開發工具軟體：**
3. 開發硬件：raspberryPI、L298N、raspberry Camera
4. 開發環境：pycharm、APP inventor2
5. 應用套件：picamera、Darknet with NNPACK
6. **產品開發所需材料：**
7. 外殼：

自走車套件組、鐵絲、壓克力板

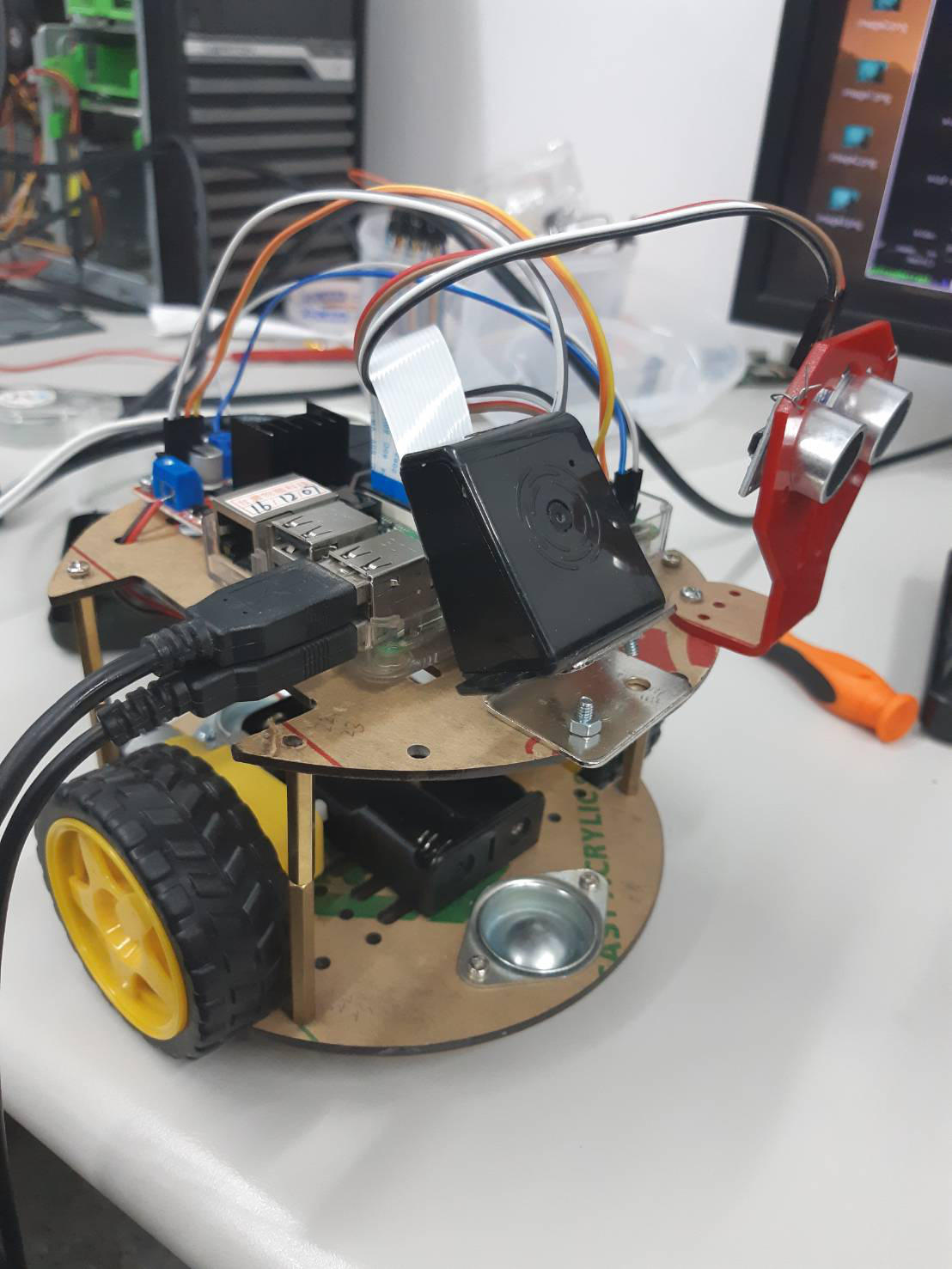
1. 區控元件：

RaspberryPi、L298N

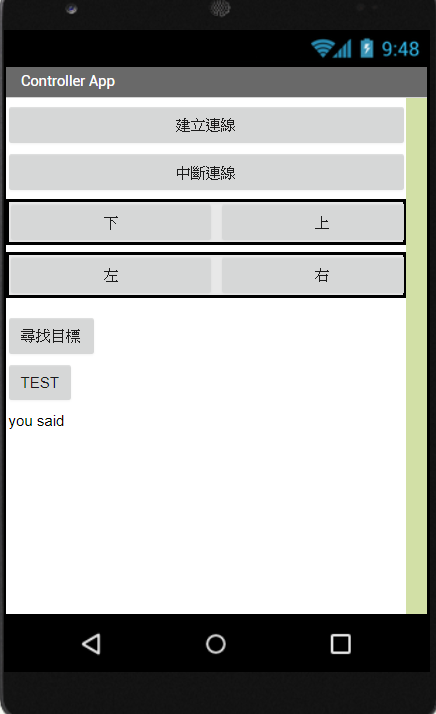
1. 其餘零件：

充電電池18650號\*2、電池供電座

1. 系統畫面
2. **垃圾桶自走車樣貌**



1. **App使用畫面**

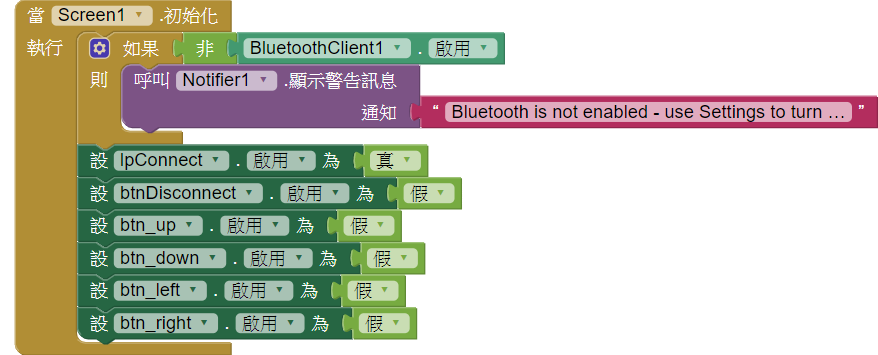
手機端為垃圾桶主要控制端。

與垃圾桶連線後，垃圾桶將會偵測周遭的環境，依指定定點前往該處。

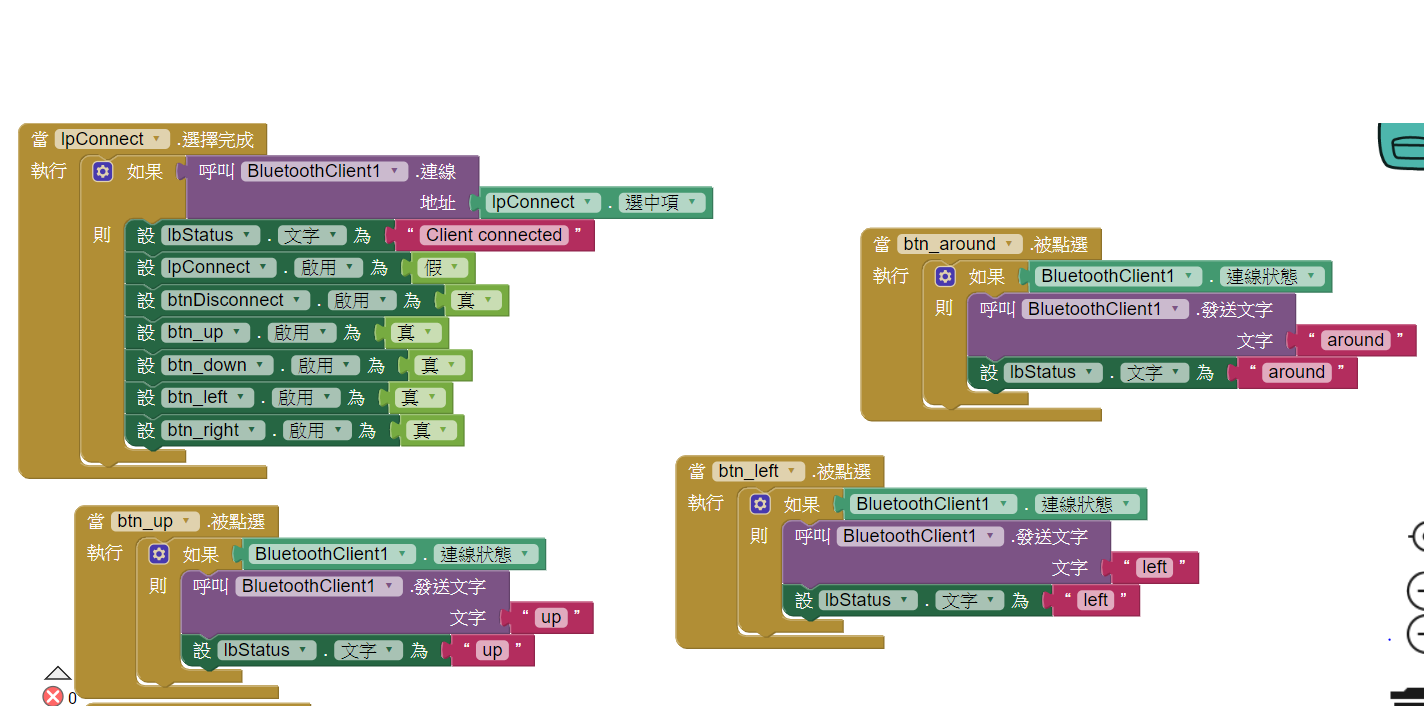
1. **系統程式碼**

|  |
| --- |
| import uuid  from bluetooth import \*  #import bluetooth  import RPi.GPIO as GPIO  import time  from time import sleep  #LED\_PIN = 18  import sys  import subprocess  from picamera import PiCamera  def main():  global Motor\_L1\_Pin  global Motor\_L2\_Pin  global Motor\_R1\_Pin  global Motor\_R2\_Pin  global camera  camera = PiCamera()  Motor\_L1\_Pin = 17  Motor\_L2\_Pin = 18  Motor\_R1\_Pin = 20  Motor\_R2\_Pin = 21  GPIO.setmode(GPIO.BCM)  GPIO.setup(Motor\_R1\_Pin, GPIO.OUT)  GPIO.setup(Motor\_R2\_Pin, GPIO.OUT)  GPIO.setup(Motor\_L1\_Pin, GPIO.OUT)  GPIO.setup(Motor\_L2\_Pin, GPIO.OUT)  #server\_socket=BluetoothSocket(RFCOMM)  server\_socket=BluetoothSocket(RFCOMM)  server\_socket.bind(("", PORT\_ANY))  server\_socket.listen(1)  port = server\_socket.getsockname()[1]  service\_id = str(uuid.uuid4())  advertise\_service(server\_socket, "LEDServer",service\_id = service\_id,service\_classes = [service\_id, SERIAL\_PORT\_CLASS],profiles = [SERIAL\_PORT\_PROFILE])  global client\_socket  try:  print("按下 Ctrl-C 可停止程式")  while True:  print('等待 RFCOMM 頻道 {} 的連線'.format(port))  client\_socket, client\_info = server\_socket.accept()  print('接受來自 {} 的連線'.format(client\_info))  try:  while True:  movement = client\_socket.recv(1024).decode().lower()  if len(movement) == 0:  break  else:  car\_move(movement)  except IOError:  pass  client\_socket.close()  print('中斷連線')  except KeyboardInterrupt:  GPIO.output(Motor\_L1\_Pin, GPIO.LOW)  GPIO.output(Motor\_L2\_Pin, GPIO.LOW)  GPIO.output(Motor\_R1\_Pin, GPIO.LOW)  GPIO.output(Motor\_R2\_Pin, GPIO.LOW)  print('中斷程式')  finally:  if 'client\_socket' in vars():  client\_socket.close()  server\_socket.close()  GPIO.cleanup()  print('中斷連線')  def camera\_stream():  print('take pic')  camera.resolution = (1024, 768)  camera.start\_preview()  camera.capture('/home/pi/darknet-nnpack/data/trash\_image.png')  camera.stop\_preview()  def enable\_object\_detection():  camera\_stream()  print('detecting')  result = subprocess.run(  ['./darknet', 'detect', 'cfg/yolov2-tiny.cfg', 'yolov2-tiny.weights', 'data/trash\_image.png'],  stdout=subprocess.PIPE)  v\_result = result.stdout.decode('utf-8')  print(v\_result)  p\_result = process\_result\_to\_car\_move(v\_result)  return p\_result  def process\_result\_to\_car\_move(v\_result):  count\_nextline = 0  for i in range(len(v\_result)):  if v\_result[i] == '\n':  count\_nextline += 1  if count\_nextline >= 1 and (i+3) < len(v\_result) and v\_result[i] == '\n':  print(v\_result[i+1:i+6]) == 'pizza'  if v\_result[i+1:i+6] == 'pizza':  GPIO.output(Motor\_L1\_Pin, GPIO.LOW)  GPIO.output(Motor\_L2\_Pin, GPIO.HIGH)  GPIO.output(Motor\_R1\_Pin, GPIO.LOW)  GPIO.output(Motor\_R2\_Pin, GPIO.HIGH)  elif v\_result[i+1:i+5] == 'stop':  return v\_result[i+1:i+5]  def car\_move(movement):  p\_move = ''  print('do func car\_move')  if movement == 'search':  print(movement)  while (p\_move != 'stop'):  GPIO.output(Motor\_L1\_Pin, GPIO.HIGH)  GPIO.output(Motor\_L2\_Pin, GPIO.LOW)  GPIO.output(Motor\_R1\_Pin, GPIO.LOW)  GPIO.output(Motor\_R2\_Pin, GPIO.HIGH)  p\_move = enable\_object\_detection()  car\_move('stop')  elif movement == 'up':  print(movement)  GPIO.output(Motor\_L1\_Pin, GPIO.LOW)  GPIO.output(Motor\_L2\_Pin, GPIO.HIGH)  GPIO.output(Motor\_R1\_Pin, GPIO.LOW)  GPIO.output(Motor\_R2\_Pin, GPIO.HIGH)  elif movement == 'right':  print(movement)  GPIO.output(Motor\_L1\_Pin, GPIO.HIGH)  GPIO.output(Motor\_L2\_Pin, GPIO.LOW)  GPIO.output(Motor\_R1\_Pin, GPIO.LOW)  GPIO.output(Motor\_R2\_Pin, GPIO.LOW)  elif movement == 'left':  print(movement)  GPIO.output(Motor\_L1\_Pin, GPIO.LOW)  GPIO.output(Motor\_L2\_Pin, GPIO.LOW)  GPIO.output(Motor\_R1\_Pin, GPIO.HIGH)  GPIO.output(Motor\_R2\_Pin, GPIO.LOW)  elif movement == 'stop':  print(movement)  GPIO.output(Motor\_L1\_Pin, GPIO.LOW)  GPIO.output(Motor\_L2\_Pin, GPIO.LOW)  GPIO.output(Motor\_R1\_Pin, GPIO.LOW)  GPIO.output(Motor\_R2\_Pin, GPIO.LOW)  else:  print('未知的指令: {}'.format(data))  if \_name\_ == '\_main\_':  main() |

1. **App程式碼**







1. 結論及未來展望

**結論：**

物聯網是個趨勢，在這次專題當中我們學習到了很多，從樹梅派的應用、Python的學習及撰寫、垃圾桶的影像辨識處理、網路連接協定、垃圾桶的硬體設計…等等。很多技術我們都是第一次，從無生有的能力也慢慢地在這次專題培養了出來，很多事情只要付出時間、努力，豐碩的果實當然是自己的，這次我們堅持了!完成屬於自己的物聯網應用-垃圾桶。

**未來展望：**

未來我們會強化外觀跟硬體的設計達到更優化，在載重上我們想要替換更適合的硬體，讓垃圾桶能乘載更多的垃圾、機動性能更高。

1. 參考資料

Darknet with NNPACK:

<https://github.com/shizukachan/darknet-nnpack>