

# OpenPiCar

OpenCV 搭配 Raspberry Pi 與 Arduino

實現純視覺自駕車

作者：崔詠荃

## 一、 摘要

本專題以電腦視覺為基礎，採用 Raspberry Pi 為運算中心，透過 OpenCV 進行標線辨識，結合 Arduino Nano 執行機電自動控制，達成具備基礎車道維持輔助系統（LKA）之能力。

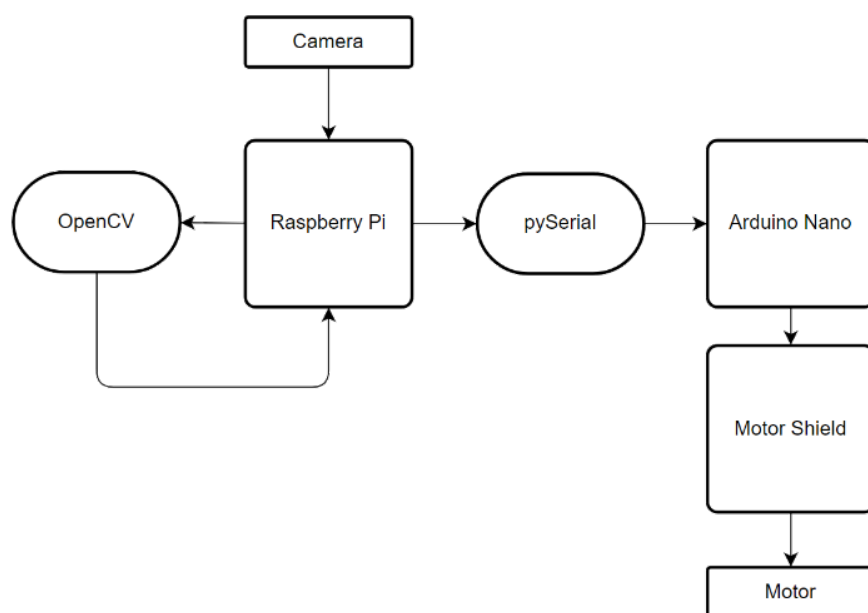
## 二、 動機及目的

以往自駕車之輔助自動駕駛能力需透過 LiDAR、攝影機、遠紅外線感應器、超音波、毫米波雷達等感應器進行 Sensor Fusion。作者過去製作之自走車尋跡功能也以紅外線、超音波感應器實現。但近年來電動車廠 Tesla 逐步向純視覺靠攏，以多顆鏡頭結合神經網路、電腦視覺實現自動輔助駕駛系統。

實務上運行自動駕駛之硬體成本高昂，故本專題目的為：使用較易取得之 Arduino 與 Raspberry Pi，嘗試以電腦視覺取代感應器，製作一自走車實現基礎自動駕駛，並具備車道維持輔助系統（LKA）與 U 型、S 型連續彎道之過彎能力。

## 三、 架構及軟、硬體介紹

### (一) 架構



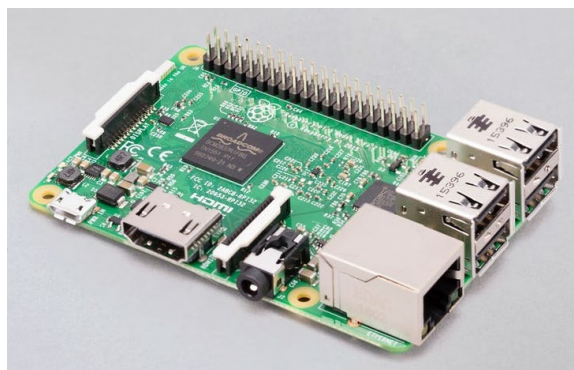
圖一、架構圖

### (二) 軟、硬體簡介

#### 1. Raspberry Pi 3B+

英國樹莓派基金會推出之小型單板電腦，具低成本、模組化與開放式優點，及影像、聲音之輸出 I/O、GPIO、USB、RJ45 等接口，

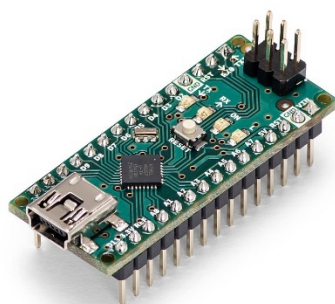
可應用於機器人、伺服器、機電控制等廣泛領域。在本專題中，其負責運行 OpenCV，將攝影機捕捉之影像經多道處理程序轉為道路偏移率與彎道曲率，藉序列埠 (Serial port) 傳送予 Arduino Nano。



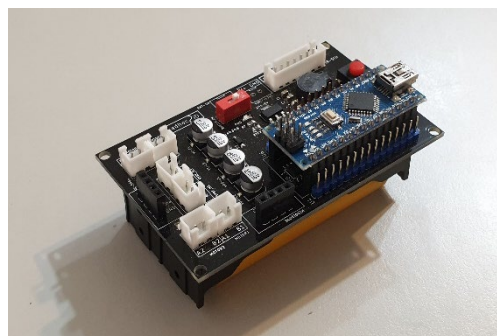
圖二、Raspberry Pi 3B+

## 2. Arduino Nano

Arduino 公司基於 Arduino Uno 板改良之迷你嵌入式硬體平台，可透過 C/C++ 語言控制數位及類比 I/O。在本專題中負責接收 Serial port 之道路偏移率與彎道曲率，另搭配 Nano 板專用之馬達驅動版 (Motor Shield) 實現四顆 N20 減速馬達之控制。



圖三、Arduino Nano



圖四、Arduino Nano 與馬達驅動板

## 3. OpenCV

全稱為 Open Source Computer Vision Library，為一跨平台、開源影像處理函式庫。本專題中於 Raspberry Pi OS 中執行 Python 版本，將攝影機辨識之道路影像經由多道處理程序，轉為道路偏移率與彎道曲率（將在後續說明）。

## 4. pySerial

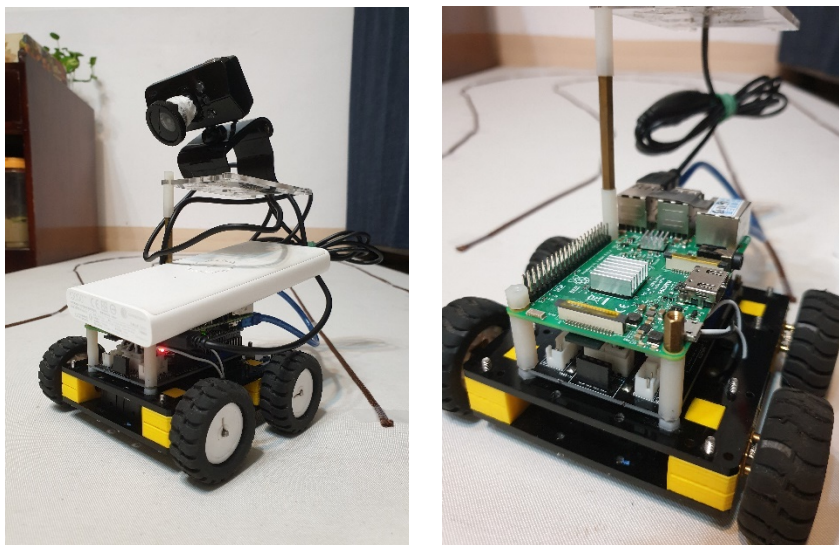
為一 Python 序列埠擴展函式庫，其提供初始化序列埠、傳送和接收序列數據之指令。因 Raspberry Pi 端 OpenCV 程式使用 Python 撰寫，Arduino 端使用 C++ 撰寫，故選擇使用 pySerial 作為兩者間溝通中介。

## 四、製作過程

### (一) 車體結構

自駕車車體為五層結構，底層為四顆 N20 減速馬達及電池空間，馬達分左右兩側控制。接著依序放置 Arduino Nano 及 Motor Shield、Raspberry Pi、行動電源空間，最上層架設攝影機。由於車輛高度較低，且攝影機之可視角較窄，故以銅柱增高，並加裝超廣角鏡片增加可視範圍。

因四顆 N20 減速馬達之工作電壓需 12V 以上，故 Arduino Nano 及驅動板使用兩顆 18650 電池額外供電，Raspberry Pi 及攝影機由行動電源供電。得益於 Raspberry Pi 具備 Wi-Fi 網卡，車輛得以連接至網路，並啟用 VNC 於遠端監控車輛動態、修改程式。

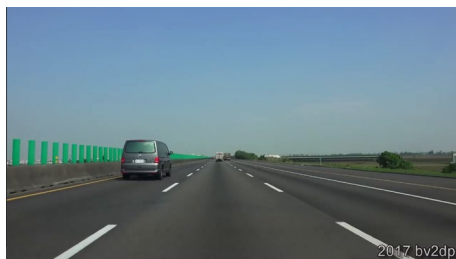


圖五、自駕車概覽

### (二) OpenCV (電腦視覺)

在 Raspberry Pi 上執行之 OpenCV 程式可大略分為下列步驟：

1. 擷取攝影機端單幀圖像傳至 Raspberry Pi。
2. 使用 Canny Edge Detection 將影像梯度計算邊緣強度，轉為二進制後輸出影像邊緣



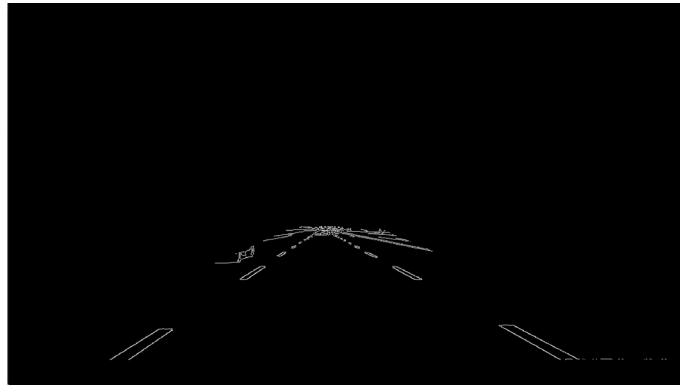
圖六、原影像



圖七、Canny Edge Detection

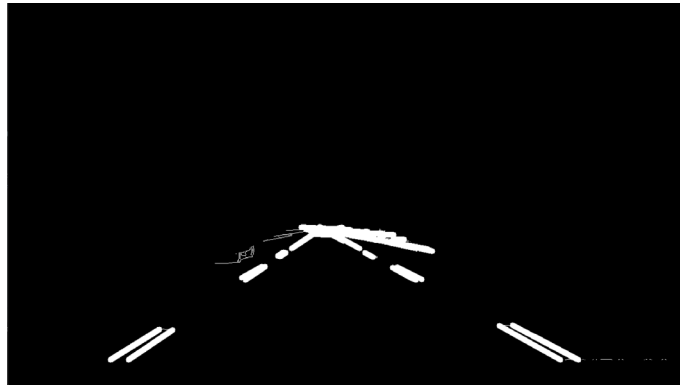
3. 定義興趣區域 (Region of Interest, 簡稱 ROI)，標示出所需辨識

之道路大致區域。



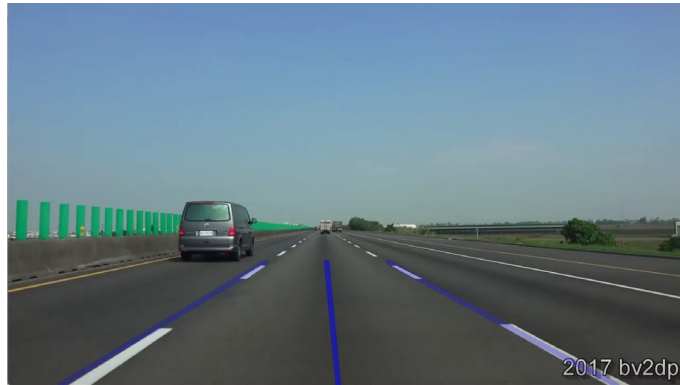
圖八、Region of Interest

4. 將圖像進行 Hough Line Transform，其運用線與點的對偶性，將原始圖像中的線轉換為參數空間的點，並通過表決尋找最大峰值。



圖九、Hough Line Transform

5. 將線依據斜率及平面座標區域分為左右兩組，分別嘗試擬合兩組，形成兩條車道線。
6. 判斷總線數，若結果為兩條直線，則取兩線之平均中線並計算偏移率；若為一條直線，則取該直線斜率回傳其為偏移線，並計算偏移率。得出之道路偏移率以 pySerial 函式寫入至序列埠。
7. 將道路標線之中線或偏移線標示回原影像中並顯示辨識結果。



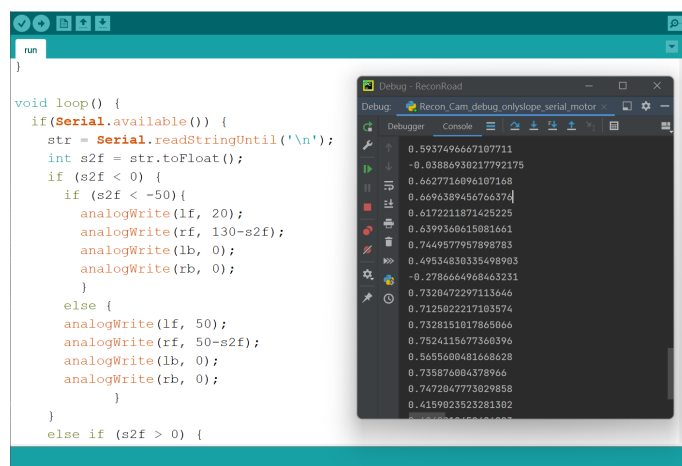
圖十、將道路標線標示回原影像

8. 重複上述步驟即得到影片。

### (三) Arduino (馬達控制)

Arduino 端根據接收到的道路偏移率調整馬達輸出，修正車輛行駛軌跡。可大略分為下列步驟：

1. 接收序列埠之道路偏移率。
2. 依據道路偏移率修正左右馬達類比訊號輸出。
3. 轉彎時若失去單側標線，則將成功辨識之直線斜率視為曲線切線斜率，增加單側馬達類比訊號輸出。

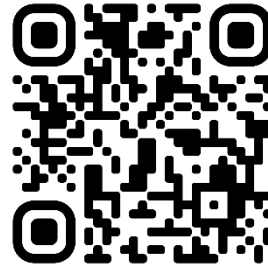


圖十一、Arduino 端程式及序列埠

## 五、 成果展示



自駕車行駛影片  
(可點擊或掃描開啟)



完整程式  
(可點擊或掃描開啟)

## 六、 結論與未來展望

本專題展示了如何使用 Raspberry Pi 應用 OpenCV、結合 Arduino Nano 實現道路辨識與標示，並可使車輛判斷自身相對道路之位置並加以修正，也實現了直線、S 彎、U 彎等道路駕駛。

由於本專題僅由筆者自行製作而無指導老師，在製作過程中花費了較多時間在克服癥結。在執行 Hough Line Transform 時，需修改 threshold, minLineLength, maxLineGap 參數，必須依據環境因素調整至適當數值以獲取最佳道路標線。專題製作之初原先計畫以無線方式將攝影機影像傳至遠端電腦運算，再傳送回 Arduino。偏移率傳遞可經由藍牙實現，但影像流量最少達每秒 5MB，若以 nRF24L01 天線為例不過 2Mbps 速率，故最終選擇使用 Raspberry Pi 在自駕車上運算。再者，受車輛大小限制，鏡頭視野相對真實汽車受限，因此最終決定將道路寬度縮減。

若硬體允許，未來預計加入多顆攝影機以達成道路建模、自動變道、距離判定煞停等功能，也將結合機器學習如 TensorFlow 實現交通號誌、車輛動態辨識。

從製作前的資料蒐集、過程中解決問題，到成功使車輛自主駕駛，不但學習到電腦視覺與自動駕駛的基礎知識，也證實即便去除紅外線、超音波、雷達等感應器，僅依靠電腦視覺也能使車輛自動駕駛。本專題作為自駕車之基礎，自許未來能整合多面向應用於自駕車領域，使車輛實現真正的自動駕駛的同時降低門檻，並回饋於社會。