

驢車 (Donkey Car) - 基於 Raspberry Pi 與機器學習 的自走車專案介紹

台灣樹莓派 <sosorry@raspberrypi.com.tw>
2019/09/20 @PyCon2019

CC (Creative Commons)

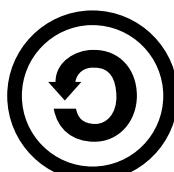
姓名標示 — 非商業性 — 相同方式分享



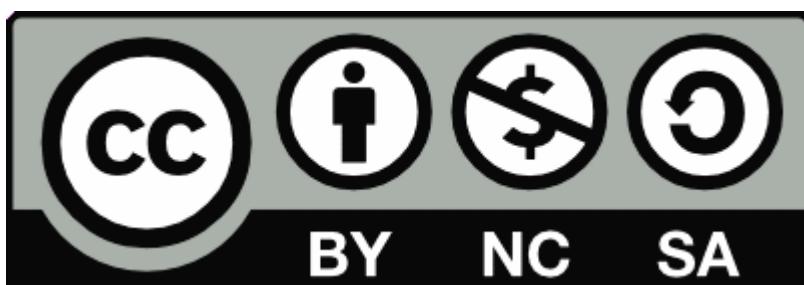
姓名標示 — 你必須給予 適當表彰、提供指向本授權條款的連結，以及 指出（本作品的原始版本）是否已被變更。你可以任何合理方式為前述表彰，但不得以任何方式暗示授權人為你或你的使用方式背書。



非商業性 — 你不得將本素材進行商業目的之使用。



相同方式分享 — 若你重混、轉換本素材，或依本素材建立新素材，你必須依本素材的授權條款來散布你的貢獻物。



關於我們

- Raspberry Pi 官方經銷商



- 專注 Raspberry Pi 應用與推廣，舉辦社群活動



本次主題

- 什麼是 Donkey Car?
 - 所需硬體
 - 軟體架構
- 使用機器學習打造自動駕駛系統
- Hack Donkey Car
- 學習資源

什麼是 Donkey Car?

從 DIY Robocars 辦的聚會開始



<https://www.meetup.com/DIYRobocars/events/234581265/>

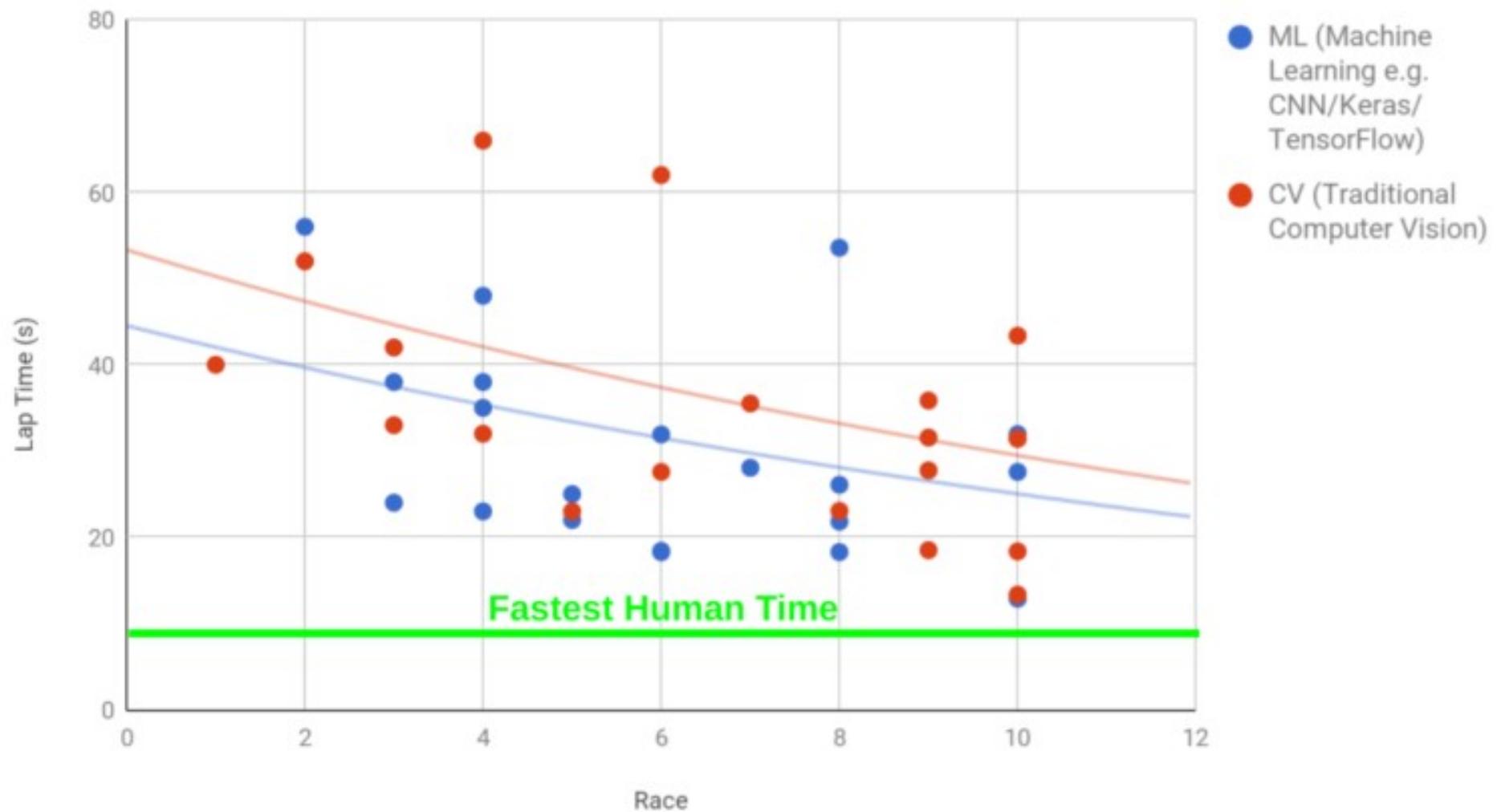


Will Roscoe
Software

Adam Conway
Hardware

ML 接近人類的開車行為

1/10 Scale Race Results



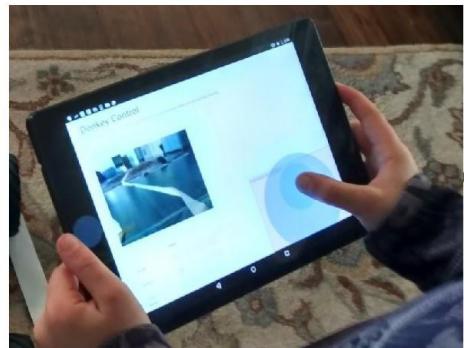
Donkey Car - 基於 Raspberry Pi 與機器學習的開源自走車平台

- 以遙控車 (RC Car) 為載體
- 在樹莓派 (Raspberry Pi) 做運算與控制
- 使用 Python 開發
- 根據神經網路 (Neural Network) 建立自走模型



<https://www.donkeycar.com/>

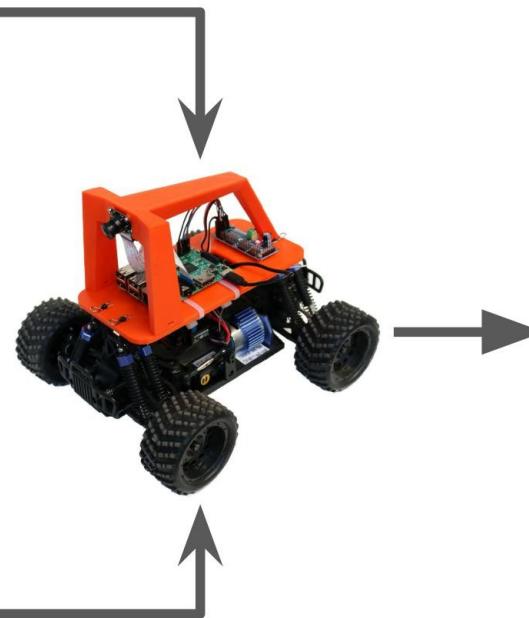
目標



Steering (-1 to 1)
Throttle (-1 to 1)
Drive Mode (manual / auto)

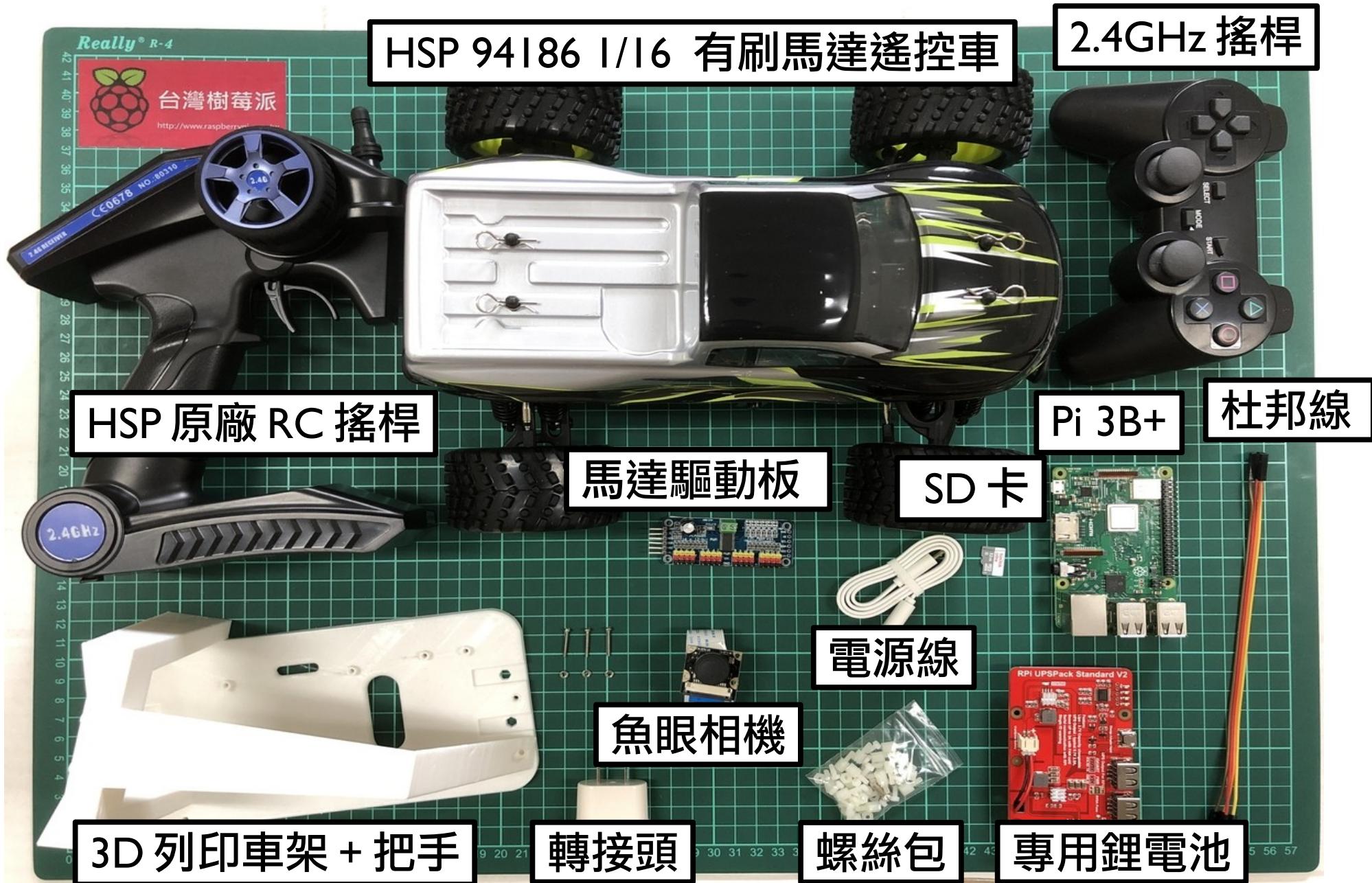


Image (120 x 160)



Drive
Perfectly.

Donkey Car 所需硬體



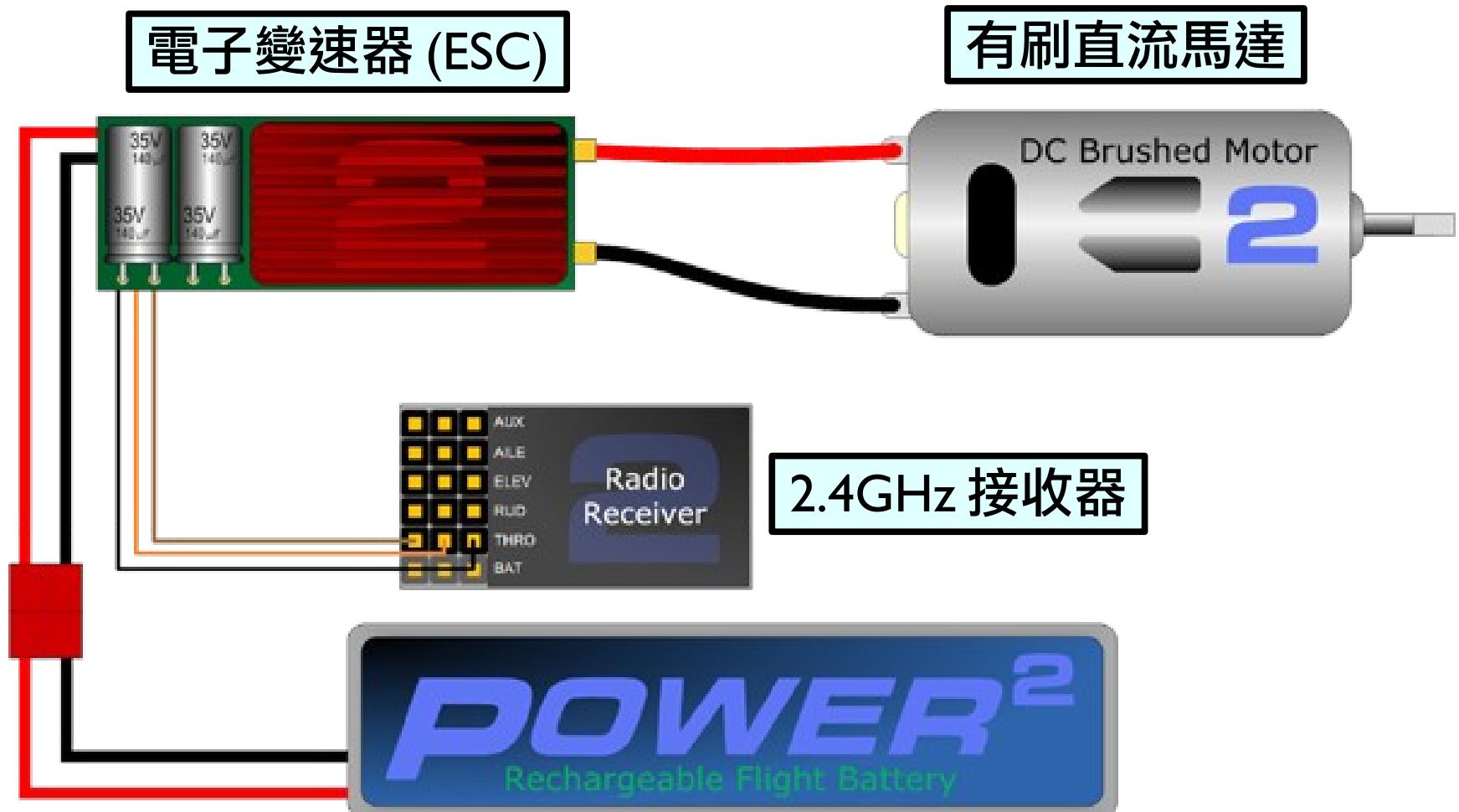
1/16 HSP(HiSPEED) RC Car



<https://www.aliexpress.com/item/32380689708.html>

動力系統架構

Brushed Motor Wiring



ESC 用來控制馬達轉速（油門）

- 將控制訊號轉為馬達用電壓，以控制馬達轉速

— = +
— = -

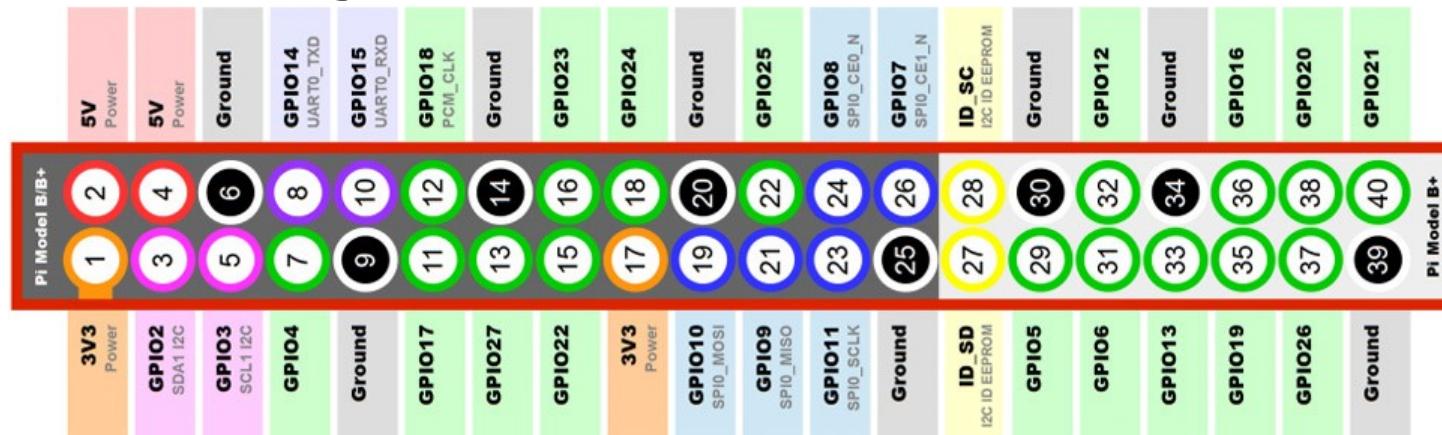
Brushed



Brushless



Raspberry Pi 3B+ 硬體規格與特色



Raspberry Pi 3 Model B+

GPIO 控制腳位

內建 WiFi 和 BT

Broadcom BCM2837B0, Cortex-A53
64-bit SoC @ 1.4GHz
with 1GB LPDDR2 SDRAM

Power over Ethernet (PoE) header
(requires separate PoE HAT)

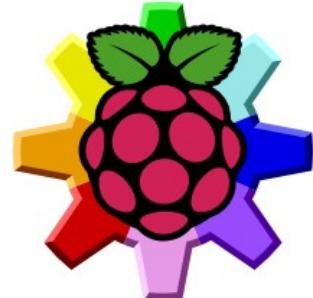
4 × USB 2.0 ports and
Faster Ethernet over USB 2.0
(maximum throughput 300Mbps)

可接相機模組



FreeBSD®

支援超過 40 種 OS



RISC OS Pi



gentoo linux™



KODI

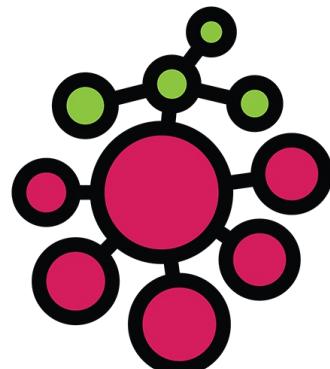


ubuntu MATE

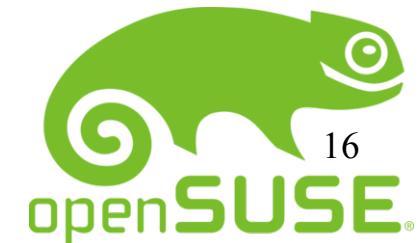
OpenWrt
Wireless Freedom



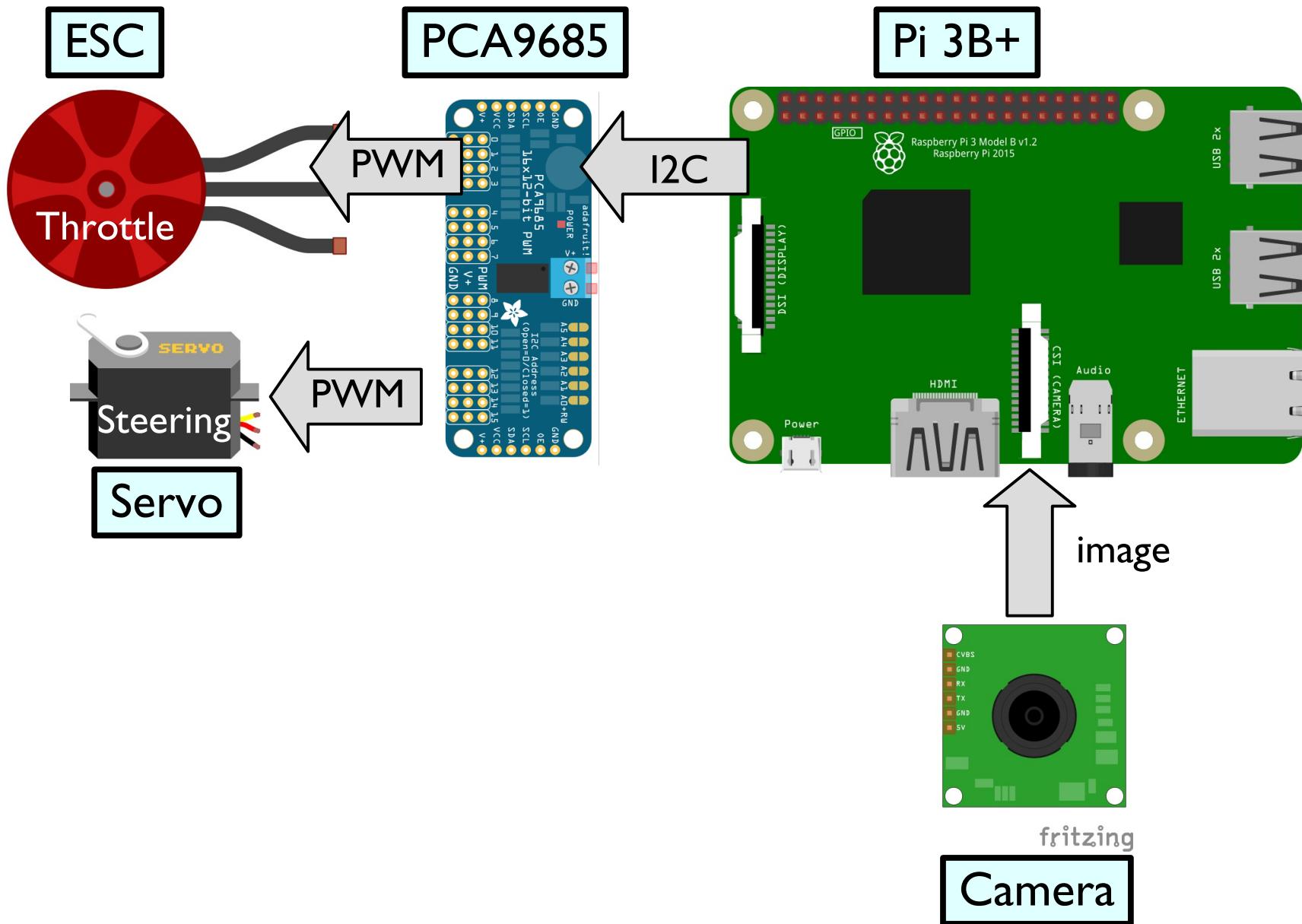
webOS



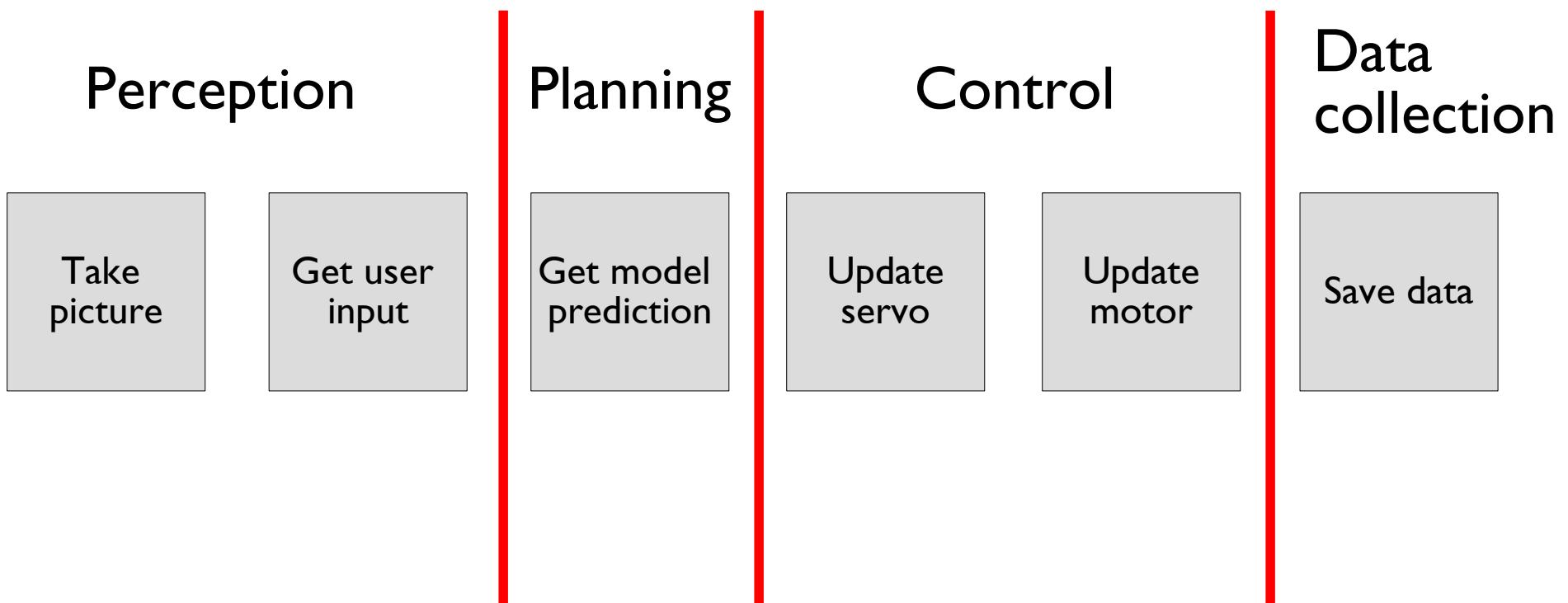
http://en.wikipedia.org/wiki/Raspberry_Pi



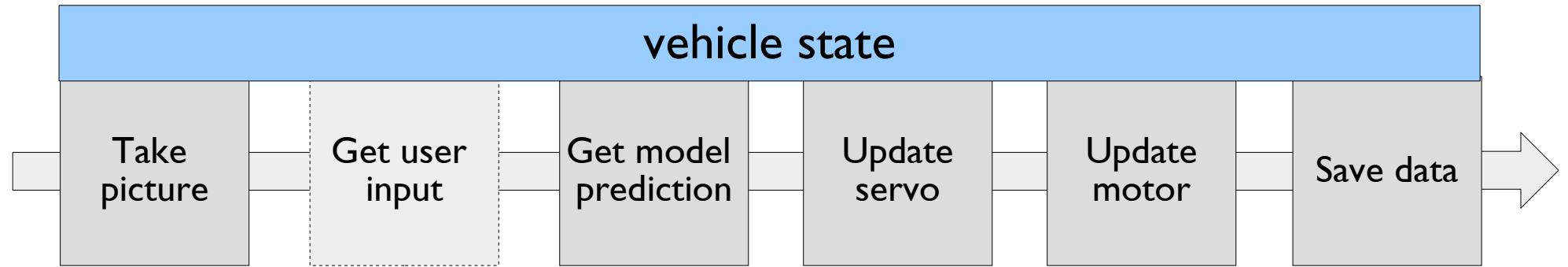
硬體架構



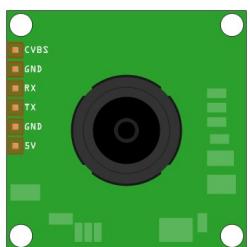
控制理論



軟體架構

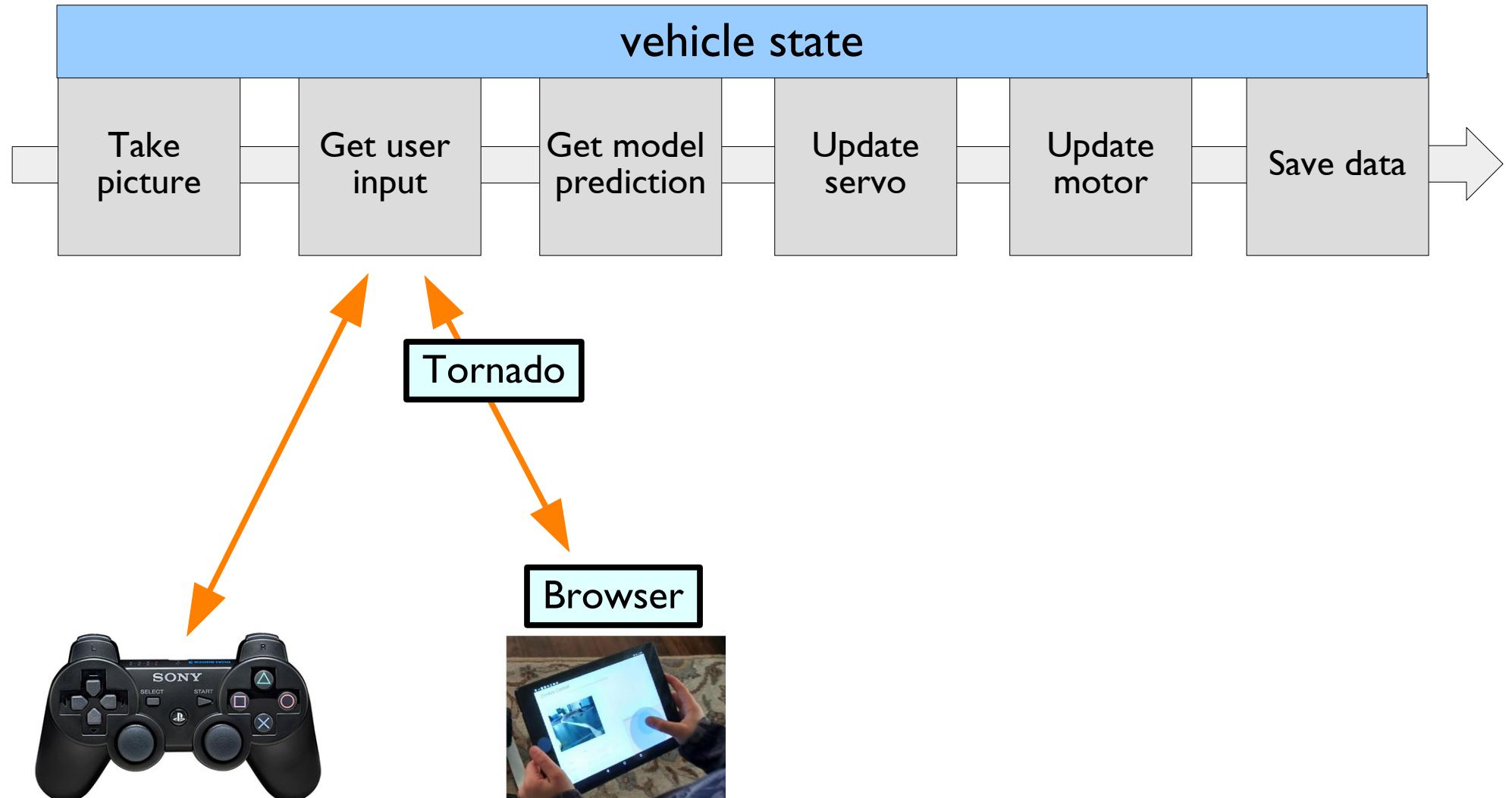


Camera

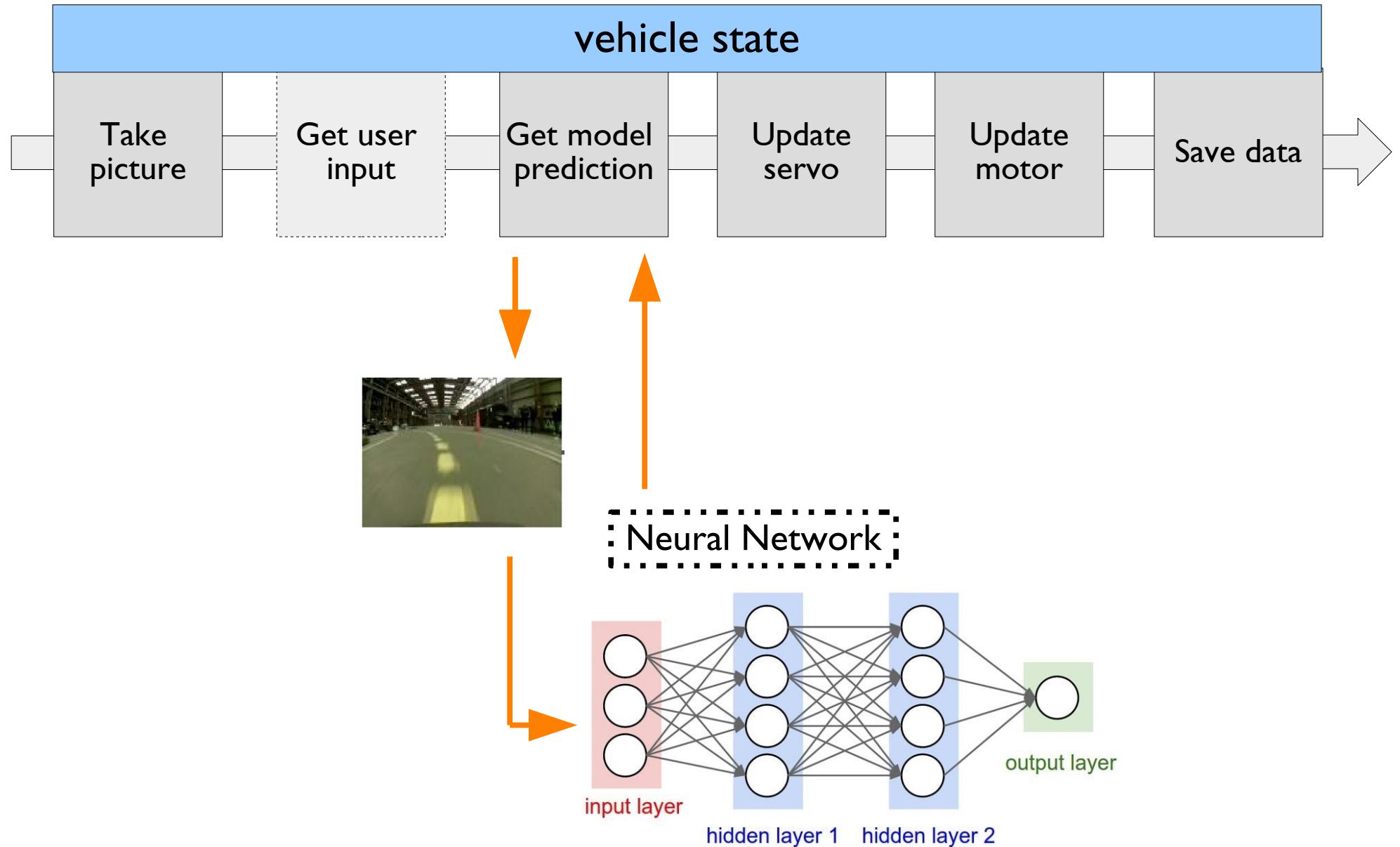


fritzing

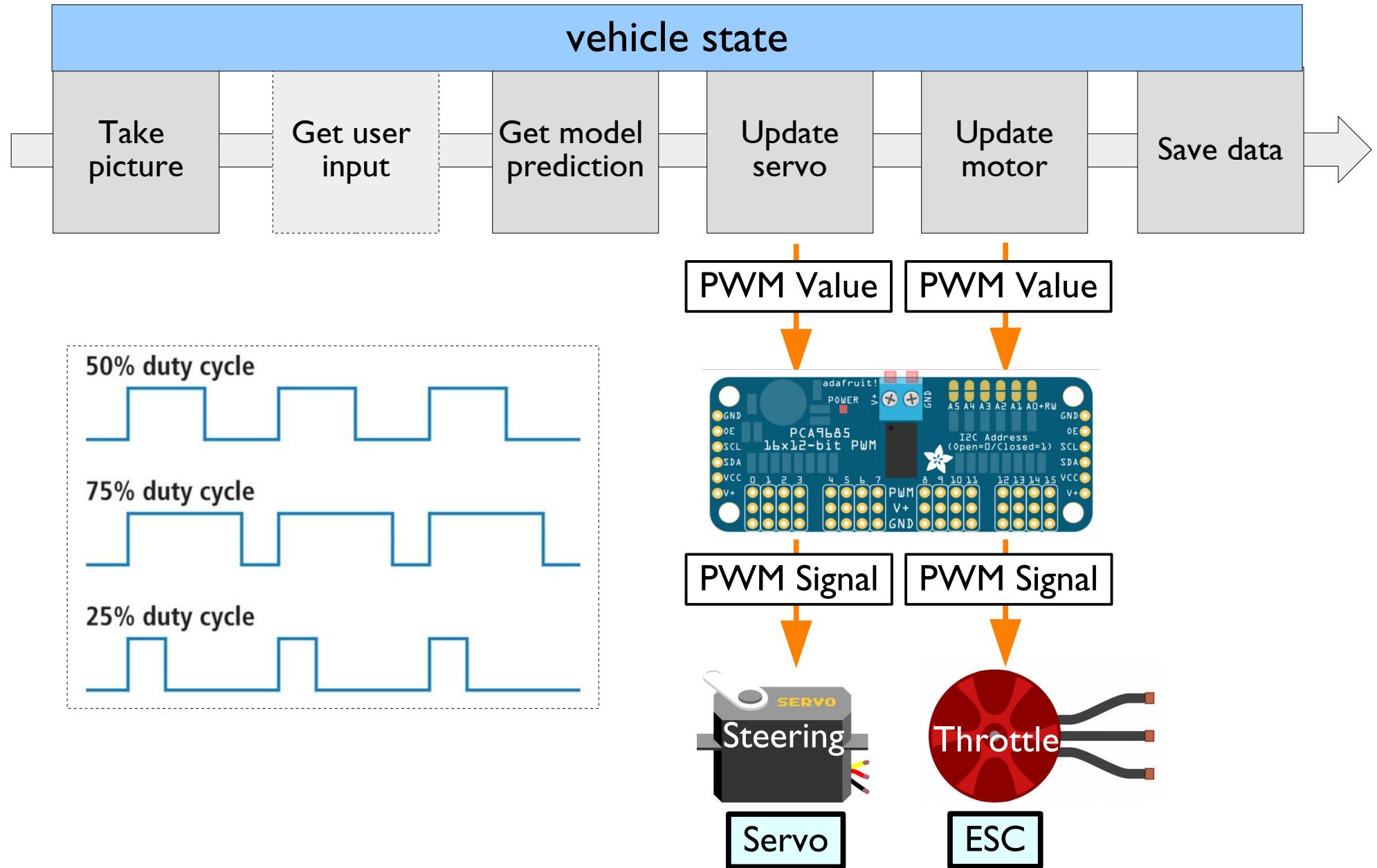
軟體架構



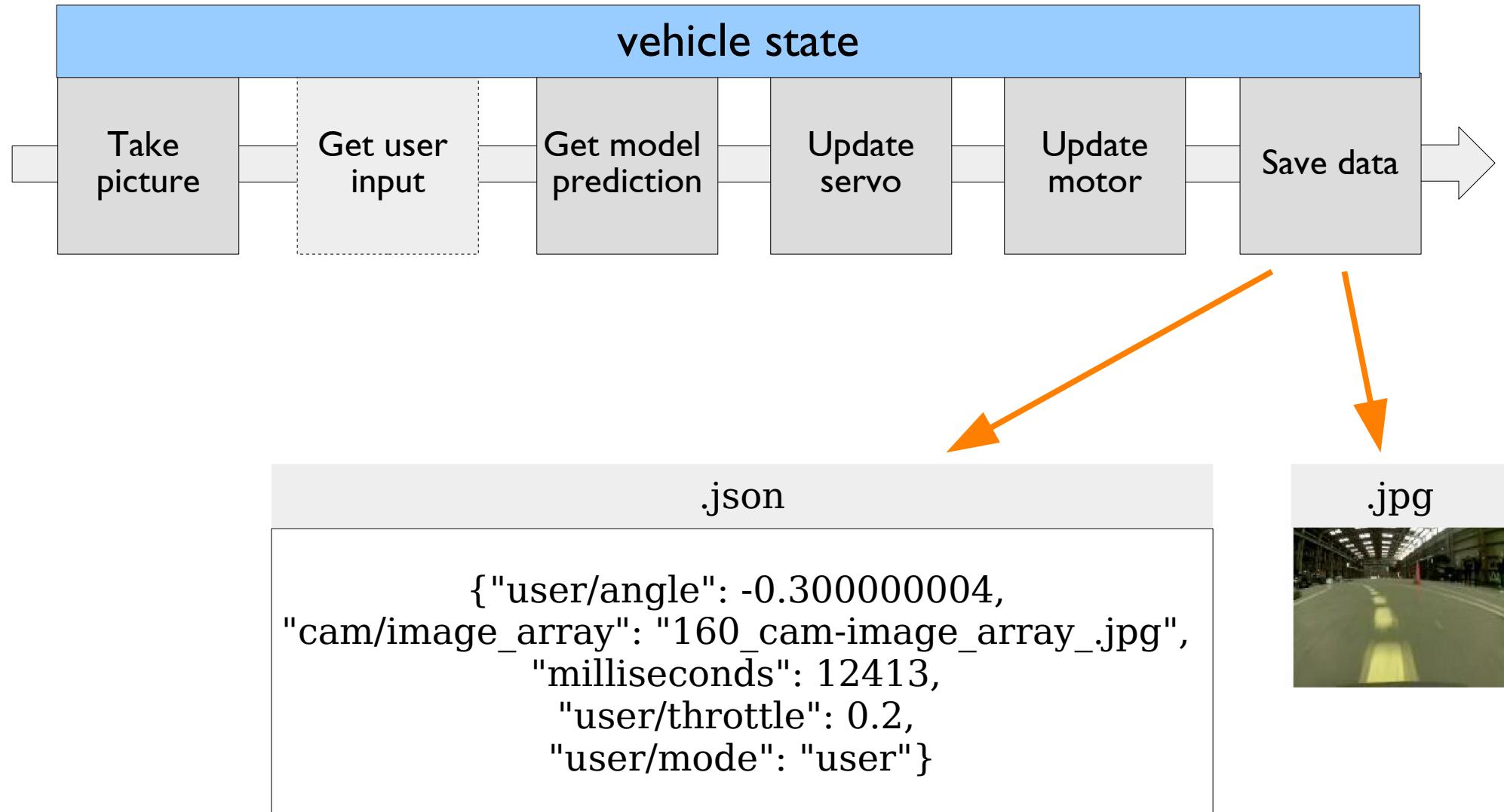
軟體架構



軟體架構



軟體架構



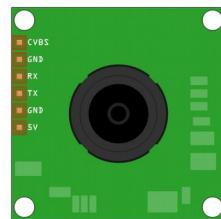
使用機器學習打造自動駕駛系統

Train Pilot

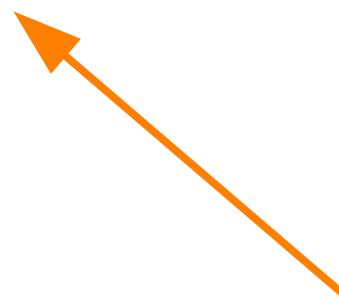
colab



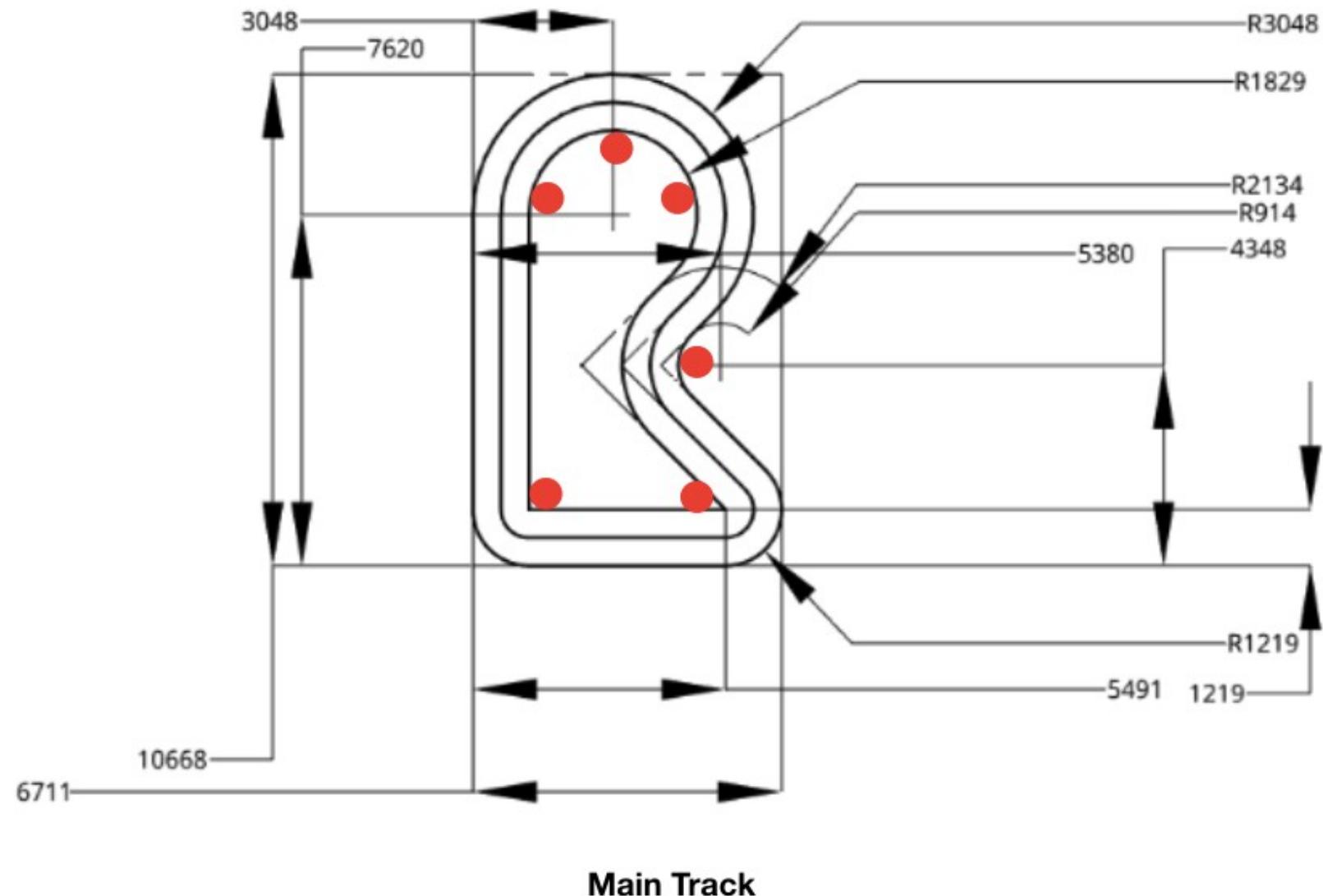
Collect Data



Test Pilot

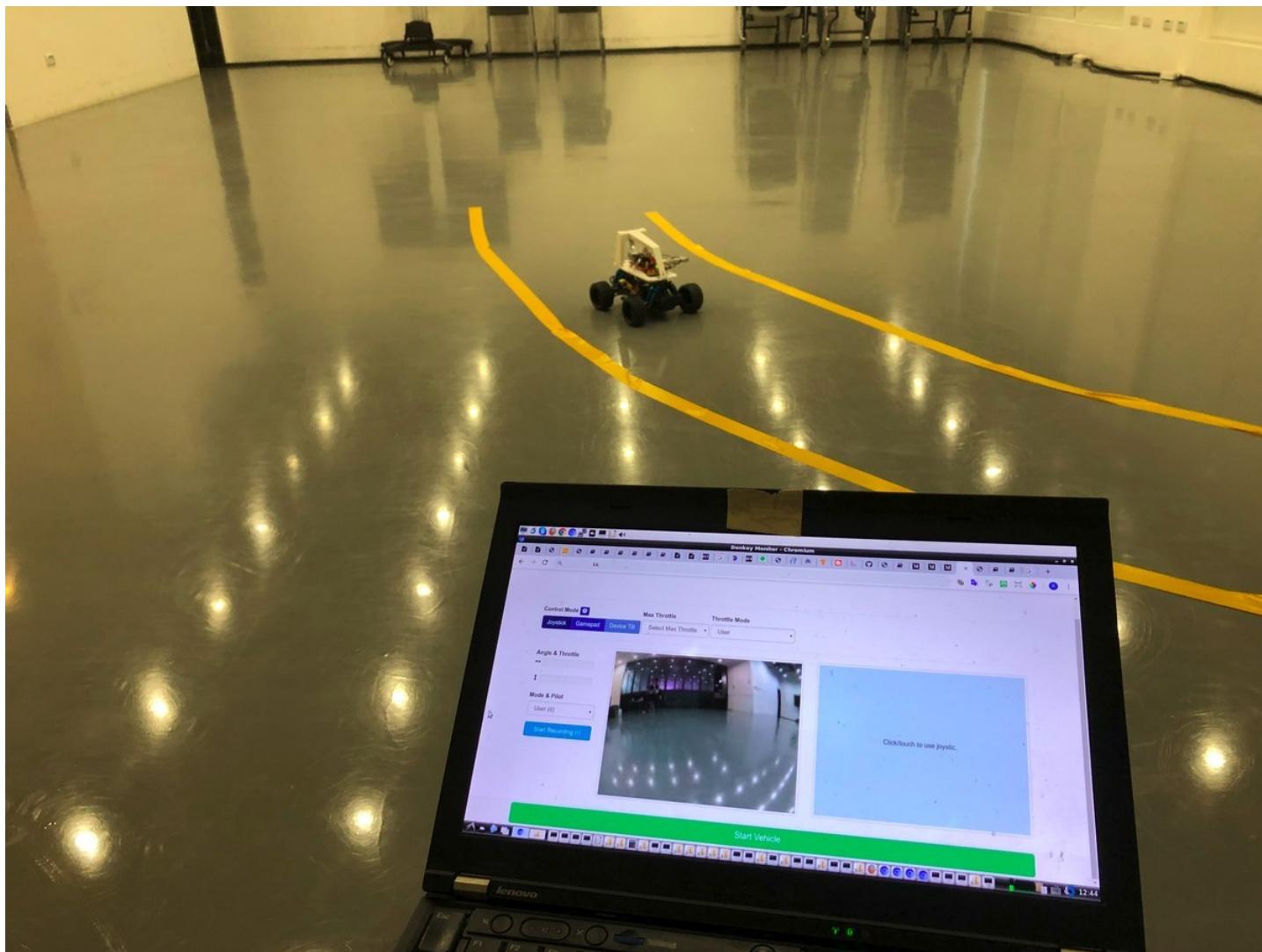


建立車道



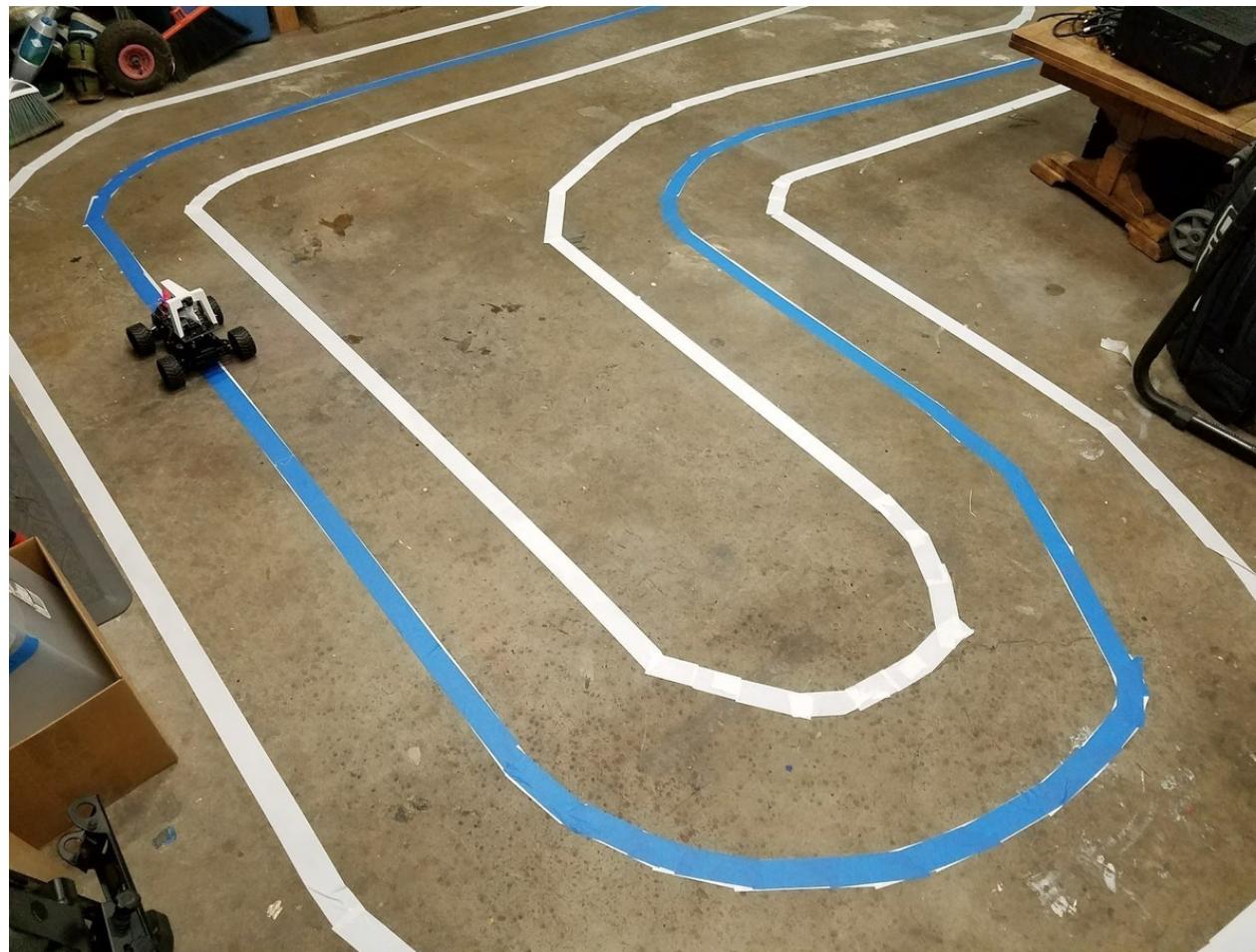
讓你的 Donkey Car 跑起來

- 使用瀏覽器或是搖桿控制



蒐集訓練資料

- 左三圈 + 右三圈，至少 5000 張照片



<http://www.sroboto.com/2017/08/donkeycar-2-install-setup-and-run.html>

清理資料

Donkey

The screenshot shows a user interface for managing training data. At the top, there is a video preview window displaying a street scene from a vehicle's perspective. To the right of the video are playback controls: a frame counter set to 0, a dropdown menu for '1x playback' (with options for 1x, 2x, 4x, and 8x), and three buttons: 'Play (→)', 'Rewind 1s (b)', and 'Split (c)'. Below the preview is a horizontal timeline showing a sequence of small thumbnail images representing individual frames. The first frame in the sequence has a play button icon. To the right of the timeline are two buttons: a blue square for 'Stop' and a grey square for 'Next'. Below the timeline are five rows of frame thumbnails, each with a play button icon on the left and a checked checkbox on the right. A red 'Save and Delete Checked Training Data!' button is located at the bottom right.

0

1x playback ▾

▶ Play (→)

◀ Rewind 1s (b)

SplitOptions (c)

Save and Delete Checked Training Data!

用 Keras/Tensorflow 訓練自動駕駛

```
Epoch 00012: val_loss improved from 0.07466 to 0.07406, saving model to models/donkeypi4.h5
16/16 [=====] - 2s 149ms/step - loss: 0.0802 - val_loss: 0.0741
Epoch 13/100
4/4 [=====] - 0s 75ms/step - loss: 0.0674

Epoch 00013: val_loss improved from 0.07406 to 0.06741, saving model to models/donkeypi4.h5
16/16 [=====] - 2s 149ms/step - loss: 0.0700 - val_loss: 0.0674
Epoch 14/100
4/4 [=====] - 0s 72ms/step - loss: 0.0579

Epoch 00014: val_loss improved from 0.06741 to 0.05786, saving model to models/donkeypi4.h5
16/16 [=====] - 2s 148ms/step - loss: 0.0620 - val_loss: 0.0579
Epoch 15/100
4/4 [=====] - 0s 71ms/step - loss: 0.0576

Epoch 00015: val_loss improved from 0.05786 to 0.05761, saving model to models/donkeypi4.h5
16/16 [=====] - 2s 146ms/step - loss: 0.0568 - val_loss: 0.0576
Epoch 16/100
4/4 [=====] - 0s 70ms/step - loss: 0.0631

Epoch 00016: val_loss did not improve from 0.05761
16/16 [=====] - 2s 143ms/step - loss: 0.0572 - val_loss: 0.0631
Epoch 17/100
4/4 [=====] - 0s 72ms/step - loss: 0.0531

Epoch 00017: val_loss improved from 0.05761 to 0.05308, saving model to models/donkeypi4.h5
16/16 [=====] - 2s 150ms/step - loss: 0.0547 - val_loss: 0.0531
Epoch 18/100
4/4 [=====] - 0s 74ms/step - loss: 0.0580
```

神經網路模型

```
img_in = Input(shape=input_shape, name='img_in')
x = img_in
x = Convolution2D(24, (5,5), strides=(2,2), activation='relu', name="conv2d_1")(x)
x = Dropout(drop)(x)
x = Convolution2D(32, (5,5), strides=(2,2), activation='relu', name="conv2d_2")(x)
x = Dropout(drop)(x)
x = Convolution2D(64, (5,5), strides=(2,2), activation='relu', name="conv2d_3")(x)
x = Convolution2D(64, (3,3), strides=(2,2), activation='relu', name="conv2d_4")(x)
x = Dropout(drop)(x)
x = Convolution2D(64, (3,3), strides=(1,1), activation='relu', name="conv2d_5")(x)
x = Dropout(drop)(x)

x = Flatten(name='flattened')(x)

x = Dense(100, activation='relu', name="fc_1")(x)
x = Dropout(drop)(x)
x = Dense(50, activation='relu', name="fc_2")(x)
x = Dropout(drop)(x)

#categorical output of the angle
angle_out = Dense(15, activation='softmax', name='angle_out')(x)

#continuous output of throttle
throttle_out = Dense(20, activation='softmax', name='throttle_out')(x)
```

實際測試驗證

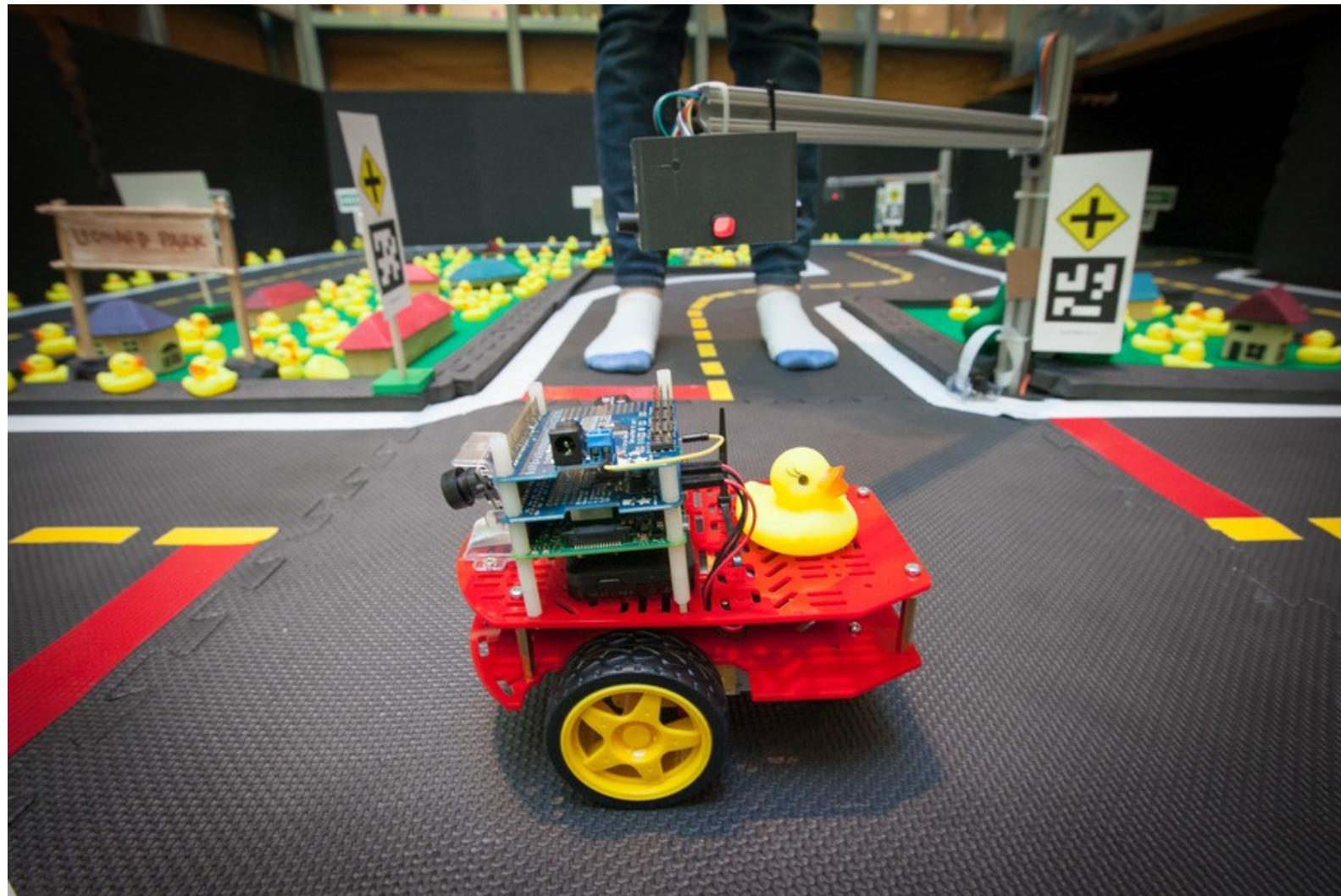
- Loop: 蒐集資料，訓練資料，測試驗證步驟



DEMO

Hack Donkey Car

在 Duckiebot 跑 Donkey Car



修改 actuator.py

```
class Adafruit_DCMotor_Hat:  
    def __init__(self, metaclass=Singleton):  
        self.mh = Adafruit_MotorHAT(addr=0x60)  
        self.mhM1 = self.mh.getMotor(1)  
        self.mhM2 = self.mh.getMotor(2)  
  
    def run(self, speed):  
        self.motor.setSpeed(self.throttle)  
        self.throttle = int(dk.utils.map_range(abs(speed), -1, 1, -255, 255))  
  
    def set_pulse(self, pulse):  
        self.pulse = pulse  
  
    def set_speed(self, speed):  
        if self.pulse > 0:  
            self.mhM2.setSpeed(speed)  
        elif self.pulse < 0:  
            self.mhM1.setSpeed(speed)
```

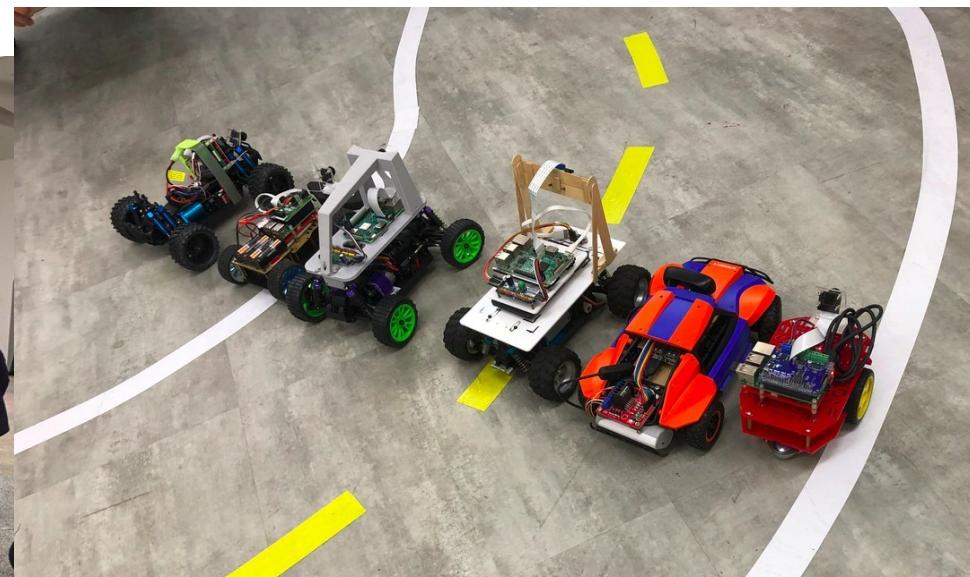
Donkey Car vs. Duckietown

	Donkey Car	Duckietown
專案開始	2016 年底	2015 年底
發源地	DIY Robocars(美國西岸)	MIT(美國東岸)
口號	小型車的開源 DIY 自駕平台	學習機器人的有趣方式
技術核心	基於機器學習 (ML)	基於機器人視覺加控制理論
特色	文件清楚 / 快速上手	從理論到實做 / 分支多 / 文件多
使用者	有興趣的玩家	學校
硬體	Pi/Jetson Nano + RC Car	Pi + 雙輪直流馬達小車
軟體	Raspbian + Python + Keras/Tensorflow	Ubuntu + ROS + Python + OpenCV
造價	USD\$250	USD\$150
commit 數	15xx	59xx
難度	***	*****

學習資源

- Donkey Car 官網文件
 - <https://docs.donkeycar.com/>
- Donkey Car 程式碼
 - <https://github.com/autorange/donkeycar>
- Donkey Car Slack
 - <https://donkeycar.slack.com>
- 組裝教學與經驗分享
 - <https://medium.com/ljlstyle/tagged/autonomous-cars>

Donkey Car 車聚 (@Fablab)



Donkey Car 工作坊

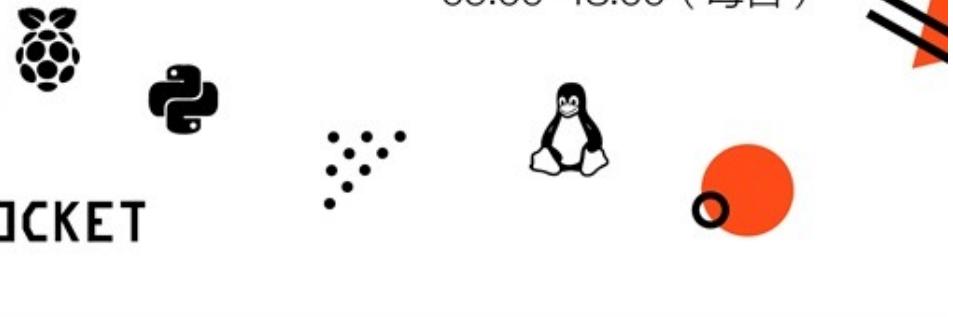


樹莓派學開車 手把手實做人工智慧自駕車

實體車架、道路設計、系統安裝、競速比賽

2019/07/20 (六) – 2019/07/21 (日)

09:00–18:00 (每日)

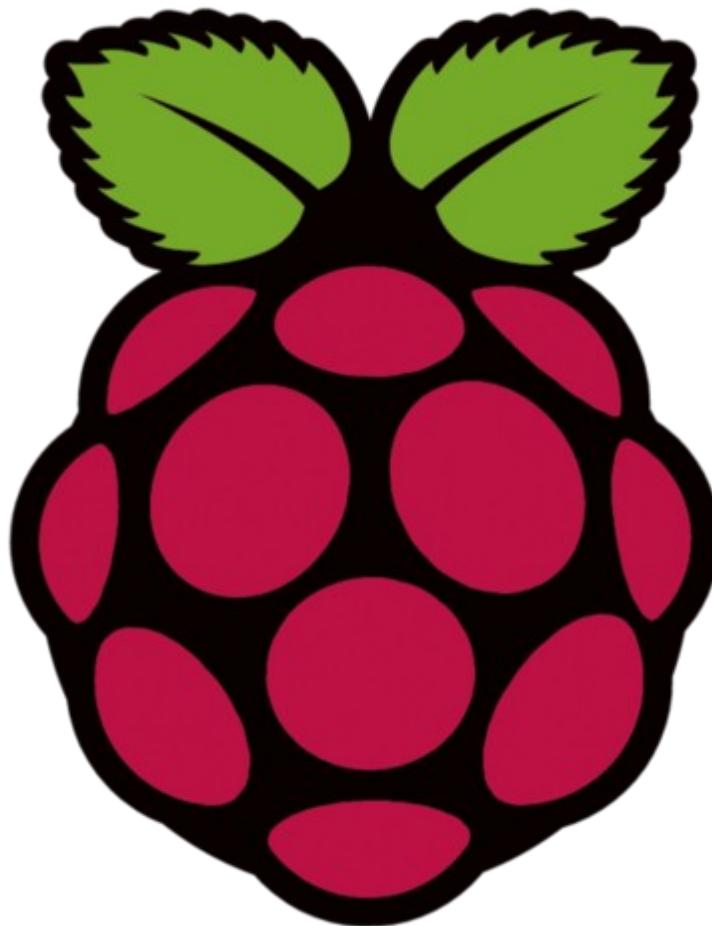


STAR ROCKET

FAQ: 為什麼取名 Donkey(驢)?

- 1. 因為驢是最早被豢養的動物
- 2. 驢對小孩很安全
- 3. 牠們偶而不遵循主人的命令
- 4. 大家對驢子的期望不會太高

Raspberry Pi Rocks the World



Thanks