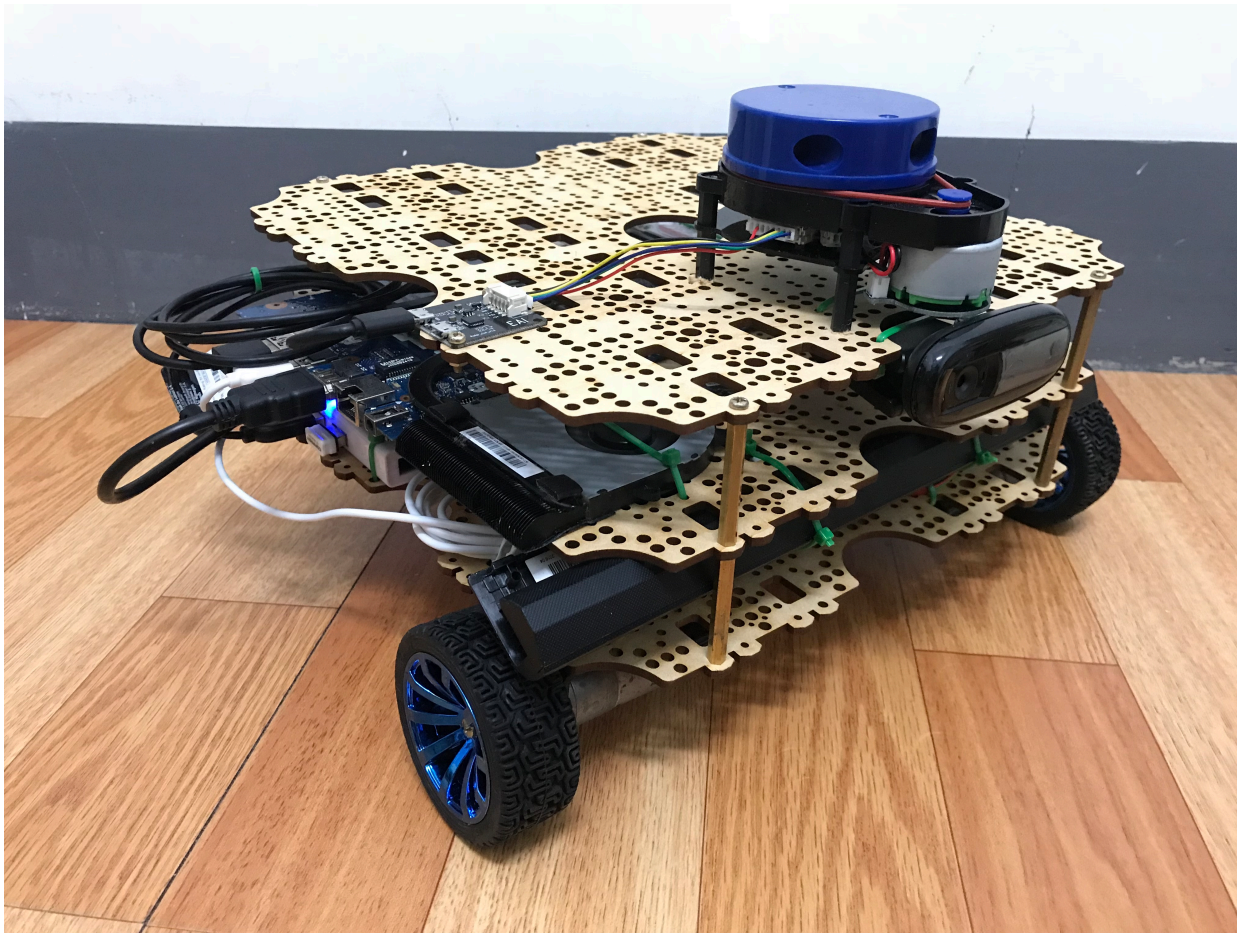


---

# Lemon minibot

## Navigation system robot car

---



Authors: Tony Guo

# Abstract

# Content

# **Chapter 1**

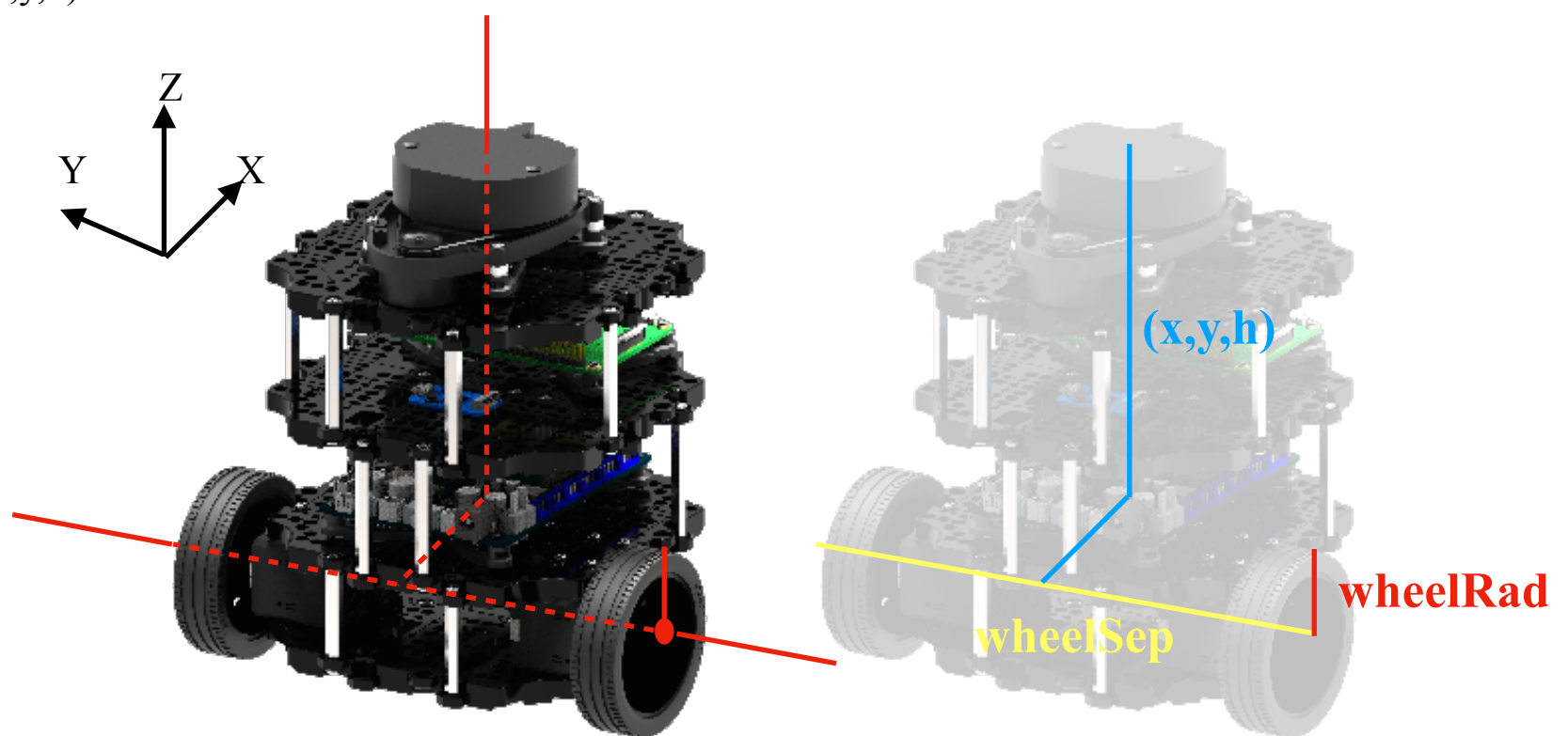
## **Introduction**

# Chapter 2

## Hardware

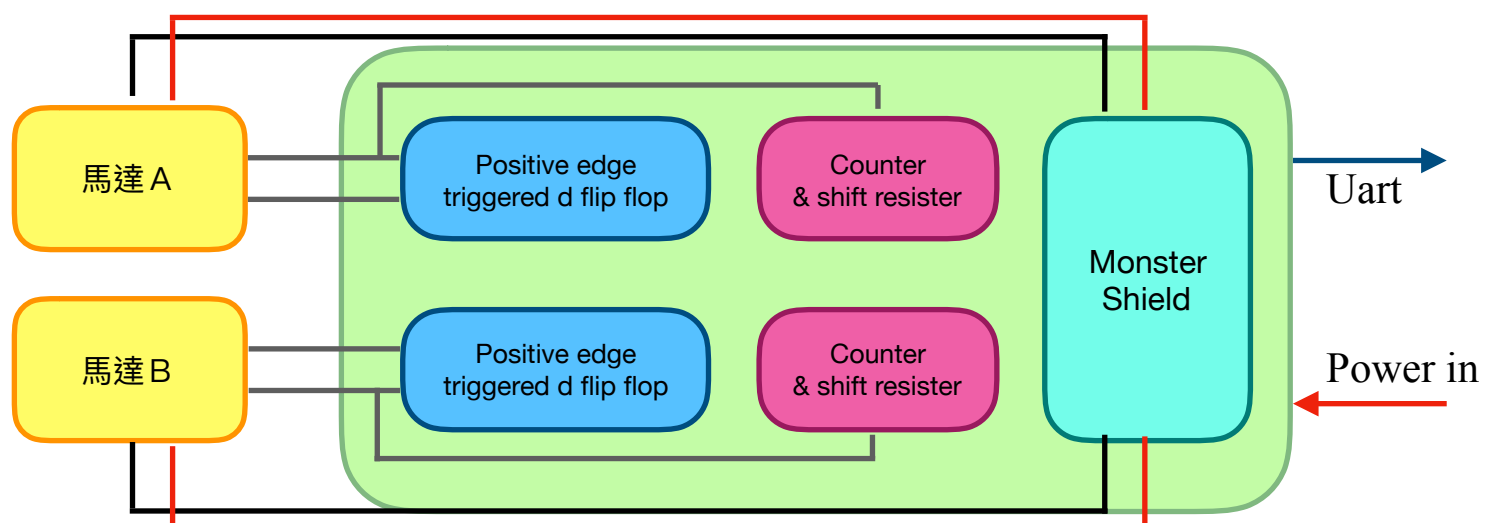
### 2.1 Custom car structure

Lemon minibot 的架構為標準兩輪差速小車(diff-drive robot car)，小車可由任何適當材料組合而成，僅需留意小車是否水平及空間是否可充足放置零組件；以turtlebot3 buger為示意圖為例，建構小車時需測量輪胎半徑( $\text{wheelRad}$ )、輪胎間距( $\text{wheelSep}$ )及雷達位移( $x,y,h$ )。



### 2.2 Motor controller (Mega 控制器)

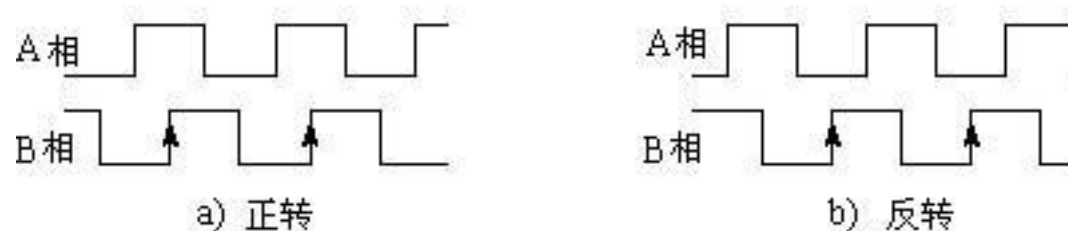
為擷取兩組馬達AB編碼器以及控制兩組馬達，因此自行設計控制器。



## 2.2.1 Motor & A/B encoder

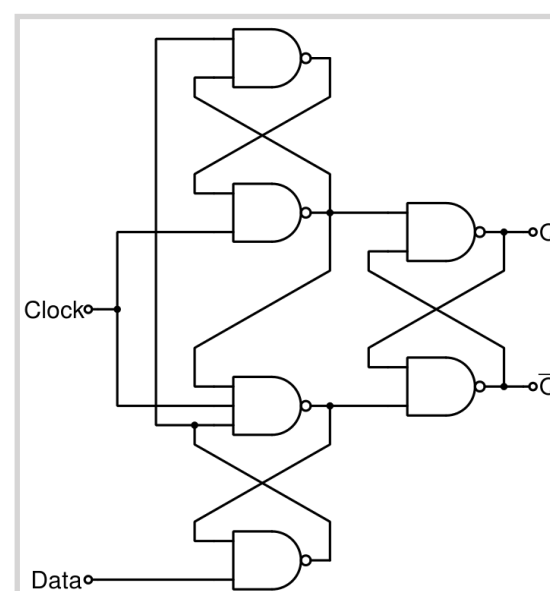
擁有AB增量型編碼器的馬達都有6pin，分別為馬達正負極、編碼器電源以及AB相輸出，接下來將說明如何使用編碼器訊號測量轉向以及轉速，在這之前也必須先知道馬達的PPR (Pulses per rev)值，意味著轉一圈會經過幾個脈衝；再者如果馬達有經過減速機，則須將PPR乘上減速比。

Lemon minibot 所使用的的編碼器為AB增量型編碼器，編碼器輸出為AB相，如下圖，A與B之間相位差為90度。



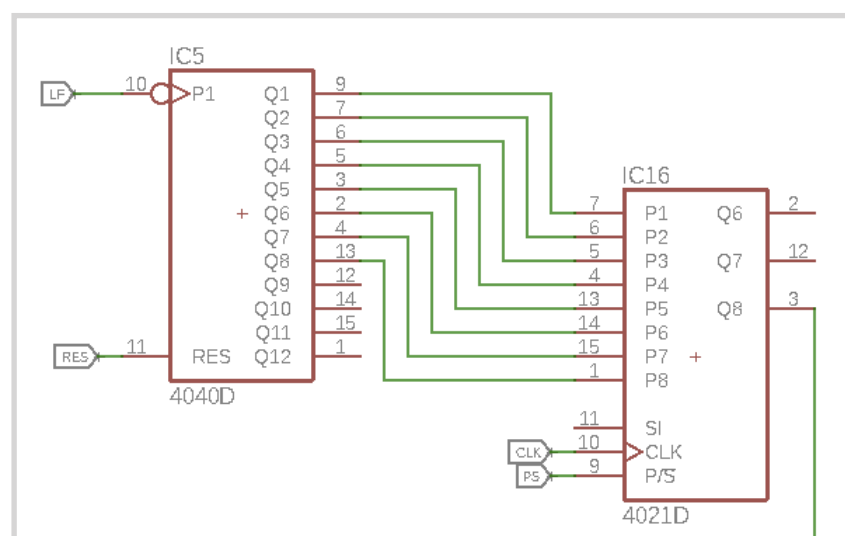
## 2.2.2 Positive edge triggered d flip flop

正緣觸發D正反器 (positive edge triggered d flip flop)，可以將AB相轉換成單一輸出的轉向；當clock背A相正緣觸發時會將Q輸出B相的值，而依照2.2.1 AB相圖可知，正反轉會輸出不同的值，並以此作為正反轉訊號。



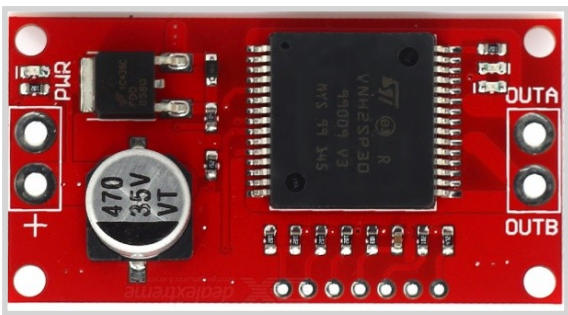
## 2.2.3 Counter & shift resister

CD4040為12bits計數器 (Counter)，CD4021為8bits的位移暫存器 (Shift resister)；解釋期運作原理，將訊號由計數器的PIN10輸入，輸入脈衝數換成二進位制由Q1~Q12輸出，再經過位移暫存器輸入至微控器。



2.2.4 Monster Shield

Monster Shield為意法半導體所生產的馬達模組，  
內含H-bridge可以控制馬達正反轉，且輸出電流可高達  
30A。



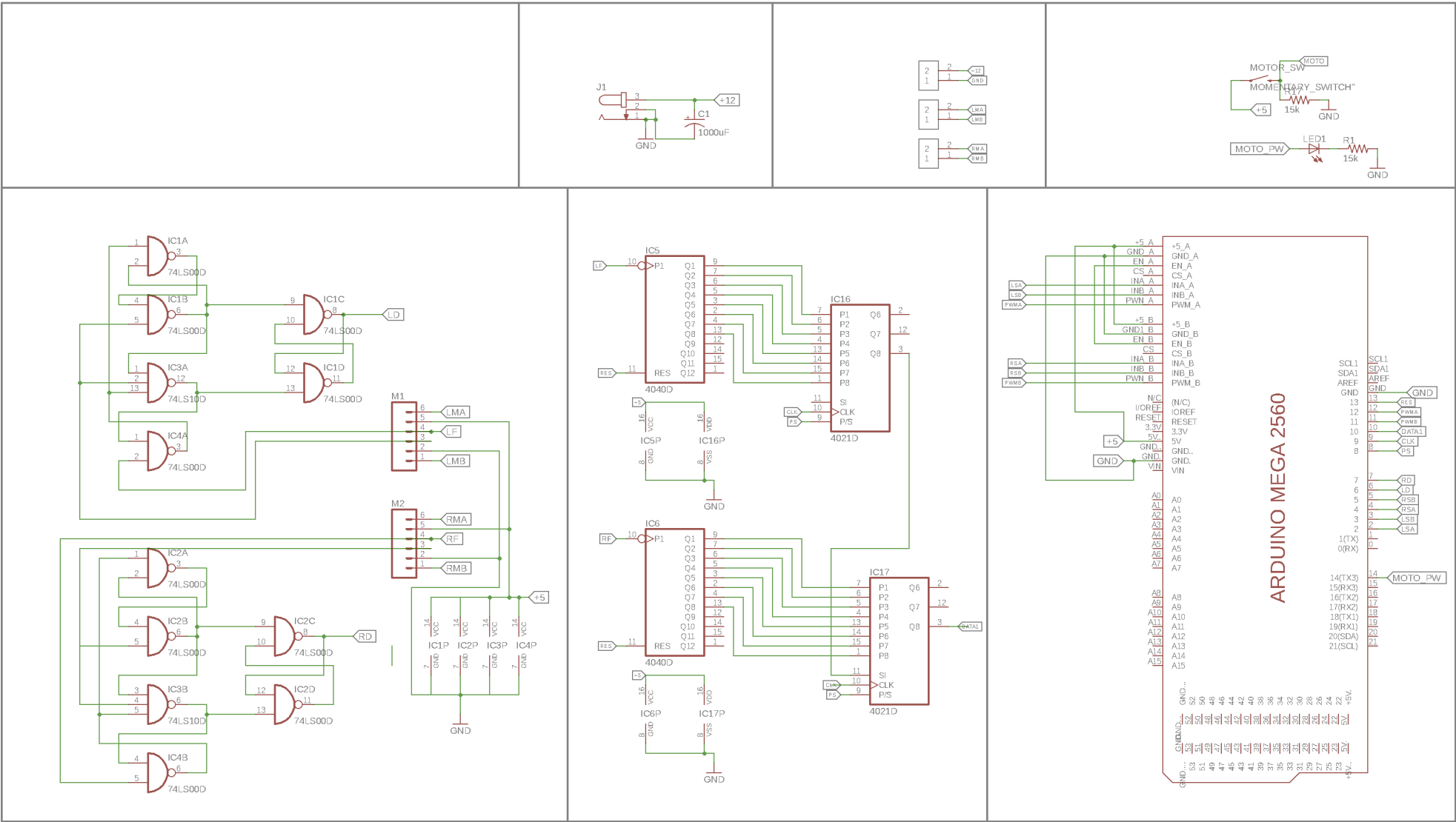
2.2.5 Power

Lemon minibot 使用一顆2 usb port 的行動電源以及一顆14.8V 2600mah的鋰電池。

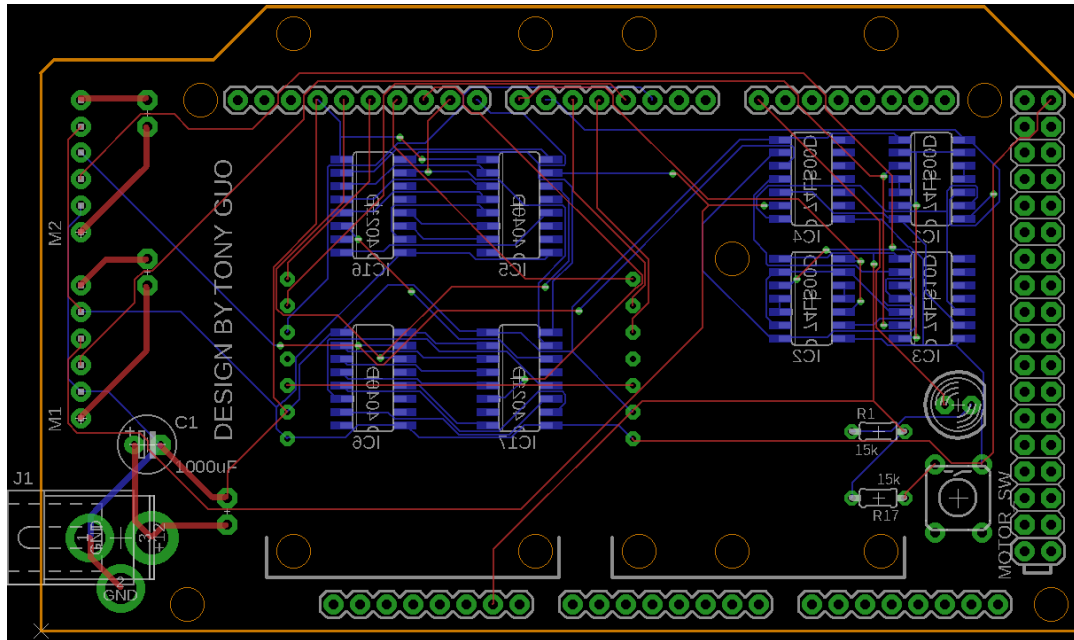
2.2.6 UART

Lemon minibot 將透過UART與raspberry pi 溝通。

2.2.7 Circuit & PCB







## 2.3 Raspberry pi 2/3

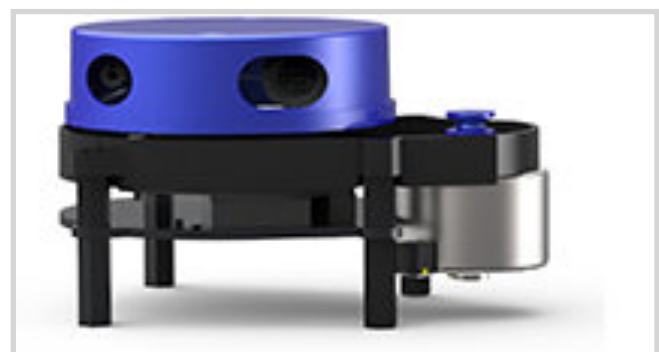
Lemon minibot 使用raspberry pi 當作主控器，運行 ubuntu mate 16.04 與 ROS (機器人作業系統) 以用來控制與接收 (電腦端)訊號。

電腦端及pi端安裝方式，請參考<專題筆記：ROS #1 安裝>，電腦端及pi端安裝方式相同，但是pi若沒有接上螢幕，則Open GL 將不會被開啟而導致無法使用 VNC開啟 GUI介面。



## 2.4 YDLIDAR X4

Lemon minibot 使用EAI的ydlidar X4光達來建置2D地圖；因為其消耗功率大，除了通訊USB 接口外，必須另外連接行動電源供電。





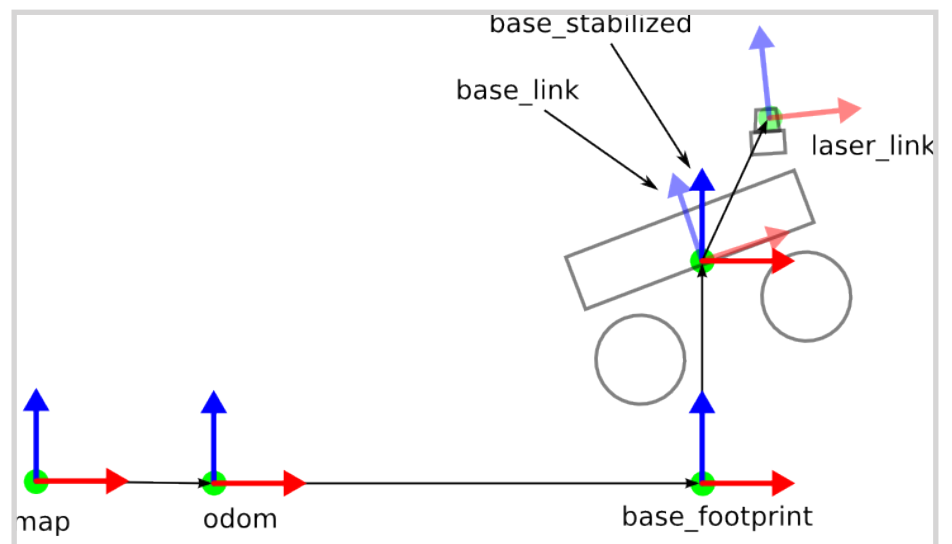
# Chapter 3

## Concept

### 3.1 Transform (TF)

為記載各個裝置位置，使用TF可以設置其相關位置。

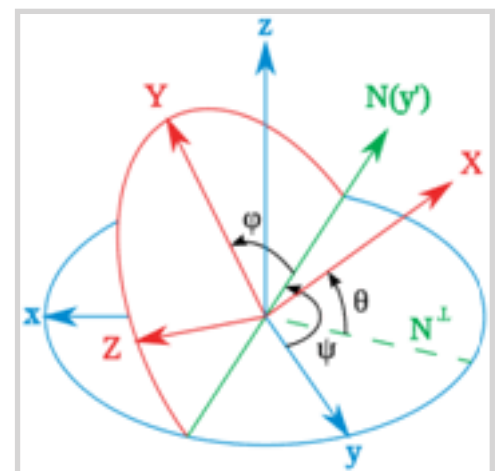
- Map : 固定的地圖
- Odom : 移動路徑
- Base\_link : 由sensor推算之座標
- Base\_footprint : Base\_link 之投影
- Laser\_link : 光達位置



### 3.2 Pose & Odometry

#### 3.2.1 Orientation 四元數

欲表現三空間的鋼體運動，常見的有歐拉角與四元數，其中歐拉角具有奇異性(例：Gimbal lock problem)，可想而知，若只使用三個變數表示三維運動，可能有多種不同的運動方式產生相同結果；因此使用具有四個變數的“四元數”來表示其運動較為恰當，且也具有歐拉角與四元數之間的轉換公式。右圖為歐拉角示意圖。



四元數表示法如下：

$$\mathbf{q} = q_0 + q_1i + q_2j + q_3k$$

四元數為一位實數以及三位虛數所構成，滿足

$$\begin{cases} i^2 = j^2 = k^2 \\ ij = k, ji = -k \\ jk = i, kj = -i \\ ki = j, ik = -j \end{cases}$$

也可使用一個純量及一個向量表示

$$\mathbf{q} = [s \quad \mathbf{v}], s = q_0 \in \mathbb{R}, \mathbf{v} = [q_1 \quad q_2 \quad q_3]^T \in \mathbb{R}^3$$

假設某個旋轉是繞著單位向量  $\mathbf{n} = [n_x \quad n_y \quad n_z]^T$  進行旋轉 $\Theta$ 角，那麼以四元數表示為

$$\mathbf{q} = \left[ \cos\left(\frac{\Theta}{2}\right) \quad n_x \sin\left(\frac{\Theta}{2}\right) \quad n_y \sin\left(\frac{\Theta}{2}\right) \quad n_z \sin\left(\frac{\Theta}{2}\right) \right]$$

接下來欲把空間一點  $\mathbf{p} = [x \quad y \quad z] \in \mathbb{R}^3$  經過  $\mathbf{q}$  做旋轉，則新座標  $\mathbf{p}'$  等於

$$\mathbf{p}' = \mathbf{q}\mathbf{p}\mathbf{q}^{-1} \text{ 化成 } \mathbf{p}' = \mathbf{R}\mathbf{p}$$

則 $\mathbf{R}$ 為

$$\mathbf{R} = \begin{bmatrix} 1 - 2q_2^2 - 2q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & 1 - 2q_1^2 - 2q_3^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & 1 - 2q_1^2 - 2q_2^2 \end{bmatrix}$$

### 3.2.2 EKF

## 3.3 Kinematic model

設左輪角速度 $\omega_l$ 、右輪角速度 $\omega_r$ ，則速度如下

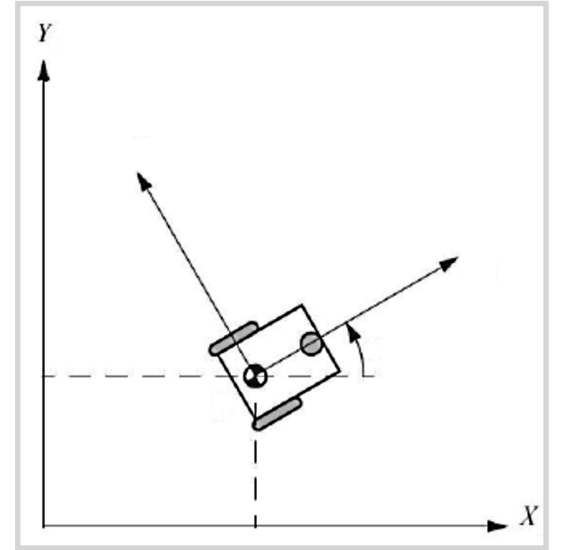
$$\begin{cases} v_l = wheelRad * \omega_l \\ v_r = wheelRad * \omega_r \end{cases}$$

$$V = \frac{v_r + v_l}{2} = wheelRad * \frac{\dot{\omega}_r + \dot{\omega}_l}{2}$$

$$\alpha = \frac{v_r - v_l}{wheelSep}$$

$$\begin{cases} V_x = V \cos \alpha \\ V_y = V \sin \alpha \end{cases}$$

$$\begin{cases} X = X + V_x * dt \\ Y = Y + V_y * dt \end{cases}$$



# Chapter 4

## Package

### 4.1 Base.py

透過Uart 連接Mega 控制器，傳送馬達控制訊號以及接收馬達轉速訊號，並積分轉換成odom

Node : /base

Subscribe : /car/cmd\_vel

Publish : /tf , /odom

### 4.2 keyboard\_teleop.py

透過鍵盤傳送速度訊號

Node : /teleop

Publish : /car/cmd\_vel

### 4.3 Joystick

使用遊戲搖桿

Node : /joy

Publish : /joy

### 4.5 teleop\_joy.cpp

遊戲搖桿轉換成控制訊號

Node : /teleop

Subscribe : /joy

Publish : /car/cmd\_vel

### 4.4 YDLIDAR X4

連接YDLIDAR X4並輸出雷射掃描資訊

Node : /ydlidar\_node

Publish : /scan

## 4.5 TF

設定座標轉換

## 4.6 rf2o

將雷射掃描資訊轉換成里程計odometry

Node : /rf2o\_laser\_odometry

Subscribe : /laser\_scan

Publish : /odom

## 4.7 robot pose ekf

使用EKF融合

Node : /rf2o\_laser\_odometry

Subscribe : /odom , /imu\_data , /vo

Publish : /robot\_pose\_ekf/odom\_combined

## 4.8 ar track alvar

## 4.9 Gmapping

# Chapter 5

## Control Mode Flow

### 5.1 Locate Ar-Tag & SLAM

Sensor	Detect	Control	Algorithm	Core
Camara	Ar tag Detector	Motor Controller	Gmapping	Core
Laser lidar	Speaker recognition	Ramdom Walk	Navigation	

### 5.2 Goal sitting through speaker recognition

Sensor	Detect	Control	Algorithm	Core
Camara	Ar tag Detector	Motor Controller	Gmapping	Core
Laser lidar	Speaker recognition	Random Walk	Navigation	

# **Chapter 6**

## **Build guide**