

新視野，未來感

新能源新世代 2025

一年一度的中學生太陽能車比賽盛事 //
組別包括大車, 細車, 太陽能船及滑翔機

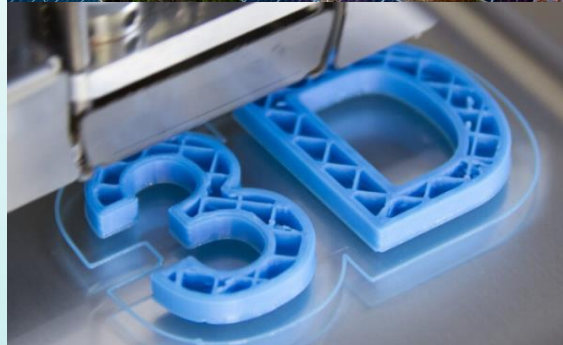
電與力工作坊I



上回重溫

- 太陽能
- 3D Print

今堂內容：細車,船及 滑翔機的延伸



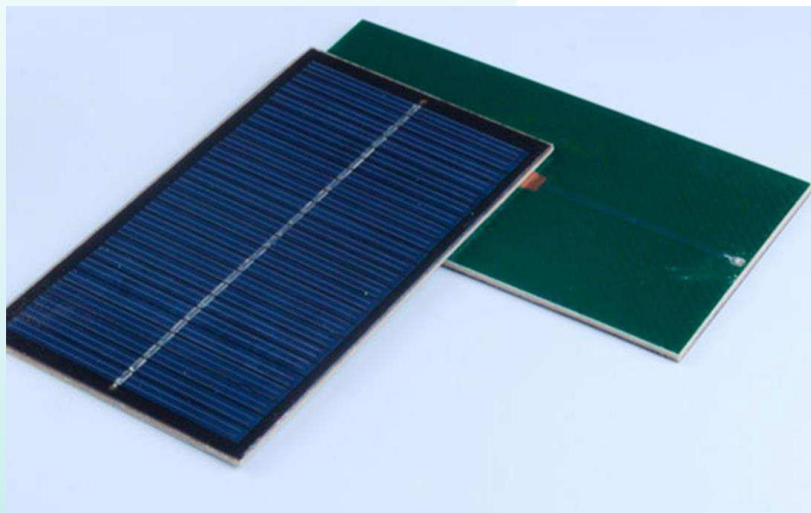
太陽能板

細車, 船及 滑翔機

輸出電壓: 2V

功率: 1W

面積: 210mm x 120mm



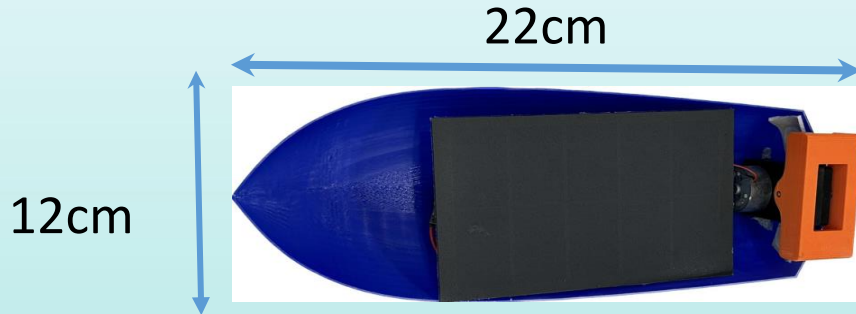


船內容

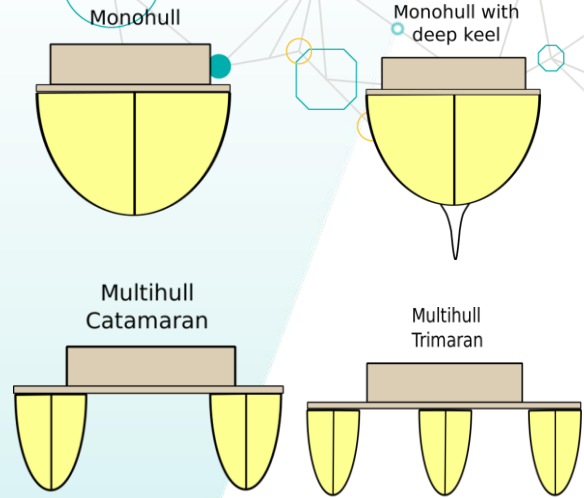
- 船身部件介紹
- 船身結構介紹

船架設計

- 船身尺寸和形狀
- 材料 (重量)
- 螺旋槳
- 船艙



10cm



船身材料

材料重量及強度

- PVC-膠樽
- Polystyrene-發泡膠
- 3D Print物料(PLA, PVA,)



滑翔機內容

- 歷史
- 原理
- 設計
- 部件





滑翔機歷史

1809年英國的喬治
凱利爵士試製了一
架滑翔機

1847年，76歲的凱
利製作了一架大型
滑翔機

德國土木工程師利
林塔爾1891年製作
了第一架固定翼滑
翔機，翼展為7米

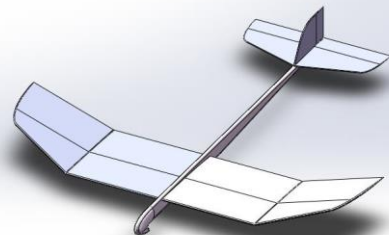
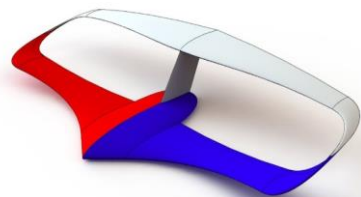
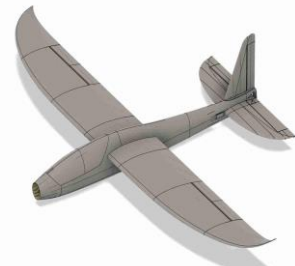
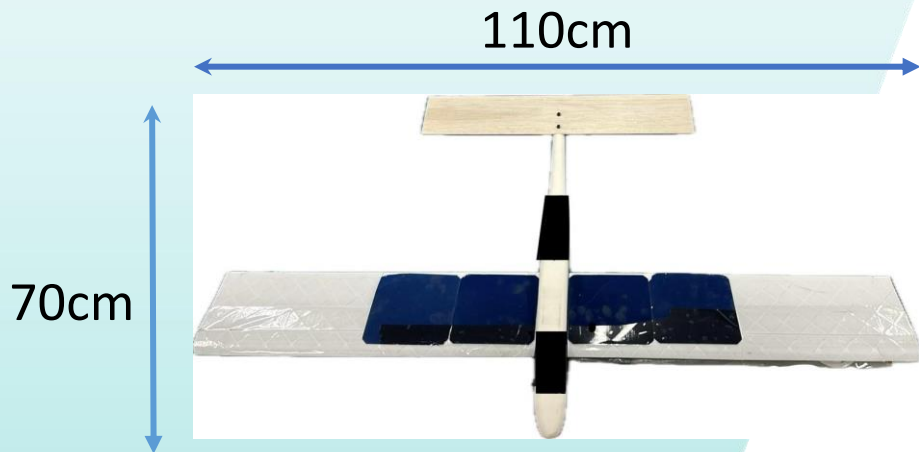
萊特兄弟飛行者一
號

1914年德國人哈斯
研製出第一架現代
滑翔機

德國空軍更是把滑
翔機用於傘兵機降
作戰並製造了歷史
上最巨大的Me 321
滑翔機

滑翔機結構

- 尺寸大小
- 滑翔機結構(機翼, 角度)
- 面積及體積

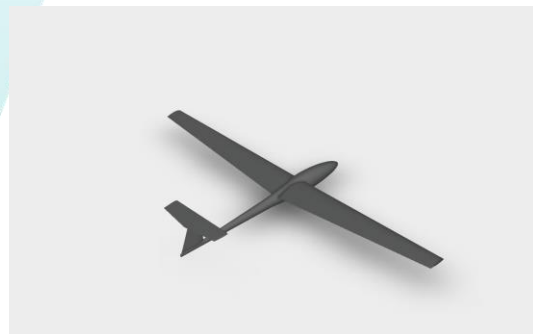




材料

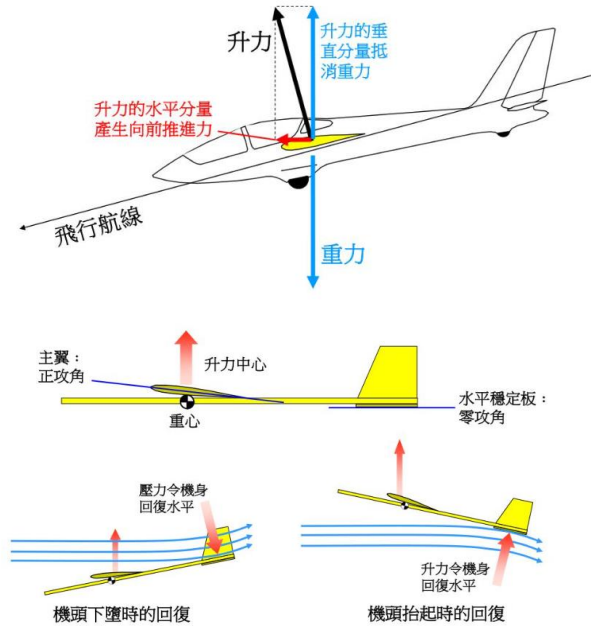
材料重量及強度

- PVC
- 木
- 3D Print物料(PLA, PVA,)



原理

- 主要依靠上升氣流進行持續飛行
- 在無風情況下，下滑飛行過程中依靠自身重力獲得前進動力
- 在有上升氣流時，可以藉此實現平飛或升高

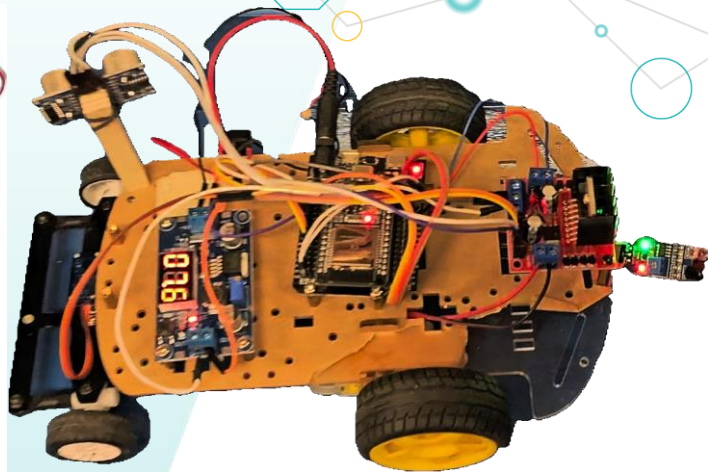
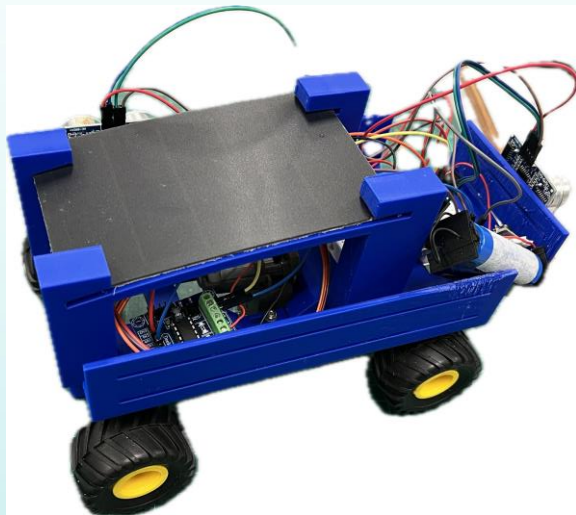




小休

細車內容

- 部件介紹
- 控制器及程式軟件
- 搖控方式
- 泊車示範影片

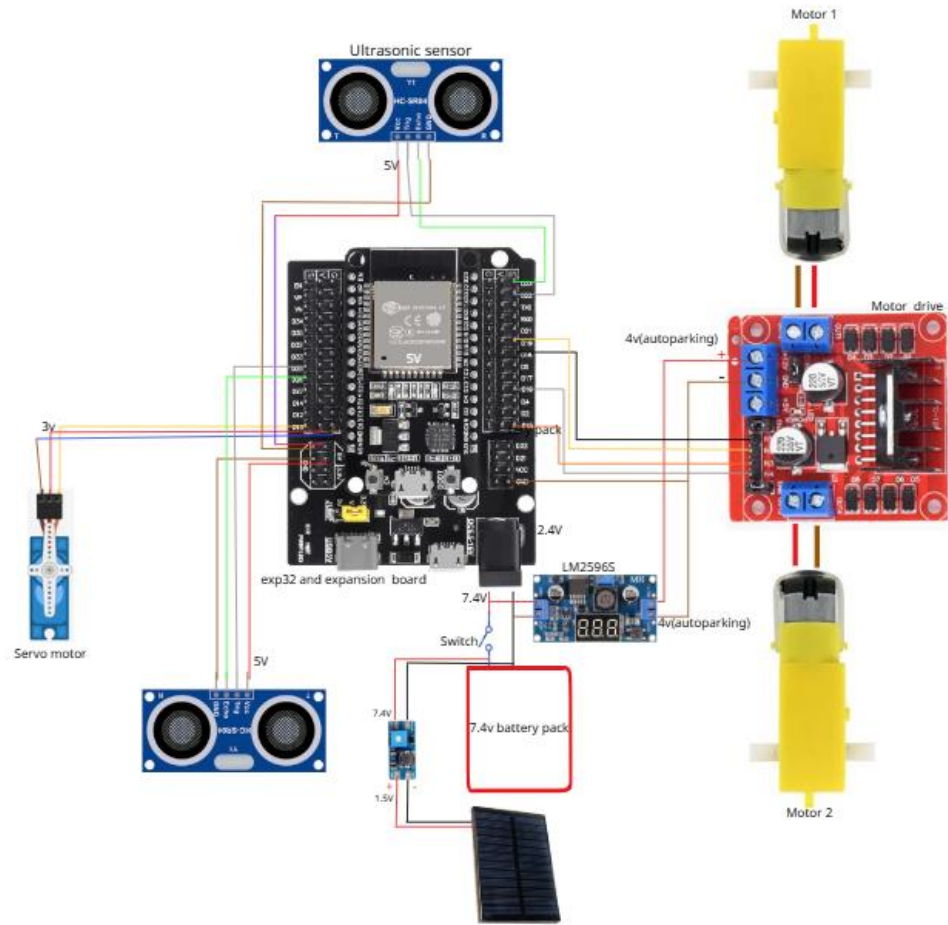
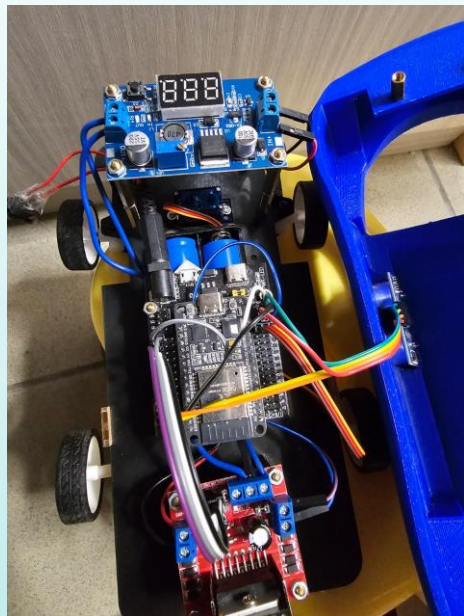


遙控車部件

- 控制器
- 感應器及驅動馬達
- 面包板
- 連接線
- 電池
- DC-DC轉換器
- 太陽能板
- 車輛結構
- 轉向結構

整一部遙控車需要咩？

電路圖

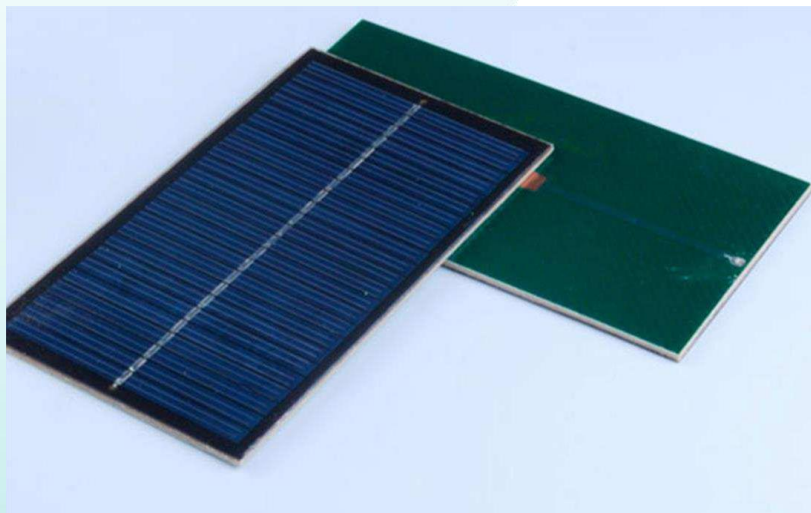


太陽能板

輸出電壓: 2V

功率: 1W

面積: 210mm x 120mm

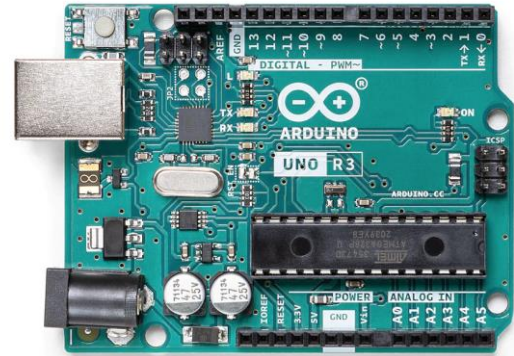


控制器

- 編寫程式方便
- 網上資源多
- 硬件及軟件開源
- 大量傳感器

官網：<https://www.arduino.cc/en/software>

Arduino



控制器

- 編寫程式方便
- 網上資源多
- 硬件及軟件開源
- 大量傳感器

官網 : <https://www.arduino.cc/en/software>



Arduino IDE 2.3.2

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits

Windows MSI installer

Windows ZIP file

Linux AppImage 64 bits (X86-64)

Linux ZIP file 64 bits (X86-64)

macOS Intel, 10.14: "Catalina" or newer, 64 bits

macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

控制器

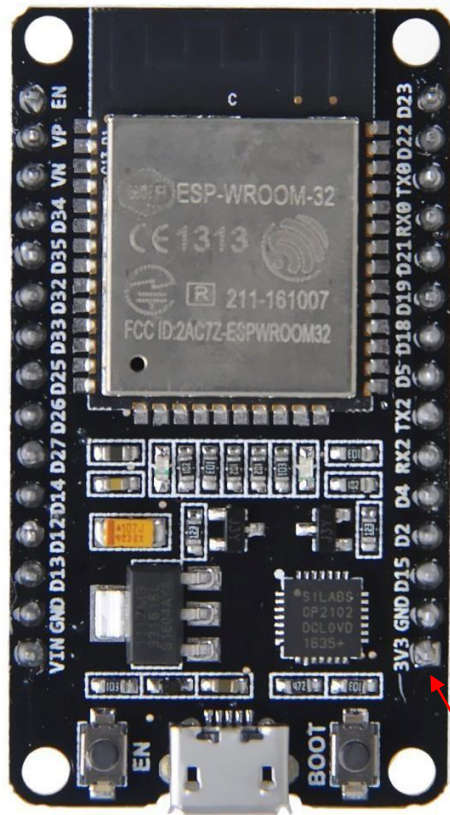
ESP32-WROOM-32

好處

- 內置藍芽功能
- 引腳多(30pin)
- 可使用Arduino IDE編程

壞處

- 需要額外下載驅動程式
- 電源只可輸出3.3V

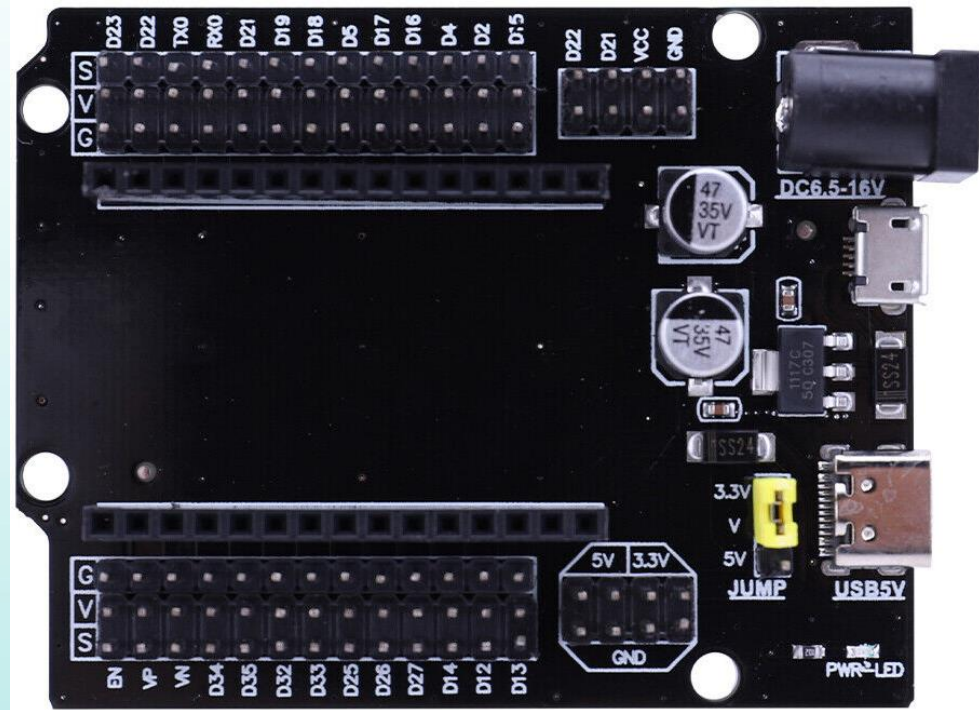


ESP32-WROOM-32

控制器擴展板

好處

- 防止芯片銷毀
- 代替麵包板，體積細
- 輸入電壓可更大
- 輸出電壓可調整 3.3v或5v





控制器驅動程式

arduino.esp8266.com/stable/package_esp8266com_index.json
dl.espressif.com/dl/package_esp32_index.json
raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json



藍芽模組

型號：HC-05

尺寸：17*44*7mm

重量：3g

電壓：3.3-6V

工作電流：30mA

通訊：USART 和 TTL兼容

制式：藍芽2.0

預設藍牙名稱：HC-05

預設密碼：1234 或 0000

預設Baud Rate：9600



超聲波傳感器

名稱：超聲波傳感測距模組

型號：HC-SR04

使用電壓：DC 5V

靜態電流：小於2mA

電平輸出：高5V

電平輸出：底0V

感應角度：不大於15度

探測距離：2CM - 450CM

高精度：可達0.3CM

PCB板大小：4.5 * 2.0CM

接線方式：VCC、trig (控制端)、echo (接收端) GND



伺服馬達

名稱：伺服馬達

型號：SG90

使用電壓：DC 4.8V - 6V

靜態電流：小於300mA

角度：最大270度

大小：23mm * 12.5mm * 29.5mm

接線方式：VCC、GND、SIGNAL



紅外線避障模組

名稱：紅外線避障模組

型號：fc-51

芯片：LM393

使用電壓：DC 3V - 5V

靜態電流：小於2mA

電平輸出：高5V

電平輸出：底0V

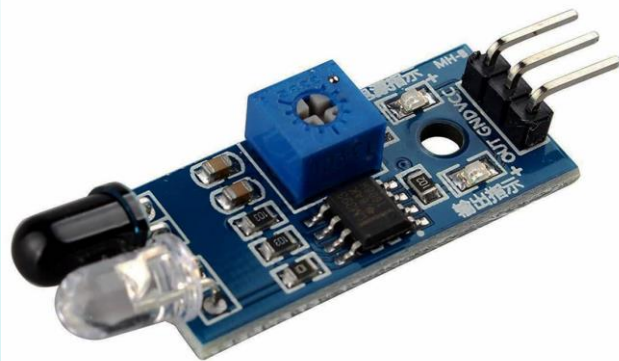
感應角度：35°

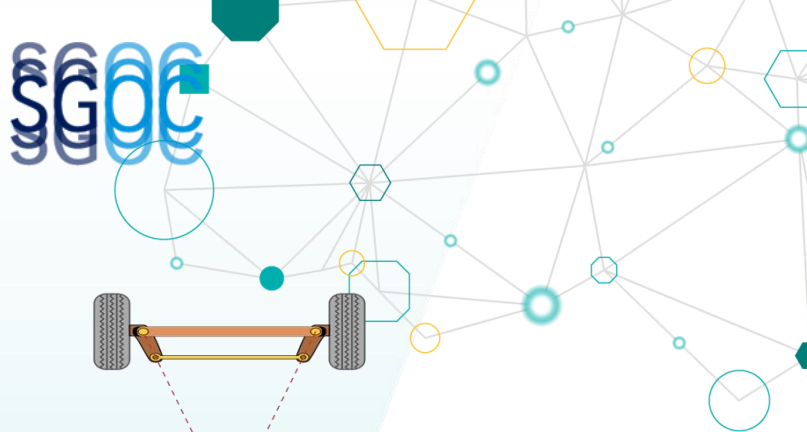
探測距離：2CM - 30CM

高精度：可達0.3CM

PCB板大小：3.2 * 1.4CM

接線方式：VCC、GND、SIGNAL





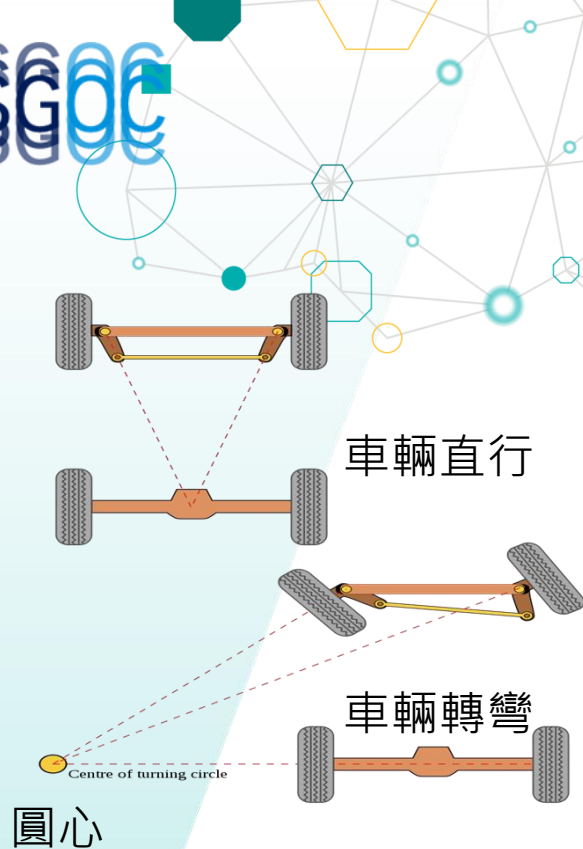
阿克曼轉向

阿克曼轉向(Ackermann Steering)

1. 車輛轉彎時通過調整車輛的轉向角度
2. 內側車輪的轉彎角度 > 外側車輪
3. 令所有車輪垂線均指向圓心



轉彎更順暢

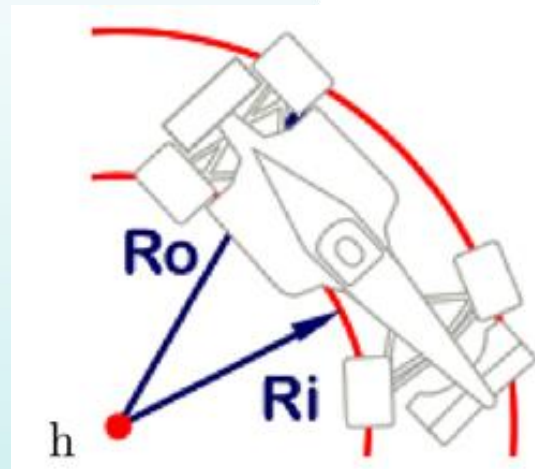




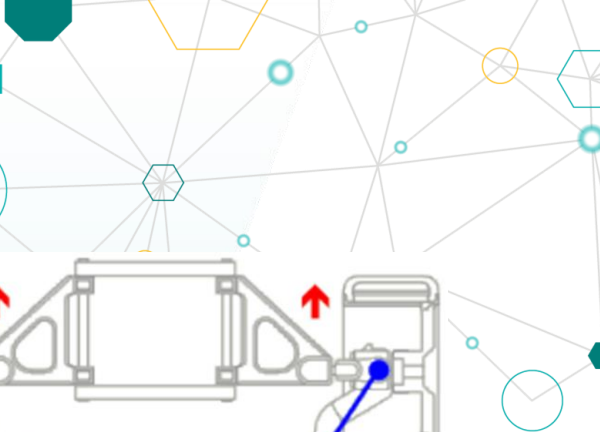
阿克曼轉向原理

當車輛轉彎時

兩個前輪以不同圓半徑轉向

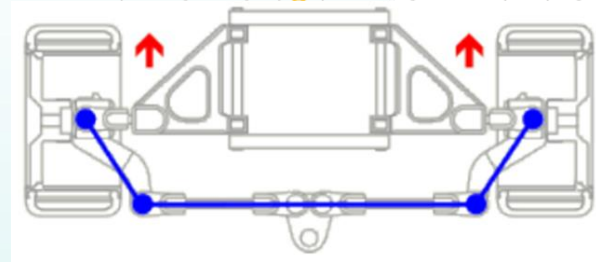


前輪半徑



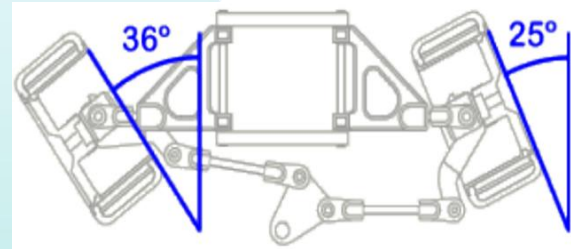
阿克曼轉向原理

車輛的轉向臂會向內傾斜



傾斜轉向臂

車輪角度以不同的速率轉向

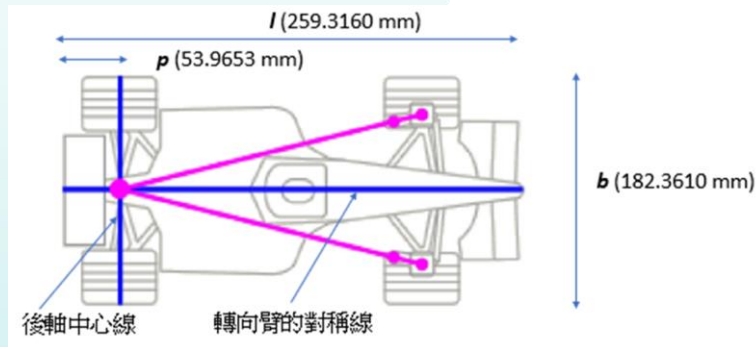


轉向角度不一



阿克曼轉向原理

1. 調整車輛轉向臂的角度
2. 轉向臂的對稱線和後軸中心線相交
3. 量度最佳的弧線半徑



車輛轉向臂的對稱線&後軸中心線相交

馬達驅動器

名稱：DC 馬達驅動器

型號：L298N

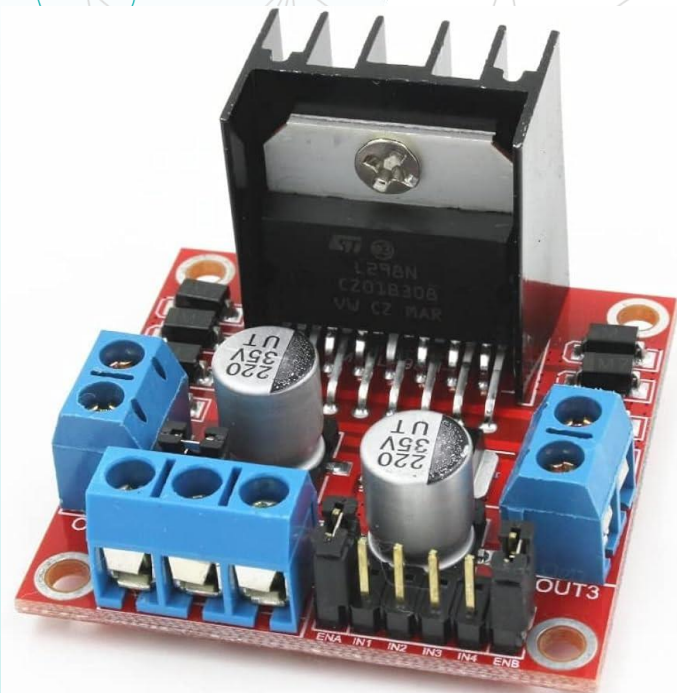
寬電源電壓範圍：3.2V - 40V

輸入電流：2A

輸出電流：600 mA

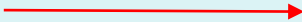
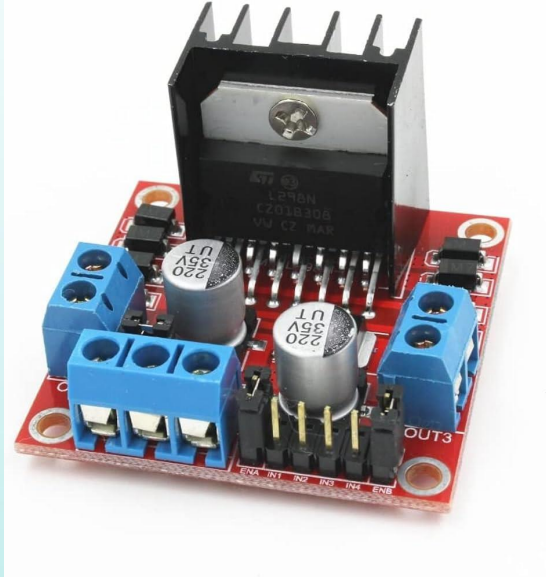
運作供電電流：2-36 mA

可控制最多兩個DC馬達





馬達驅動器



升壓模組

名稱：2A DC-DC升壓模組

型號：MT3608

最大輸出電流:2A

輸入電壓:2V ~24V

最大輸出電壓:>28V

效率:>93%

產品尺寸:36毫米* 17毫米* 14毫米



降壓模組

名稱：2A DC-DC降壓模組

型號：LM296S

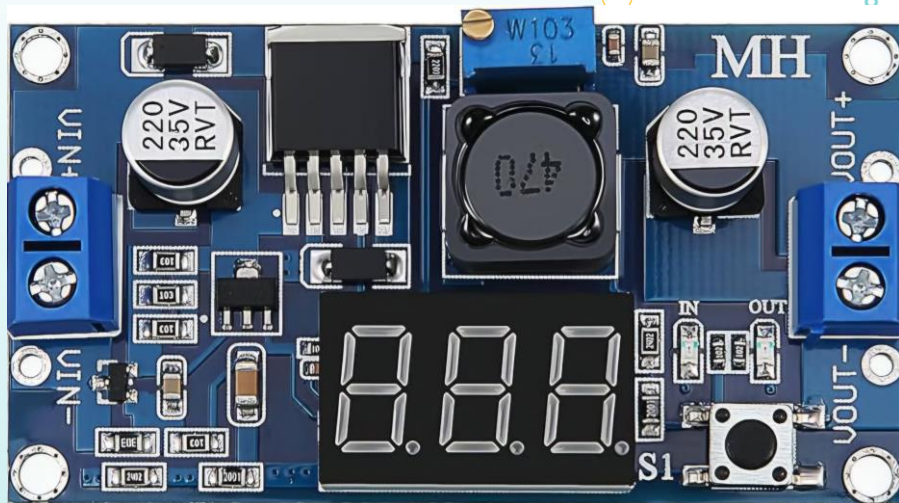
最大輸出電流:2A-3A

輸入電壓:4V-40V

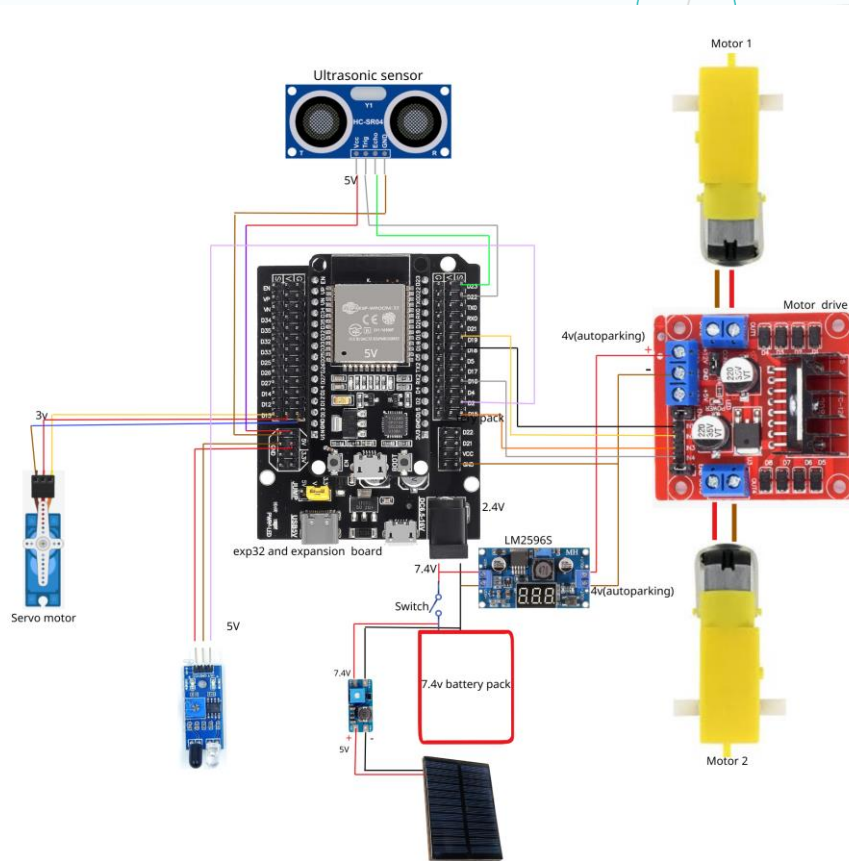
最大輸出電壓:1.25V-37V

效率:>92%

產品尺寸:17.8毫米* 66毫米* 17.8毫米



連接圖





小休

貼士程式碼

```
//引入數據庫
#include "BluetoothSerial.h"
#include <ESP32Servo.h>
//藍芽設定
BluetoothSerial SerialBT;
//馬達驅動器設定
int motorFoward = 5; //馬達向前
int motorBack = 18; //馬達向前
//伺服馬達設定
#define PIN_Servo 13 //SG90 DATA PIN
Servo myServo;
//超聲波模組設定
int trig = 22; //發送超聲波腳
int echo = 23; //接收超聲波腳
```

```
//聲音與距離參數
#define SOUND_SPEED 0.034
long duration_us;
float distanceCm;
bool autoparking = false;
//ESP32-WROOM-32 設定
void setup() {
  Serial.begin(115200);
  SerialBT.begin("ESP32"); //遙控車名
  pinMode(motorFoward, OUTPUT);
  pinMode(motorBack, OUTPUT);
  myServo.attach(PIN_Servo);
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
}
```

貼士程式碼

//藍芽控制

```
void readBluetoothData() {  
  if (SerialBT.available()) {  
    char c = SerialBT.read();  
    SerialBT.println(c);  
  }  
}
```

```
if (c == '8') { //向前  
  Serial.println("前");  
  digitalWrite(motorForward, HIGH);  
  digitalWrite(motorBack, LOW);  
} else if (c == '2') { //向後  
  Serial.println("後");  
  digitalWrite(motorForward, LOW);  
  digitalWrite(motorBack, HIGH);  
} else if (c == '4') { //向左  
  Serial.println("左");  
  myServo.write(60); //可更改角度  
} else if (c == '6') { //向右  
  Serial.println("右");  
  myServo.write(120); //可更改角度  
}
```

```
else if (c == '0') { //自動泊車  
  Serial.println("泊車中");  
  Autoparking();  
} else if (c == '9') { //停止  
  Serial.println("停");  
  myServo.write(90);  
  digitalWrite(motorForward, LOW);  
  digitalWrite(motorBack, LOW);  
}  
}  
}
```



貼士程式碼

```
//CAR SELF-PARKING PROGRAM  
void Autoparking(void) {  
  autoparking = true;  
  while (autoparking == true) {  
    digitalWrite(motorFoward, HIGH);  
    digitalWrite(motorBack, LOW);
```

```
    if (distanceCm <= 3.5) { //根據情況更改  
      digitalWrite(motorFoward, LOW);  
      digitalWrite(motorBack, LOW);  
      delay(10);  
      myServo.write(90);  
      digitalWrite(motorFoward, HIGH);  
      digitalWrite(motorBack, LOW);  
      delay(700); //根據情況更改  
      digitalWrite(motorFoward, LOW);  
      digitalWrite(motorBack, LOW);  
      delay(100);  
      myServo.write(140); //可更改角度  
      delay(100);  
      digitalWrite(motorFoward, LOW);  
      digitalWrite(motorBack, HIGH);  
      delay(2600); //根據情況更改
```

```
      digitalWrite(motorFoward, LOW);  
      digitalWrite(motorBack, LOW);  
      delay(100);  
      myServo.write(40); //可更改角度  
      delay(100);  
      digitalWrite(motorFoward, LOW);  
      digitalWrite(motorBack, HIGH);  
      delay(1900); //根據情況更改  
      digitalWrite(motorFoward, LOW);  
      digitalWrite(motorBack, LOW);  
      autoparking = false;  
      break;  
    }  
  }  
}
```



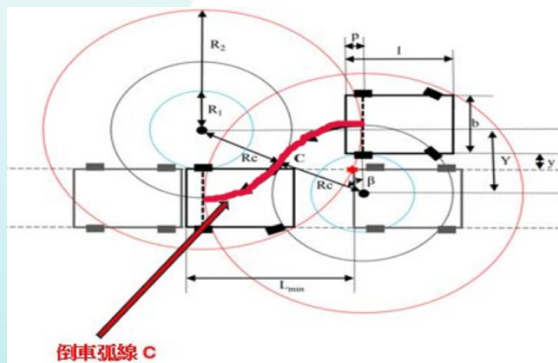
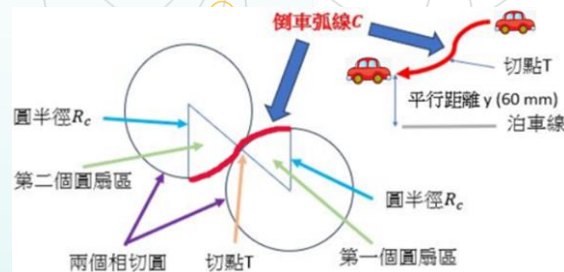
貼士程式碼

```
//ESP32-WROOM-32 SET UP
void loop() {
  readBluetoothData();
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  duration_us = pulseIn(echo, HIGH);
  distanceCm = duration_us * SOUND_SPEED / 2; // 計算距離
  Serial.print("右前 (cm): ");
  Serial.println(distanceCm);
  delayMicroseconds(10);
  delay(10);
}
```

超聲波模組運行

泊車路徑

1. 車輛泊車時先與泊車線保持平行距離 Y
2. 倒車並沿著第一個圓扇區的圓周行至切點 T
3. 車輪會轉向相反的角度
4. 繼續沿著第二個圓扇區的圓周倒車直至與泊車線平行為止
5. 倒車路徑 C 是一條弧線由二條以圓半徑為 R_c 相切圓的兩個圓扇區所結合的弧線



泊車路徑的距離



公式計算

假設這輛模型車是一個長方形，長度是 l mm，闊度是 b mm，則最小泊車長度計算如下：

1.

$$L_{min} = p + \sqrt{R_2^2 - R_1^2}$$

其中 R_1 和 R_2 的計算如下：

2.

$$R_1 = R_c - \frac{b}{2}$$

和

3.

$$R_2 = \sqrt{\left(R_c + \frac{b}{2}\right)^2 + (l - p)^2}$$

計算結果是最小泊車長度約為 400 毫米，或者是這輛模型車長度的 1.54 倍。

這輛模型車沿圓弧線倒車時所行駛的距離 C 計算如下：

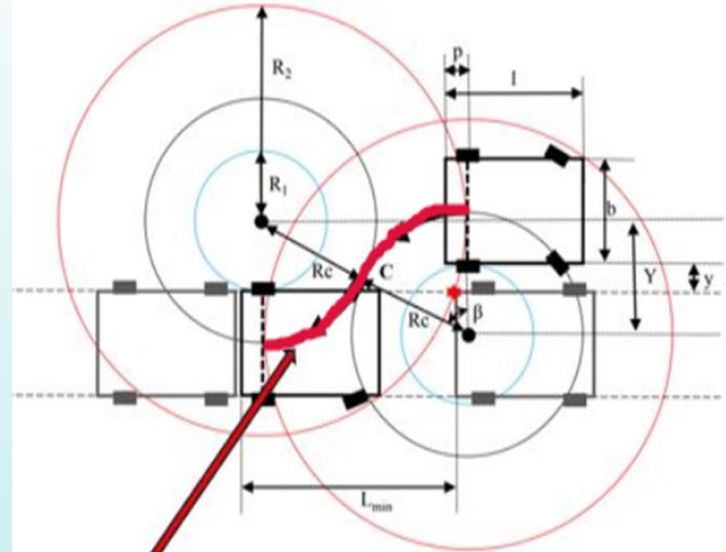
4.

$$C = 2\beta R_c$$

其中 β 角度的計算公式如下：

5.

$$\beta = \cos^{-1}\left(\frac{Y}{2R_c}\right)$$

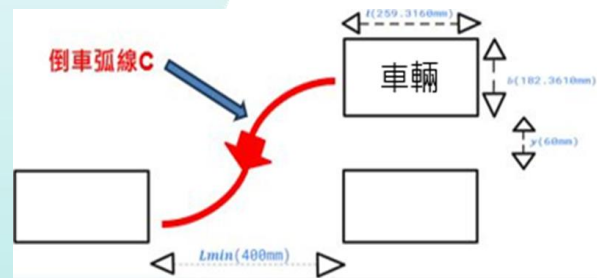
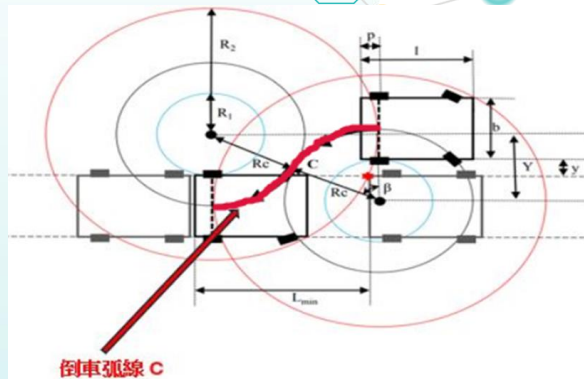


倒車弧線 C

數值

數值的代表：

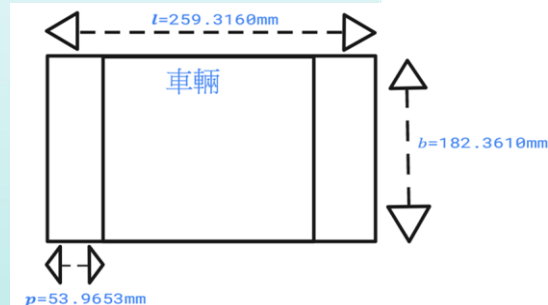
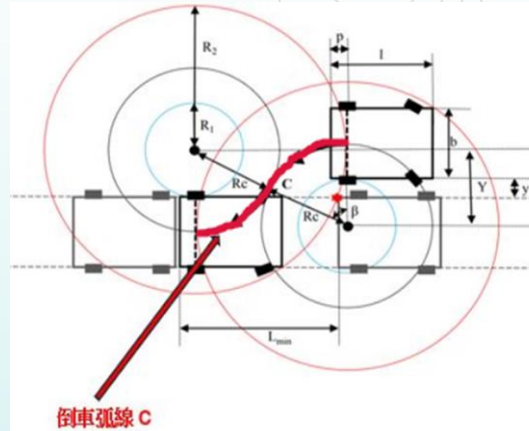
- l 代表車輛長度。
- b 代表車輛闊度。
- p 代表車輛橫軸與車尾距離。
- L_{min} 代表最小泊車長度。
- R_1 代表最細圓的半徑。
- R_2 代表最大圓的半徑。
- R_c 代表中間圓的半徑。
- Y 用於計算 β 角度。
- y 代表車輛與泊車位的平行距離。



數值

設數值:

- l = 259.3160 mm
- b = 182.3610 mm
- p = 53.9653 mm
- L_{min} = 400 mm
- R_1 = 121.5064 mm
- R_2 = 366.7482 mm
- R_c = 212.6869 mm
- Y = 195.8692 mm
- y = 60 mm



計算公式

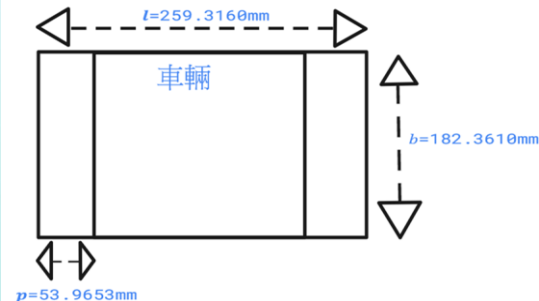
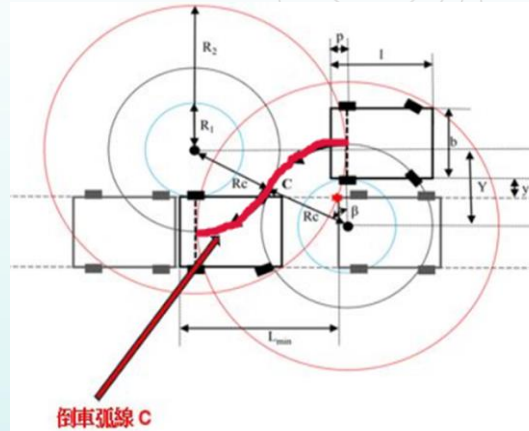
計算 R_1 的數值

$$R_1 = R_c - \frac{b}{2}$$

$$\text{L.H.S} = R_1 = 121.5064 \text{ mm}$$

$$\begin{aligned} \text{R.H.S} &= R_c - \frac{b}{2} \\ &= 212.6869 - \frac{182.3610}{2} \\ &= 121.5064 \text{ mm} \end{aligned}$$

因此，L.H.S = R.H.S



計算公式

計算 R_2 的數值

$$R_2 = \sqrt{\left(R_c + \frac{b}{2}\right)^2 + (l - P)^2}$$

$$\text{L.H.S} = R_2 = 366.7482 \text{ mm}$$

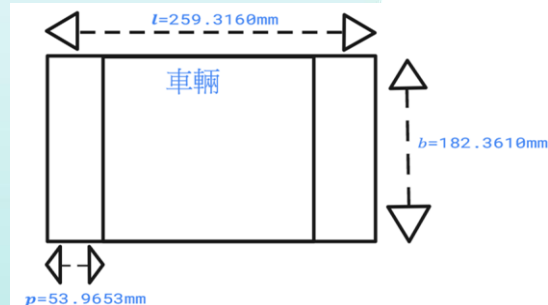
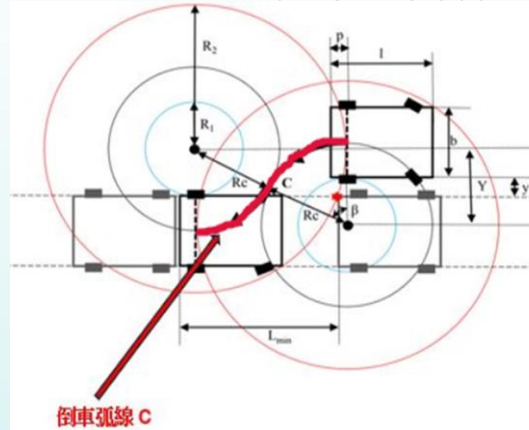
$$\text{R.H.S} = \sqrt{\left(R_c + \frac{b}{2}\right)^2 + (l - P)^2}$$

$$= \sqrt{\left(212.6869 + \frac{182.3610}{2}\right)^2 + (259.3160 - 53.9653)^2}$$

$$= 366.7482880304 \text{ mm}$$

$$\approx 366.7482 \text{ mm}$$

因此，L.H.S = R.H.S



計算公式

計算 L_{min} 的數值

$$L_{min} = p + \sqrt{R_2^2 - R_1^2}$$

$$\text{L.H.S} = L_{min} = 400 \text{ mm}$$

$$\text{R.H.S} = p + \sqrt{R_2^2 - R_1^2}$$

$$= 53.9653 + \sqrt{(366.7482)^2 - (121.5064)^2}$$

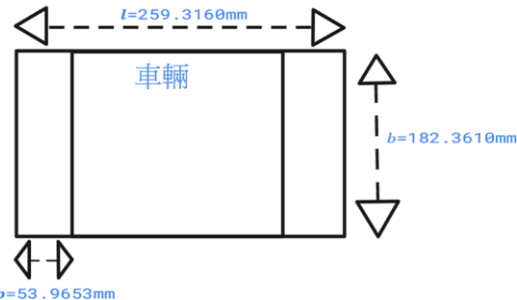
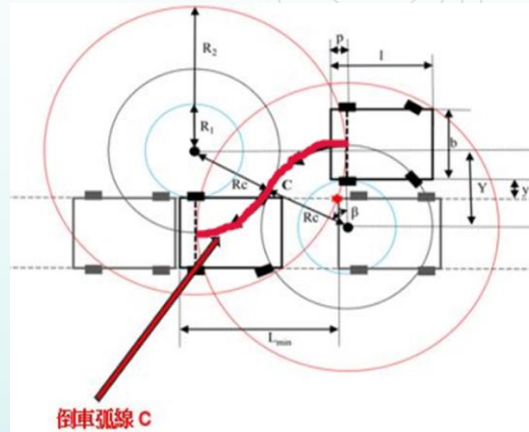
$$= 400.00061172 \text{ mm}$$

$$\approx 400 \text{ mm}$$

因此，L.H.S = R.H.S

備註：最小泊車長度(L_{min})的值是400 mm或約等於1.54倍車輛長度(l),

$$\text{即 } L_{min} = 1.54l = 1.54 \times 259.3160 = 399.34664 \text{ mm} \approx 400 \text{ mm}$$



$p = 53.9653 \text{ mm}$

計算公式

計算 β 角度的數值

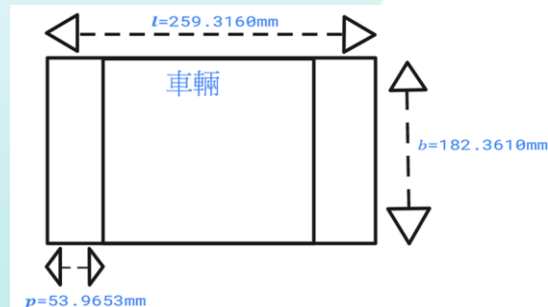
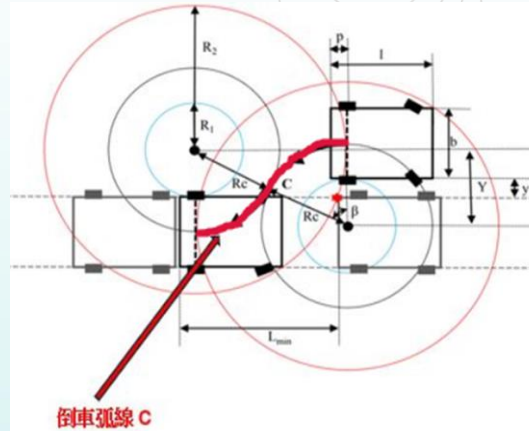
$$\beta = \cos^{-1} \left(\frac{Y}{2R_c} \right)$$

$$\beta = \cos^{-1} \left(\frac{195.8692}{2 \times 212.6869} \right)$$

$$= \cos^{-1}(0.4604637145)$$

$$= 1.09227881 \text{ rad 或 } 62.582966^\circ$$

$$\approx 1.0923 \text{ rad 或 } 62.583^\circ$$

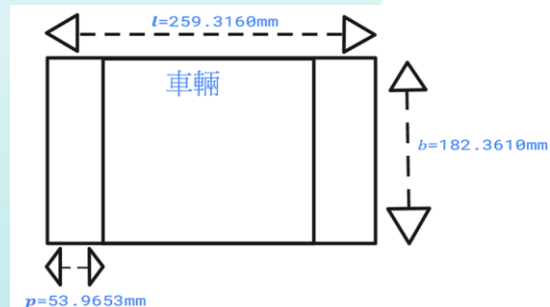
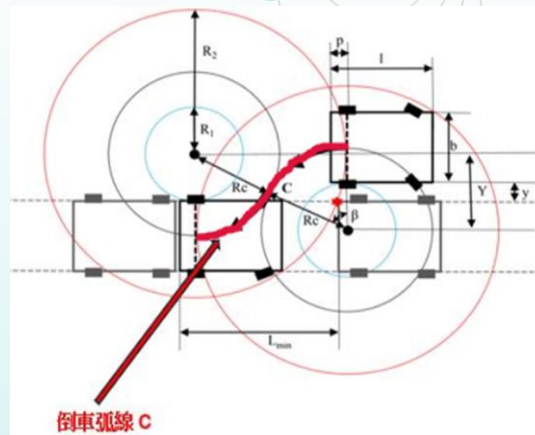
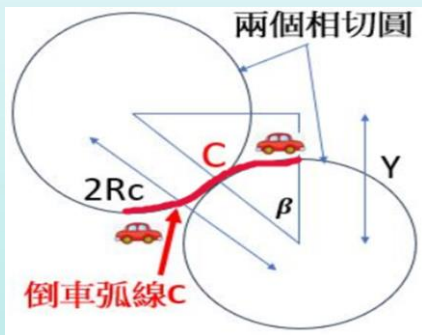


計算公式

計算弧線長度C的數值

$$C = 2\beta R_c$$

$$\begin{aligned} C &= 2 \times 1.09227881 \times 212.6869 \\ &= 464.6267880692 \text{ mm} \\ &= 0.4646 \text{ m} \end{aligned}$$



如何控制

Google Play 遊戲 應用程式 電影 圖書 兒童

Arduino Bluetooth Control

broxcode

含廣告內容

10萬+
次下載

3+
3歲以上

安裝在更多裝置上

分享

這個應用程式與你的裝置不相容



如何控制

- 可自訂按鍵
- 實現到基本控制
- 不支援IOS



如何控制



Arduino Car

ONE DAY OF CODE

包含廣告 • 應用程式內購買

超過 500K 次

下載次數

3+

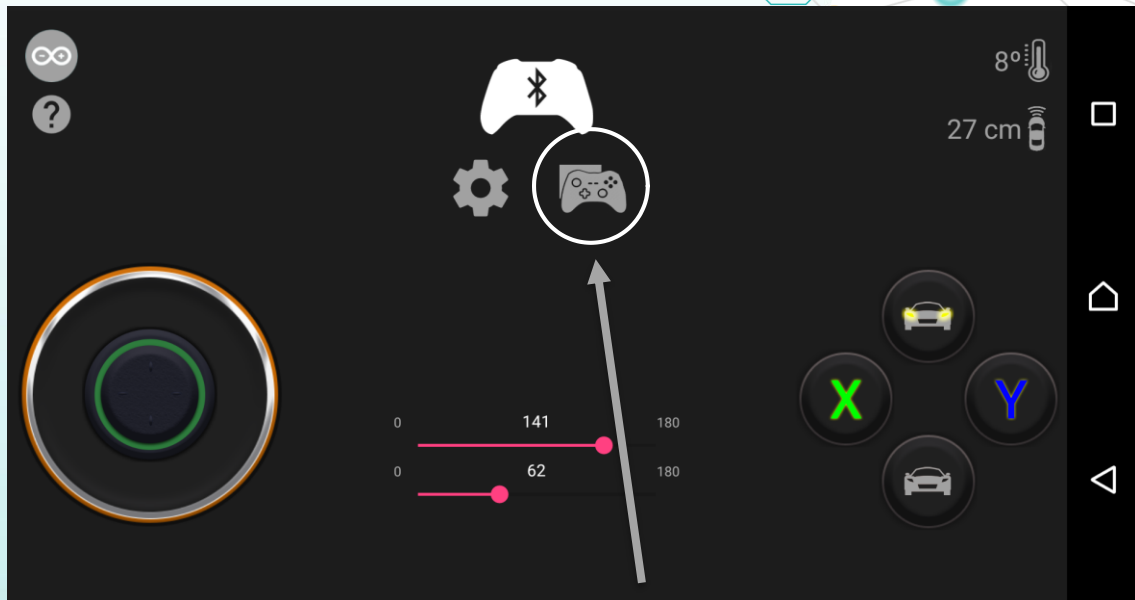
3 歲或以上 ⓘ

解除安裝

開啟

如何控制

- 可自訂按鍵
- 實現到基本控制
- 不支援IOS

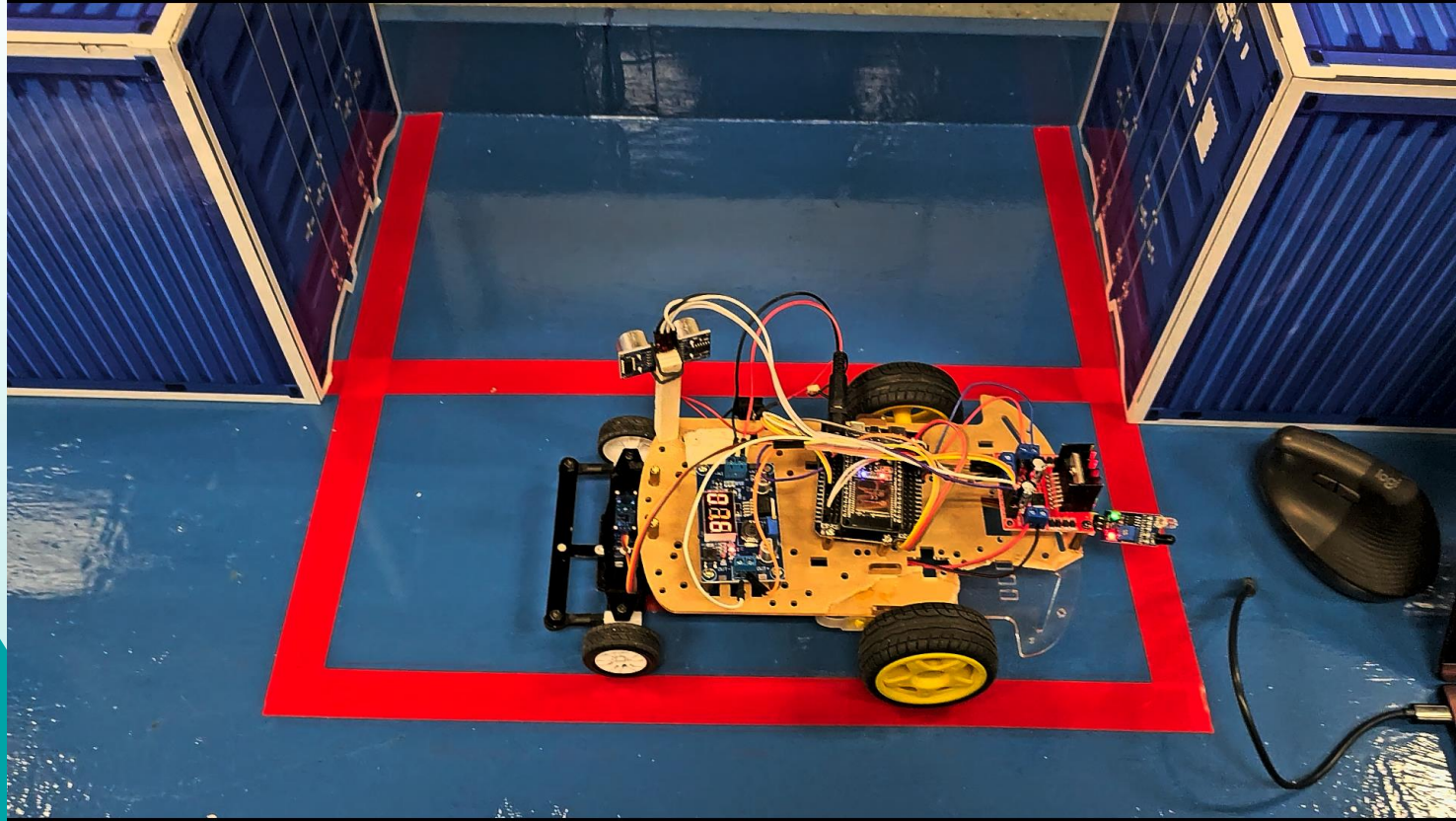


如何控制

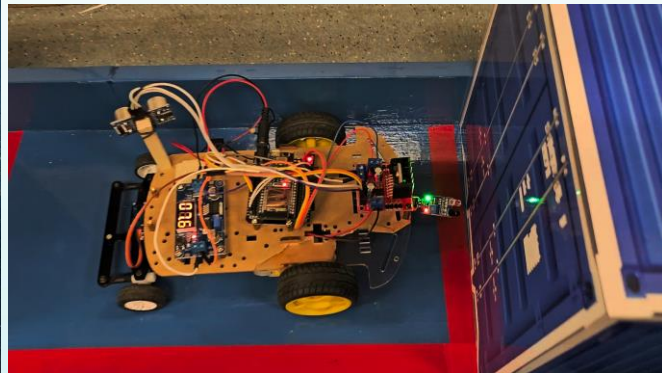
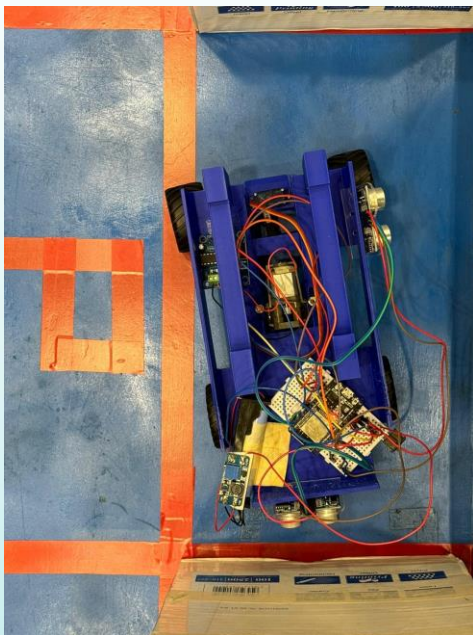
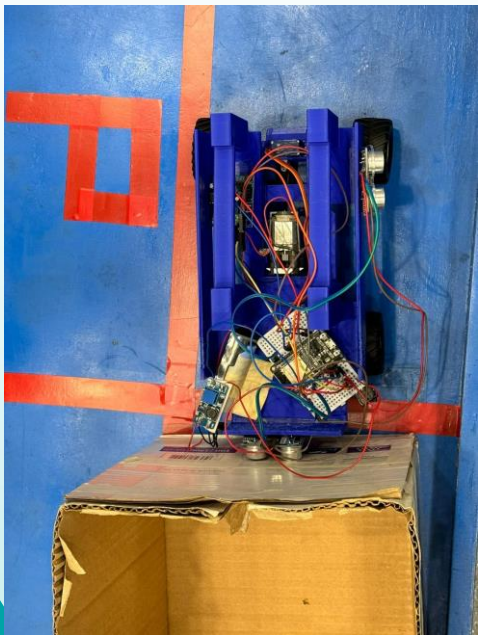
- 可自訂按鍵
- 實現到基本控制
- 不支援IOS



播放泊車示範影片



成功





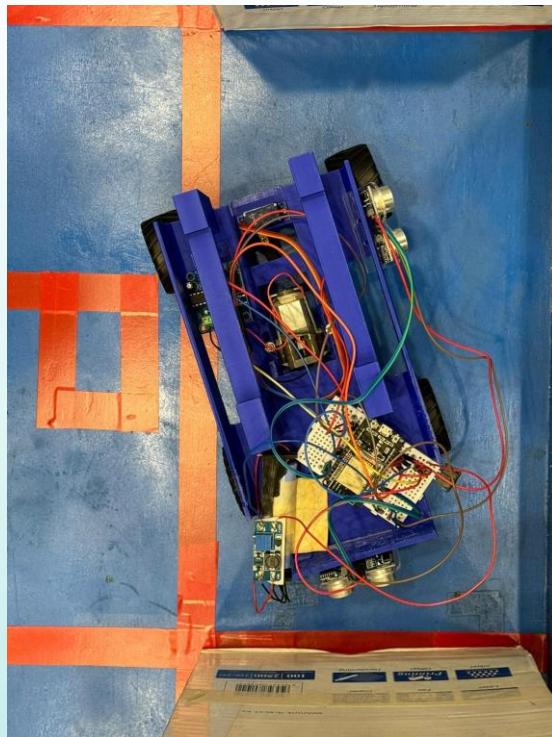
IVE

Engineering
工程

SOPHIE



不成功



建議

- 下一堂前，請大家分組買相關細車材料開始砌車

最新資訊

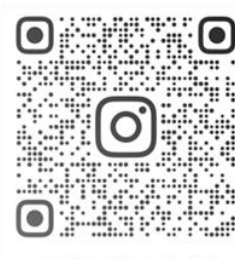


VTC Solar Car Team

Smart Grid Operation Centre



Instagram



@SGOC.IVE



亮