

#### 111-2 電資工程入門設計與實作

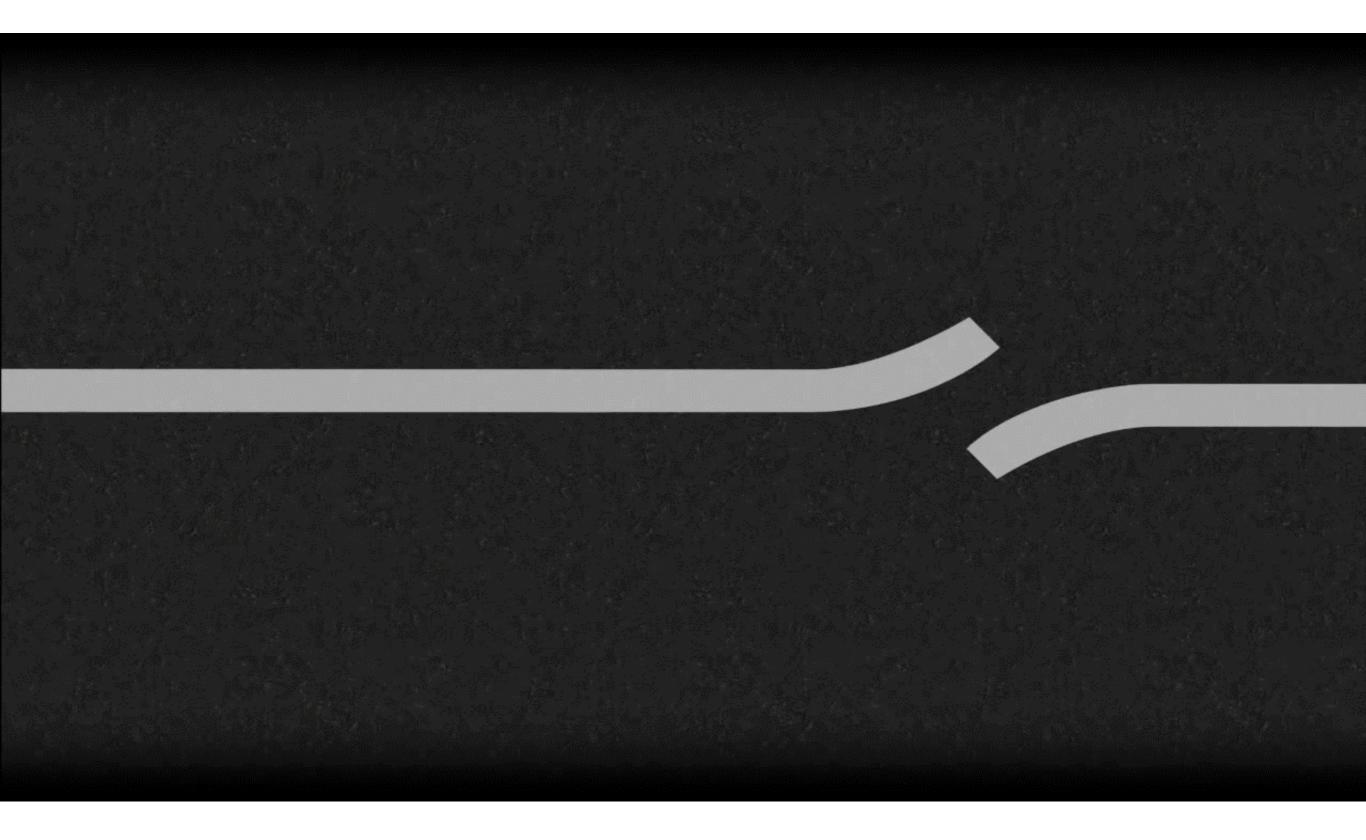
# Unit 6 循跡自走車

授課教師:陳士元

課前準備:劉容均、鄧笙敔、謝明圜



#### 2013 University of Manchester, UK



#### 2019 Japan Robotrace Contest 1st Prize Winner



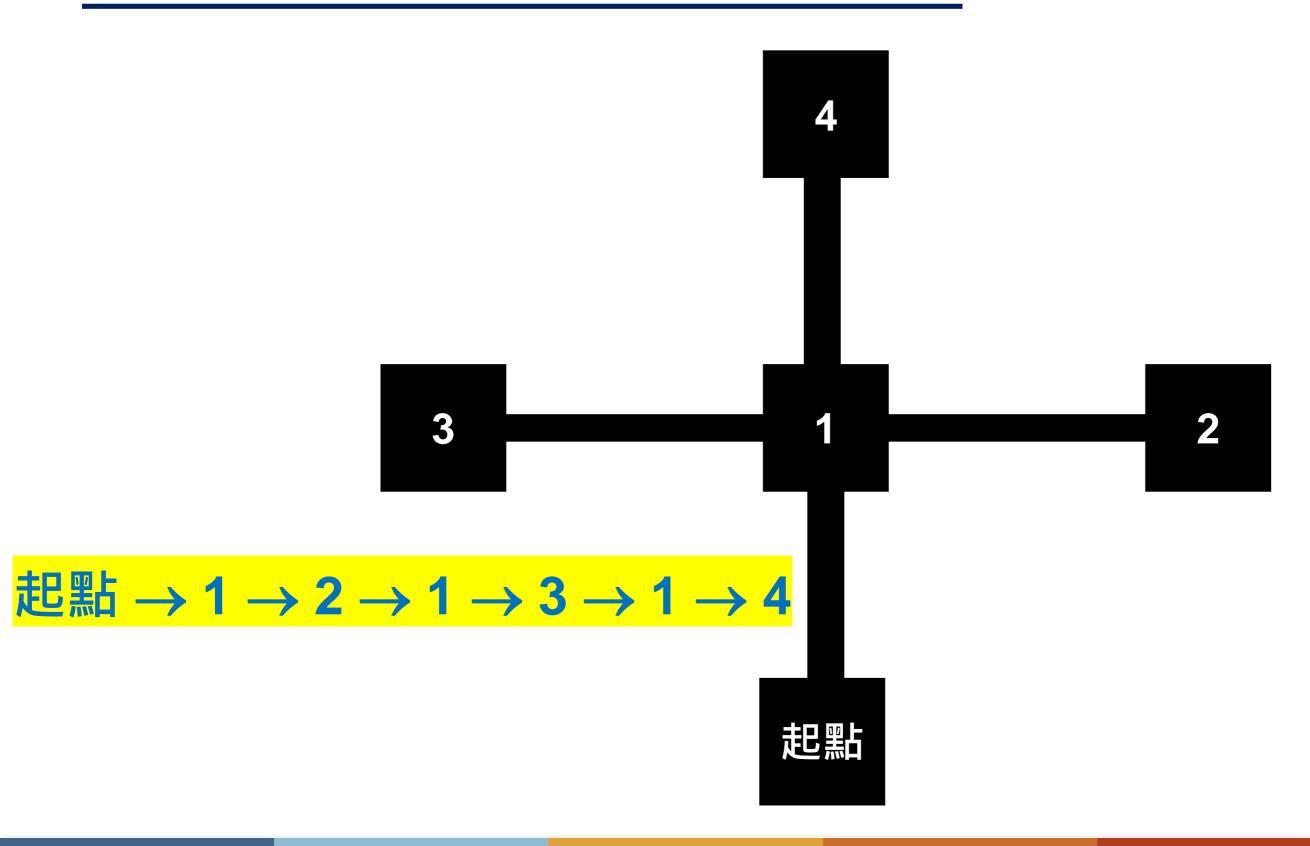
## 學習目標

了解循跡控制的基本原理

讓車車變成循跡自走車

### 本週任務:循跡走直線、曲線

## 進階任務:十字形地圖



#### 循跡自走車必備

充飽電的電池

大腦: Arduino

眼:紅外線感測器

腳:馬達

系統整合

循跡演算法

#### 循跡自走車必備

充飽電的電池

大腦: Arduino

眼:紅外線感測器

腳:馬達

系統整合

循跡演算法

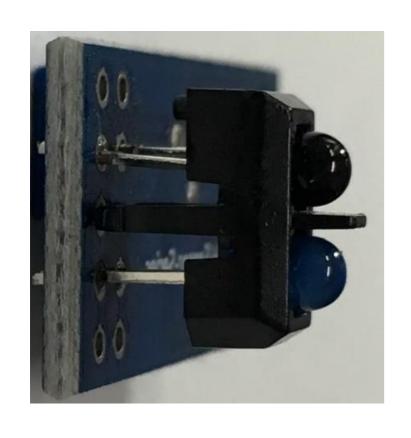
#### 紅外線感測器

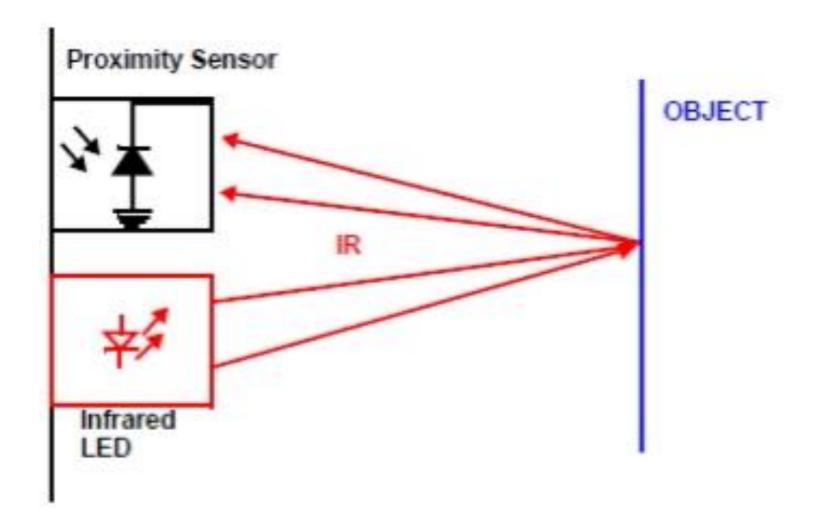
判別黑白: digitalRead()

黑色吸光、反射少 ⇒ 輸出高電位

白色反光、反射多 ⇒ 輸出低電位



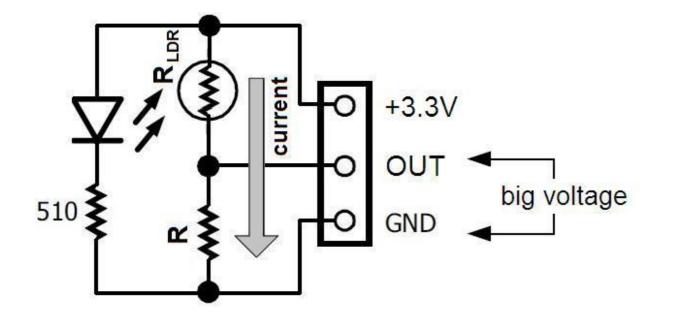


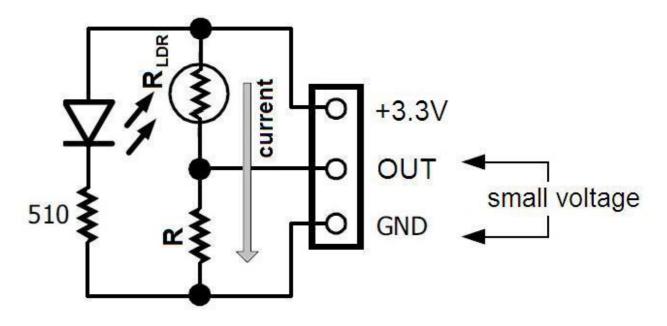


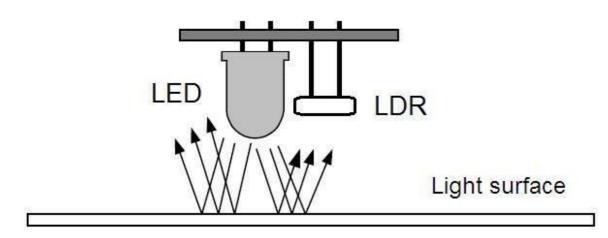
#### 紅外線感測器:延伸閱讀

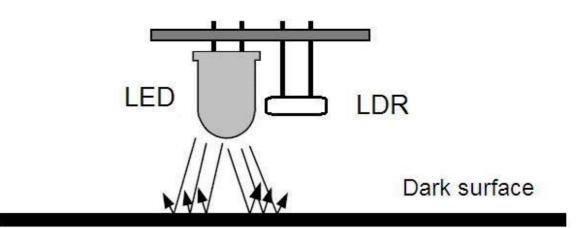
#### IR LED

#### Light Dependent Resistor (LDR)







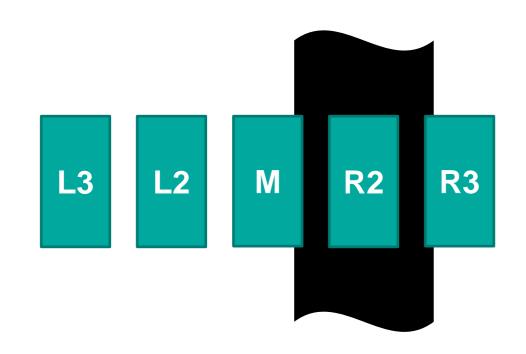


#### 紅外線感測器:測試與調整

<mark>測試</mark> 當車上的五顆紅外線感測器分別位於地圖黑 線的正上方時,是否都能偵測到?

如何確認?

如何調整靈敏度?





### 循跡自走車必備

充飽電的電池

大腦:Arduino

眼:紅外線感測器

腳:馬達

系統整合

循跡演算法

#### 馬達:左右輪控制函式

MotorWriting(vL, vR) 指定左右輪馬達的轉速/轉向 double vL 左輪馬達的PWM及正反轉方向 double vR 右輪馬達的PWM及正反轉方向

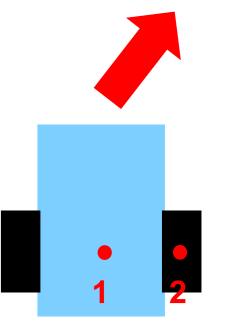
digitalWrite 指定兩個馬達的接線跟轉動方向的關係

analogWrite(MotorL\_PWML, vL) analogWrite(MotorR\_PWMR, vR)

養成Coding好習慣,別忘了檢查

在 loop() 中加入 MotorCheck()

例如:前進、後退、左轉、右轉 依序各2秒



### 馬達:左右輪控制函式

MotorWriting(vL, vR) 範例程式碼

void MotorWriting(double vL, double vR) {

analogWrite(MotorL\_PWML, vL);
analogWrite(MotorR\_PWMR, vR);

## 馬達:左右輪控制函式

完成後,請測試當 VL = VR = 200 時,車子如何行進?

#### 循跡自走車必備

充飽電的電池

大腦: Arduino

眼:紅外線感測器

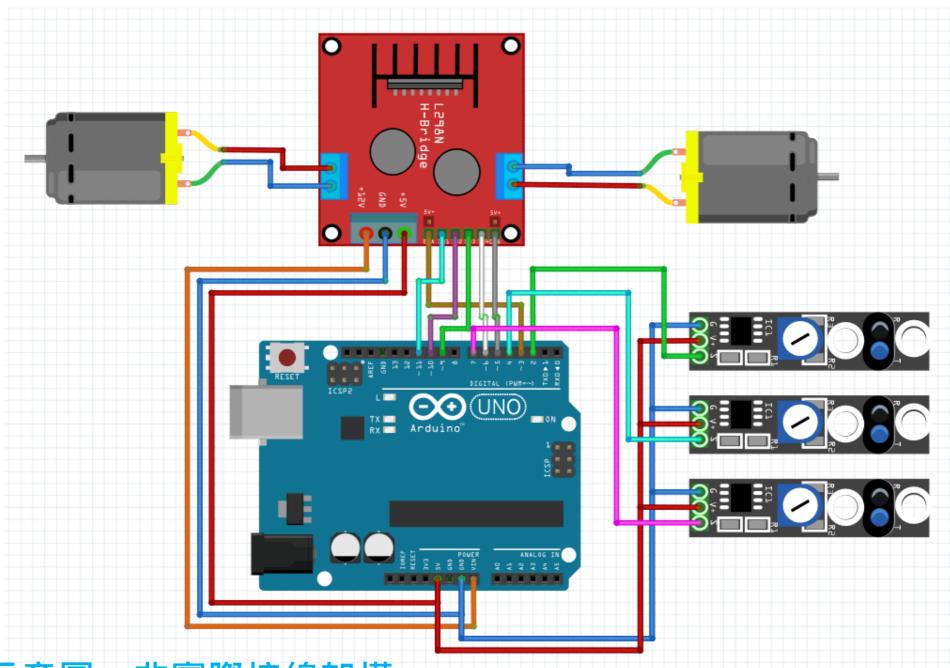
腳:馬達

系統整合

循跡演算法

### 循跡子系統整合

第一步:完成前面兩部分



此為示意圖,非實際接線架構

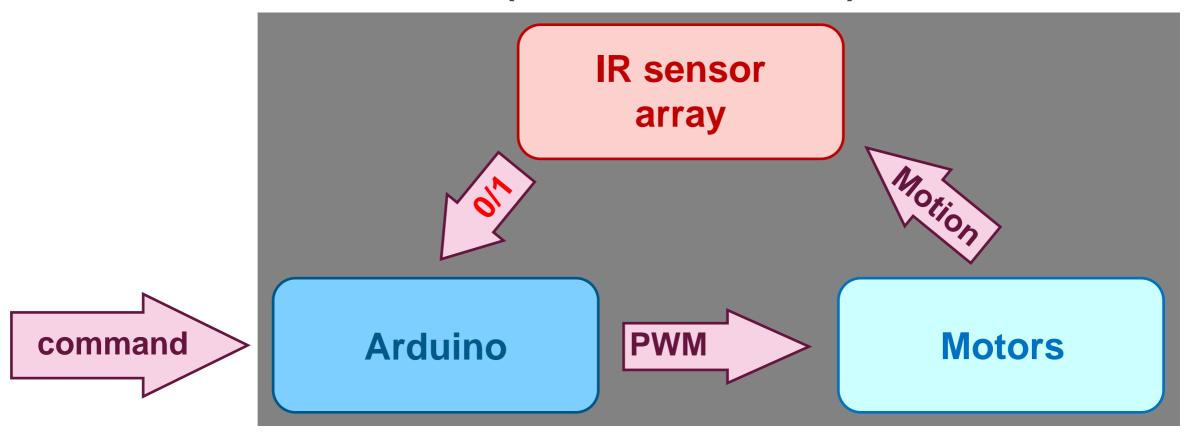
### 循跡子系統整合

第二步:利用 IR感測器的讀值 及 MotorWriting(vL, vR) 完成循跡控制。

第三步: Debug & Fine-tune

這就是迴授控制!

(Feedback control)



#### 循跡自走車必備

充飽電的電池

大腦: Arduino

眼:紅外線感測器

腳:馬達

系統整合

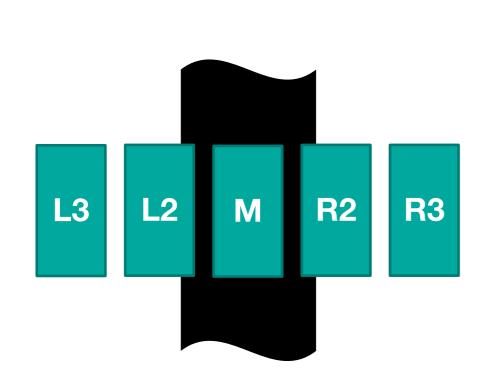
循跡演算法

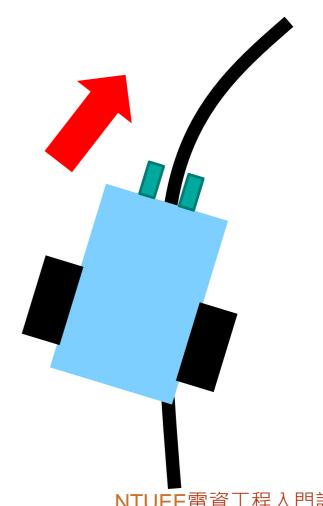
## 循跡演算法

假設車車硬體是完美的,如何讓它循跡?

請設計一個流程 (流程圖或pseudo code) 依照當下五 個IR感測器的讀值組合,調整車身運動狀態

請各組討論並於工作紀錄簿寫下你們的想法





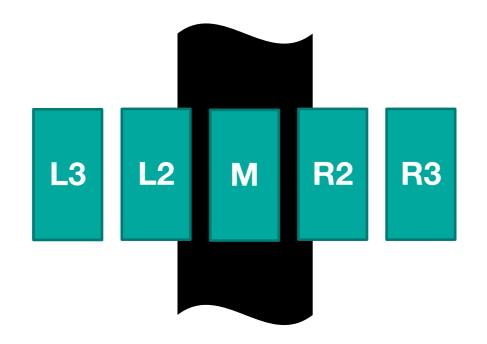
### 循跡演算法

假設車車硬體是完美的,如何讓它循跡?

試試看「窮舉法」

If (...) {...} else if (...) {...} else if (...) {...} else if ...

請依據你們的想法寫成code教Arduino該怎麼做完成後請試走直線(拼圖地圖) 並錄影紀錄



## 循跡演算法 範例程式碼

```
void Tracking() {
   int 13 = digitalRead(L3);
   int 12 = digitalRead(L2);
   int m = digitalRead(M);
   int r2 = digitalRead(R2);
   int r3 = digitalRead(R3);
```



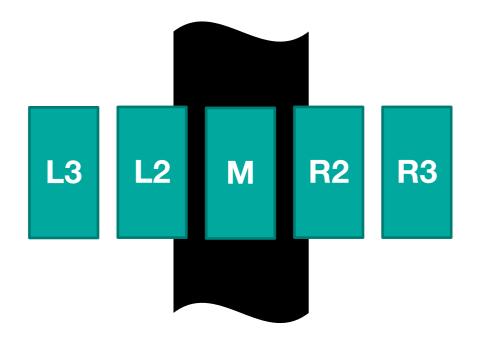
### 循跡演算法

假設車車硬體是完美的,如何讓它循跡?

除了「窮舉法」

If (...) {...} else if (...) {...} else if (...) {...} else if ...

也可根據當下的位置偏差量計算決定車身運動方式



#### 循跡演算法

如何利用這五個感測器當下的讀值計算決定vL、vR?

定義一個描述循跡線偏移量的參數 error

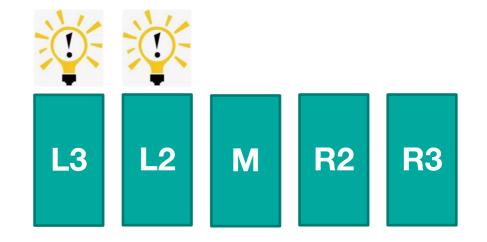
以 I3, I2, m, r2, r3 的讀值來定義

由 error 計算得到 vL, vR

MotorWriting(vL, vR)

### 循跡演算法 範例程式碼

```
void Tracking() {
    int 13 = digitalRead(L3);
    int 12 = digitalRead(L2);
    int m = digitalRead(M);
    int r2 = digitalRead(R2);
```



## 循跡演算法

這樣的循跡控制方式稱為 P control (Proportional)

#### 測試與觀察

調整 vL, vR 的修正量跟 error 之間的比例常數 Kp (100) 調整直行時的車速 Tp (150)

請錄影記錄不同參數的狀況,並比較討論

除了循跡走直線,請換成橢圓形地圖試試看!

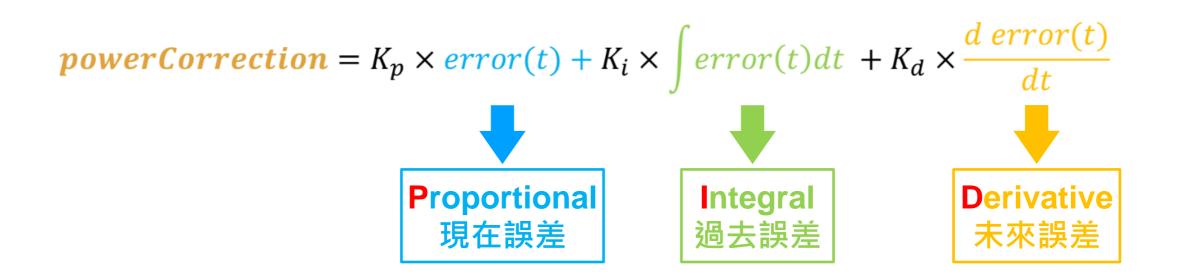
#### 進階循跡: PID control

#### P control 是 PID control 的一個特例

Integral: 考慮過去error的加總 (過去誤差)

Derivative: 考慮與前次error之差值 (誤差趨勢 ⇒ 未來誤差)

調整馬達的出力: Tp ± powerCorrection



#### 進階循跡: PID control

#### PD control 是 PID control 的另一個特例

Derivative: 考慮與前次error之差值 (誤差趨勢 ⇒ 未來誤差)

#### 理論上

```
\frac{d\;error(t)}{dt} = \lim \frac{error(t+\delta t) - error(t)}{\delta t} \approx \frac{error(t+\Delta t) - error(t)}{\Delta t}
```

 $\Delta t$  更新error的時間間隔

#### 實作時

```
double Kd; // 參數,手動調整
double lastError; // 前次偏移誤差
double dError = error - lastError;
double powerCorrection = Kp*error + Kd*dError;
```

#### 進階循跡: PID control

#### PID control

Integral: 考慮過去的error的加總 (過去誤差)

#### 理論上

$$sumError = \int_{t_1}^{t_2} error(t)dt \approx \Delta t \sum_{t=t_1}^{t_2} error(t)$$

誤差可以從t=0開始累積或過去一段時間的誤差加總

#### 實作時

```
double Ki; // 參數,手動調整,先不要調太大
double sumError; // 累計偏移誤差
double sumError += error; // 每個loop都要更新一次
double powerCorrection = Kp*error + Kd*dError
+ Ki*sumError;
```

# 本週任務

車子循跡走直線(拼圖地圖),錄影上傳Youtube

車子正反各繞橢圓形地圖兩圈,錄影上傳

Bonus: 車子連續自動來回走完十字拼圖地圖,錄影上傳當走到 T 字形死巷 ⇒ 判斷是死巷,然後迴轉

Deadline: 4/8(六)下午五點 (與本週工作紀錄簿死線相同)

接下來2次Open lab: 3/29(三)晚上、4/8(六)上午

### 本週Checklist

分數	評分標準	
1	MotorWriting 正常	
2	輪子轉速會隨 IR 變化	
3	直線循跡	
4	正反各繞橢圓形地圖兩圈 正走兩圈、手動迴轉、反走兩圈	
Bonus	完整走完十字地圖 起點 $\rightarrow$ 1 $\rightarrow$ 2 $\rightarrow$ 1 $\rightarrow$ 3 $\rightarrow$ 1 $\rightarrow$ 4	

# 總結

#### 循跡自走車六寶

電

巡

眼

腳

眼腳協調

循跡控制

循跡控制的兩種基本方法

窮舉法

P control ⇒ 進階循跡 PD/PID control

週次	日期	Lecture	LAB	HW	講授教師
1	2/20	果程介紹、課程提醒事項、團隊破冰 演算法簡介討論		觀看影片: 焊接、演算法理論,上 傳個人notion連結	陳和麟
2	2/27	和平紀念日(放假,不上課)	一放假三上下午依舊上課 (實體+錄影)、團建活動、工作紀 錄的重要性	工作紀錄簿撰寫與批改 (線上)	莊哲明、 李紋霞
3	3/6	演算法、Arduino	紅外線感測模組、RFID 規定分組 (2人1組)	個人工作紀錄簿開始記錄	陳和麟、 陳君朋
4	3/13	Arduino	驅動馬達、藍芽 規定分組 (2人1組)	影片觀看 上傳各組notion連結	陳君朋
5	3/20	系統設計概念、車子組裝	組車:自行分組 3人1組,至少1 人會C語言,1人會python	車子前進、 調整車子、 拼圖完成	盧奕璋、 林坤佑
6	3/27	循跡控制	循跡走直線、橢圓、十字形地圖	Unit 6 Checklist	陳士元
7	4/3	春假(放假,不上課)	3 Open Labs: 3/29 (三) 晚上 4/8 (六) 下午 Weekly report死線	1. 準備5-min進度報告 2. 完成指定題Checklist 3. 期季影片:	
8	4/10	期中考週(不上課)	4/15 (六) 下午 另,可前往NTUEE Maker Space	指定專題介紹	
9	4/17	指定專題設計與進度報告、 工程開發流程方法、Checkpoint	上台進度報告		莊哲明
10	4/24	自選題介紹說明/創新方法與流程			莊哲明
11	5/1	指定專題展示、競賽	依競賽成績排序給分		
12	5/8	自選題Proposal報告	上台報告		
13	5/15	自選題進度報告	上台進度報告		
14	5/22	自選題進度報告	上台進度報告		
15	5/29	自選題進度報告	上台進度報告		
16	6/5	期末考週(不上課)			
17	6/12				TAs
18	6/19	自選題聯合展示、競賽	6/19 (-) 13:00~17:00		

#### 進度報告準備小撇步

各組只有五分鐘,請把握重點及時間!!!

建議包含項目(不限於此)

Checklist列表,可標註已完成/進行中/待辦項目

遇到的主要問題及解決過程或預計如何解決

後續時程安排 (如甘特圖)

團隊分工情形

當場由老師、助教、同學共同評分,評分向度有

報告內容完整度

簡報表現及台風

工作規劃合理性

團隊合作

# 下次上課(4/17)前請完成:

分數	評分標準	
1	電腦有連到藍牙	
2	上行/下行至少一個成功	
3	上行/下行都成功	
4	用藍牙遙控車子(任意形式)	

分數	評分標準	
1	刷RFID後print出UID	
2	在行走中刷到RFID	
3		

分數	評分標準	
1	使用模擬地圖,從A走到B,無冗餘步驟	
2	車子在中地圖實際從A走到B,無冗餘步驟	
3	車子在中地圖實際從A走到B,並回傳B點UID給server	
4	回傳地圖中所有UID給server,且 server 有分數	