



機械人技術與人工智能(AI)編程

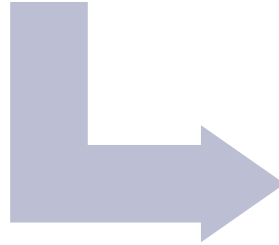


人工智能



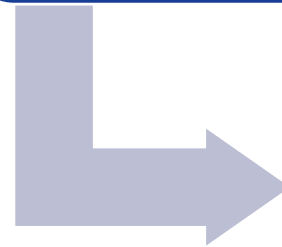
數據收集

• 傳感器



機械學習

• 智能判斷

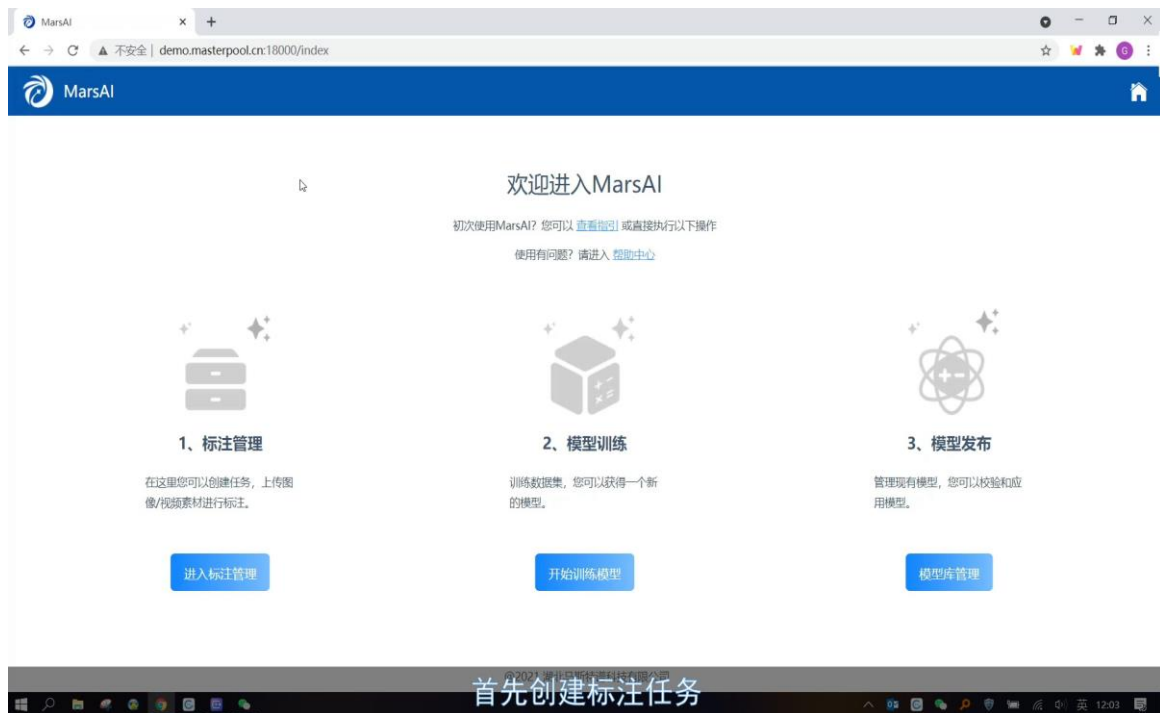


數據應用
及反饋

• 反應動作



機械學習模組製作



MarsAI

欢迎进入MarsAI

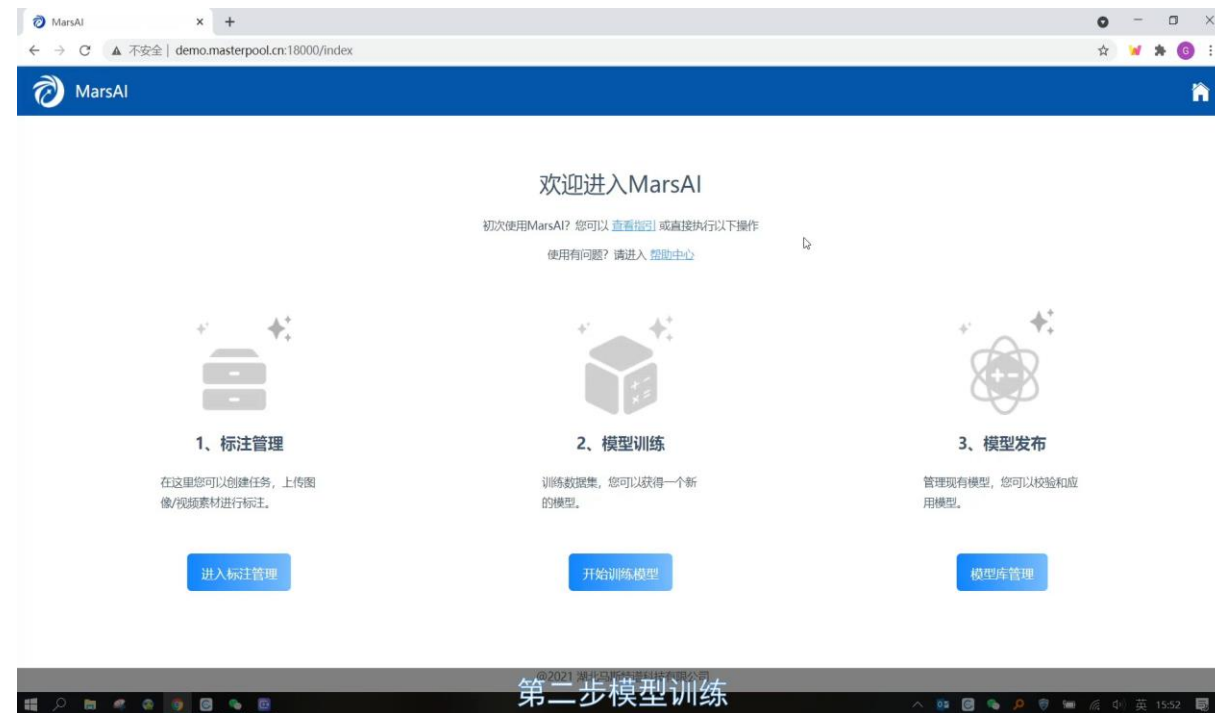
初次使用MarsAI? 您可以 [查看指引](#) 或直接执行以下操作

使用有问题? 请进入 [帮助中心](#)

- 1、标注管理**
在这里您可以创建任务, 上传图片/视频素材进行标注。
[进入标注管理](#)
- 2、模型训练**
训练数据集, 您可以获得一个新的模型。
[开始训练模型](#)
- 3、模型发布**
管理现有模型, 您可以校验和应用模型。
[模型库管理](#)

©2021 廣州阿斯克智能科技有限公司

首先创建标注任务



MarsAI

欢迎进入MarsAI

初次使用MarsAI? 您可以 [查看指引](#) 或直接执行以下操作

使用有问题? 请进入 [帮助中心](#)

- 1、标注管理**
在这里您可以创建任务, 上传图片/视频素材进行标注。
[进入标注管理](#)
- 2、模型训练**
训练数据集, 您可以获得一个新的模型。
[开始训练模型](#)
- 3、模型发布**
管理现有模型, 您可以校验和应用模型。
[模型库管理](#)

©2021 廣州阿斯克智能科技有限公司

第二步模型训练

機械學習模組製作



機械學習例子Machine Learning



機械學習例子Machine Learning

目标跟随无人机

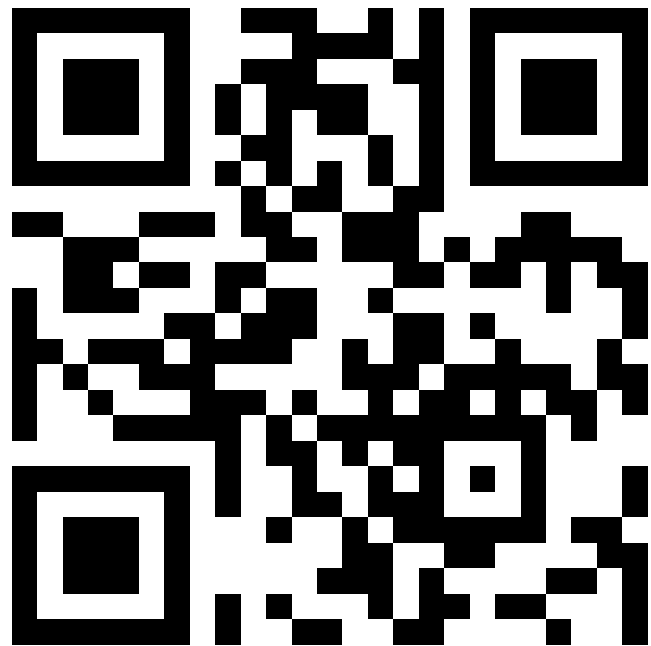


機械學習例子Machine Learning

自动识别夹取两个目标



dji EDUCATION HUB

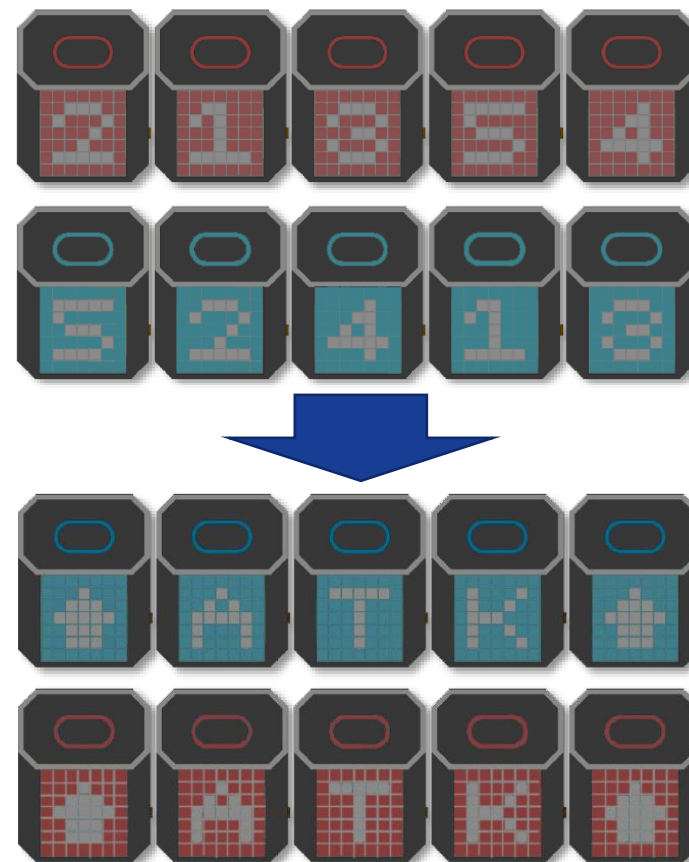


項目一

辨識任務

啟動規則

- 透過 5 個電子視覺標籤隨機產生一串由 1 ~ 5 組成的隨機數，機械人需要在 1 米以外的距離，按照 1 到 5 的順序依次擊中 5 個視覺標籤上方的裝甲模組，即可視為成功啟動。
- 啟動成功會有標記。成功啟動後，電子視覺標籤會顯示攻擊力加成視覺標籤，機械人識別到該視覺標籤後全隊獲得 2 倍攻擊力加成。



瞄準方式

- 實行方式一般分為兩種：
- **第一種是「自動瞄準模組」**（簡易）
- 第二種是「PID 控制器」（進階）

怎樣使用自動瞄準模組來瞄準

- 智能大類—>「識別到（紅心）並瞄準」。



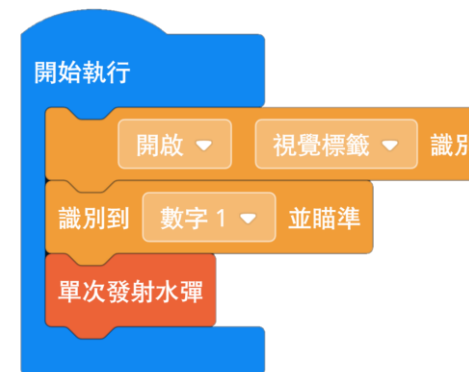
智能



- 使用這個模組可以簡單地自動瞄準目標，支援包含圖形、數字和字母的視覺標籤。



- 在程式中，啟用視覺標籤識別後，當視野中出現設定的視覺標籤後，機械人可以自動瞄準。



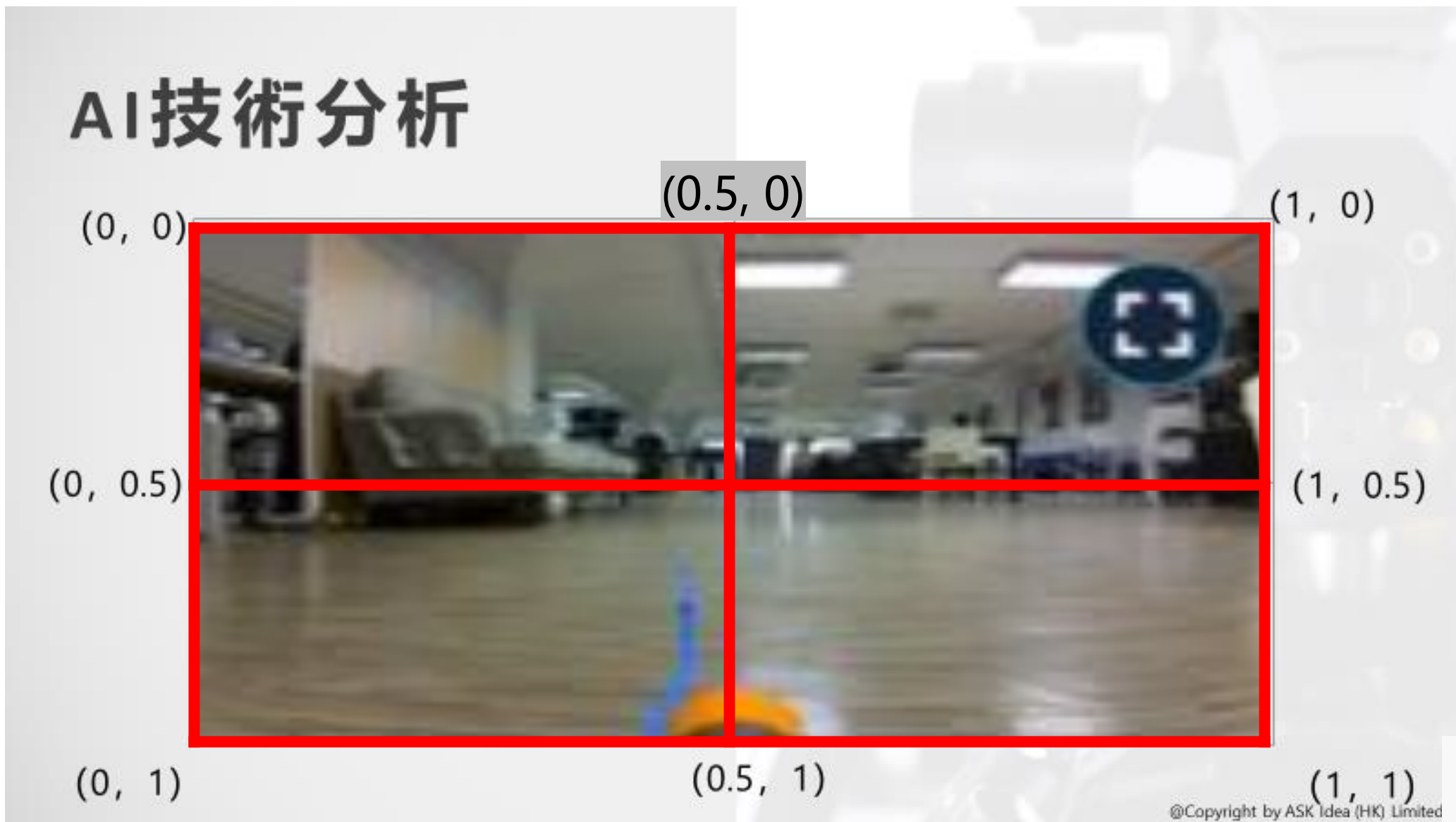
- **需要注意：**如果在執行自動瞄準模組時，機械人的視野中沒有需要瞄準的視覺標籤，將不會有相應的動作。

影響識別效果的情況

- 如果視覺標籤處於機械人視野中心，識別到的信息是準確的，但有其他情況會對識別效果造成影響：

情況	旋轉	顛倒	透視	移動過快	光線不足	遮擋
影響	無影響	無影響	較大影響	較大影響	較大影響	無法識別

AI技術分析



瞄準方式

- 實行方式一般分為兩種：
- 第一種是「自動瞄準模組」 (簡易)
- **第二種是「PID 控制器」 (進階)**

獲取視覺標籤資訊

- 當我們啟用視覺標籤識別後，相應的模組「**識別到的視覺標籤資訊**」會傳回一系列的即時資訊，我們可以將這些數據儲存在清單中。



- 與程式中的變數相同，清單資訊也會即時顯示在 FPV 視窗中。按照下圖所示編寫程式，點擊執行後開啟 FPV 視窗，並讓視覺標籤出現在機械人的視野當中再觀察：



- 機械人識別到視覺標籤後，傳回的數據以六個為一組，識別到多少視覺標籤就會傳回多少組數據，這些視覺信息還會實時更新。



視覺信息的格式

- 創建一個名為「MarkerList」的列表來儲存視覺標籤信息。每一個視覺標籤出現在視野時，會產生 6 個特徵信息：
 - 視覺標籤的數量
 - 視覺標籤的 ID
 - 標籤在視野中的橫坐標、縱坐標
 - 標籤的寬度、高度



例子一：1個標籤



N	ID_1	X_1	Y_1	W_1	H_1	ID_2	X_2	H_n
表示識別到的視覺標籤數目	第1個視覺標籤的 ID	第 1 個視覺標籤在視野中的橫座標	第 1 個視覺標籤在視野中的縱座標	第 1 個視覺標籤在視野中的寬度	第 1 個視覺標籤在視野中的高度	第 2 個視覺標籤的 ID	第 2 個視覺標籤在視野中的橫座標		第 n 個視覺標籤在視野中的高度

視覺信息的格式

- 創建一個名為「MarkerList」的列表來儲存視覺標籤信息。每一個視覺標籤出現在視野時，會產生 6 個特徵信息：
 - 視覺標籤的數量
 - 視覺標籤的 ID
 - 標籤在視野中的橫坐標、縱坐標
 - 標籤的寬度、高度



例子二：2個標籤
例子二：2個標籤



N	ID_1	X_1	Y_1	W_1	H_1	ID_2	X_2	Y_2	W_2	H_2	H_n
表示識別到的視覺標籤數目	第1個視覺標籤的ID	第1個視覺標籤在視野中的橫坐標	第1個視覺標籤在視野中的縱坐標	第1個視覺標籤在視野中的寬度	第1個視覺標籤在視野中的高度	第2個視覺標籤的ID	第2個視覺標籤在視野中的橫坐標	第2個視覺標籤在視野中的縱坐標	第2個視覺標籤在視野中的寬度	第2個視覺標籤在視野中的高度		第n個視覺標籤在視野中的高度

(進階)使用 PID 控制器



	優點	缺點
PID 控制器	可以優化瞄準效果，按照需求修改自動瞄準射擊的功能。	模組較多，實行起來比較複雜；需要對 PID 控制器參數設定有一定的基礎認識。

```

開始執行
  開啟 視覺標識 識別
  設定整機運轉 自由模式
  設定 PID 控制器 Yaw 的參數 Kp 110 Ki 0 Kd 65
  設定 PID 控制器 Pitch 的參數 Kp 70 Ki 0 Kd 20
  一直
    將 MarketList 設為 識別到的視覺標識資訊
    如果 MarketList 的第 1 項 == 1 然後
      將 X 設為 MarketList 的第 3 項
      將 Y 設為 MarketList 的第 4 項
      設定 PID 控制器 Yaw 的誤差為 X - 0.5
      設定 PID 控制器 Pitch 的誤差為 0.5 - Y
      控制雲台以 PID 控制器 Yaw 的輸出 度/秒繞航向軸旋轉 PID 控制器 Pitch 的輸出 度/秒繞俯仰軸旋轉
      等待 0.05 秒
    否則
      控制雲台以 0 度/秒繞航向軸旋轉 0 度/秒繞俯仰軸旋轉
  
```

什麼是 PID 控制器

- PID 控制器是一個反饋控制部件，常用於溫度控制、摩打控制等
- 什麼是「反饋控制」？
執行一系列動作後，將實際結果和期望結果進行比較，從而調整下一步行動

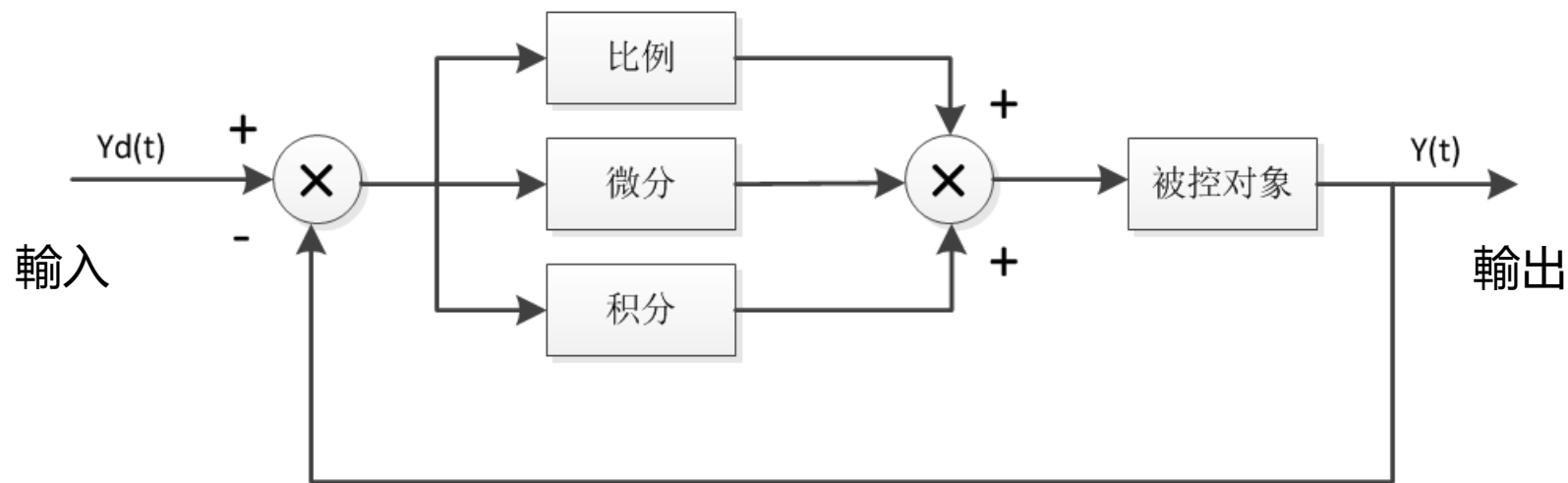


- 現實例子：
老師按特定方式授課 → 學生從課堂中學習 → 期望學生能掌握全部知識 → 以考試測量成效 → 老師按照學生成績放慢 / 加快 / 維持往後講解速度



什麼是 PID 控制器

- PID 控制器，由比例單元(P)、積分單元(I) 和微分單元(D)組成
- 透過調整三個單元的增益數值 K_p 、 K_i 和 K_d 來達到其特勻



模拟PID控制系统原理框图

PID 控制器的 P (Proportional) – 比例單元

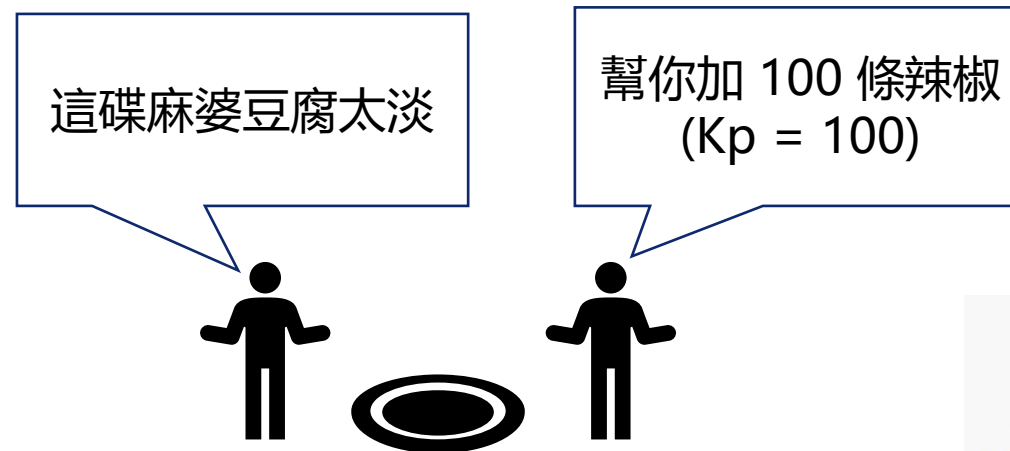
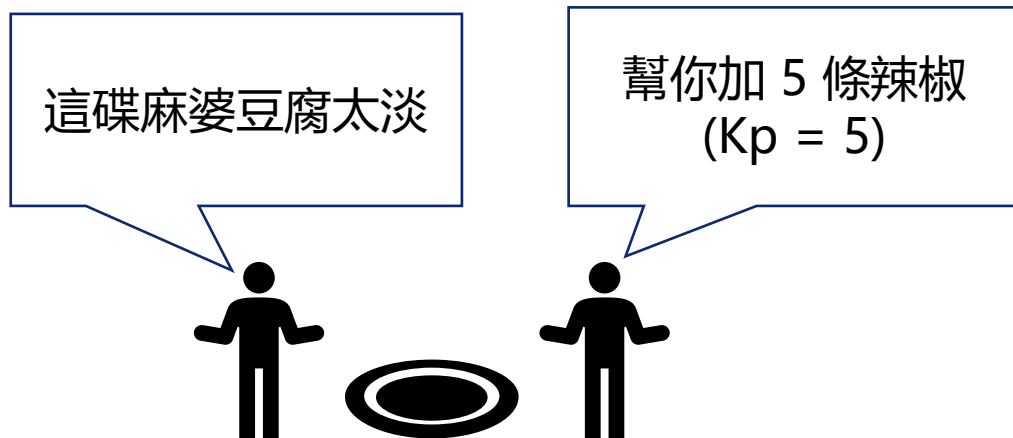
P (Proportional) 比例單元

- 透過增益數值 K_p 調控
- 最基礎、最常用
- 將誤差以比例放大來控制 (雲台 / 摩打)
- K_p 愈大, 反應速度愈快
- 例子 1:

* 注意:

K_p 過大 (例如 9999) 會令 PID 過份反應, 遍離控制

● 例子 2:



PID 控制器的 I (Integral) – 積分單元

PID 控制器的 D (Derivative) – 微分單元

I (Integral) 積分單元

- 透過增益數值 K_i 調控
- 將誤差累積乘以 K_i ，以消除誤差很小時的左右 / ，上下抖動
- 例子：

用 4~8 條辣椒的麻辣豆腐銷量不錯，下次應該都這樣做



K_i

D (Derivative) 微分單元

- 透過增益數值 K_d 調控
- 將誤差的變動乘以 K_d ，控制變動範圍
- 例子：

幫客人加 100 條辣椒
($K_p = 100$)



LV.100 廚師

不可以，太辣了，最多加 20 條
($K_d = 20$)



LV.20 經理

簡介 PID 控制器的模塊使用

設定 PID 控制器 Pitch 的參數 Kp 0 Ki 0 Kd 0

- 使用此模塊進行 Kp、Ki、Kd 參數的修改與調試，完成 PID 參數初始化設置

設定 PID 控制器 Pitch 的誤差為 0

- 使用此模塊將誤差輸入 PID 控制器中

PID 控制器 Pitch 的輸出

- 獲得一次 PID 運算的輸出，每調用一次該模塊，就會根據當前誤差值重新計算一次輸出

調控 PID 控制器

Kp、Ki、Kd 的數值需要透過不斷嘗試去找出理想設定值

- 建議：

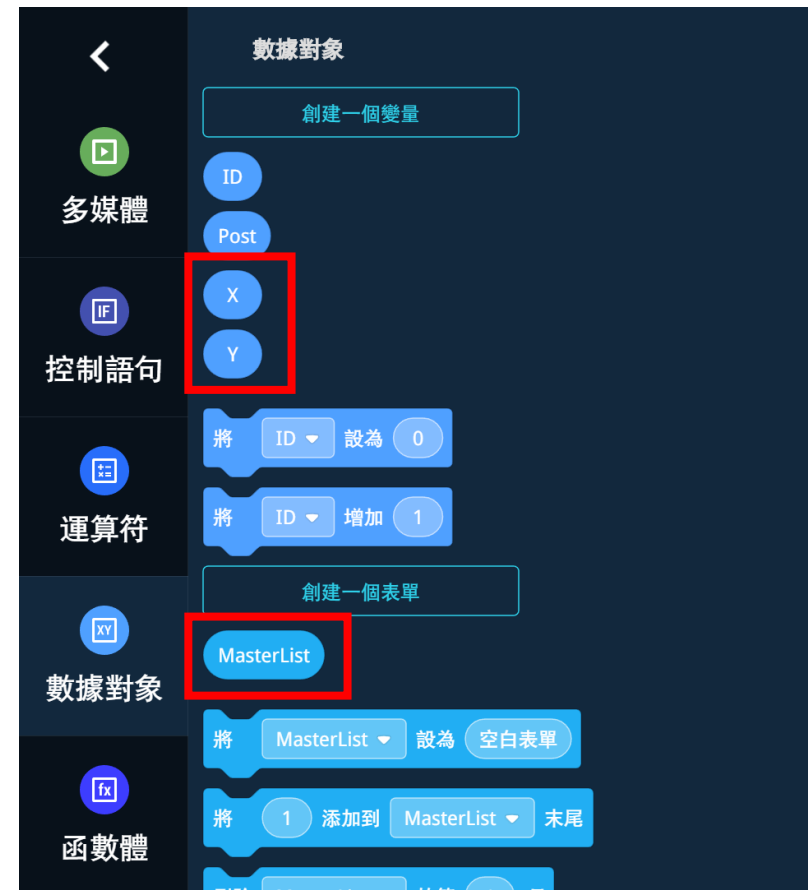
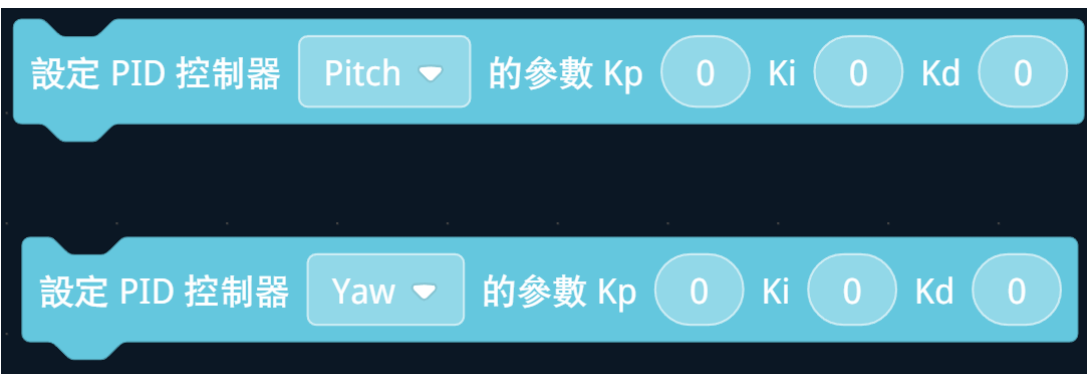
使用圖表記錄不同PID控制器參數下的表現，方便進行觀察及比較，例如：

Kp值	Ki值	Kd值	平均時間 (單位: 秒)
108	1	4	6.0834
118	1	4	5.7544
128	1	4	5.3844
138	1	4	5.1072
148	1	4	5.751
148	1	14	4.9714
148	1	24	5.124
143	1	4	5.799
143	1	14	5.3226
143	1	24	5.1722
140	1	4	5.0334

(進階)使用 PID 控制器 – 程式



- 配置設定：
 1. 創建所需 變數、表單、PID 控制器：
 - X: 用作儲存當時標籤的橫坐標，作計算用途
 - Y: 用作儲存當時標籤的縱坐標，作計算用途
 - 創建表單 MarkerList，用作儲存標籤資訊
 - 創建 PID 控制器 Yaw (控制雲台左右旋轉)、Pitch (控制雲台上下旋轉)



(進階)使用 PID 控制器 – 程式



- 初始執行設定 (只執行一次) :
2. 開啟 視覺識別 及 自由模式 (雲台能單獨轉動 → 更快)
 3. 設定 PID 控制器參數 K_p 、 K_i 、 K_d
(透過不斷測試調整數值)



(進階)使用 PID 控制器 – 程式



- 控制 (重複執行) :

5. 獲取視覺標籤資訊 並 放入表單
6. 檢查標籤目標的資訊是否存在
7. 將橫坐標、縱坐標放入變數 X、Y

```

一直
  將 MarketList 設為 識別到的視覺標籤資訊
  如果 MarketList 的第 1 項 == 1 然後
    將 X 設為 MarketList 的第 3 項
    將 Y 設為 MarketList 的第 4 項
  
```

第ID-1項 第ID+2項

N	ID_1	X_1	Y_1	W_1	H_1
表示識別到的視覺標籤數目	第1個視覺標籤的 ID	第 1 個視覺標籤在視野中的橫座標	第 1 個視覺標籤在視野中的縱座標	第 1 個視覺標籤在視野中的寬度	第 1 個視覺標籤在視野中的高度

(進階)使用 PID 控制器 - 程式

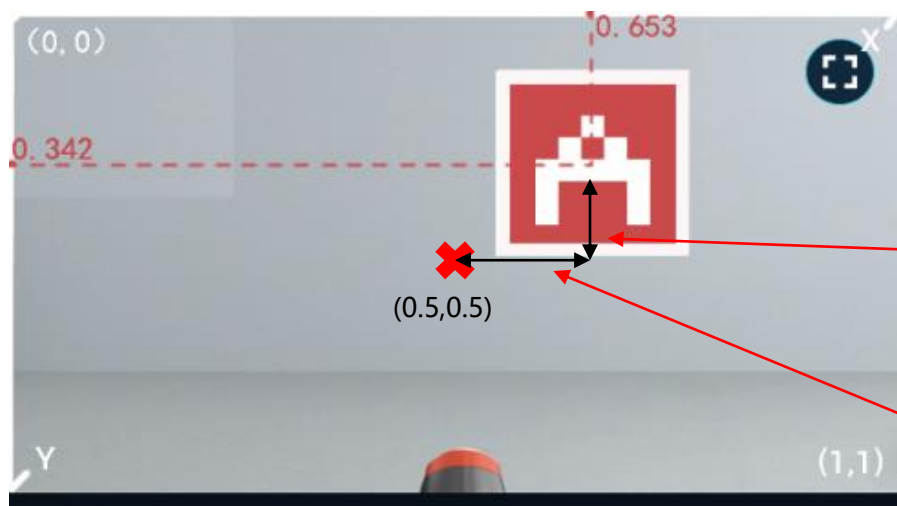


● 控制 (重複執行) :

8. 計算標籤位置與中心點之誤差並將將誤差放入 PID 控制器計算

Yaw 誤差: $X - 0.5$; $Pitch$ 誤差: $0.5 - Y$

9. 以 PID 控制器的輸出轉動雲台



Pitch 誤差 = $0.653 - 0.5 = 0.153$



向上轉動 0.153

Yaw 誤差 = $0.5 - 0.342 = 0.158$



向右轉動 0.158

項目二

自動巡線項目

巡線原理

目的：讓機械人一直沿著指定路線前進

首先要讓機械人知道「線在哪」，需要通過傳感器（視像鏡頭）獲取線的位置信息，用顏色線來設置指定路線

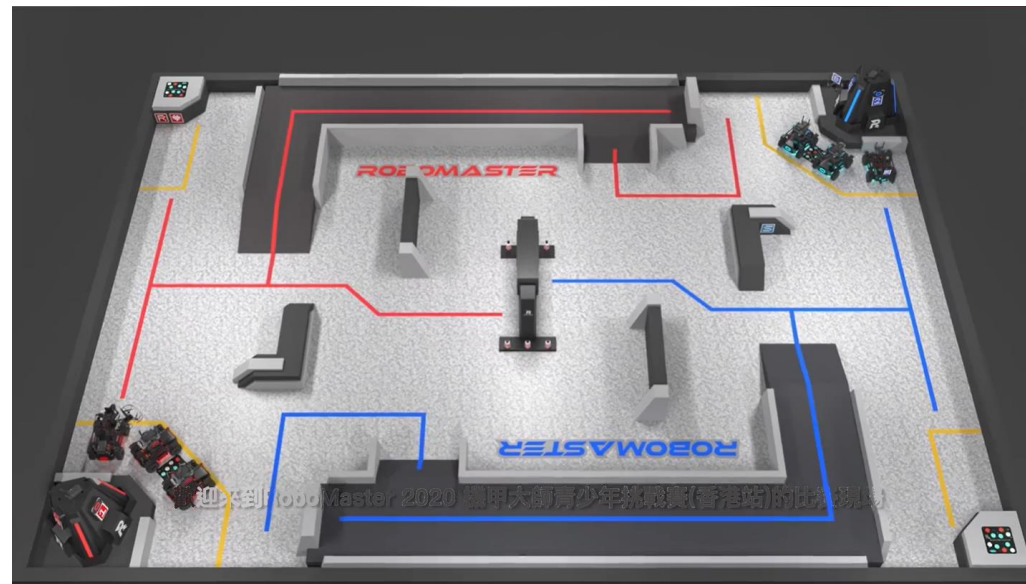
巡線的目標是使機械人一直處於線的中間位置，如有偏離則立即自我調整

例如：機械人從線的信息得知「偏右」，便會向左移



場地自動引導線地圖

- 在青少年挑戰賽中開始的頭一分鐘內，地面機械人可以自動執行按照事先預定的程式，透過循線和岔路判斷到達特定地點執行特定動作，並且為比賽的手動操作階段作好準備。



路線資訊檢視

- 智能大類 —> 「識別到的單線資訊」。



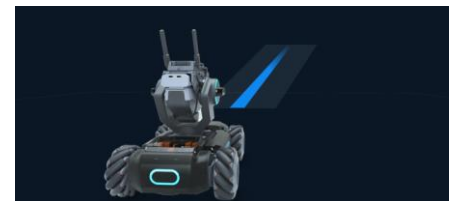
智能


識別到的單線信息

- 在程式中，啟用視線識別後，當視野中出現設定的顏色線路資訊後，機械人可以傳回線路資訊，並將線路資訊儲存於表單 **LineList**。



- **需要注意：**如果在啟動線路識別後，機械人視野中沒有線路時，將不會傳回線路資訊。

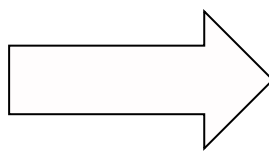


- 點擊執行後，開啟右邊的 FPV 視窗 ，可以看到即時的線路資訊。



體驗活動：

- 編程使用列表記錄下藍線的數據
- 使用FPV介面觀察列表中的數據



狀態

固定模式	速度	俯仰	航向
雲台跟隨底盤模式	2.4m/s	20.0°	10.0°

變量

LineList 長度: 42 ^

1	10	2	1
3	0.503125	4	0.794444
5	-3.598248	6	0
7	0.503125	8	0.766667
9	-4.580367	10	-0.032737
11	0.5	12	0.738889
13	-0.700007	14	-0.700007
15	0.5	16	0.711111
17	-0.700007	18	0.002858
19	0.5	20	0.683333
21	-5.397032	22	-0.156504
23	0.496875	24	0.655556
25	-3.591615	26	0.056703
27	0.496875	28	0.627778

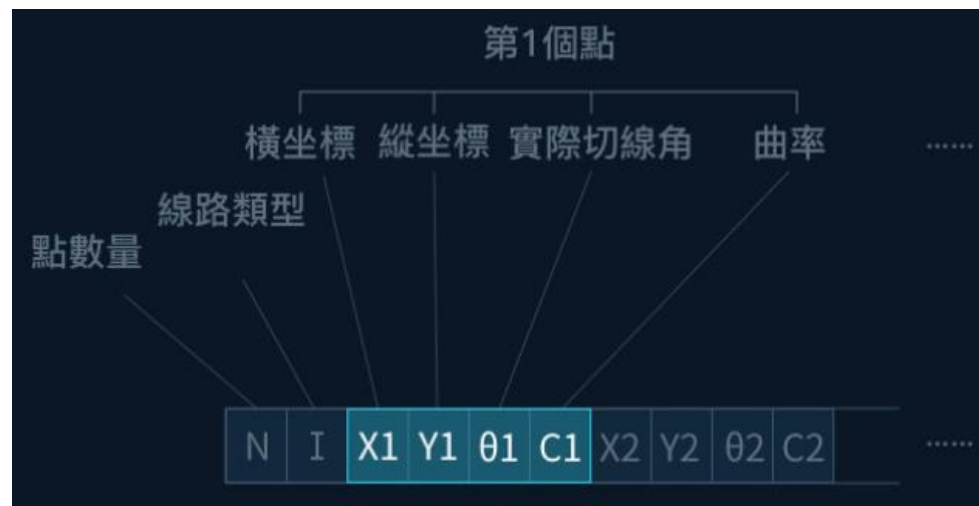
路線資訊分析

狀態			
固定模式	速度	俯仰	航向
雲台跟隨底盤模式	2.4m/s	20.0°	10.0°

變量			
LineList 長度: 42 ^			
1	10	2	1
3	0.503125	4	0.794444
5	-3.598248	6	0
7	0.503125	8	0.766667
9	-4.580367	10	-0.032737
11	0.5	12	0.738889
13	-0.700007	14	-0.700007
15	0.5	16	0.711111
17	-0.700007	18	0.002858
19	0.5	20	0.683333
21	-5.397032	22	-0.156504
23	0.496875	24	0.655556
25	-3.591615	26	0.056703
27	0.496875	28	0.627778

● 當機械人識別到線路後，會傳回共 42 項數據，這 42 項數據分別為：

1. 線上點的數量
2. 線路類型
- 3.~42. 10 個點（視野中由下至上從線路等距提取）的座標和角度資訊等詳細數據



數據分析:

- 第一項數據: N 點數目, 預設為 10 或 0; 10 代表識別到了線路, 0 代表識別不到線路。

LineList		長度: 42 ^	
1	10	2	1
3	0.503125	4	0.794444
5	-3.598248	6	0
7	0.503125	8	0.766667
9	-4.580367	10	-0.032737
11	0.5	12	0.738889
13	-0.700007	14	-0.700007
15	0.5	16	0.711111
17	-0.700007	18	0.002858
19	0.5	20	0.683333
21	-5.397032	22	-0.156504
23	0.496875	24	0.655556
25	-3.591615	26	0.056703
27	0.496875	28	0.627778

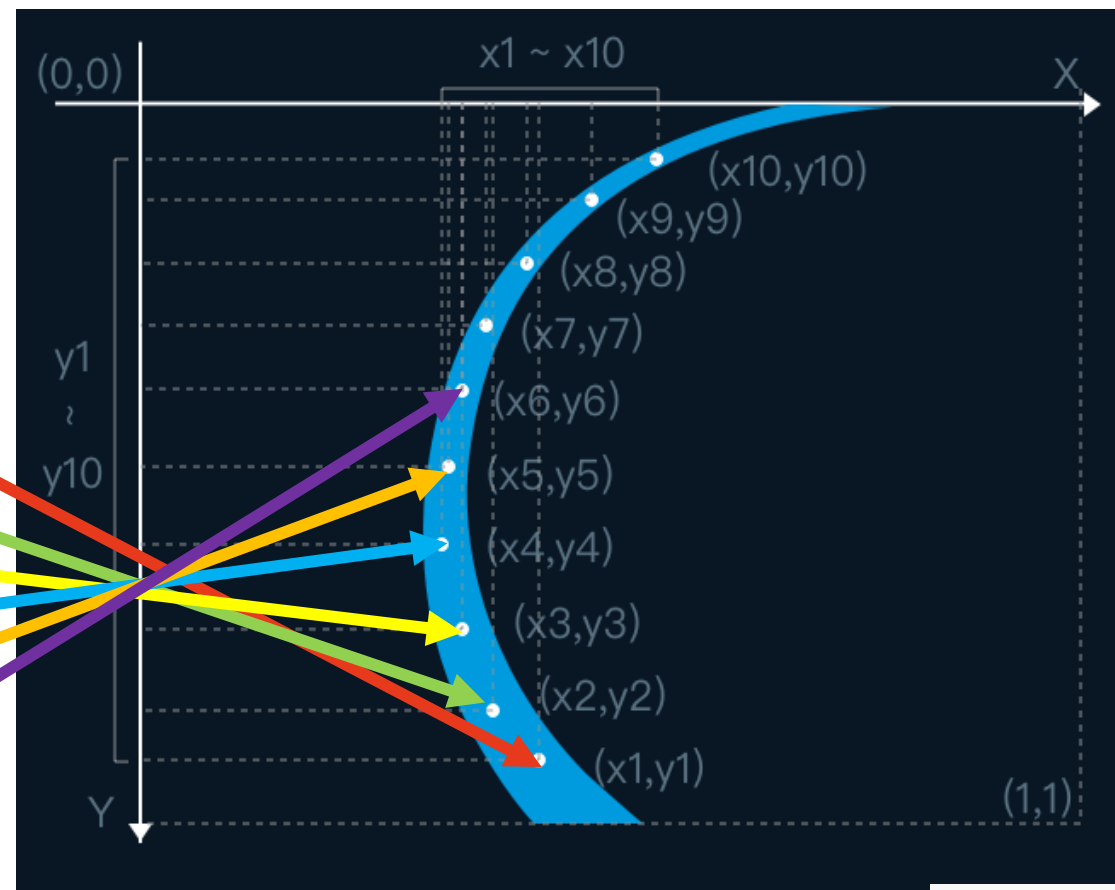
	視野內無線	視野內一條線	視野內Y型路	視野內十字路
情況				
數據	0	1	2	3

- 第二項數據: l 代表線路類型: 數字 0 代表沒有識別到線路, 1、2 和 3 分別對應三種線路的情況, 如下圖所示; 我們可以根據傳回的值確定目前的線路情況, 從而作出選擇。

數據分析：

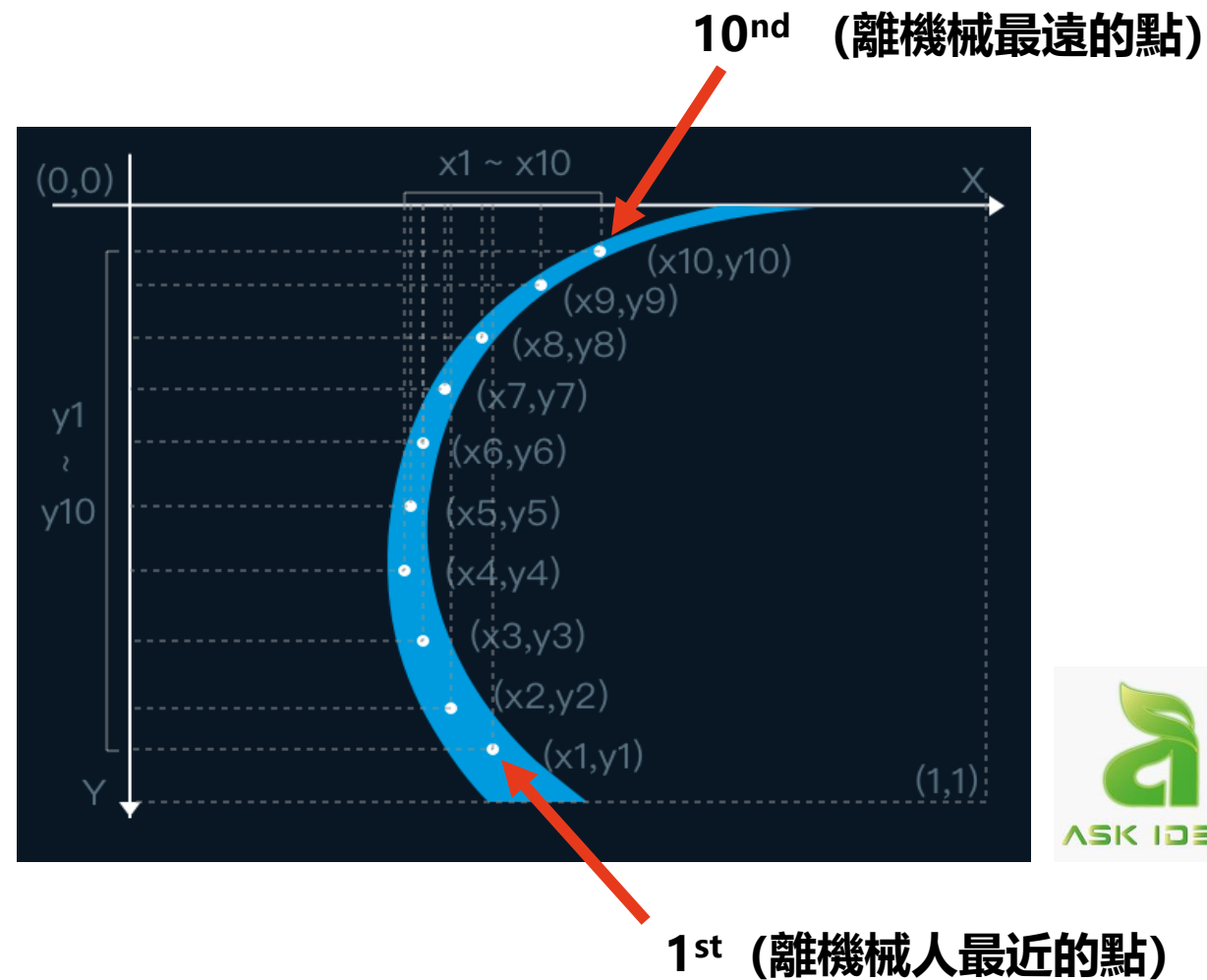
LineList 長度: 42 ^

1	10	2	1
3	0.503125	4	0.794444
5	-3.598248	6	0
7	0.503125	8	0.766667
9	-4.580367	10	-0.032737
11	0.5	12	0.738889
13	-0.700007	14	-0.700007
15	0.5	16	0.711111
17	-0.700007	18	0.002858
19	0.5	20	0.683333
21	-5.397032	22	-0.156504
23	0.496875	24	0.655556
25	-3.591615	26	0.056703
27	0.496875	28	0.627778



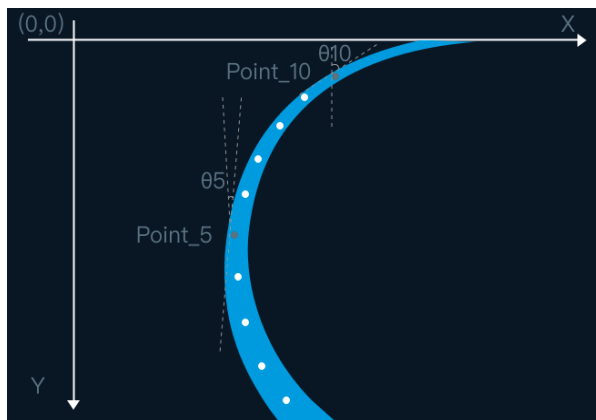
路線資訊分析

- 接下來的數據項目以四個數據為一組，分別代表視野中十個點的橫向座標 x 、縱向座標 y 、實際切線角 θ 以及曲率 C ；第三至第六位代表的就是第一個點的這四項數據，第七至十位就是第二點，如此類推。
- 第 1 個點位於視野最下端，也就是最接近機械人的點。
- 機械人視野為十字坐標系統，視野左上角為 $(0, 0)$ ，右下角為 $(1, 1)$

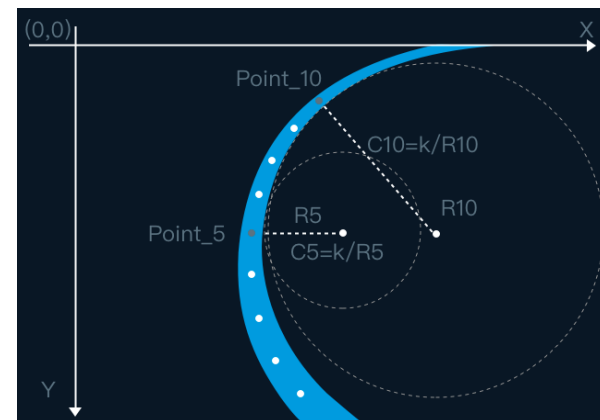


路線資訊分析

- 當切線角為 0 時，意味着此點所在的線路為直線；切線角為 90 時，證明此點所在的線路非常彎曲。



- 外接圓越小，證明此點附近的線路弧度較大；外接圓越大，證明此點附近的線路弧度越小。傳回的值 $c=k/R$ 中，數據經過處理，取值範圍為 1-10；1 證明弧度小，10 證明弧度很大。



實行巡線主要分為兩個技術點

- 實行巡線主要分為兩個技術點：

控制系統	A	B
建構方向控制系統	比例控制	PID 控制器
建構速度控制系統	低速固定速度	根據彎度調速

步兵機械人巡線常式



● 編程分析：

- ① 主要是使用底盤跟隨雲台模式，控制雲台轉動使線路保持在路線中央。
- ② 有關步兵機械人巡線部分的程式設計操作，可以參考大師之路第 7 關。

```
開始執行
將 V_average 設置為 1
將 K 設置為 0.08
設定雲台運轉 雲台跟隨底盤模式
控制雲台 向下 旋轉 20 度
開啟 線識別
設定線識別顏色為 藍
設定 PID 控制器 Follow_Line 的參數 Kp 200 Ki 0 Kd 25
一直
將 Linelist 設置為 識別到的單線信息
如果 Linelist 的項目數 == 42 然後
    如果 Linelist 的第 2 項 == 1 然後
        將 x 設置為 Linelist 的第 19 項
        設定 PID 控制器 Follow_Line 的誤差為 x - 0.5
        控制雲台以 PID 控制器 Follow_Line 的輸出 度/秒繞航向軸旋轉 0 度/秒繞俯仰軸旋轉
        將 v 設置為 V_average - K * 絕對值 Linelist 的第 37 項 / 180
        設定底盤平移速率 v 公尺/秒
        控制底盤向 0 度平移
    否則
        控制雲台以 0 度/秒繞航向軸旋轉 0 度/秒繞俯仰軸旋轉
```

步兵機械人巡線常式



- 編程分析：
- 配置設定
- 1. 創建所需 變數、表單、PID 控制器：
 - V_average: 平均速度
 - V: 計算後的移動速度
 - K: 用作根據切線角調節速度的比例系數
 - X : 用作儲存當時標籤的橫坐標，作計算用途
 - 創建表單 LineList, 用作儲存線的資訊
 - 創建 PID 控制器 Follow_Line (控制雲台左右旋轉)



步兵機械人巡線常式



	大	小
V_average	速度快	速度慢
K	速度變化大, 但穩定	速度均勻, 但容易出界

- 初始執行設定 (只執行一次) :
- 1. 設定速度控制常數: V_average 和 K
- 2. 設定雲台跟隨底盤模式 (保持鏡頭向前)
- 3. 雲台向下 20 度 (望向地上的線)
- 4. 開啟線識別
- 5. 設定 PID 控制器參數 Kp、Ki、Kd (透過不斷測試調整數值)

速度控制常數

底盤模式

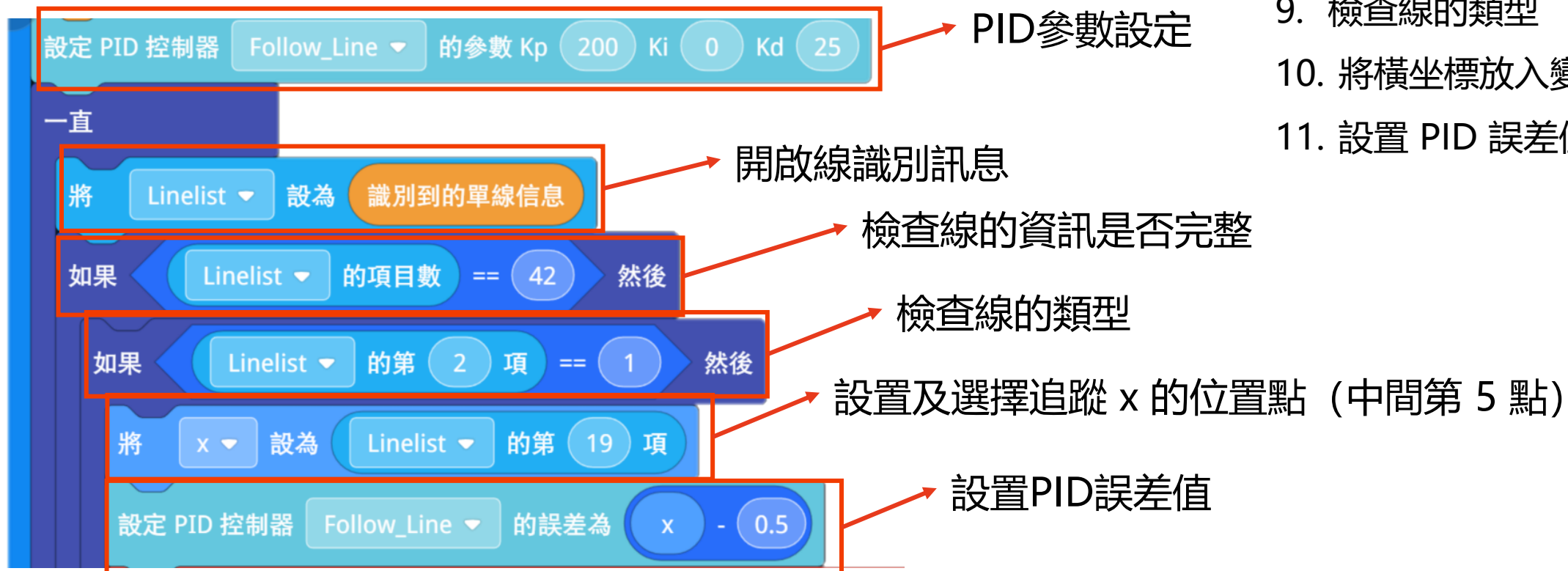
線識別控制

PID參數設定

步兵機械人巡線常式



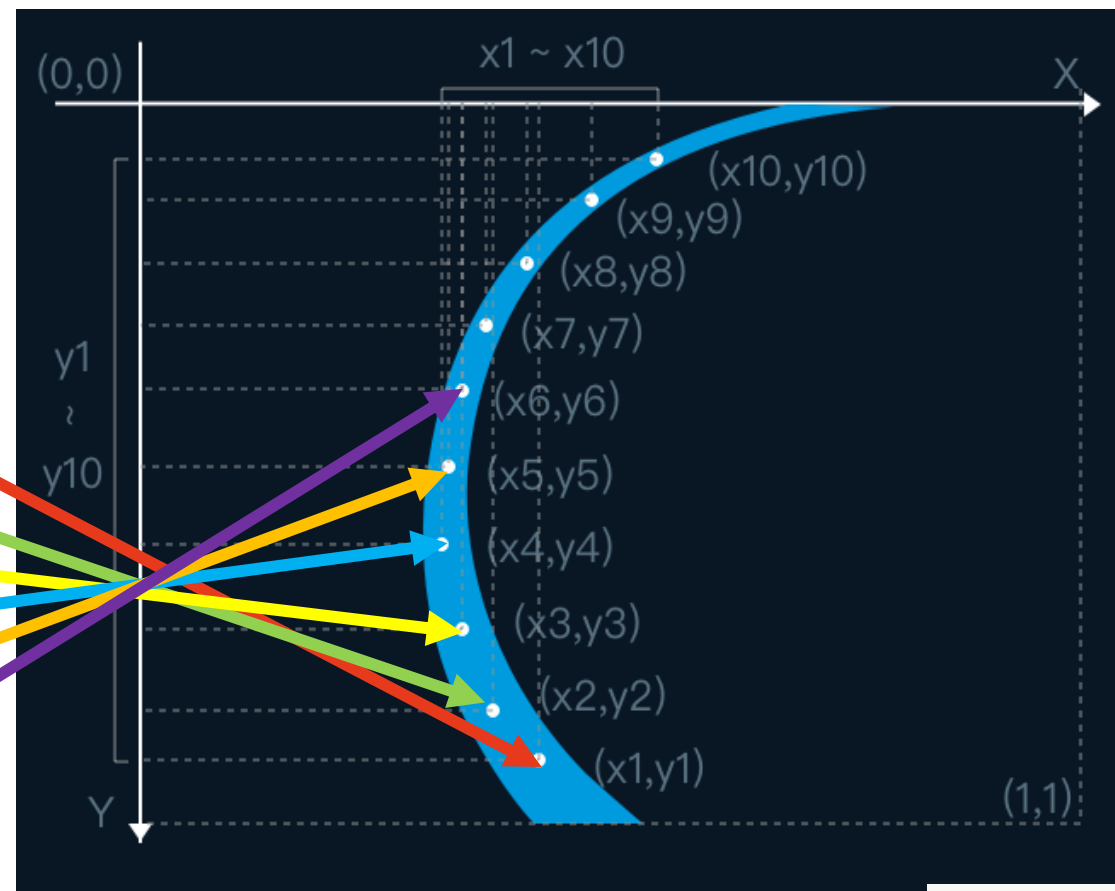
- 控制（重複執行）：
 7. 獲取線的資訊 並 放入表單
 8. 檢查線的資訊是否完整
 9. 檢查線的類型
 10. 將橫坐標放入變數 X
 11. 設置 PID 誤差值



數據分析：

LineList 長度: 42 ^

1	10	2	1
3	0.503125	4	0.794444
5	-3.598248	6	0
7	0.503125	8	0.766667
9	-4.580367	10	-0.032737
11	0.5	12	0.738889
13	-0.700007	14	-0.700007
15	0.5	16	0.711111
17	-0.700007	18	0.002858
19	0.5	20	0.683333
21	-5.397032	22	-0.156504
23	0.496875	24	0.655556
25	-3.591615	26	0.056703
27	0.496875	28	0.627778



步兵機械人巡線常式



- 控制（重複執行）：
 12. 設置 PID 誤差值
 13. 以 PID 輸出作雲台追蹤控制
 14. 計算 並 輸出速度控制

設置PID誤差值

雲台追蹤控制

設置及選擇速度控制

設定 PID 控制器 Follow_Line 的誤差為 $x - 0.5$

控制雲台以 PID 控制器 Follow_Line 的輸出 度/秒繞航向軸旋轉 0 度/秒繞俯仰軸旋轉

將 v 設為 $V_average - K * \text{絕對值}(\text{Linelist 的第 } 37 \text{ 項} / 180)$

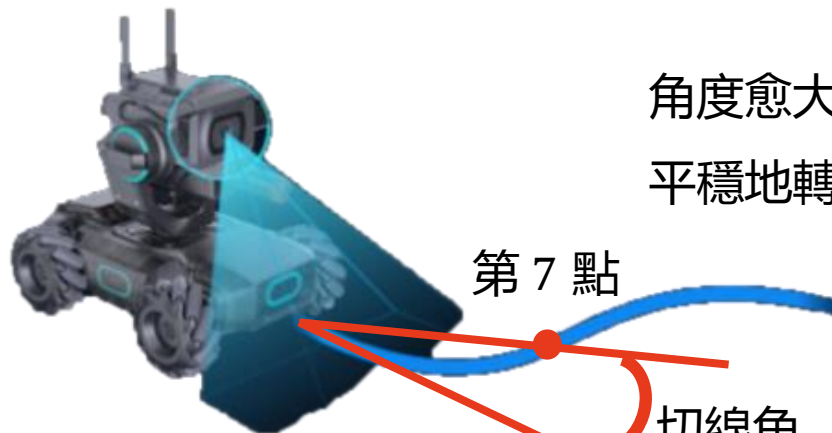
設定底盤平移速率 v 公尺/秒

控制底盤向 0 度平移

否則

控制雲台以 0 度/秒繞航向軸旋轉 0 度/秒繞俯仰軸旋轉

步兵機械人巡線常式



角度愈大，速度愈慢，
平穩地轉彎



設定 PID 控制器 Follow_Line 的誤差為 $x - 0.5$

控制雲台以 PID 控制器 Follow_Line 的輸出 度/秒繞航向軸旋轉 0 度/秒繞俯仰軸旋轉

將 v 設為 $V_average - K * \text{絕對值}(\text{Linelist 的第 } 37 \text{ 項} / 180)$

設定底盤平移速率 v 公尺/秒

控制底盤向 0 度平移

否則

控制雲台以 0 度/秒繞航向軸旋轉 0 度/秒繞俯仰軸旋轉

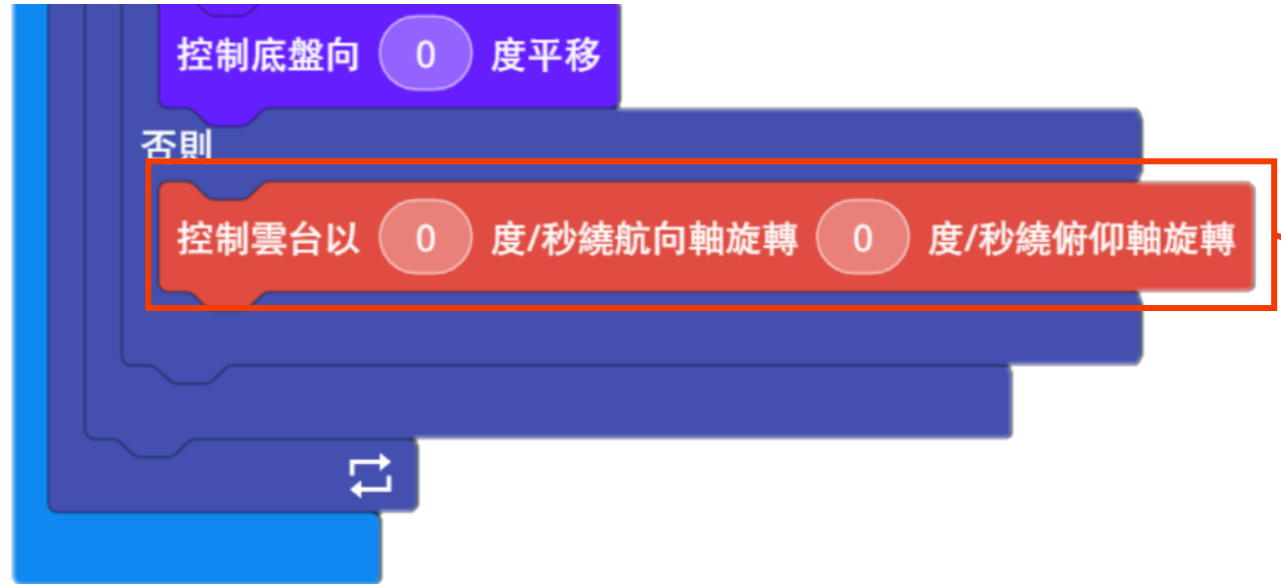
速度公式：

$$\text{速度} = \text{平均速度} - K \times |\text{偏移角}|$$

$$\text{偏移角} = \frac{\text{切線角}}{180^\circ}$$

假如切線角 = 0 (直線)，
以最快速度移動， $V = V_average$

步兵機械人巡線常式



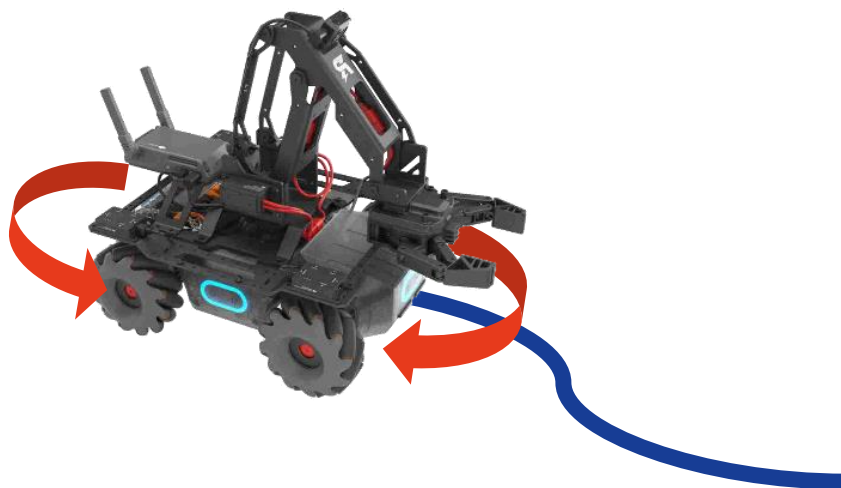
- 控制（重複執行）：
15. 如果線的資訊不正確
→ 保持原本位置

工程機械人巡線常式



● 常式分析：

與步兵機械人執行巡線不同的是，由於工程機械人沒有雲台，所以不可以使用雲台旋轉與底盤跟隨雲台的模式，需要使用底盤直接旋轉。



```
開始執行
將 V_average 設為 1
將 K 設為 0.3
開啟 線識別
設定線識別顏色為 藍
設定 PID 控制器 Follow_Line 的參數 Kp 100 Ki 0 Kd 15
一直
將 LineList 設為 識別到的單線信息
如果 LineList 的項目數 == 42 然後
  如果 LineList 的第 2 項 == 1 然後
    將 X 設為 LineList 的第 11 項
    設定 PID 控制器 Follow_Line 的誤差為 X - 0.5
    將 V 設為 V_average - K * 絕對值 LineList 的第 37 項 / 180
    控制底盤以 V 公尺/秒沿x軸平移 0 公尺/秒沿Y軸平移 PID 控制器 Follow_Line 的輸出 度/秒繞Z軸旋轉
  否則
    控制底盤以 0 公尺/秒沿x軸平移 0 公尺/秒沿Y軸平移 0 度/秒繞Z軸旋轉
```

工程機械人巡線常式

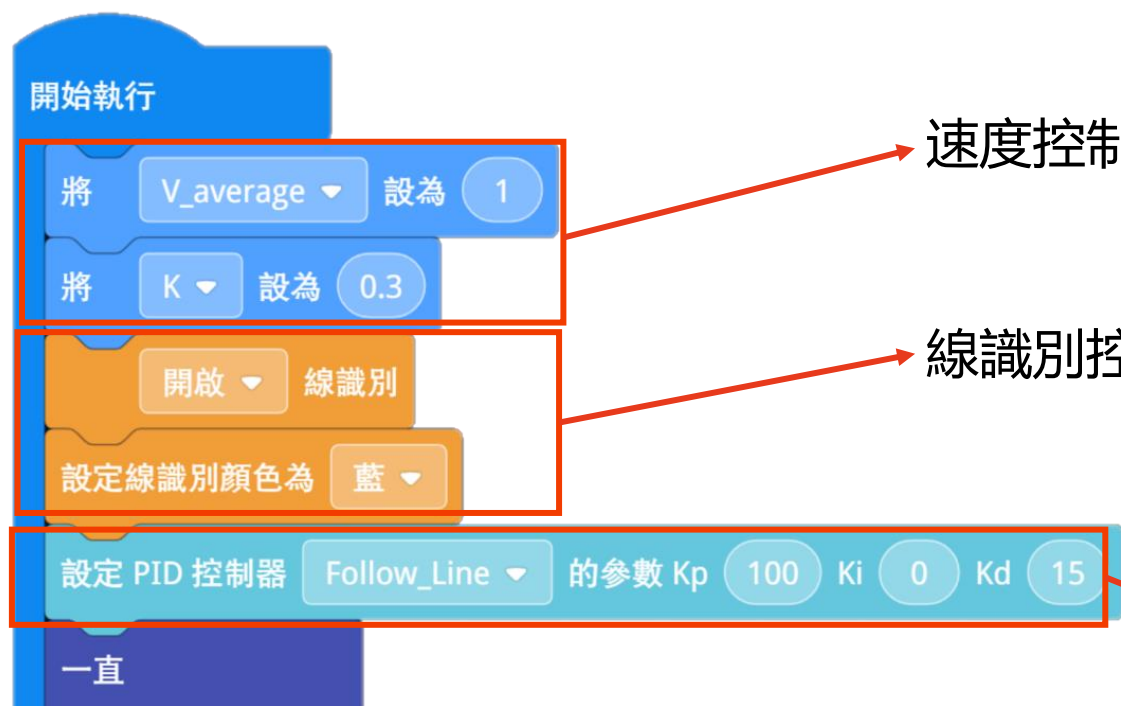


- 編程分析：
- 配置設定（與步兵機械人一樣）
 1. 創建所需變數、表單、PID 控制器：
 - V_average：平均速度
 - V：計算後的移動速度
 - K：用作根據切線角調節速度的比例系數
 - X：用作儲存當時標籤的橫坐標，作計算用途
 - 創建表單 LineList，用作儲存線的資訊
 - 創建 PID 控制器 Follow_Line（控制底盤左右旋轉）



設定 PID 控制器 Follow_Line 的參數 Kp 0 Ki 0 Kd 0

工程機械人巡線常式



- 初始執行設定（只執行一次）：
（與步兵機械人相似）

2. 設定速度控制常數：V_average 和 K

設定雲台跟隨底盤模式

雲台向下 20 度（望向地上的線）

3. 開啟線識別

4. 設定 PID 控制器參數 Kp、Ki、Kd

（透過不斷測試調整數值）

工程機械人巡線常式



- 控制（重複執行）：
（與步機械人一樣）

5. 獲取線的資訊 並 放入表單
6. 檢查線的資訊是否完整
7. 檢查線的類型
8. 將橫坐標放入變數 X
9. 設置 PID 誤差值



工程機械人巡線常式



- 控制（重複執行）：
（與步兵機械人相似）
- 10. 設置 PID 誤差值
- 11. 計算 並 輸出速度控制
- 12. 以 V 及 PID輸出作底盤追蹤控制

設定 PID 控制器 Follow_Line 的誤差為 X - 0.5

設置PID誤差值

將 V 設為 V_average - K * 絕對值 LineList 的第 37 項 / 180

設置及選擇速度控制

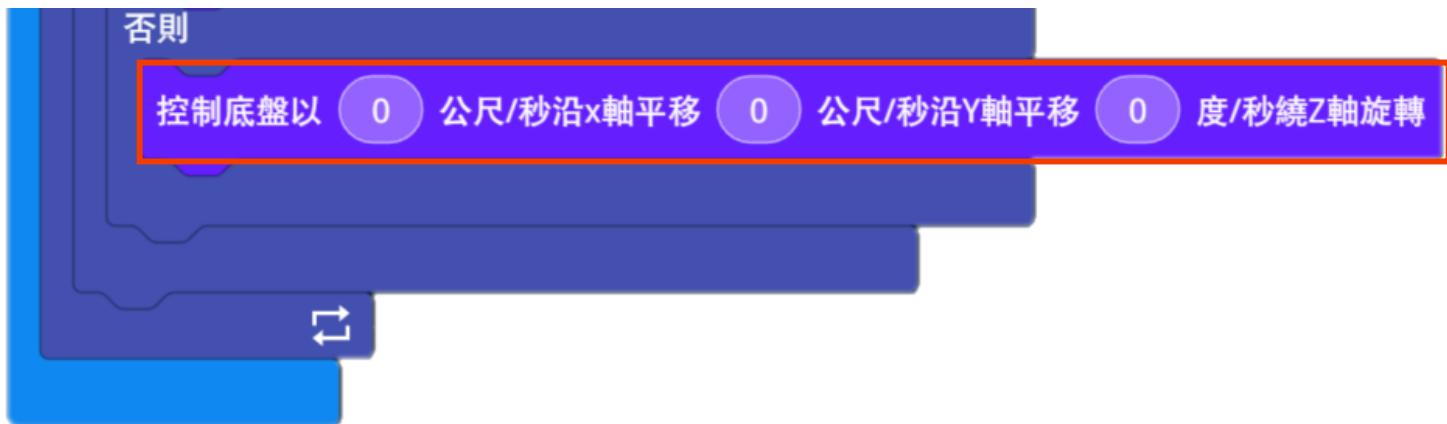
控制底盤以 V 公尺/秒沿x軸平移 0 公尺/秒沿Y軸平移 PID 控制器 Follow_Line 的輸出 度/秒繞Z軸旋轉

底盤追蹤控制

工程機械人巡線常式



- 控制（重複執行）：
（與步機械人相似）
- 13. 如果線的資訊不正確
→ 保持原本位置



數據錯誤時的處理