

|           |             |
|-----------|-------------|
| 教具名稱      | 循跡履帶車       |
| 課程名稱      | 循規不蹈矩—小車繞圈圈 |
| 運算思維/創意實作 | 演算法步驟       |
| 編撰教師      | 潘浚琪         |
| 編撰基地      | 高雄市鳳山區忠孝國小  |
| 課程影片      | 無           |
| 建議授課節數    | 4節          |

# 大綱

1. 情境主題及目的
2. 情境分析
3. 演算法步驟(學生填空用)
4. 演算法步驟(教師用)
5. 演算法步驟 vs 積木程式堆疊
6. 注意事項

# 情境主題及目的

- (1) 情境主題：循規不蹈矩 - 小車繞圈圈
- (2) 情境目的：讓小車沿著黑線移動。

# 情境分析

## (3) 情境分析：

1. 透過類比腳位讀取左、右循跡數值；
2. 計算兩邊差異，等於左循跡數值減去右循跡數值。如果兩邊差異很大，代表兩邊的顏色深淺有明顯差異。此判斷數值可以自行調整，如果要敏感點就把數值設小一點，如果要精準點就把數值設大一點。(以下皆以200為例)
3. 如果兩邊差異 $>200$ ，代表左邊壓線，那麼小車左轉。
4. 如果兩邊差異 $<-200$ ，代表右邊壓線，那麼小車右轉。
5. 如果以上皆非，代表沒壓線，小車直走。

# 演算法步驟(學生用)

## (4) 演算法步驟：

重複

01. 讀取左循跡數值

02. 讀取右循跡數值

03. 計算兩邊差異 = 左循跡 - 右循跡

04. 判斷 如果 (兩邊差異 > 200 )

05. 那麼

06. 否則再判斷 如果 ( 兩邊差異 < -200 )

07. 那麼

08. 否則

重複結束

# 演算法步驟(教師用)

## (4) 演算法步驟：

重複

01.讀取左循跡數值

02.讀取右循跡數值

03.計算兩邊差異 = 左循跡 - 右循跡

04.判斷 如果 (兩邊差異 > 200 )

05. 那麼 左轉

06. 否則再判斷 如果 ( 兩邊差異 < -200 )

07. 那麼 右轉

08. 否則 前進

重複結束

# 演算法步驟 vs 積木程式堆疊

## (4) 演算法步驟：

## (5) 積木程式堆疊

重複

01. 讀取左循跡數值

02. 讀取右循跡數值

03. 計算兩邊差異 = 左循跡 - 右循跡

04. 判斷 如果 (兩邊差異 > 200)

05.           那麼 左轉

06.           否則再判斷 如果 (兩邊差異 < -200)

07.                           那麼 右轉

08.                           否則 前進

重複結束

程式 1



# 演算法步驟 vs 積木程式堆疊

## (4) 演算法步驟：

重複  
01. 讀取左循跡數值  
02. 讀取右循跡數值  
03. 計算兩邊差異 = 左循跡 - 右循跡  
04. 判斷 如果 (兩邊差異 > 200)  
05.           那麼 左轉  
06.           否則再判斷 如果 (兩邊差異 < -200)  
07.                           那麼 右轉  
08.                           否則 前進  
重複結束

## (5) 積木程式堆疊

### 程式 2



此數值可自行調整、測試



# 演算法步驟 vs 積木程式堆疊

## (4) 演算法步驟：

重複  
01. 讀取左循跡數值  
02. 讀取右循跡數值  
03. 計算兩邊差異 = 左循跡 - 右循跡  
04. 判斷 如果 (兩邊差異 > 200)  
05.           那麼 左轉  
06.           否則再判斷 如果 (兩邊差異 < -200)  
07.                   那麼 右轉  
08.                   否則 前進  
重複結束

## (5) 積木程式堆疊

### 程式 3



此數值可自行調整、測試

# 演算法步驟 vs 積木程式堆疊

## (4) 演算法步驟：

重複  
01. 讀取左循跡數值  
02. 讀取右循跡數值  
03. 計算兩邊差異 = 左循跡 - 右循跡  
04. 判斷 如果 (兩邊差異 > 200)  
05.           那麼 左轉  
06.       否則再判斷 如果 (兩邊差異 < -200)  
07.           那麼 右轉  
08.           否則 前進  
重複結束

## (5) 積木程式堆疊



程式 2+3

# 演算法步驟 vs 積木程式堆疊

## (4) 演算法步驟：

重複

01. 讀取左循跡數值

02. 讀取右循跡數值

03. 計算兩邊差異 = 左循跡 - 右循跡

04. 判斷 如果 (兩邊差異 > 200)

05.           那麼 左轉

06.           否則再判斷 如果 (兩邊差異 < -200)

07.                           那麼 右轉

08.                           否則 前進

重複結束

## (5) 積木程式堆疊

重複無限次

變數 左循跡 設為 讀取類比腳位 A1 註

變數 右循跡 設為 讀取類比腳位 A0 註

變數 兩邊差異 設為 左循跡 - 右循跡

如果 兩邊差異 > 200 那麼

配對直流馬達 左轉 轉速為 快

否則

如果 兩邊差異 < -200 那麼

配對直流馬達 右轉 轉速為 快

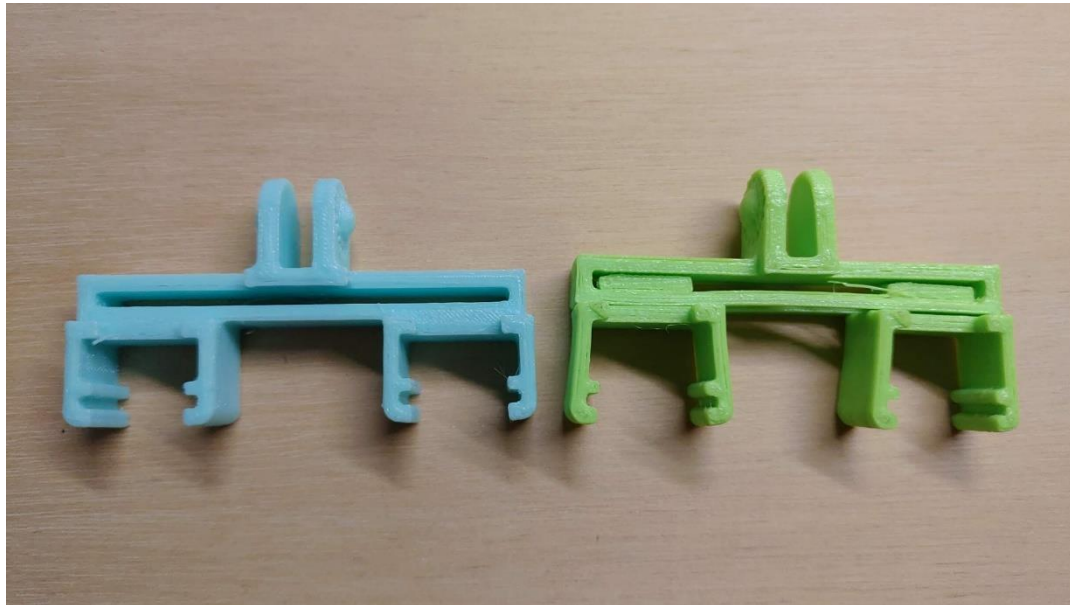
否則

配對直流馬達 前進 轉速為 快

程式 1+2+3  
重複無限次

# 注意事項

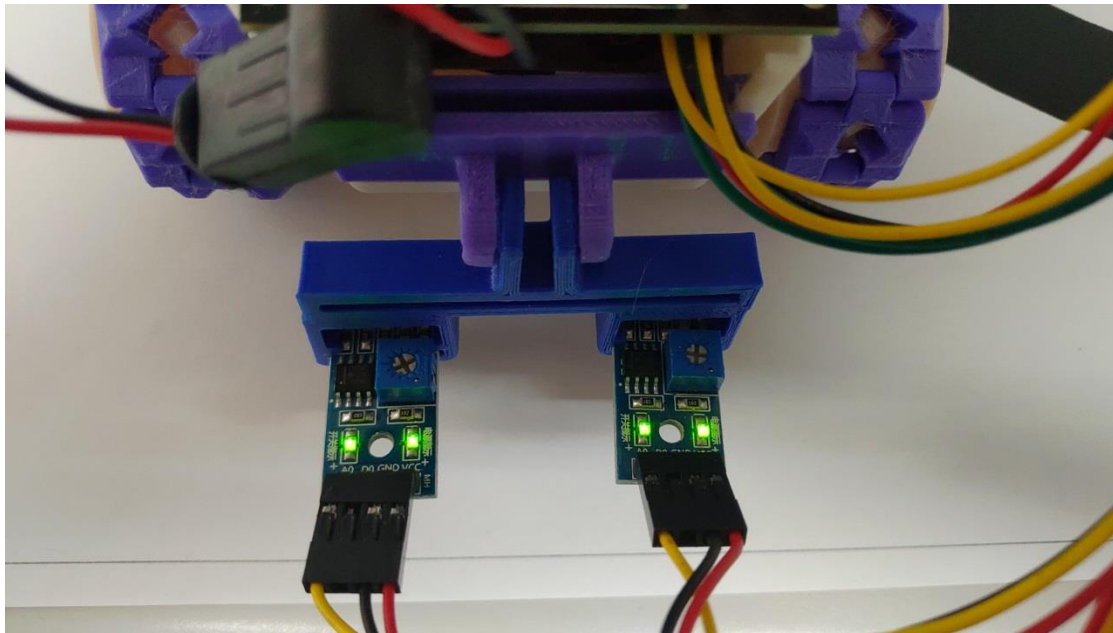
1. 原廠的循跡機構是活動式的，但因為一來組裝較麻煩，二來每次在測試循跡前都要先檢查是否有推到兩邊。所以我稍微調整設計，將其組件合併成一個。



左邊是我改過的，右邊是原廠，

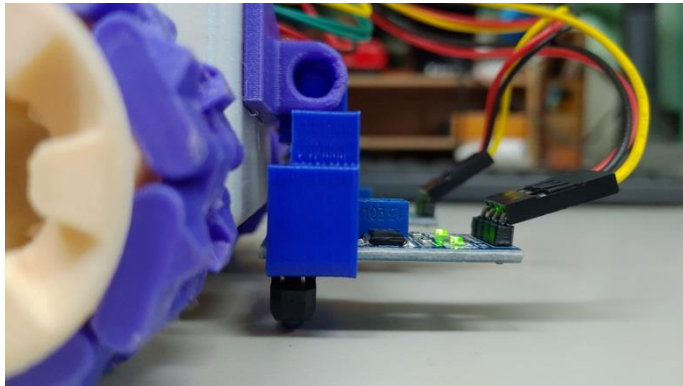
## 注意事項

2. 我的循跡模組與機構的安裝方向與原廠不同。原本的安裝方式很容易被彈開而影響判讀，而且也無法安裝三個以上的循跡模組，反過來裝還能看到模組上的燈號顯示。

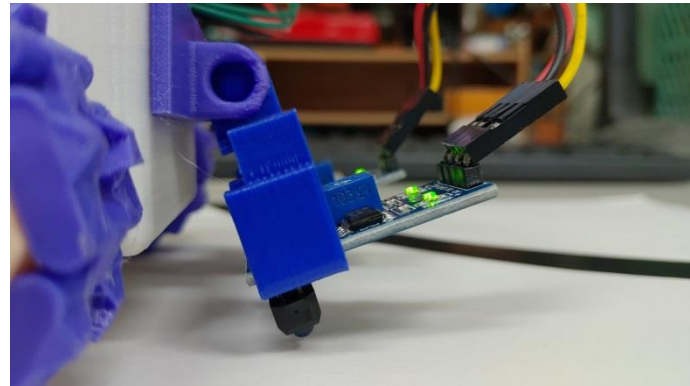


## 注意事項

3. 循跡模組要和地面保持一個微妙的距離，不能太近也不能太遠。如果依我的安裝方式，垂直朝著地面數值會偏高，稍微往上轉動一格的距離最為理想。黑線的數值超過800，白色區域的數值小於100。



這角度的數值會偏高



這角度比較適合

## 注意事項

4. 如果因為車速太快，小車來不及反應，可以切換到無限制幀數模式。但有時還是會遇到wifi突然延遲的問題(lag)，如果想要完全避免這問題，可能要改用燒錄模式。

