

教具名稱	馬達與感測器教具
課程名稱	鐵捲門
運算思維	演算法步驟
編撰教師	高自造
編撰基地	高雄市●●高中行星基地
課程影片(有/無)	有

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

情境主題及目的

- (1) 情境主題：鐵捲門
- (2) 情境目的：結合搖桿與馬達的運轉，利用搖桿左移、右移及按壓鍵模擬鐵捲門上移、下移及停止功能。

情境分析

(3) 情境分析：

- 1.讀取搖桿方向與按鍵值；
- 2.如果搖桿左移(X軸)，且X值 < 50 ，則馬達正轉(鐵捲門上移)，綠燈亮，此時將搖桿右移，仍需維持目前狀態，必須在馬達停止運轉後才可改變狀態；
- 3.如果搖桿右移(X軸)，且X值 > 800 ，則馬達逆轉(鐵捲門下移)，藍燈亮，此時將搖桿左移，仍需維持目前狀態，必須在馬達停止運轉後才可改變狀態；
- 4.按壓搖桿按鍵，則馬達停止運轉，LED燈滅。

演算法步驟(學生填空用)

(4) 演算法步驟：

01. 讀取搖桿方向值 X 及按鍵值 Button

02. 判斷 ($X < 50$)

03. 成立：馬達正轉(上移)，綠燈亮，

04. 否則：

05. 不成立：再判斷 ($X > 800$)

06. 成立：馬達逆轉(抽風)，藍燈亮，

07. 否則：

08. 不成立：再判斷 Button = 1

09. 成立：風扇停止，LED燈滅，

10. 不成立：回到步驟1

演算法步驟(教師用)

(4) 演算法步驟：

01. 讀取搖桿方向值 X 及按鍵值 Button

02. 判斷 ($X < 50$ 且 mode 不等於 2)

03. 成立：馬達正轉(上移)，綠燈亮，設定 mode=1

04. 否則：

05. 不成立：再判斷 ($X > 800$ 且 mode 不等於 1)

06. 成立：馬達逆轉(抽風)，藍燈亮，設定 mode=2

07. 否則：

08. 不成立：再判斷 Button = 1

09. 成立：風扇停止，LED 燈滅，設定 mode=3

10. 不成立：回到步驟 1

演算法步驟 vs 積木程式堆疊

(4) 演算法步驟：

01. 讀取搖桿方向值 X 及按鍵值 Button
02. 判斷 ($X < 50$ 且 mode 不等於 2)
03. 成立：馬達正轉(上移)，綠燈亮，設定 mode=1
04. 否則：
05. 不成立：再判斷 ($X > 800$ 且 mode 不等於 1)
06. 成立：馬達逆轉(抽風)，藍燈亮，設定 mode=2
07. 否則：
08. 不成立：再判斷 Button = 1
09. 成立：風扇停止，LED 燈滅，設定 mode=3
10. 不成立：回到步驟 1

(5) 積木程式堆疊



程式 1

演算法步驟 vs 積木程式堆疊

(4) 演算法步驟：

01. 讀取搖桿方向值 X 及按鍵值 Button
02. 判斷 ($X < 50$ 且 mode 不等於 2)
03. 成立：馬達正轉(上移)，綠燈亮，設定 mode=1
04. 否則：
05. 不成立：再判斷 ($X > 800$ 且 mode 不等於 1)
06. 成立：馬達逆轉(抽風)，藍燈亮，設定 mode=2
07. 否則：
08. 不成立：再判斷 Button = 1
09. 成立：風扇停止，LED 燈滅，設定 mode=3
10. 不成立：回到步驟 1

(5) 積木程式堆疊



程式 2

演算法步驟 vs 積木程式堆疊

(4) 演算法步驟：

01. 讀取搖桿方向值 X 及按鍵值 Button
02. 判斷 ($X < 50$ 且 mode 不等於 2)
03. 成立：馬達正轉(上移)，綠燈亮，設定 mode=1
04. 否則：
05. 不成立：再判斷 ($X > 800$ 且 mode 不等於 1)
06. 成立：馬達逆轉(抽風)，藍燈亮，設定 mode=2
07. 否則：
08. 不成立：再判斷 Button = 1
09. 成立：風扇停止，LED 燈滅，設定 mode=3
10. 不成立：回到步驟 1

(5) 積木程式堆疊



程式 3

演算法步驟 vs 積木程式堆疊

(4) 演算法步驟：

01. 讀取搖桿方向值 X 及按鍵值 Button
02. 判斷 ($X < 50$ 且 mode 不等於 2)
03. 成立：馬達正轉(上移)，綠燈亮，設定 mode=1
04. 否則：
05. 不成立：再判斷 ($X > 800$ 且 mode 不等於 1)
06. 成立：馬達逆轉(抽風)，藍燈亮，設定 mode=2
07. 否則：
08. 不成立：再判斷 Button = 1
09. 成立：風扇停止，LED 燈滅，設定 mode=3
10. 不成立：回到步驟 1

(5) 積木程式堆疊



程式 4

演算法步驟 vs 積木程式堆疊

(4) 演算法步驟：

01. 讀取搖桿方向值 X 及按鍵值 Button
02. 判斷 ($X < 50$ 且 mode 不等於 2)
03. 成立：馬達正轉(上移)，綠燈亮，設定 mode=1
04. 否則：
05. 不成立：再判斷 ($X > 800$ 且 mode 不等於 1)
06. 成立：馬達逆轉(抽風)，藍燈亮，設定 mode=2
07. 否則：
08. 不成立：再判斷 Button = 1
09. 成立：風扇停止，LED 燈滅，設定 mode=3
10. 不成立：回到步驟 1

重複無限次

(5) 積木程式堆疊

The image shows a block-based programming script for a motor control system. The script is organized into a loop structure. At the top, there is a 'Repeat indefinitely' block. Inside this loop, the first block is 'Set variable X to read joystick position A0'. This is followed by a 'Set variable Button to read digital pin 7'. Then, there is an 'If' block with the condition 'X < 50 and mode != 2'. If true, it sets the motor speed to 255, sets digital pin 10 to high (green light), and digital pin 11 to low (blue light), and sets mode to 1. If false, it goes to another 'If' block with the condition 'X > 800 and mode != 1'. If true, it sets the motor speed to -255, sets digital pin 10 to low (green light), and digital pin 11 to high (blue light), and sets mode to 2. If false, it goes to a third 'If' block with the condition 'Button = 1'. If true, it sets the motor speed to 0, sets digital pin 10 to low (green light), and digital pin 11 to low (blue light), and sets mode to 3. If false, it loops back to the beginning of the 'Repeat indefinitely' block.