

Motion Studio 教程 3

為程序加上互動, 優化 “XY 平台點對點運動” 程序

說明

在上一個教程(教程 2 我的第一支程式) 實際上只能算是測試程序, 因為它單純只是讓機台順序運行到 P1, P2. 而設備運行時有 2 個重點是不能忽略的.

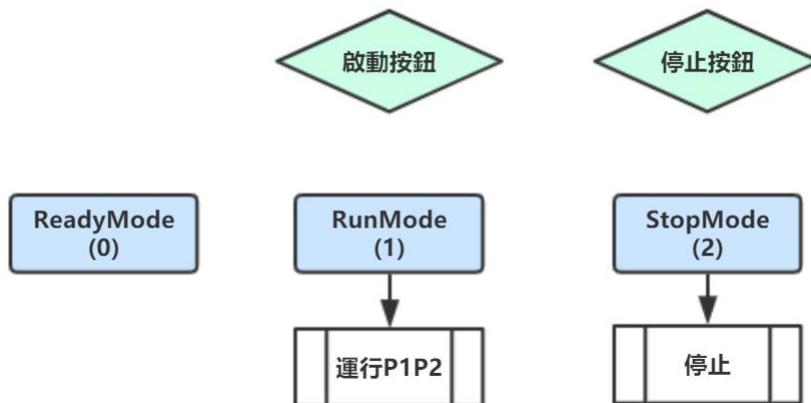
1. 不能忽略人機的互動處理: 例如動作是由啟動按鈕來觸發
2. 不能忽略設備動作的停止處理: 程序必須考慮突發狀況, 隨時可以停止.

我的第一支程式 - XY 平台點對點運動程序代碼	
BASE 0, 1	'選擇軸 0,1 參與運動
MOVE 10000, 5000	'對軸 0,1 下 MOVE 移動命令, 相對移動到 P1 點
WAIT DONE	'等待軸 0,1 運動停止
SLEEP 2000	'延遲 2000ms, 並等待延遲時間到
LINE 10000, 5000	'對軸 0,1 下 LINE 移動命令, 相對移動到 P2 點
WAIT DONE	'等待軸 0,1 運動停止

接下來在動作不變情況下, 我們替程序加上互動,

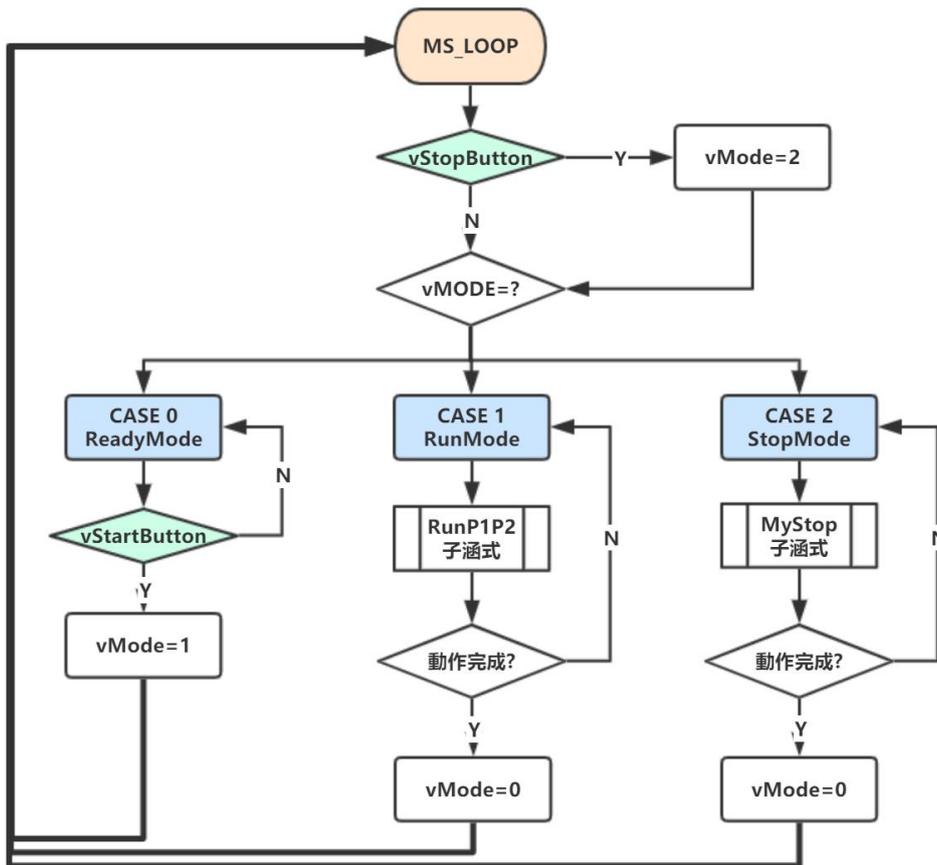
目標:

1. 系統有 3 種模式狀態 : (ReadyMode, RunMode, StopMode)
2. 可以知道系統處於甚麼模式狀態
3. ReadyMode 可以接收**啟動按鈕**訊號, 執行 P1P2 動作
4. 可以隨時接收**停止按鈕**訊號, 執行停止



我們設計了 2 個虛擬按鈕, 3 種模式, 並使用 VR 紀錄模式與運行步驟, 如下表格與流程

別名	VR	用途
vMODE	VR(0)	紀錄目前模式
vRunStep	VR(1)	紀錄目前運行步驟
vStopStep	VR(2)	紀錄目前停止步驟
vRunButton	VR(3)	虛擬啟動按鈕
vStopButton	VR(4)	虛擬停止按鈕



Ready Mode (0):

一開始進入的模式, 等待啟動按鈕按下

Run Mode (1):

運行 P1P2 模式, 動作完成後切換到 Ready Mode (0)

Stop Mode (2):

停止所有運動, 動作完成後切換到 Ready Mode (0)

主步驟

全部代碼共 60 行, 可直接跳到步驟 5 看完整代碼.

- 步驟 1 – 替虛擬按鈕 VR 定義別名
- 步驟 2 – 主程序架構
- 步驟 3 – RunP1P2() 子函式
- 步驟 4 – MyStop() 子函式
- 步驟 5 – 完成代碼
- 步驟 6 – 測試

步驟 1 – 替虛擬按鈕 VR 定義別名

為了增加程序可讀性, 我們先在程式開頭將用到的 VR 定義程可識別的名稱. 這樣我們就可以在 bas 程序中直接使用下面已定義的別名.

```
#DEFINE vMODE      VR(0)  '紀錄目前模式
#DEFINE vRunStep   VR(1)  '紀錄目前運行步驟
#DEFINE vStopStep  VR(2)  '紀錄目前停止步驟
#DEFINE vRunButton VR(3)  '虛擬啟動按鈕
#DEFINE vStopButton VR(4) '虛擬停止按鈕
```

步驟 2 – 主程序架構

1. 先建立主循環流程:

使用 MS_LOOP(10)循環指令, 其中 10 代表每次循環週期是 10ms. 在 MS_LOOP()...MS_LEND 範圍內的程序, 都會 10ms 執行一次.

```
MS_LOOP(10)      'Task 主流程循环
...
MS_LEND
```

2. 設計 3 種系統模式

重頭戲, 我們設計了 3 種狀態, 0: Ready, 1:Run, 2:Stop. 並使用 SELECT CASE 來處理這三種狀態. 並在程序一開始將 vMode 狀態設為 0: Ready

```
vMODE=0          'vMode 初始狀態設為 0: Ready
MS_LOOP(10)      'Task 主流程循环
  SELECT CASE CINT(vMODE)
    CASE 0       'Ready Mode
    CASE 1       'Run Mode
    CASE 2       'Stop Mod
  END SELECT
MS_LEND
```

3. 狀態切換: 運行 P1P2 處理

- 架構完成後, 就可以在 CASE 0(Ready Mode)加入偵測啟動按鈕 **vRunButton** 的處理, 這樣在 Ready Mode 下, 只要按下 vRunButton(0 變 1), 就將 vMode 設為 1, 下一個循環就會跳到 CASE 1 (Run Mode),
- 在 CASE 1(Run Mode) 加入自定義子涵式 RunP1P2(). 當進入 CASE 1 (Run Mode) 會持續執行 RunP1P2() 直到 vRunStep 等於 STEP_DONE 時將 vMode=0 離開 CASE1, 重新進入 CASE0.

Note: 由於 RunP1P2() 內包含多個動作, 因此需要持續執行, 直到動作完成.

```

vMODE=0
MS_LOOP(10)          'Task 主流程循环
  SELECT CASE CINT(vMODE)
  CASE 0             'Ready Mode
    IF MS_EDGER(vRunButton) THEN
      vMODE=1       'RunButton 发生时將模式切换到 1(Run)
      vRunStep=0    '進入此模式將 RunStep 步驟清為 0
    END IF
  CASE 1             'Run Mode'
    RunP1P2()
    IF vRunStep=1000 THEN vMODE=0  '如果步驟完成，回到 ReadyMode
  END SELECT
MS_LEND

```

4. 狀態切換: 動作的停止處理, 讓設備隨時可以停止

為了讓停止有最高優先權, 在 MS_LOOP 循環的開始, 我們加入對 vStopButton 停止命令的處理, 並使用 MS_EDGER 指令偵測 vStopButton 有無變化 如果有變化, 立刻將 vMode 模式切換成 2(Stop), 這樣在 SELECT CASE 中就會執行跳到 CASE 2 並執行 STOP()停止子程序

同時在 CASE2 中加入 MyStop()子函式, 讓程序進入 CASE2 後會持續執行 MyStop(), 直到 vStopStep=1000 時將 vMode=0 離開 CASE1, 重新進入 CASE0.

```

vMODE=0
MS_LOOP(10)          'Task 主流程循环
  IF MS_EDGER(vStopButton) THEN
    vMODE=2         'StopButton 发生时'將模式切换到 Stop
    vStopStep=0    '進入此模式將停止流程步驟清為 0
  END IF

  SELECT CASE CINT(vMODE)
  CASE 0 ...
  CASE 1 ...
  CASE 2           'Stop Mode
    MyStop()
    IF vStopStep=1000 THEN vMODE=0  '如果步驟完成，回到 ReadyMode
  END SELECT
MS_LEND

```

5. 完成的主架構代碼

我們先完成主架構，讓程序的輪廓先出來，接下來再進行 RunP1P2() 與 MyStop() 兩個子函式。

```
主架構代碼
VRCLEAR(0,10)      '清除 VR(0) - VR(10)
vMODE=0
MS_LOOP(10)        'Task 主流程循環
  IF MS_EDGER(vStopButton) THEN
    vMODE=2        'StopButton 发生时'將模式切換到 Stop
    vStopStep=0    '進入此模式將停止流程步驟清為 0
  END IF

  SELECT CASE CINT(vMODE)
  CASE 0           'Ready Mode
    IF MS_EDGER(vRunButton) THEN
      vMODE=1      'RunButton 发生时將模式切換到 1(Run)
      vRunStep=0   '進入此模式將 RunStep 步驟清為 0
    END IF
  CASE 1           'Run Mode'
    RunP1P2()
    IF vRunStep=1000 THEN vMODE=0 '如果步驟完成，回到 ReadyMode
  CASE 2           'Stop Mode
    MyStop()
    IF vStopStep=1000 THEN vMODE=0 '如果步驟完成，回到 ReadyMode
  END SELECT
MS_LEND
```

步驟 3 – 子函式：RunP1P2()

1. 定義子程序 RunP1P2()

使用 SUB 來定義子程序 RunP1P2, 在 SUB ... END SUB 範圍內的程序都是屬於這個子程序

```
SUB RunP1P2
...
END SUB
```

2. 實現子程序 RunP1P2()

在 SUB RunP1P2 裡,我們使用 SELECT CASE CINT(vRunStep)來控制 SUB RunP1P2 的動作執行順序, 當 CASE0 動作執行完, 就切換成 CASE10, 最後在 CASE 30 完成動作。

CASE 0: '初始化 → CASE 10: '移動到 P1 点 → CASE 20: '延遲 2 秒 → CASE 30: '移動到 P2 点

重點說明: 動作執行與切換 - 以 CASE 10 為例:

重點 1: 由於運動指令 MOVEABS 只需執行一次, 因此我們使用 MS_PULSE(RunStep)來讓 MOVEABS 只執行一次, MS_PULSE 會偵測 vRunStep 值是否有發生變化, 當有變化時他會回傳 1. 沒有變化回傳 0. 因此

搭配 IF 判斷式, 就能讓 MOVEABS 10000, 5000 只執行一次.

重點 2: 當 MOVEABS 10000, 5000 執行後, 我們使用 AxisReady() 來判斷軸是否是 Ready 狀態, 當 Ready 狀態時, 就將 RunStep 切換到 20, 這樣下個循環週期就會跳到 CASE 20

Note: CASE n 使用 0, 10, 20,...的原因: 從實際的經驗來看, 開發的後期往往需要依實際狀況添加新動作, 因此我們預留空間, 讓後面可以在原有的 CASE10, CASE20 動作間, 插入 CASE11, CASE12...的動作.

RunP1P2 子程序完整代碼

```
SUB RunP1P2
  SELECT CASE CINT(vRunStep)
  CASE 0 '初始化
    BASE 0,1
    Timer_A.Reset() '重置 Timer_A
    vRunStep=10
  CASE 10 '移動到 P1 点
    IF MS_PULSE(vRunStep) THEN MOVEABS 10000, 5000
    IF AxisReady(0,1)=1 THEN vRunStep=20
  CASE 20 '延遲 2 秒
    IF Timer_A.On(2000) THEN vRunStep=30
  CASE 30 '移動到 P2 点
    IF MS_PULSE(vRunStep) THEN LINEABS 20000, 10000
    IF AxisReady(0,1)=1 THEN vRunStep=1000 '完成
  END SELECT
END SUB
```

重點 1

重點 2

< 注意 >

我們可以看到 在上一個步驟的主流程代碼中, 在調用 RunP1P2() 前有一行代碼,
IF MS_PULSE(vMODE) THEN vRunStep=0 '進入此模式將 RunStep 步驟清為 0
目的是為了在執行 RunP1P2()前將 vRunStep=0, 好讓 RunP1P2()裡面的順序能夠從 CASE 0 開始

```
...
CASE 1 'Run Mode
  IF MS_PULSE(vMODE) THEN vRunStep=0 '進入此模式將 RunStep 步驟清為 0
  RunP1P2()
  IF vRunStep=STEP_DONE THEN vMODE=0
...
```

步驟 4 – 子函式 : MyStop()

1. 定義 MyStop()子程序

使用 SUB 來定義子程序 MyStop，在 SUB ... END SUB 範圍內的程序都是屬於這個子程序

```
SUB MyStop
...
END SUB
```

2. 實現 MyStop()子程序

在 SUB MyStop 裡，我們一樣使用 SELECT CASE CINT(vRunStep)來控制 MyStop()的動作執行順序，當 CASE0 動作執行完，就切換成 CASE10 完成動作。

CASE 0: '停止 → CASE 10: '停止動作完成

MyStop 子程序完整代碼

```
SUB MyStop '停止动作
SELECT CASE CINT(vStopStep)
CASE 0 '停止
BASE 0,1
STOPDEC
vStopStep=10
CASE 10 '等待停止完成
IF AxIsReady(0,1)=1 THEN vStopStep=1000 '完成
END SELECT
END SUB
```

< 注意 >

我們可以看到 在步驟 2 的主流程代碼中，在按鈕按下後，除了將模式 vMode=2 外，也會將 vStopStep=0，目的是為了在執行 MyStop()前先將 vStopStep=0，好讓 MyStop()裡面的順序能夠從 CASE 0 開始

```
MS_LOOP(10) 'Task 主流程循环
IF MS_EDGER(vStopButton) THEN
vMODE=2 'StopButton 发生时'將模式切換到 Stop
vStopStep=0 '進入此模式將停止流程步驟清為 0
END IF
```

步驟 5 – 完成代碼 (全部代碼)

```
#DEFINE vMODE VR(0) '紀錄目前模式
#DEFINE vRunStep VR(1) '紀錄目前運行步驟
#DEFINE vStopStep VR(2) '紀錄目前停止步驟
#DEFINE vRunButton VR(3) '虛擬啟動按鈕
#DEFINE vStopButton VR(4) '虛擬停止按鈕
#DEFINE STEP_DONE 1000

DIM SHARED as TMR Timer_A '定義一個計時器 Timer_A
```

```

SUB RunP1P2
SELECT CASE CINT(vRunStep)
CASE 0 '初始化
    BASE 0,1
    VH=3000
    Timer_A.Reset() '重置 Timer_A
    vRunStep=10
CASE 10 '移動到 P1 点
    IF MS_PULSE(vRunStep) THEN MOVEABS 10000, 5000
    IF AxIsReady(0,1)=1 THEN vRunStep=20
CASE 20 '延遲 2 秒
    IF Timer_A.On(2000) THEN vRunStep=30
CASE 30 '移動到 P2 点
    IF MS_PULSE(vRunStep) THEN LINEABS 20000, 10000
    IF AxIsReady(0,1)=1 THEN vRunStep=1000 '完成
END SELECT
END SUB

SUB MyStop '停止动作
SELECT CASE CINT(vStopStep)
CASE 0 '初始化
    BASE 0,1
    STOPEMG
    vStopStep=10
CASE 10
    IF AxIsReady(0,1)=1 THEN vStopStep=1000 '完成
END SELECT
END SUB

VRCLEAR(0,10) '清除 VR(0) - VR(10)
vMODE=0
MS_LOOP(10) 'Task 主流程循环
    IF MS_EDGER(vStopButton) THEN
        vMODE=2 'StopButton 发生时將模式切换到 Stop
        vStopStep=0 '進入此模式將停止流程步驟清為 0
    END IF

    SELECT CASE CINT(vMODE)
    CASE 0 'Ready Mode
        IF MS_EDGER(vRunButton) THEN
            vMODE=1 'RunButton 发生时將模式切换到 1(Run)
            vRunStep=0 '進入此模式將 RunStep 步驟清為 0
        END IF
    CASE 1 'Run Mode'
        RunP1P2()
        IF vRunStep=1000 THEN vMODE=0 '如果步驟完成, 回到 ReadyMode
    CASE 2 'Stop Mode
        MyStop()
        IF vStopStep=1000 THEN vMODE=0 '如果步驟完成, 回到 ReadyMode
    END SELECT
MS_LEND

```

步驟 6 – 測試

編譯正常後，即可開始調試。Motion Studio 提供下面方法快速調試

1. 模擬按鈕動作: 可藉由 VR 工具, 改變 VR [當前值]欄位, 來模擬 0 到 1 的變化
2. 觀察 VR 變化: 由 VR 工具[當前值]欄位的數值來觀察變化
3. 觀察位移變化: 藉由軸狀態工具 DPOS 欄位來觀察位移變化

請切換成定時更新模式 (Red arrow pointing to the refresh icon in the VR table)

1. 改變VR(3), VR(4)當前值 來模擬按鈕按下 (Blue arrow pointing to VR(3) and VR(4) in the table)

2. 由VR(0)-VR(3)當前值 來觀察模式與步驟是否正確 (Blue arrow pointing to VR(0) through VR(3) in the table)

3. 由DPOS來觀察位移變化 (Blue arrow pointing to DPOS values in the axis status table)

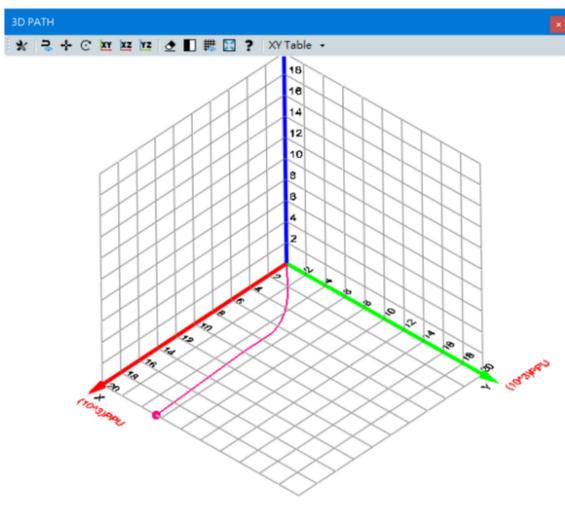
```

4 #DEFINE vRunButton VR(3)
5 #DEFINE vStopButton VR(4)
6 #DEFINE STEP_DONE 1000
7
8 DIM SHARED as TMR Timer_A
9
10 SUB RunP1P2
11   SELECT CASE CINT(vRunStep)
12   CASE 0 '初始化
13     BASE 0,1
14     VH=3000
15     Timer_A.Reset()
16     vRunStep=10
17   CASE 10 '移動到P1点
18     IF MS_PULSE(vRunStep) THEN MOVEABS 10000, 5000
19     IF AxisReady(0,1)=1 THEN vRunStep=20
20   CASE 20 '延遲2秒
21     IF Timer_A.On(2000) THEN vRunStep=30
    
```

名称	当前值	描述	初
范围[0-1...			
VR(0)	0	MODE	
VR(1)	1000	RunStep	
VR(2)	1000	StopStep	
VR(3)	1	RunButton	
VR(4)	0	StopButton	
VR(5)	0		
VR(6)	0		
VR(7)	0		
VR(8)	0		
VR(9)	0		
VR(10)	0		

轴	STATE	SVON	EL+	EL-	ORG	ALM	SEL+	SEL-	INP	EMG	DPOS	MPOS	DSPEED	R
轴(0)	READY	■	■	■	■	■	■	■	■	■	20,000	20,000	0	
轴(1)	READY	■	■	■	■	■	■	■	■	■	10,000	10,000	0	
轴(2)	READY	■	■	■	■	■	■	■	■	■				
轴(3)	READY	■	■	■	■	■	■	■	■	■				

或是開啟 3D Path 軌跡工具觀看運行效果



Motion Studio 官網：<http://mas.advantech.com.cn>