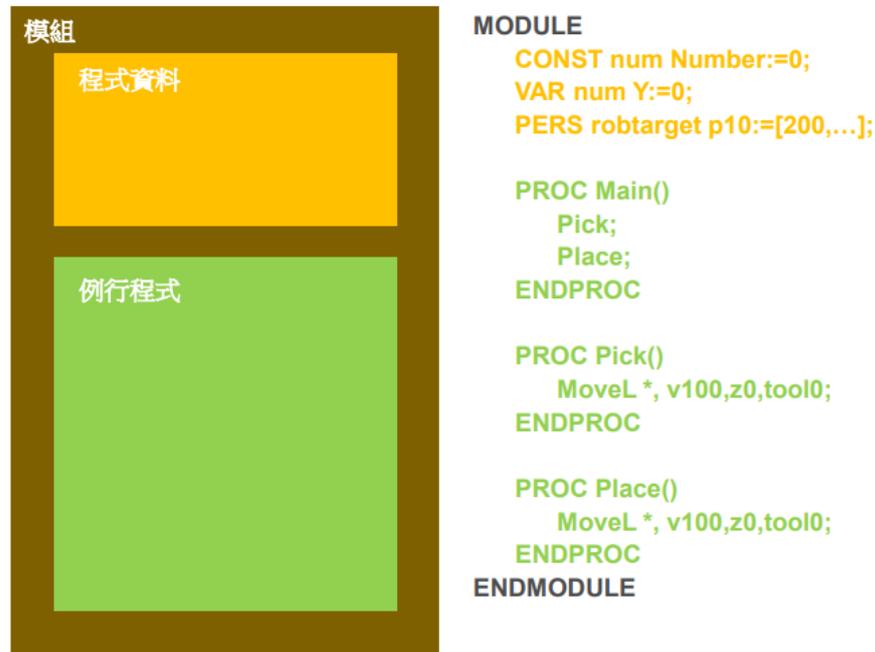


程式概念



機器人程序儲存器中，只允許存在一個主程序；所有例行程序與數據無論存在於哪個模塊，全部被系統共享；所有例行程序與數據除特殊定義外，名稱必須是唯一的。

註:只會有一個主程式 main

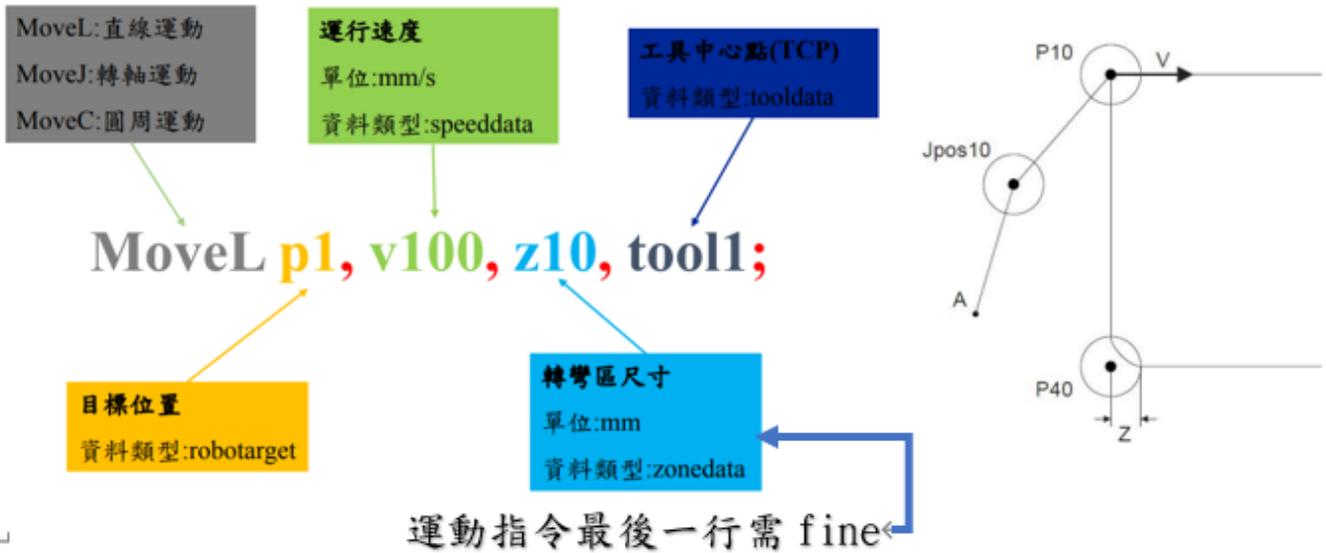
程式指令種類:

- Common 常用的指令
- Prog.Flow (Controlling the program flow) 控制程序流程類
 1. 调用另一程序（無返回值程序）並在執行該程序後，按指令繼續執行；
 2. 基于是否滿足給定條件，執行不同指令；
 3. 重複某一指令序列多次，直到滿足給定條件；
 4. 移至同一程序中的某一標籤；
 5. 終止程序執行過程。
- Various (Various instructions) 各種指令
 1. 給數據賦值；

2. 等待一段指定時間或等到滿足條件時；
3. 在程序中插入注釋；
4. 加載編程模塊。

- Settings (Motion settings) 運動設置
- Motion&Proc/ Motion Adv./ MotionSetAdv 運動
- I/O (Input and output signals) 輸入輸出
- Communicate (Communication) 通信
- Interrupts 中斷
- Error Rec. (Error recovery) 錯誤恢復
- System & time 系統與時間
- Mathematics 數學

程式指令說明-運動指令



基礎指令

MoveL：機器人從當前目標點位置以線性方式移動至指令目標點。

MoveJ：機器人從當前目標點位置以軸運動方式移動至指令目標點。

MoveC：機器人從當前目標點位置以圓運動方式移動至指令目標點。

MoveAbsJ：機器人從當前位置單軸移動至指令設定的各軸角度。

IF：條件判斷。

FOR：迴圈控制。

WHIEL：條件判斷，條件成立會在條件內執行程式，直到條件不成立跳出迴圈。

WaitDI：等待一個輸入訊號。

WaitTime：等待時間。

WaitUntil：複合式條件判斷。

運動控制指令(進階影片 01、02)

Accset: 機器手臂運行速度改變時，對所產生的相應加速度進行限制。

AccSet Acc, Ramp;

註: ACC : 機器手臂加速度百分比(num)，最小 20

Ramp: 機器手臂加速度坡度(num)，最小 10

Velset: 速度控制

velset override max;

註: override: 機器手臂速度百分比(num)

max : 機器手臂速度最大值(num)

EX:

Velset 50, 800;

Movel p1, V1000, z10, tool1; $1000 * 50 = 500 \text{mm/s}$

Movel p2, V2000, z10, tool1; $2000 * 50 = 1000 > 800 \rightarrow 800 \text{mm/s}$

ConfJ: 使用 MOVEJ 時對手臂運行姿態進行限制與調整，系統默認為 on。

ConfL: 使用 MOVEL 時對手臂運行姿態進行限制與調整，系統默認為 on。

SingArea: 位置方位調整

SingArea[\Wrist][\off];

註:

Wrist:

啟用位置方位調整。(switch) 手臂運行時，為了避免死機，位置點允許其方位有些改變。例: 奇異點

off:關閉位置方位調整。(switch)默認值

SoftAct:軟化手臂各關節軸或外軸系統，軟化值範圍 0%~100%，而 $\geq 100\%$ 需與指令 SoftDeact 同時使用，通常不使用在工作位置。

SoftAct[\MechUnit,]Axis, Softness[\Ramp];

註:

[\MechUnit]:軟化外軸名稱

Axis:軟化外軸號碼

Softness:軟化值%

[\Ramp]:軟化坡度%

SoftDeact:用於使軟化手臂主機或外軸伺服系統 SoftAct 失效。

SoftDeact[\Ramp];

[\Ramp]:軟化坡度%, $\geq 100\%$

外軸開通指令(進階影片 03)

ActUnit:使手臂外軸開啟。

DeactUnit:使手臂外軸關閉。

計數指令(進階影片 03)

Add:在一個數字數據上增加相應值，可用賦值指令代替。

Add Name, AddValue;

註:

Name:數據名稱

AddValue:增加的值

Clear:將一個數值數據值歸 0，可用賦值指令代替。

Clear Name;

Incr:將一個數值數據值增加 1，可用賦值指令代替。

Incr Name;

Decr: 將一個數值數據值減少 1，可用賦值指令代替。

Decr Name;

輸入輸出指令(進階影片 04)

InvertD0: 手臂輸入信號值反轉，0 為 1，1 為 0。

InvertDo Signal;

註:

Signal: 輸出信號名稱

PluseD0: 手臂輸出數字脈衝信號，一般作為運輸鏈完成信號或計數信號。

PluseD0[\High][\Plenggh]Signal;

註:

[\High]: 輸出脈衝信號時，輸出信號可以處在高電平

[\Plenggh]: 脈衝長度，0.1s~32s, 默認值為 0.2s

Signal: 輸出信號名稱

Reset: 將相對應的訊號設定為 0

Reset signal;

Set: 將相對應的訊號設定為 1

Set signal;

SetA0: 手臂當前模擬量輸出信號輸出相應值。

SetA0 signal, Value;

註:

signal: 模擬量輸出信號名稱 (signaldo)

Value: 模擬量輸出信號值 (num)

SetD0: 設置手臂相應數字輸出信號值。

SetDO[\Sdelay]signal, Value;

註:

[\Sdelay]:延遲輸出時間 s (num)

signal: 輸出信號名稱 (signaldo)

Value: 輸出信號值 (num)

SetGO: 設置手臂相應組合數字輸出信號值。

SetGO[\Sdelay]signal, Value;

註:

[\Sdelay]:延遲輸出時間 s (num)

signal: 輸出信號名稱 (signaldo)

Value: 輸出信號值 (num)

WaitDI: 等待數字輸入信號滿足相應值，達到通訊目的。

WaitDI Signal, Value [\MaxTime][\TimeFlag];

註:

signal: 輸出信號名稱 (signaldo)

Value: 輸出信號值 (dionum)

[\MaxTime]:最長等待時間 (num)

[\TimeFlag]:超出邏輯量 (bool)

WaitDO: 等待數字輸入信號滿足相應值，達到通訊目的。

WaitDO Signal, Value [\MaxTime][\TimeFlag];

註:

signal: 輸出信號名稱 (signaldo)

Value: 輸出信號值 (dionum)

[\MaxTime]:最長等待時間 (num)

[\TimeFlag]:超出邏輯量 (bool)

程序運行停止指令(進階影片 04)

break: 手臂在當前指令行立刻停止運行，程式運行指針停在下一行指令，可 Start 繼續運行。

Exit: 手臂在當前指令行立刻停止運行，並程序重置，程式運行指針回主程式第一行。

STOP: 手臂在當前指令行立刻停止運行，程式運行指針停在下一行指令，可 Start 繼續運行，如果手臂停止時間被手動移動後，然後直接啟動手臂，手臂將警告確認路徑。

```
Stop[\NoRegain];
```

ExitCycle: 手臂在當前指令行立刻停止運行，並且設定當前循環結束，手臂自動從主程式第一行繼續運行下一個循環。

例行程序指令(進階影片 04)

ProcCall: 呼叫例行程式。

```
Procedure{Argument};
```

註:

Procedure: 例行程序名稱 (Identifier)

{Argument}: 例行程序參數 (all)

CallByvar: 呼叫例行程式，但無法調用帶有參數的例行程序。

```
CallByVar Name, Number;
```

註:

Name: 例行程序名稱第一部分 (string)

Number: 例行程序名稱第二部分 (num)

計時指令(進階影片 05)

ClkReset: 將手臂相應德時鐘復位，常用於紀錄循環時間或手臂跟蹤輸鍊。

註:

Clock1: 時鐘名稱(clock)

註: 實例

```
ClkReset clock1;
```

```
ClkStart clock1;
```

```
RunCycle;
```

```
ClkStop clock1;
```

```
nCycleTime:=ClkRead(clock1);
```

```
TPWrite "Last Cycle Time:" \Num:=nCycleTime
```

ClkStart: 啟動手臂相應時鐘，常用於紀錄循環時間或手臂跟蹤輸鍊。

ClkStop: 手臂時鐘計時超過 4,294,967 秒，既 49 天 17 小時 2 分 47 秒，手臂將出錯。

註:

Clock1: 時鐘名稱(clock)

註: 實例

```
ClkReset clock1;
```

```
ClkStart clock1;
```

```
RunCycle;
```

```
ClkStop clock1;  
nCycleTime:=ClkRead(clock1);  
TPWrite "last Cycle Time:" \Num:=nCycleTime;
```

中斷指令(進階影片 05)

CONNECT: 將手臂相應中斷數據連接到相應的中斷處理程序，是手臂中斷功能必不可少的組成部分，必須同指

令: ISignalDI, ISignalDO, ISignalAI, ISignalAO 或 ITimer 聯合使用

註:

Interrupt: 中斷數據名稱(intnum)

Trap routine: 中斷數據程序(identifier)

IDelete: 將手臂相應中斷數據與相應的中段處理程序之間元連接去除。

ISignalDI: 使用相應的數字信號輸入信號觸發相應的中斷功能，必須同指令 CONNECT 聯合使用。

註:

[\Single]: 單次中斷開關(switch)

Signal: 觸發中斷信號(signaldi)

TriggValue: 觸發信號值(dionum)

Intreeupu: 中斷數據名稱(intnum)

ISignalD0: 使用相應的數字信號輸出信號觸發相應的中斷功能，必須同指令 CONNECT 聯合使用。

ISignalAI: 使用相應的模擬量輸入信號觸發相應的中斷功能，必須同指令 CONNECT 聯合使用。

註:

Condition: 中斷觸發狀態(adotrigg)

HighValue: 最大邏輯值(num)

LowValue: 最小邏輯值(num)

DeltaValue: 中斷復位差值(num)

[\Dpos]: 正值中斷開關(switch)

[\Dneg]: 負值中斷開關(switch)

ISignalA0: 使用相應的模擬量輸出信號觸發相應的中斷功能，必須同指令 CONNECT 聯合使用。

ISleep: 手臂相應中斷數據暫時失效，直到執行指令 IWatch 後才恢復。

Iwatch: 激活手臂已失效的相應中斷數據，正常情況下，與指令 ISleep 配合使用。

IDisable: 手臂相應中斷功能暫時不執行，直到執行指令 IEnable 後，才進入中斷處理程序，此指令使用於手臂正在執行不希望被打斷操作時。

IEnable:

ITimer: 定時處理手臂相應中斷數據，此指令常用於通過通訊信口讀寫數據場合。

註:

Time: 觸發中斷時間(num)

通信指令(人機對話)(進階影片 06)

TPErase: 清屏指令。

註: 實例

TPErase;

TPWrite “ ABB Robotics ” ;

TPWrite “ _____ ” ;

TPWrite: 在示教器上屏幕上顯示相應字符串。

註:

String: 屏幕顯示的字符串 (string)

[\Num]: 屏幕顯示數字數據值 (string)

[\Bool]: 屏幕顯示邏輯量數據 (string)

[\POS]: 顯示位置值 XYZ (string)

[\Orient]: 顯示方位 q1, q2, q3, q4 (string)

TPReadFK: 在示教器上屏幕上顯示相應字符串。

註:

Answer: 數字賦值 1-5 (num)

Text: 屏幕字符串 (string)

FKx: 功能鍵字符串 (string)

[\MaxTime]: 最長等待時間 (num)

[\DIBreak]: 輸入信號控制 (signal di)

[\BreakFlag]: 指令狀態控制 (signal di)

TPReadNum: 在示教器上屏幕上顯示相應字符串。

註：

Answer: 數字賦值 (num)

String: 屏幕字符串 (string)

[\MaxTime]: 最長等待時間 (num)

[\DIBreak]: 輸入信號控制 (signal di)

[\BreakFlag]: 指令狀態控制 (signal di)

ErrWrite: 在示教器屏幕上顯示標準出錯介面，錯誤代碼為 80001。

註：

[\w]: 事件紀錄開關 (switch)

Header: 錯誤信息標題 (string)

Reason: 錯誤信息原因 (string)

[\RL2]: 附加錯誤信息原因 (string)

[\RL3]: 附加錯誤信息原因 (string)

[\RL4]: 附加錯誤信息原因 (string)

TPShow: 手臂在示教器屏幕顯示介面，通常與手臂附加軟體 ScreenViewer 配合使用。

註：

Window : 顯示相應示教器窗口 (tpnum)

出錯處理常用指令(進階影片 09)

EXIT: 當出現無法處理時將程序指令停止執行

RAISE: 當定制出錯處理時，用於激活出錯處理

RETRY: 再次執行激活出錯處理的指令

TRYNEXT: 執行激活出錯處理的下一句指令

RETURN: 回到之前的子程序

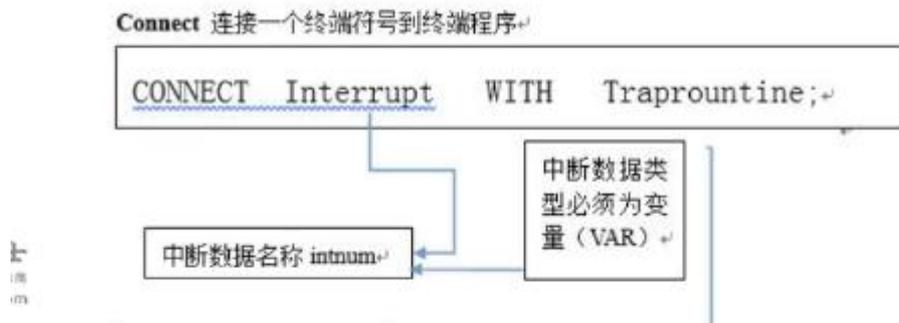
ResetRetryCount: 復位重試的次數

Offs 指令: 目標點位置偏移。為了精確確定目標點，可以採用函數

offs。Offs (p, x, y, z) 代表離 p 點 X 軸偏差量為 x, Y 軸偏差量為 y, Z 軸偏差量為 z

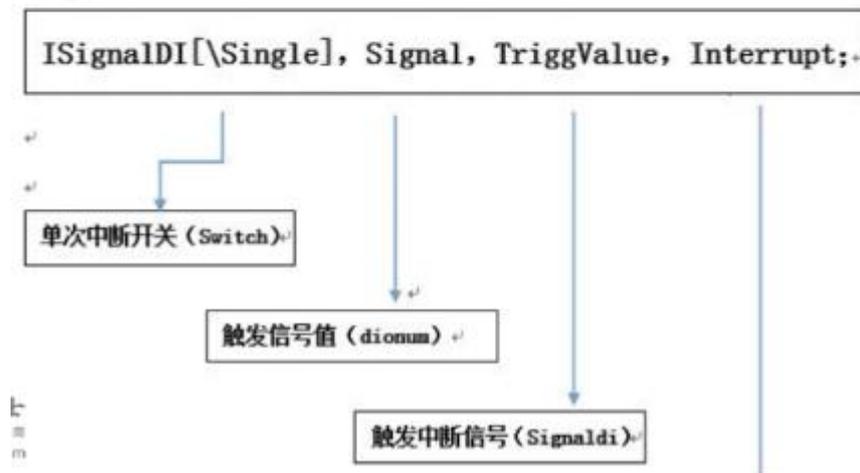
中斷指令

ISignalDI/Connect /IDelete



2) **IDelete** 取消中断

3) **ISignalDI** 使用一个数字输入信号触发中断



ISignalDO 使用一个数字输出信号触发中断

現以一個傳感器的信號進行實時監控為例編寫一個中斷程序：

① 在正常情況下，nCountBule 的信號的 0.

② 如果 nCountBule 的 信號從 0 變成 1，就對 nCountBule 數據進行加 1 的操作。操作步驟：



1. 新建例行程序，設定名稱，類型選擇“中斷”，然後單擊“確定”。



2. 在中斷程序中，添加如圖中所示的指令。



3. 在例行程序“InitAll()”中，添加指令“IDelete”（取消指定的中斷）。

4. 選擇“intnol”（如果沒有的話，就新建一個），然後單擊“確定”。

5. 選擇“CONNECT”指令（連接一個中斷符號到中斷程序）。

6. 雙擊“”進行設定，選中“intnol”，然後單擊“確定”。

7. 雙擊“”進行設定，選擇要關聯的中斷程序“rInterrupt”，然後單擊“確定”。

8. 選擇指令 “ISignalDI” （使用一個數字輸入信號觸發中斷）。

選擇 “DI_CHECKOK” ，然後單擊 “確定” 。

9. 雙擊此指令（ * 提示：ISignalDI 中的 Single 參數啟用，則此中斷只會響應

10. DI_CHECKOK 一次；若要重複響應，則將其去掉，單擊 “可選變量” 。

11. 單擊 “\Single” 進入設定畫面。

12. 選中 “\Single” ，然後單擊 “不使用”，單擊 “關閉” 。

13. 設定完成，此中斷程序只需在初始化例行程序 rInitAll 中執行一遍，即在程序執行的整個過程中都生效。