

範本框架使用

Version 1.0

概述

Motion Studio新建工程時，可以選擇新建“空的框架”和新建“範本框架”兩種方式。“空的程式框架”裡沒有任何代碼，由使用者自行編寫整個程式，而用“範本框架”新建工程後，Motion Studio裡就已經創建了兩個Task，並在系統內部封裝了一些函數、變數，在ProjectDefine.bi和兩個Task創建了部分程式以及創建了一部分VR變數區域。這些共同組成了“範本框架”。

“範本框架”定義了機器接受命令的介面和機器運行的狀態切換，使用者在程式框架下添加代碼，就可以很快完成一套設備運行的程式，讓設備穩定的運行。該“範本框架”特別適用順序流水作業的運動控制相關設備。

目的

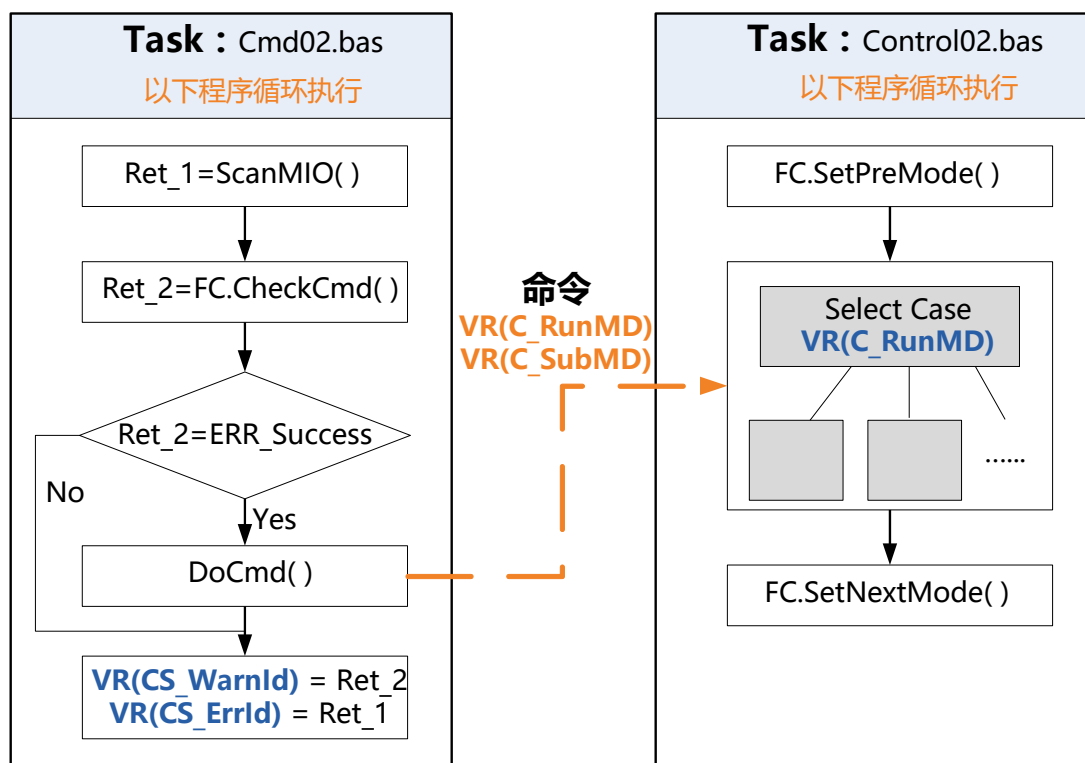
“範本框架”可以讓使用者快速搭建穩定可靠的程式，避免了開發者隨意開發導致的程式結構不佳，運行效率不高以及不易調試、維護等問題。同時，很好的使用該框架，可以用同一個框架應對各種非標設備，大幅縮短開發週期。

框架流程

Task 組成

框架由 2 個 Task 構成，兩個 Task 的主體結構都是 While 迴圈。

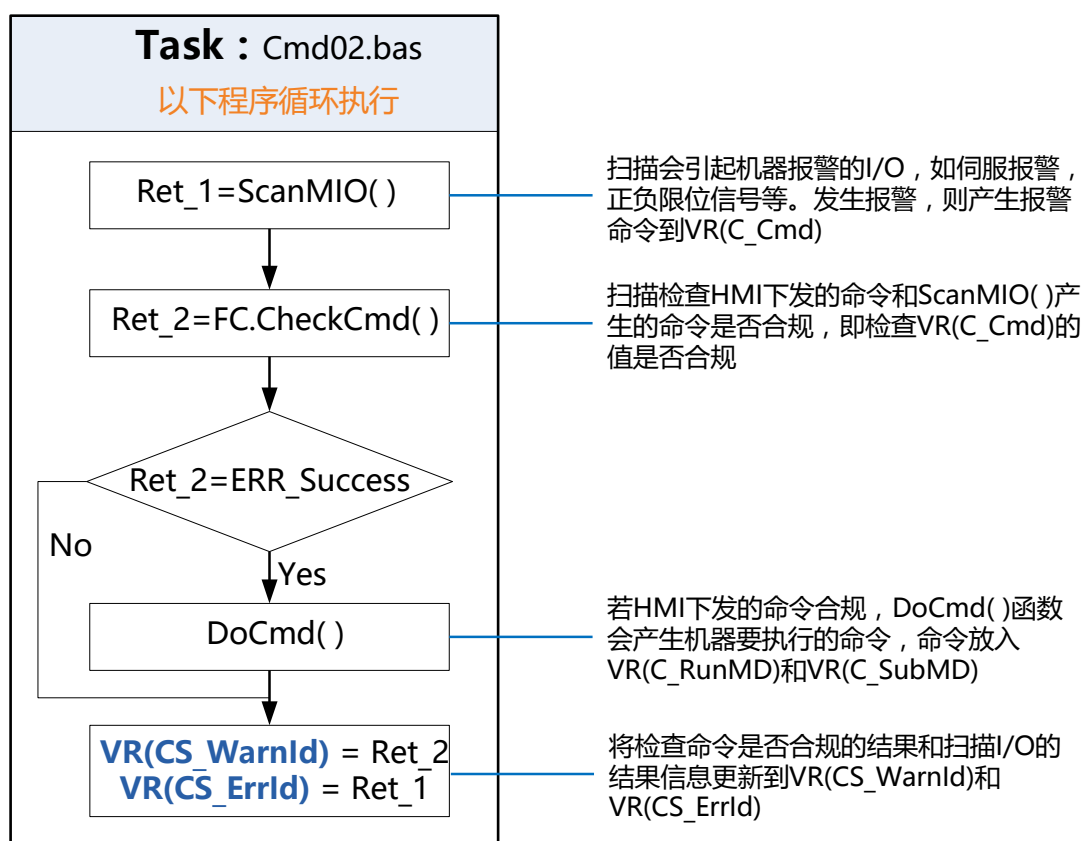
1. Cmd02.bas：負責掃描外部資訊 (HMI 下發的命令和 I/O) —>產生機器動作命令。
2. Control02.bas：負責機器動作執行—>執行命令。



Cmd02.bas 主流程解析

Cmd02.bas 這個 Task 流程很簡單，說明如下：

- 掃描有沒有伺服報警、限位元報警等錯誤發生
- 掃描 HMI 下發的命令和 ScanMIO()產生的命令 VR(C_Cmd)是否合規
 - a) 沒有命令，則不產生動作命令
 - b) 命令不合規，則不產生動作命令
 - c) 命令合規，則產生動作命令到VR(C_RunMD)和VR(C_SubMD)
- 更新以上兩個掃描結果資訊到VR(CS_WarnId)和VR(CS_ErrId)

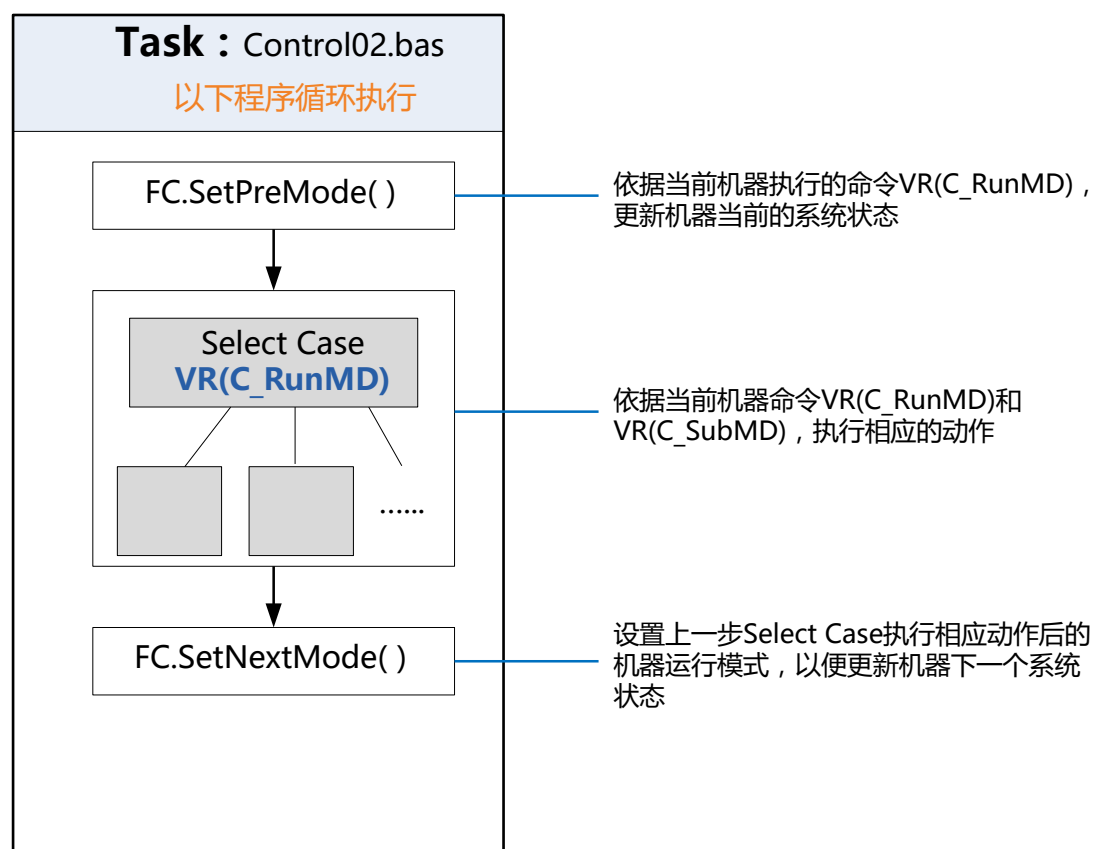


Control02.bas 主流程解析

Control02.bas 這個 Task 的流程說明如下：

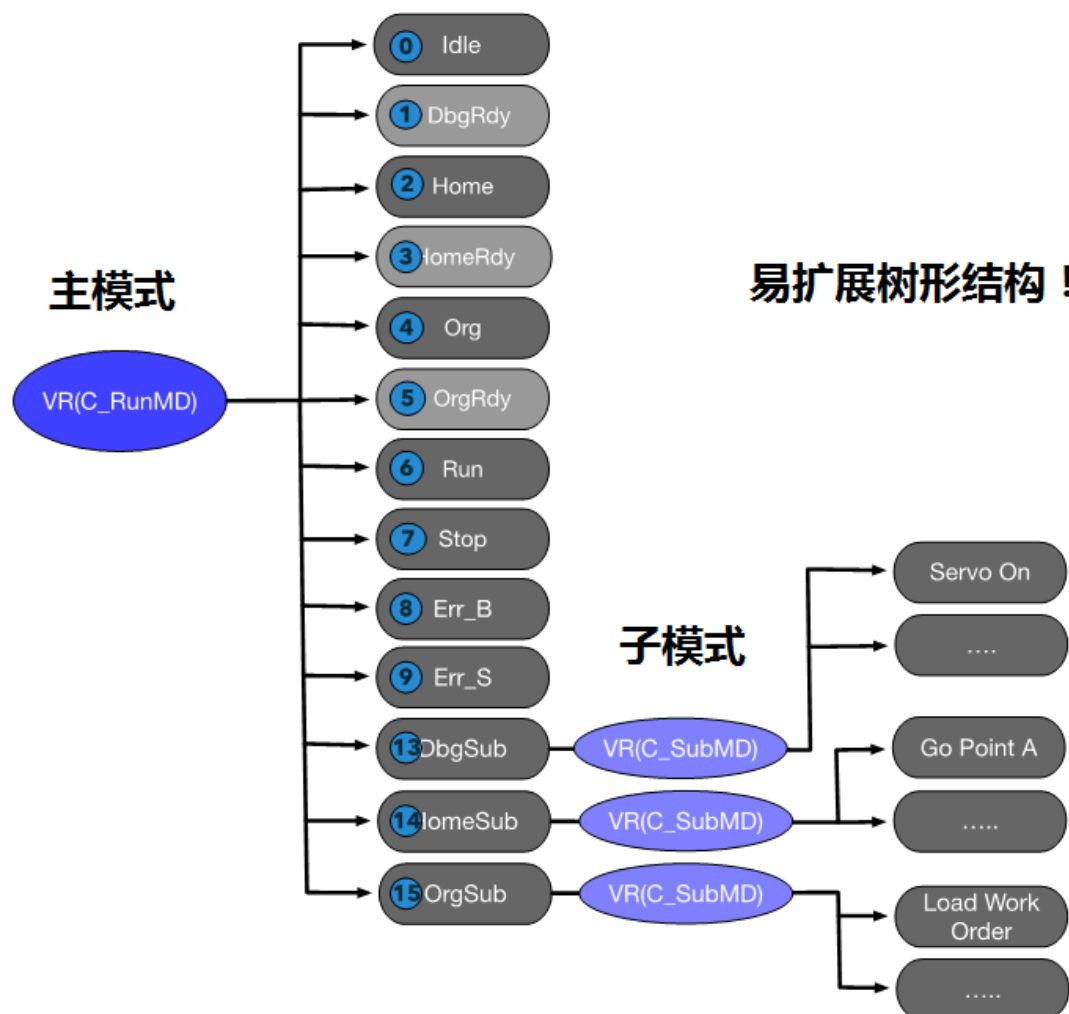
- 更新當前機器的系統狀態
- 根據 Cmd02.bas 產生的機器執行命令，執行相應的動作
- 設置機器下一個運行模式，以便更新機器下一個系統狀態

注意：使用範本框架的情況下，系統內部定義了機器狀態機的運行模式，機器需按照系統定義好的狀態機運行。具體請參照“狀態機與機器運行模式”章節。



狀態機與機器運行模式

狀態機由 Select Case 實現運行模式管理。框架內部已定義好狀態機，根據機器得到的命令 VR(C_RunMD)，系統會自動切換狀態機。



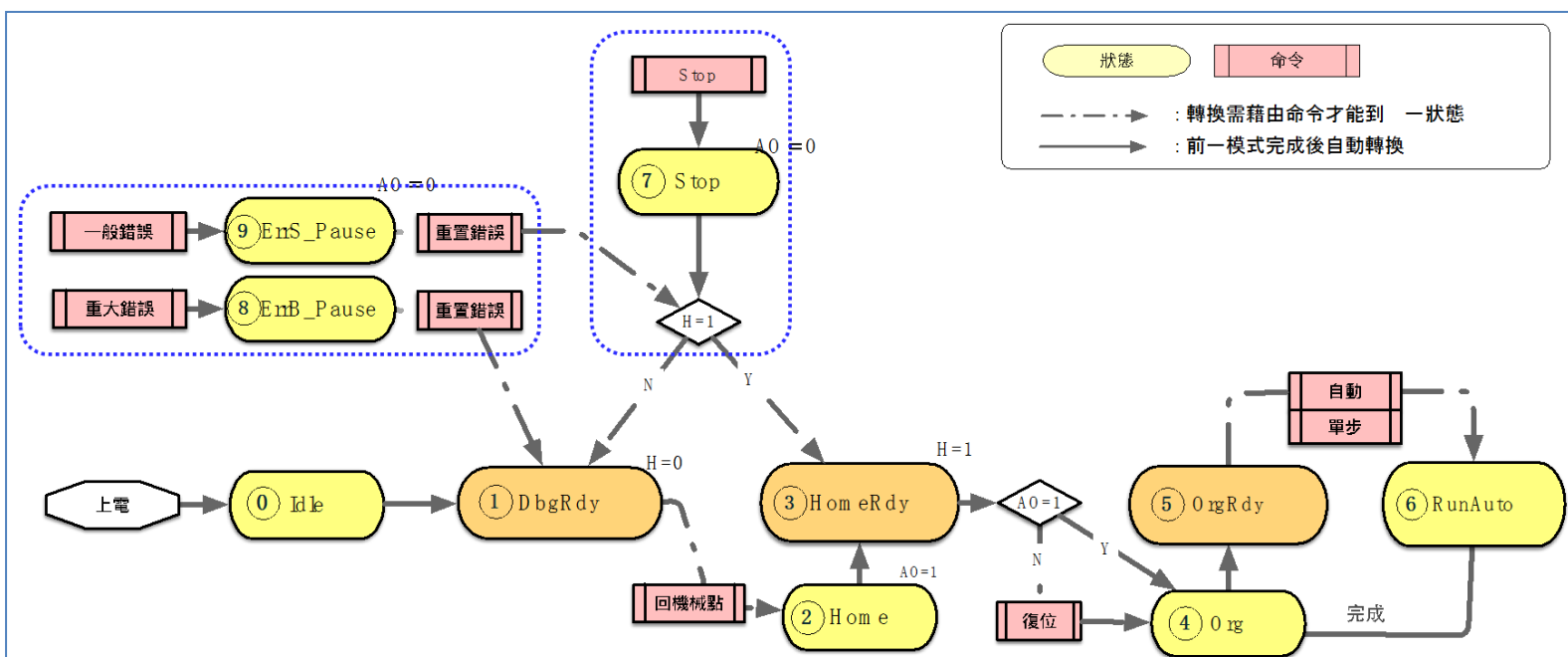
狀態機說明如下：

狀態	說明
Idle	機器剛上電的狀態
DbgRdy	機器上電後經過初始化後的狀態，稱為調試模式等候狀態
Home	回機械原點運動中的狀態
HomeRdy	回完機器原點的狀態，稱為原點模式等候狀態
Org	回工作原點運動中的狀態
OrgRdy	回完工作原點的狀態，稱為工作原點模式等候狀態
Run	自動運行中的狀態，即自動流程運行中的狀態
Stop	自動流程運行被停止後的狀態

Err_B	機器出現重大錯誤後的狀態
Err_S	機器出現一般錯誤後的狀態
DbgSub	調試模式等候狀態下的子動作在執行時的狀態
HomeSub	原點模式等候狀態下的子動作在執行時的狀態
OrgSub	工作原點模式等候狀態下的子動作在執行時的狀態

機器狀態機切換如下圖

- 三種等候狀態：①DbgRdy、③HomeRdy、⑤OrgRdy。等候狀態下，機器無動作，處於等待接收命令狀態
- 框架中的自動流程動作支援自動/單步/暫停運行
- 依據不同命令，自動切換狀態機模式
- 發生停止命令或一般錯誤的重置錯誤命令後，如果機器曾回過機械原點，會自動進入 HomeRdy 狀態
- 分兩種錯誤處理：一般錯誤 Err_S 和重大錯誤 Err_B。發生重大錯誤，重置錯誤後，系統會進入 DbgRdy，需重回機械原點，才能進行後續的回工作原點和自動流程動作。



全域介面變數

系統保留了 VR(0)~VR(15)的使用，這 16 個變數的 VR 編號內部都做了巨集定義。使用者可以針對這些介面，查詢 VR 表來調試機器，說明如下：

VR	巨集定義	說明
VR(0)	VR(C_RunMD)	系統執行的動作模式
VR(1)	VR(CS_RunSTA)	系統狀態
VR(2)	預留	
VR(3)	VR(CS_RunStepCount)	自動流程動作中當前運行的步驟號
VR(4)	VR(C_SubMD)	系統執行的動作子模式
VR(5)	VR(C_Cmd)	外部下發的命令。例如：HMI 下發的命令
VR(6)	預留	
VR(7)	預留	
VR(8)	VR(CS_WarnId)	系統警告代碼
VR(9)	VR(CS_ErrId)	系統錯誤代碼
VR(10)	VR(CF_Home)	機械原點的標誌位元
VR(11)	VR(CF_Stop)	停止命令的標誌位元
VR(12)	VR(CF_Pause)	暫停命令的標誌位元
VR(13)	VR(CF_Step)	單步命令的標誌位元
VR(14)	VR(CF_Err)	錯誤的標誌位元
VR(15)	VR(CF_AutoOrg)	機械原點後自動執行到工作原點的標誌位元

VR(C_Cmd)

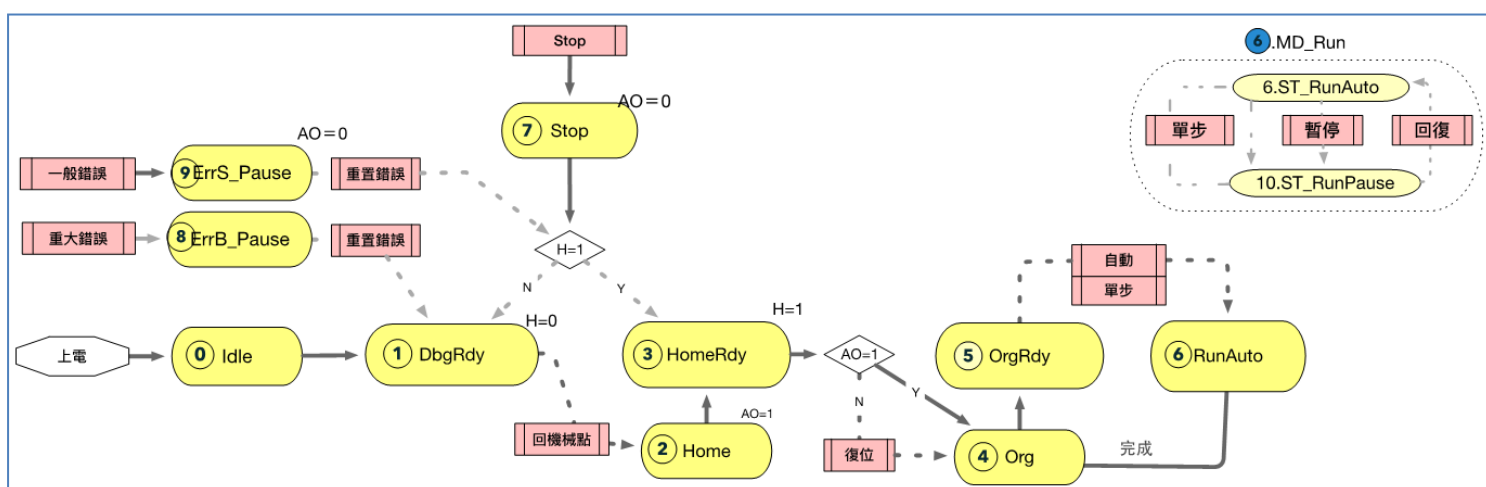
VR(C_Cmd)作為接收外部命令的介面，可以是 HMI 下發命令改變 VR(C_Cmd)的值，也可以通過程式執行賦值改變 VR(C_Cmd)的值。

VR(C_Cmd)數值說明

數值	系統內巨集定義	說明
0	N_NO_CMD	無命令
1	N_CMD_Home	回機械原點命令
2	N_CMD_Org	回工作原點命令
3	N_CMD_Run	自動流程運行命令
4	N_CMD_Stop	停止命令
5	N_CMD_Pause	暫停命令
6	N_CMD_Step	單步運動命令
7	N_CMD_Resume	恢復命令，繼續運行被暫停的自動流程
8	N_CMD_OnErrB	重大錯誤處理命令
9	N_CMD_OnErrS	一般錯誤處理命令
10	N_CMD_ResetErr	重置錯誤狀態命令
101~199	用戶自訂...	DbgRdy 模式下自訂動作命令
201~299	用戶自訂...	HomeRdy 模式下自訂動作命令
301~399	用戶自訂...	OrgRdy 模式下自訂動作命令
401~499	用戶自訂...	RunAuto 模式下自訂動作命令(暫保留)

VR(CS_RunSTA)回饋的是當前機器處於的系統狀態。系統狀態會根據機器動作的執行，系統內自動切換。

數值	系統內巨集定義	說明
0	N_ST_Idle	初始空閒狀態
1	N_ST_DbgRdy	完成初始化，在 DbgRdy 模式下等候狀態
2	N_ST_Home	正在回機械原點中...
3	N_ST_HomeRdy	完成回機械原點的等候狀態
4	N_ST_Org	正在到工作原點中...
5	N_ST_OrgRdy	完成到工作原點的等候狀態
6	N_ST_RunAuto	正在自動流程動作運行中...
7	N_ST_Stop	正在停止運行中...
8	N_ST_ErrB_Pause	發生重大錯誤，處於暫停狀態
9	N_ST_ErrS_Pause	發生一般錯誤，處於暫停狀態
10	N_ST_RunPause	自動流程動作運行處於暫停狀態
11	N_ST_Dbg_SubActions	正在執行 DbgRdy 模式下自訂動作...
12	N_ST_Home_SubActions	正在執行 HomeRdy 模式下自訂動作...
13	N_ST_Org_SubActions	正在執行 OrgRdy 模式下自訂動作...
14	N_ST_Run_SubActions	正在執行 RunAuto 模式下自訂動作...



框架的動作組成

用框架開發，系統把整個設備的動作分為“基本動作”和“用戶自訂動作”

- 基本動作：系統已定義好，具體裡面的動作細節由使用者根據機台實際應用編寫。
- 用戶自訂動作：使用者根據實際應用，可以自訂動作，放入三個等候狀態模式下，即框架保留了用戶自訂動作的擴展空間。



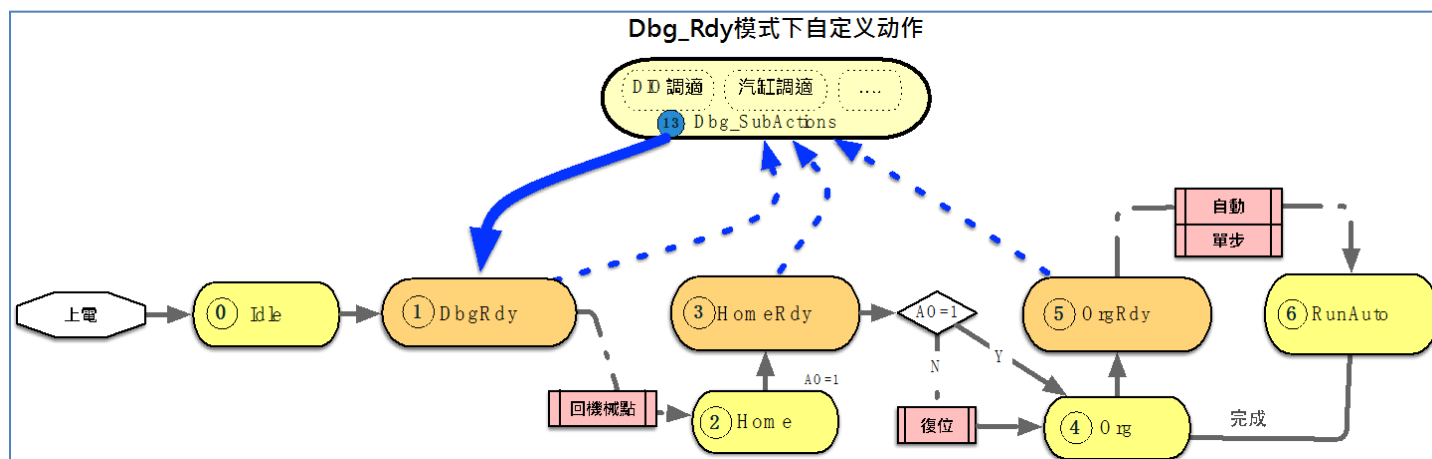
基本動作

	說明
系統定義中的基本動作 命令 VR(C_Cmd) 的 編號 範圍 為:0~10	<ol style="list-style-type: none"> 1. 系統定義了常見設備中需用到的基本動作 2. 基本動作執行會引起系統狀態機切換，從而實現動作有效管理 3. 例如：回機械原點，回工作原點，自動流程動作

用戶自訂動作

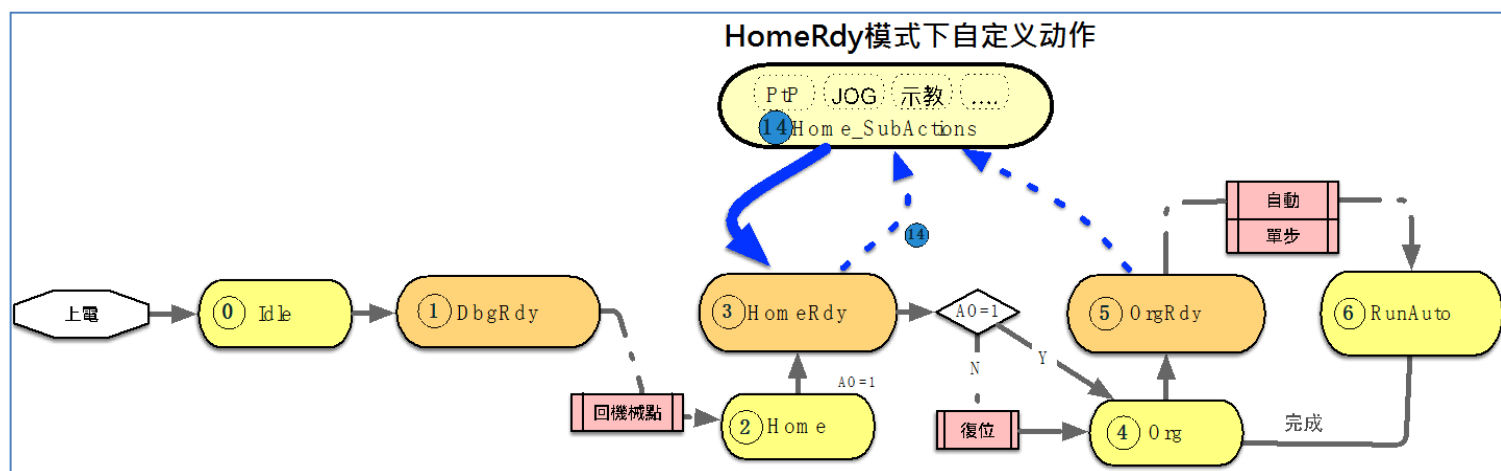
- DbgRdy 模式下的自訂動作：Dbg_SubActions

	說明
DbgRdy 模式下自訂動作 命令 VR(C_Cmd) 的編號範圍為:101~199	4. DbgRdy 模式可以執行的是調試動作,不需要回機械原點就可以執行的動作放在此模式下。 5. 當系統處於①DbgRdy、③HomeRdy、⑤OrgRdy 的模式下，才可以執行 6. 該模式下的自訂動作執行完，系統會自動進入 DbgRdy 狀態模式 7. 例如：DIO 手動調試，軸調試



● HomeRdy 模式下的自訂動作：Home_SubActions

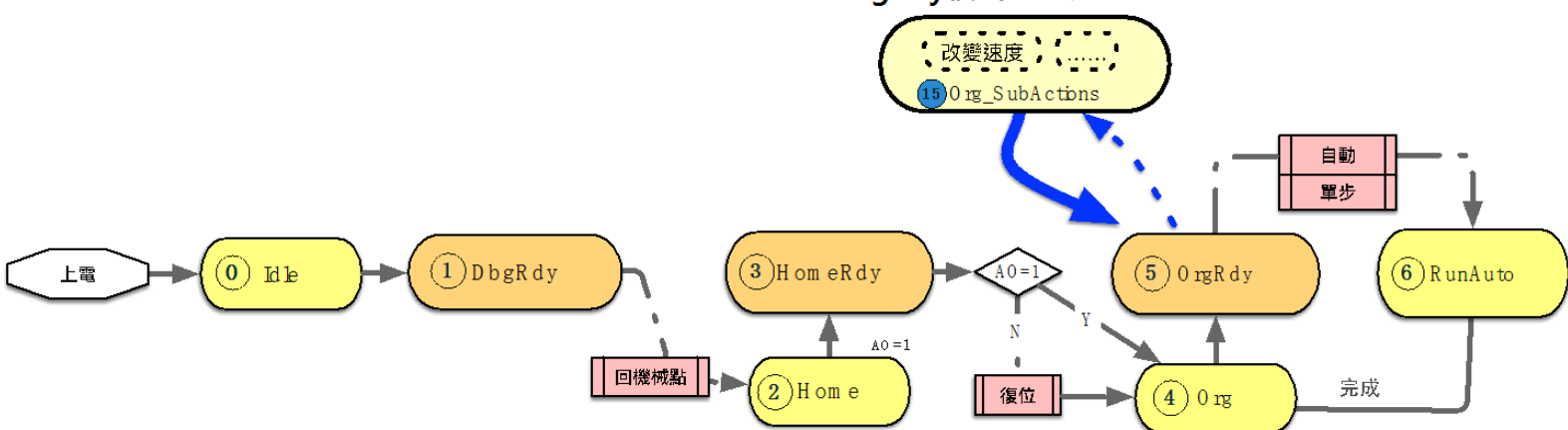
	說明
HomeRdy 模式下自訂動作 命令 VR(C_Cmd) 的編號範圍為:201~299	<ol style="list-style-type: none"> 1. 需要回機械原點才可執行的動作放在此模式下。 2. 當系統處於③HomeRdy、⑤OrgRdy 的模式下，才可以執行。 3. 該模式下的自訂動作執行完，系統會自動進入 HomeRdy 狀態模式 4. 例如：示教、手動定位



● OrgRdy 模式下的自訂動作：Org_SubActions

	說明
OrgRdy 模式下自訂動作 命令 VR(C_Cmd) 的編號範圍為:301~399	1. 需要回工作原點才可執行的動作放在此模式下。 2. 當系統處於⑤OrgRdy 的模式下，才可以執行。 3. 該模式下的自訂動作執行完，系統會自動進 OrgRdy 狀態模式 4. 例如：改變加工速度

OrgRdy模式下自定义动作

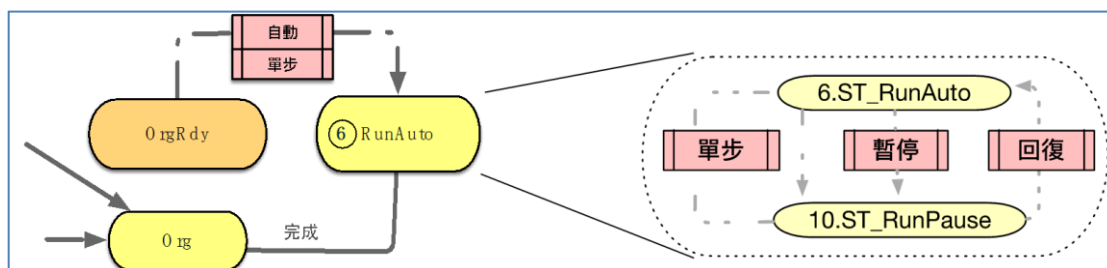


自動/單步/暫停

自動流程動作 (Run) 可以實現自動循序執行、單步執行以及暫停操作。

- 自動流程順序動作需寫在 Sub ModeRun 的函數體裡
- 一般情況下，使用者定義的設備元件 (如 XYZ Table) 下的 WaitDone() 方法作為一個單步節點
- 如果需要動作步驟間實現暫停操作，需在步驟節點間插入 FC.CheckPause

```
'Purpose：運行模式
SUB ModeRun
'用戶自定義動作代碼，請編寫於此運行模式下...
'示範動作：
XYZ.MOVE_XYZ 5000, 6000, 7000      'path 0
XYZ.WaitDone()
FC.ChecknPause() ← 暫停節點
XYZ.MOVE_XYZ 15000, 16000, 17000  'path 1
XYZ.WaitDone()
FC.ChecknPause() ← 暫停節點
END SUB
```



如何使用框架

框架已具備的功能

- 14 種設備系統狀態自動轉換
- 發生重大錯誤與一般錯誤時暫停動作
- 特定狀態，拒絕不合法的操作(Ex: 機器還沒有回零，不能操作 RUN)
- 9 種基本動作處理：
 - a) 回原點，到工作點，
 - b) 自動運行，暫停，單步運行，停止
 - c) 重大錯誤，一般錯誤，重置錯誤



兩個步驟開發設備

步驟 1：添加基本動作處理

添加步驟	說明
1. 初始化	在 Init()，編寫您的初始化動作...
2. 回機械原點	在 ModeHome ()，編寫您的回機械原點動作...
3. 到工作原點	在 ModeOrg ()，編寫您的回工作原點包含的動作...
4. 自動運行	在 ModeRun ()，編寫您的自動運行動作...
5. 停止	1. 在 UserStop()，編寫如何停止您的設備動作 2. 在 ModeStop()，編寫停止後，您希望執行的動作...
6. 錯誤處理	重大錯誤: 1. 在 UserErrB(),編寫發生錯誤時您要採取的動作(Ex: 停止) 2. 在 ModeErrB(), 編寫 Motion 錯誤停止後，您希望執行的動作... 3. 當發生錯誤，系統會暫停 Control02.bas 的執行，並等待清除錯誤命令 4. 請在 UserResetErr(),編寫您的清除錯誤動作
	一般錯誤: 1. 在 UserErrS(),編寫發生錯誤時您要採取的動作(Ex: 停止) 2. 在 ModeErrS(), 編寫 Motion 錯誤停止後，您希望執行的動作... 3. 當發生錯誤，系統會暫停 Control02.bas 的執行，並等待清除錯誤命令 4. 請在 UserResetErr(),編寫您的清除錯誤動作

步驟 2：添加基本動作處理

- 步驟 2-1：在 DbgRdy 模式下添加可執行的自訂動作

添加步驟

1. 在 DoCmd()，定義您的命令編號 1xx ... 命令編號範圍：101-199
2. 在 DoCmd() 設定子動作模式編號 VR(C_SubMD)= n
3. 在 ModeDbg_SubActions()，添加當子動作模式 VR(C_SubMD)= n 時，要執行的動作

範例：如需在 DbgRdy 模式下添加“對軸 0、1、2 三個軸的伺服使能”動作。

```
SUB DoCmd()
SELECT CASE INT(VR(C_Cmd))
CASE 101                                '指令編號(Debug Ready 模式下)
    VR(C_SubMD)=1                      '設定子動作模式編號 VR(C_SubMD)= n
    Ret =Err_Success
END SELECT
```

```
SUB ModeDbg_SubActions()
SELECT CASE INT(VR(C_SubMD))
CASE 1                                  '當子動作模式 VR(C_SubMD) = n 時
    BASE 0,1,2                          '示範動作...對軸 0,1,2 使能伺服
    SVON
END SELECT
```

- 步驟 2-2：在 HomeRdy 模式下添加可執行的自訂動作

添加步驟

1. 在 DoCmd()，定義您的命令編號 2xx ... 命令編號範圍：201-299
2. 在 DoCmd() 設定子動作模式編號 VR(C_SubMD)= n
3. 在 ModeHome_SubActions()，添加當子動作模式 VR(C_SubMD)= n 時，要執行的動作

範例：如需在 HomeRdy 模式下添加“做軸 0、1、2 三軸的直線插補”動作。

```
SUB DoCmd()
SELECT CASE INT(VR(C_Cmd))
CASE 201                '指令編號(Home Ready 模式下)
    VR(C_SubMD)=2        '設定子動作模式編號 VR(C_SubMD)= n
    Ret = Err_Success
END SELECT
```

```
SUB ModeHome_SubActions()
SELECT CASE INT(VR(C_SubMD))
CASE 2                  '當子動作模式 VR(C_SubMD)= n 時
    BASE 0,1,2          '示範動作...到指定點 A
    LINEABS 1000, 1000, 1000
    WAIT DONE
END SELECT
```

- 步驟 2-3：在 OrgRdy 模式下添加可執行的自訂動作

添加步驟

1. 在 DoCmd()，定義您的命令編號 3xx ... 命令編號範圍：301-399
2. 在 DoCmd() 設定子動作模式編號 VR(C_SubMD)= n
3. 在 ModeOrg_SubActions()，添加當子動作模式 VR(C_SubMD)= n 時，要執行的動作

範例：如需在 OrgRdy 模式下添加“載入工單”動作。

```
SUB DoCmd()
SELECT CASE INT(VR(C_Cmd))
CASE 301                                '指令編號(Org Ready 模式下)
    VR(C_SubMD)=3                      '設定子動作模式編號 VR(C_SubMD)= n
    Ret = Err_Success
END SELECT
```

```
SUB ModeOrg_SubActions()
SELECT CASE INT(VR(C_SubMD))
CASE 3                                '當子動作模式 VR(C_SubMD)= n 時
    LoadWorkOrder()                  '示範動作...載入工單, 準備加工。
END SELECT
```

系統函數說明

ScanMIO()

格 式： Function ScanMIO() AS INTEGER

描 述： 掃描 Motion I/O 的狀態。預設框架下，掃描軸上 MIO.ALM、MIO.PEL、MIO.NEL、MIO.EMG 四個 Motion I/O 的狀態。需檢測其他 Motion I/O 的狀態，需在 ScanMIO()函數體中自行添加

返回值：

- ERR_AxisAlmError：發生伺服報警時，返回 ERR_AxisAlmError
- ERR_AxisPelError：發生正向硬體限位元報警時，返回 ERR_AxisPelError
- ERR_AxisNelError：發生負向限位報警時，返回 ERR_AxisNelError
- ERR_AxisEmgError：發生負向限位報警時，返回 ERR_AxisEmgError
- ERR_Success：沒有發生軸錯誤時返回 ERR_Success

FC.CheckCmd()

格 式： FUNCTION FlowCtrl.CheckCmd() AS INTEGER

描 述： 檢測 VR(C_Cmd)命令是否合規。系統內部已做巨集定義，VR(C_Cmd)即 VR(5)。

VR(C_Cmd)中的命令值說明如下表：

命令	對應值	說明
N_NO_CMD	0	無命令
N_CMD_Home	1	回機械原點命令
N_CMD_Org	2	到工作原點命令
N_CMD_Run	3	自動流程運行命令
N_CMD_Stop	4	停止運動命令
N_CMD_Pause	5	暫停自動流程命令
N_CMD_Step	6	自動流程單步執行命令
N_CMD_Resume	7	暫停後恢復自動流程運行命令
N_CMD_OnErrB	8	重大錯誤後處理命令
N_CMD_OnErrS	9	一般錯誤後處理命令

N_CMD_ResetErr	10	清除軸錯誤狀態命令
N_CMD_DbgSub	101~199	DebugReady 模式下的副程式執行命令
N_CMD_HomeSub	201~299	HomeReady 模式下的副程式執行命令
N_CMD_OrgSub	301~399	OrgReady 模式下的副程式執行命令
N_CMD_RunSub	401~499	Run 模式下的副程式執行命令
其他命令	以上值之外	不合規命令

返回值：

- ERR_NoCommand：檢測到 VR(C_Cmd)的值為 0
- ERR_InvalidOperation：檢測到 VR(C_Cmd)的值不符合機器當前狀態允許的操作，比如當前機器正在執行自動流程動作，這時候 HMI 下發一個回原點命令，機器不會執行回原點命令。FC.CheckCmd() 函數的返回值則為 ERR_InvalidOperation
- ERR_UnknowCommand：檢測到 VR(C_Cmd)的值非 (0~10) 或 (101~499)
- ERR_Success：檢測到 VR(C_Cmd)的值非以上三種情況時，返回 ERR_Success

DoCmd()

格 式： FUNCTION DoCmd() AS INTEGER

描 述： 根據檢測到的 VR(C_Cmd)值，系統內部會改變 VR(C_RunMD)和 VR(C_SubMD)的值。系統內部已做巨集定義，VR(C_RunMD)即 VR(0)，VR(C_SubMD)即 VR(4)。

根據 VR(C_Cmd)的值，產生的 VR(C_RunMD)和 VR(C_SubMD)的值如下表

VR(C_Cmd)值	產生的 VR(C_RunMD)值	產生的 VR(C_SubMD)值
N_NO_CMD	無變化	無變化
N_CMD_Home	N_MD_Home : 2	無變化
N_CMD_Org	N_MD_Org : 4	無變化
N_CMD_Run	N_MD_Run : 6	無變化
N_CMD_Stop	N_MD_Stop : 7	無變化
N_CMD_Pause	無變化，但程式會執行到 FC.ChecknPause()這行停下	無變化
N_CMD_Step	無變化，但程式會執行到下一個	無變化

	FC.ChecknPause()行停下	
N_CMD_Resume	無變化，但程式會恢復執行，繼續未執行完的自動流程動作	無變化
N_CMD_OnErrB	N_MD_ErrB : 8	無變化
N_CMD_OnErrS	N_MD_ErrS : 9	無變化
N_CMD_ResetErr	<ul style="list-style-type: none"> ErrB : N_MD_DbgRdy ErrS : 曾回過機械原點，則 N_MD_HomeRdy ErrS : 未回過機械原點，則 N_MD_DbgRdy 	無變化
N_CMD_DbgSub	N_MD_Dbg_SubActions : 13	依照使用者在該狀態模式下的子動作定義的編號
N_CMD_HomeSub	N_MD_Home_SubActions : 14	依照使用者在該狀態模式下的子動作定義的編號
N_CMD_OrgSub	N_MD_Org_SubActions : 15	依照使用者在該狀態模式下的子動作定義的編號
N_CMD_RunSub	N_MD_Run_SubActions : 16	依照使用者在該狀態模式下的子動作定義的編號
其他命令	無變化	無變化

返回值：

ERR_UnknownCommand：檢測到 VR(C_Cmd)的值非 (0~10) 或 (101~499)

ERR_Success：檢測到 VR(C_Cmd)的值合規

FC.SetPreMode()

格 式： FlowCtrl.SetPreMode()

描 述： 根據機器得到的動作命令 VR(C_RunMD)，更新設置機器當前的系統狀態 VR(CS_RunSTA)。

VR(C_RunMD)與 VR(CS_RunSTA)的對應關係如下：

命令 VR (C_RunMD)	設置的系統狀態 VR (CS_RunSTA)
N_MD_Idle : 0	N_ST_Idle : 0

N_MD_DbgRdy : 1	N_ST_DbgRdy : 1
N_MD_Home : 2	N_ST_Home : 2
N_MD_HomeRdy : 3	N_ST_HomeRdy : 3
N_MD_Org : 4	N_ST_Org : 4
N_MD_OrgRdy : 5	N_ST_OrgRdy : 5
N_MD_Run : 6	N_ST_RunAuto : 6
N_MD_Stop : 7	N_MD_Stop : 7
N_MD_ErrB : 8	N_ST_ErrB_Pause : 8
N_MD_ErrS : 9	N_ST_ErrS_Pause : 9
N_MD_Dbg_SubActions : 13	N_ST_Dbg_SubActions : 13
N_MD_Home_SubActions : 14	N_ST_Home_SubActions : 14
N_MD_Org_SubActions : 15	N_ST_Org_SubActions : 15
N_MD_Run_SubActions : 16	N_ST_Run_SubActions : 16

FC. SetNextMode()

格 式： FlowCtrl.SetNextMode()

描 述： 目前模式動作完成後，依據目前的模式，決定了下一個模式。即當前動作命令 VR(C_RunMD)的模式下動作完成後，用該函數設置更新下一個 VR(C_RunMD)，這個運行模式的管理是依據框架系統內部定義的狀態機實現的，請參考“狀態機與機器運行模式”章節。

當前 VR(C_RunMD)模式與下一個 VR(C_RunMD)模式的對應關係如下：

當前 VR (C_RunMD) 模式	下一個 VR (C_RunMD) 模式
N_MD_Idle : 0	N_MD_DbgRdy : 1
N_MD_DbgRdy : 1	N_MD_DbgRdy : 1
N_MD_Home : 2	N_MD_HomeRdy : 3
N_MD_HomeRdy : 3	VR(CF_AutoOrg)為 1，則 N_MD_Org : 4 VR(CF_AutoOrg)為 0，則 N_MD_HomeRdy : 3
N_MD_Org : 4	N_MD_OrgRdy : 5
N_MD_OrgRdy : 5	N_ST_OrgRdy : 5

N_MD_Run : 6	N_MD_Org : 4
N_MD_Stop_Insert : 10	N_MD_Stop : 7
N_MD_Stop : 7	回過機械原點 : N_MD_HomeRdy : 3 未回過機械原點 : N_MD_DbgRdy : 1
N_MD_ErrB_Insert : 11	N_MD_ErrB : 8
N_MD_ErrB : 8	N_MD_DbgRdy : 1
N_MD_ErrS_Insert : 12	N_MD_ErrS : 9
N_MD_ErrS : 9	回過機械原點 : N_MD_HomeRdy : 3 未回過機械原點 : N_MD_DbgRdy : 1
N_MD_Dbg_SubActions : 13	N_MD_DbgRdy : 1
N_MD_Home_SubActions : 14	N_MD_HomeRdy : 3
N_MD_Org_SubActions : 15	N_MD_OrgRdy : 5
N_MD_Run_SubActions : 16	N_MD_Run : 6

FlowCtrl

FlowCtrl 是系統內部定義的一個類，這個類裡面包含了一些流程控制用到的 SUB 和 Funtion。用戶結合本文檔的說明，無需瞭解該類裡的方法細節即可完成開發。該類裡的一些需要使用者瞭解內部細節的方法如 FC.CheckCmd()、FC.SetPreMoe()、FC.SetNextMoe()等已在本章節說明。

XYZ_TABLE

XYZ_TABLE 是框架下系統定義的一個示例類，這個類代表 XYZ 直角坐標平臺這樣的一個元件，該類包含了 XYZ 直角坐標平臺下的一些方法：如三軸移動，回原點，WatiDone,停止移動，清除錯誤等。

該類定義在 ProjectDefine.bi 裡面，如果使用者的設備也是 XYZ 直角坐標平臺，使用者可以根據實際應用，在 TYPE XYZ_TABLE 裡擴展方法、屬性。如果使用者的設備不是 XYZ 直角坐標平臺，用戶可以仿照 TYPE XYZ_TABLE 定義自己的類。