模板框架使用

Version 1.0

概述

Motion Studio新建工程时，可以选择新建“空的框架”和新建“模板框架”两种方式。“空的程序框架”里没有任何代码，由用户自行编写整个程序，而用“模板框架”新建工程后，Motion Studio里就已经创建了两个Task，并在系统内部封装了一些函数、变量，在ProjectDefine.bi和两个Task创建了部分程序以及创建了一部分VR变量区域。这些共同组成了“模板框架”。

“模板框架”定义了机器接受命令的接口和机器运行的状态切换，用户在程序框架下添加代码，就可以很快完成一套设备运行的程序，让设备稳定的运行。该“模板框架”特别适用顺序流水作业的运动控制相关设备。

目的

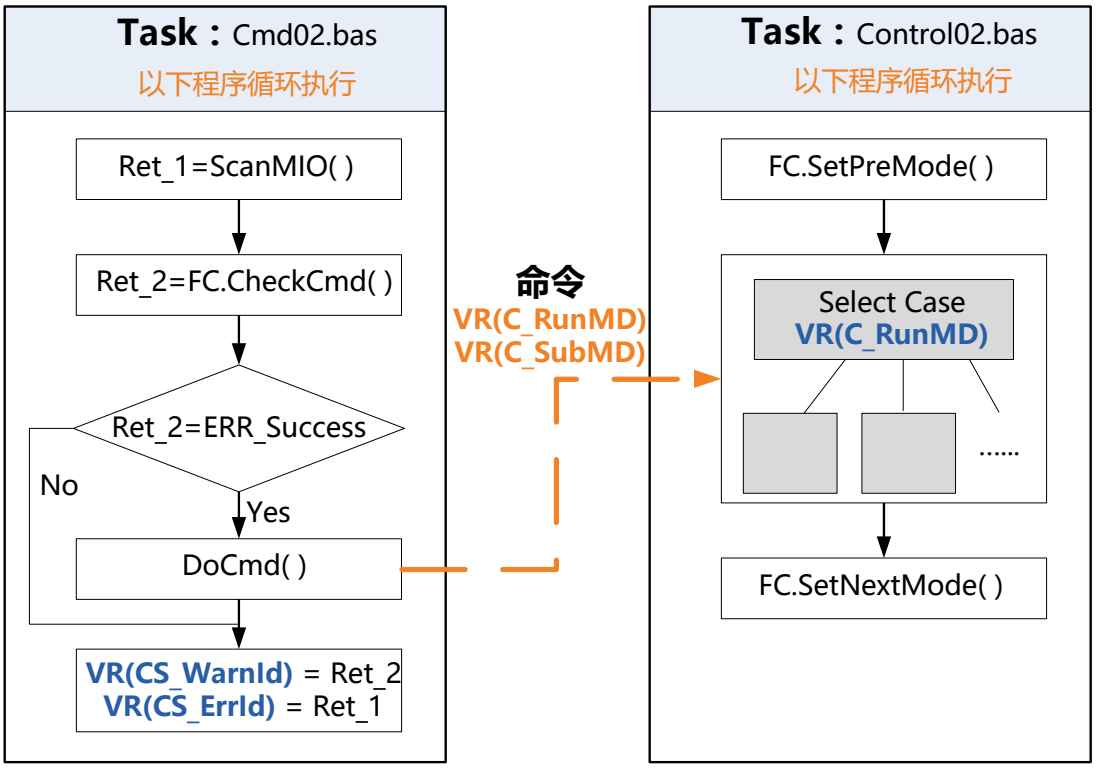
“模板框架”可以让用户快速搭建稳定可靠的程序，避免了开发者随意开发导致的程序结构不佳，运行效率不高以及不易调试、维护等问题。同时，很好的使用该框架，可以用同一个框架应对各种非标设备，大幅缩短开发周期。

框架流程

Task 组成

框架由 2 个 Task 构成，两个 Task 的主体结构都是 While 循环。

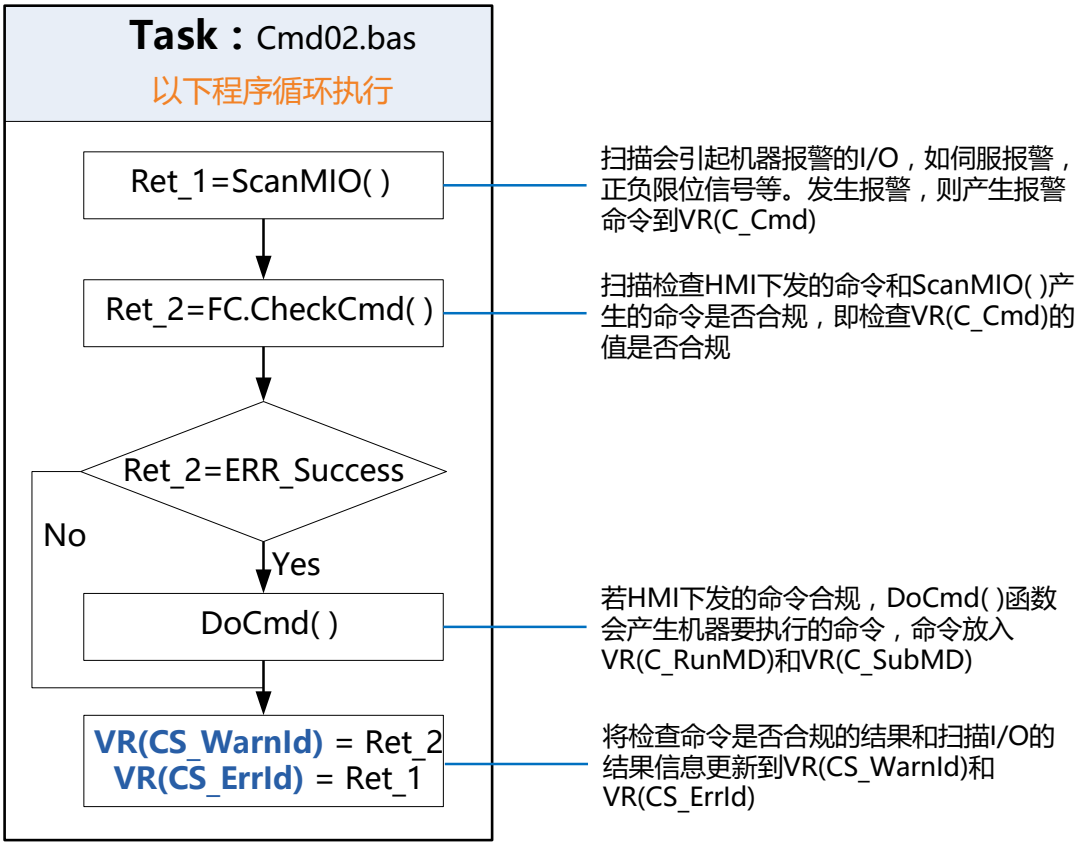
1. Cmd02.bas：负责扫描外部信息（HMI 下发的命令和 I/O）—>**产生机器动作命令。**
2. Control02.bas： 负责机器动作执行—>**执行命令。**



Cmd02.bas 主流程解析

Cmd02.bas 这个 Task 流程很简单，说明如下：

- 扫描有没有伺服报警、限位报警等错误发生
- 扫描 HMI 下发的命令和 ScanMIO()产生的命令 VR(C_Cmd)是否合规
  a) 没有命令，则不产生动作命令
  b) 命令不合规，则不产生动作命令
  c) 命令合规，则产生动作命令到VR(C_RunMD)和VR(C_SubMD)
- 更新以上两个扫描结果信息到VR(CS_WarnId)和VR(CS_ErrId)

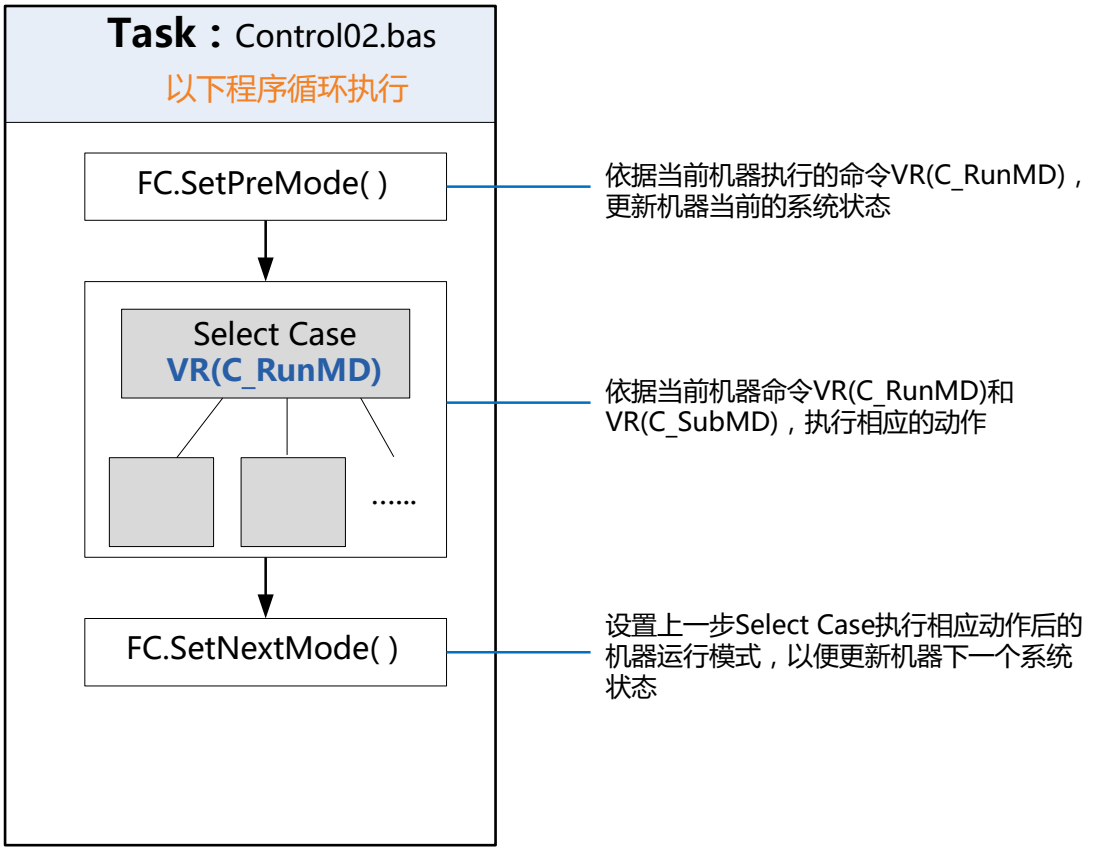


Control02.bas 主流程解析

Control02.bas 这个 Task 的流程说明如下：

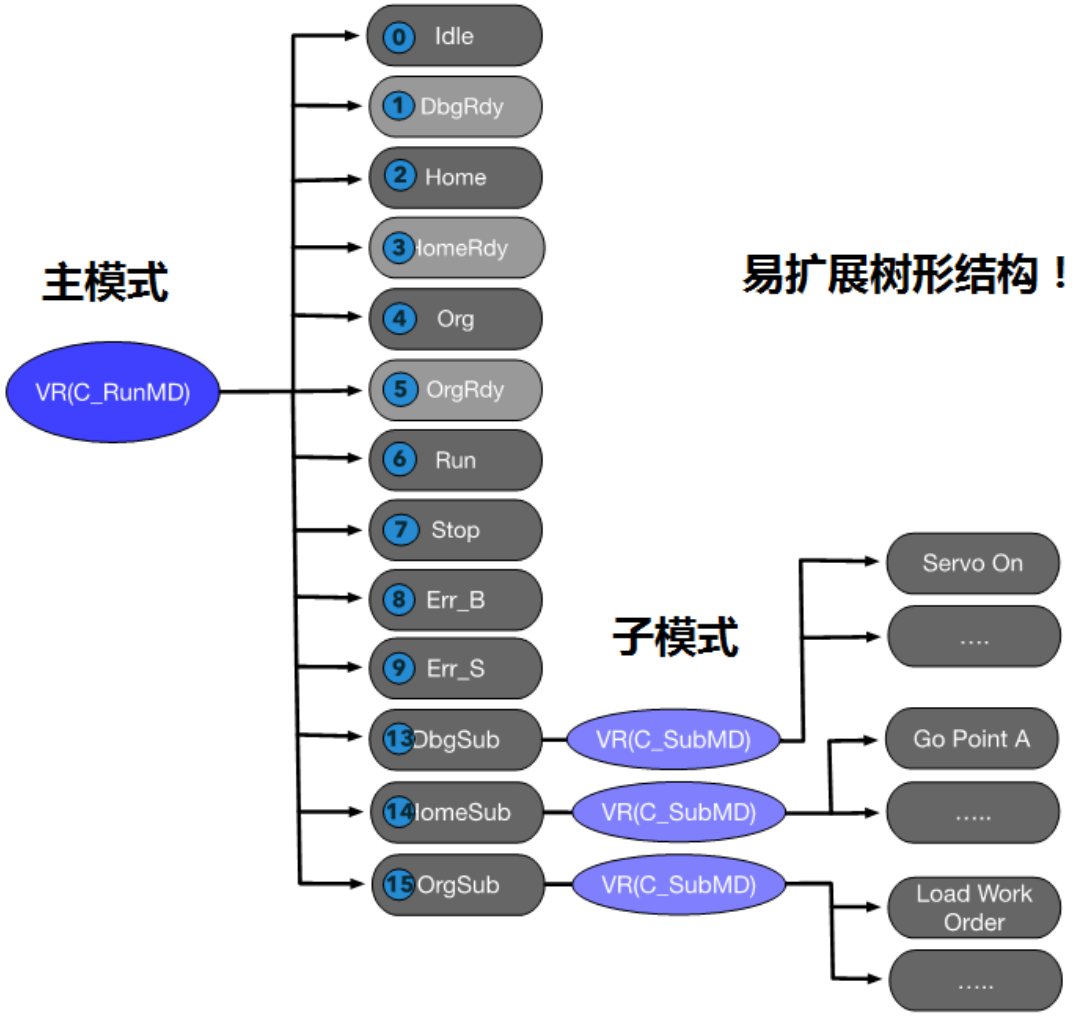
- 更新当前机器的系统状态
- 根据 Cmd02.bas 产生的机器执行命令，执行相应的动作
- 设置机器下一个运行模式，以便更新机器下一个系统状态

注意：使用模板框架的情况下，系统内部定义了机器状态机的运行模式，机器需按照系统定义好的状态机运行。具体请参照“状态机与机器运行模式”章节。



状态机与机器运行模式

状态机由 Select Case 实现运行模式管理。框架内部已定义好状态机，根据机器得到的命令 VR(C_RunMD)，系统会自动切换状态机。



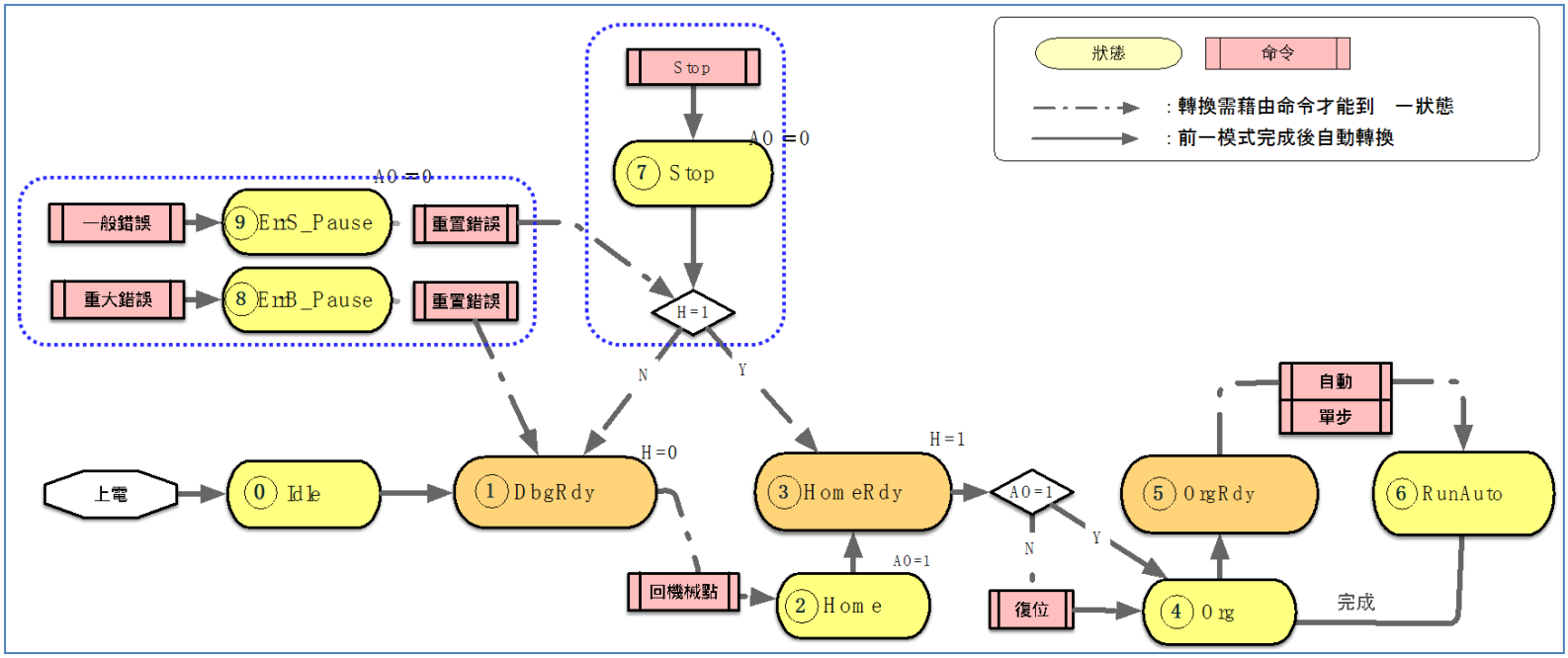
状态机说明如下：

状态	说明
Idle	机器刚上电的状态
DbgRdy	机器上电后经过初始化后的状态，称为调试模式等待状态
Home	回机械原点运动中的状态
HomeRdy	回完机器原点的状态，称为原点模式等待状态
Org	回工作原点运动中的状态
OrgRdy	回完工作原点的状态，称为工作原点模式等待状态
Run	自动运行中的状态，即自动流程运行中的状态
Stop	自动流程运行被停止后的状态

Err_B	机器出现重大错误后的状态
Err_S	机器出现一般错误后的状态
DbgSub	调试模式等待状态下的子动作在执行时的状态
HomeSub	原点模式等待状态下的子动作在执行时的状态
OrgSub	工作原点模式等待状态下的子动作在执行时的状态

机器状态机切换如下图

- 三种等待状态：①DbgRdy 、③HomeRdy、⑤OrgRdy。等待状态下，机器无动作，处于等待接收命令状态
- 框架中的自动流程动作支持自动/单步/暂停运行
- 依据不同命令，自动切换状态机模式
- 发生停止命令或一般错误的重置错误命令后，如果机器曾回过机械原点，会自动进入 HomeRdy 状态
- 分两种错误处理：一般错误 Err_S 和重大错误 Err_B。发生重大错误，重置错误后，系统会进入 DbgRdy，需重回机械原点，才能进行后续的回工作原点和自动流程动作。



全局接口变量

系统保留了 VR(0)~VR(15)的使用，这 16 个变量的 VR 编号内部都做了宏定义。用户可以针对这些接口，查询 VR 表来调试机器，说明如下：

VR	宏定义	说明
VR(0)	VR(C_RunMD)	系统执行的动作模式
VR(1)	VR(CS_RunSTA)	系统状态
VR(2)	预留	
VR(3)	VR(CS_RunStepCount)	自动流程动作中当前运行的步骤号
VR(4)	VR(C_SubMD)	系统执行的动作子模式
VR(5)	VR(C_Cmd)	外部下发的命令。例如：HMI 下发的命令
VR(6)	预留	
VR(7)	预留	
VR(8)	VR(CS_WarnId)	系统警告代码
VR(9)	VR(CS_ErrId)	系统错误代码
VR(10)	VR(CF_Home)	机械原点的标志位
VR(11)	VR(CF_Stop)	停止命令的标志位
VR(12)	VR(CF_Pause)	暂停命令的标志位
VR(13)	VR(CF_Step)	单步命令的标志位
VR(14)	VR(CF_Err)	错误的标志位
VR(15)	VR(CF_AutoOrg)	机械原点后自动执行到工作原点的标志位

VR(C_Cmd)

VR(C_Cmd)作为接收外部命令的接口，可以是 HMI 下发命令改变 VR(C_Cmd)的值，也可以通过程序执行赋值改变 VR(C_Cmd)的值。

VR(C_Cmd)数值说明

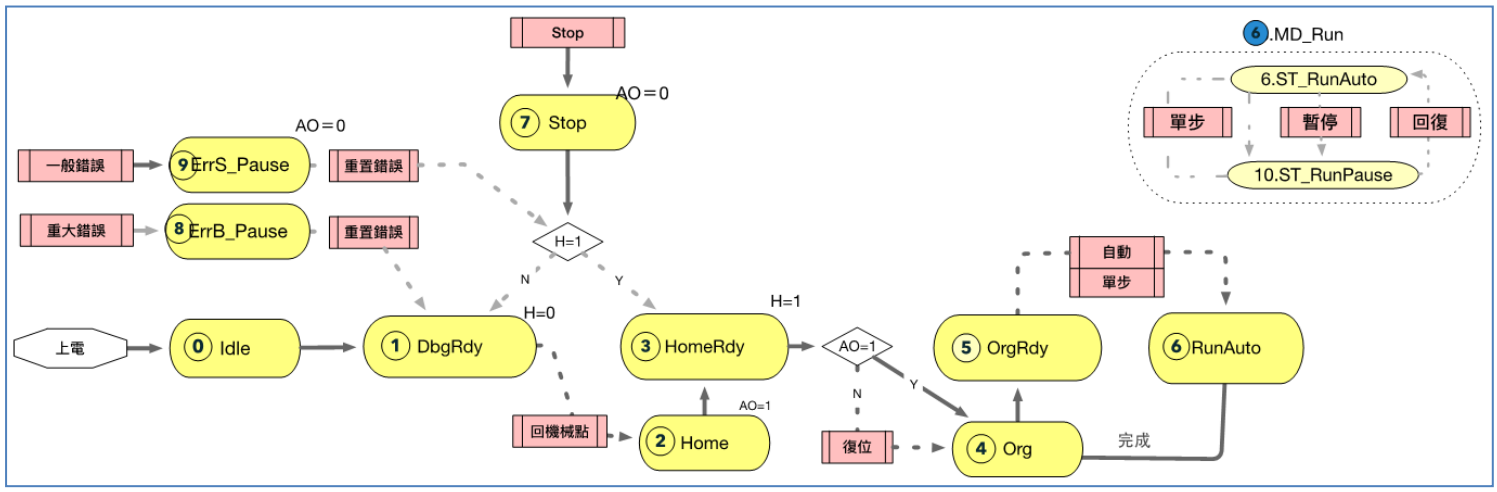
数值	系统内宏定义	说明
0	N_NO_CMD	无命令
1	N_CMD_Home	回机械原点命令
2	N_CMD_Org	回工作原点命令
3	N_CMD_Run	自动流程运行命令
4	N_CMD_Stop	停止命令
5	N_CMD_Pause	暂停命令
6	N_CMD_Step	单步运动命令
7	N_CMD_Resume	恢复命令，继续运行被暂停的自动流程
8	N_CMD_OnErrB	重大错误处理命令
9	N_CMD_OnErrS	一般错误处理命令
10	N_CMD_ResetErr	重置错误状态命令
101~199	用户自定义...	DbgRdy 模式下自定义动作命令
201~299	用户自定义...	HomeRdy 模式下自定义动作命令
301~399	用户自定义...	OrgRdy 模式下自定义动作命令
401~499	用户自定义...	RunAuto 模式下自定义动作命令(暂保留)

VR(CS_RunSTA)

VR(CS_RunSTA)反馈的是当前机器处于的系统状态。系统状态会根据机器动作的执行，系统内自动切换。

VR(CS_RunSTA)数值说明

数值	系统内宏定义	说明
0	N_ST_Idle	初始空闲状态
1	N_ST_DbgRdy	完成初始化，在 DbgRdy 模式下等待状态
2	N_ST_Home	正在回机械原点中...
3	N_ST_HomeRdy	完成回机械原点的等待状态
4	N_ST_Org	正在到工作原点中...
5	N_ST_OrgRdy	完成到工作原点的等待状态
6	N_ST_RunAuto	正在自动流程动作运行中...
7	N_ST_Stop	正在停止运行中...
8	N_ST_ErrB_Pause	发生重大错误，处于暂停状态
9	N_ST_ErrS_Pause	发生一般错误，处于暂停状态
10	N_ST_RunPause	自动流程动作运行处于暂停状态
11	N_ST_Dbg_SubActions	正在执行 DbgRdy 模式下自定义动作...
12	N_ST_Home_SubActions	正在执行 HomeRdy 模式下自定义动作...
13	N_ST_Org_SubActions	正在执行 OrgRdy 模式下自定义动作...
14	N_ST_Run_SubActions	正在执行 RunAuto 模式下自定义动作...



框架的动作组成

用框架开发，系统把整个设备的动作分为“基本动作”和“用户自定义动作”

- 基本动作：系统已定义好，具体里面的动作细节由用户根据机台实际应用编写。
- 用户自定义动作：用户根据实际应用，可以自定义动作，放入三个等待状态模式下，即框架保留了用户自定义动作的扩展空间。

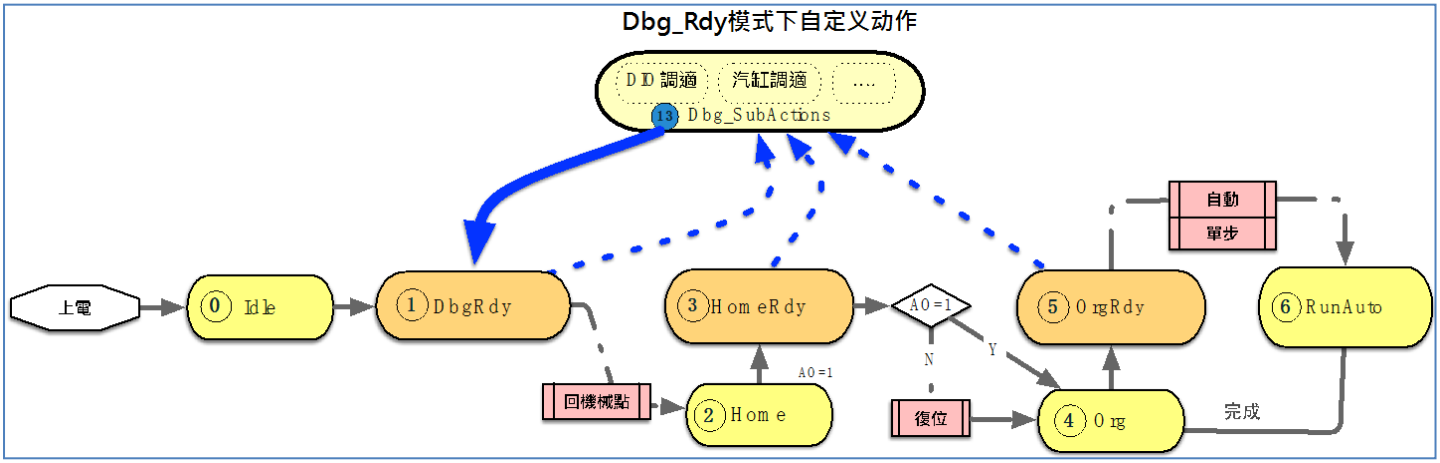
基本动作

	说明
系统定义中的基本动作 命令 VR(C_Cmd)的编号范围为:0~10	1. 系统定义了常见设备中需用到的基本动作 2. 基本动作执行会引起系统状态机切换，从而实现动作有效管理 3. 例如：回机械原点，回工作原点，自动流程动作

用户自定义动作

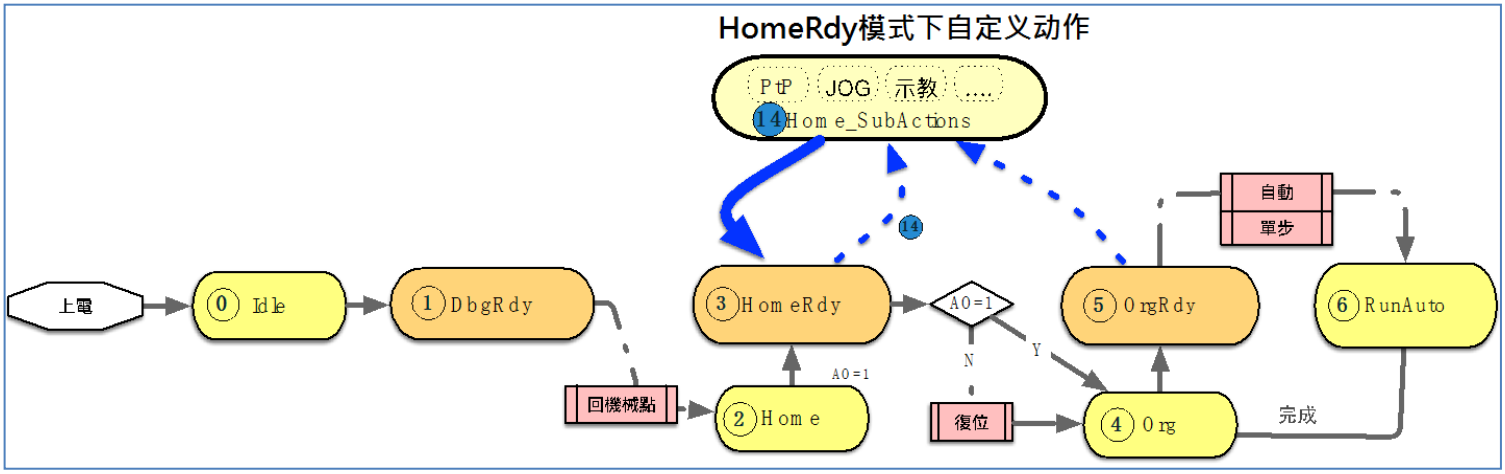
- DbgRdy 模式下的自定义动作：Dbg_SubActions

	说明
DbgRdy 模式下自定义动作 命令 VR(C_Cmd)的编号范围为:101~199	4. DbgRdy 模式可以执行的是调试动作,不需要回机械原点就可以执行的动作放在此模式下。 5. 当系统处于①DbgRdy 、③HomeRdy、⑤OrgRdy 的模式下，才可以执行 6. 该模式下的自定义动作执行完，系统会自动进入 DbgRdy 状态模式 7. 例如：DIO 手动调试，轴调试



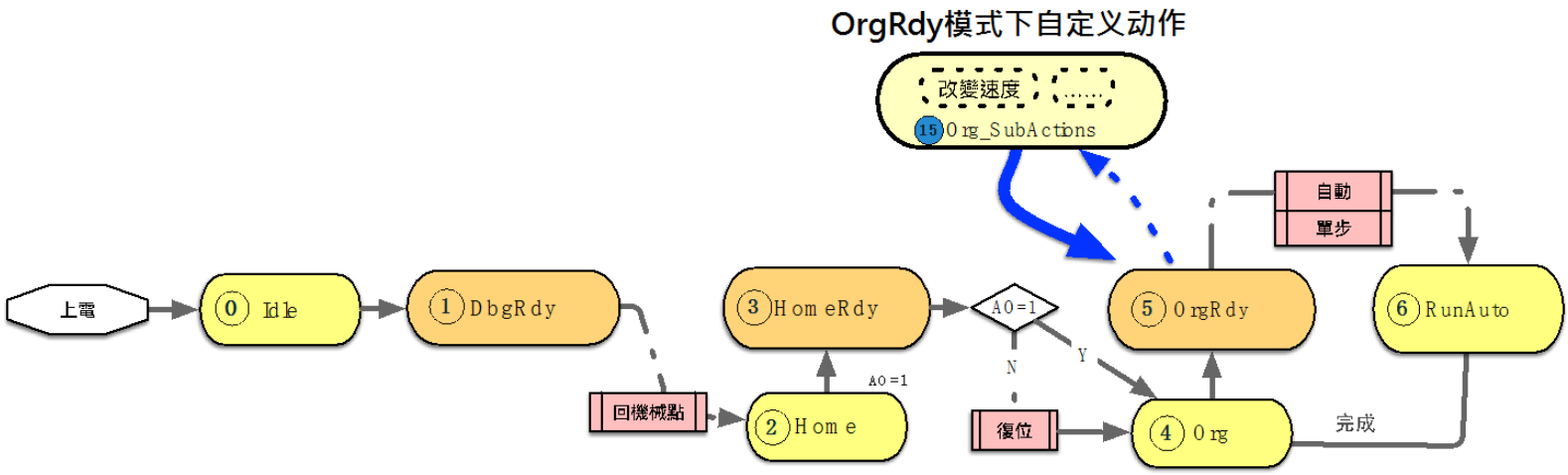
- HomeRdy 模式下的自定义动作：Home_SubActions

	说明
HomeRdy 模式下自定义动作 命令 VR(C_Cmd)的编号范围为:201~299	1. 需要回机械原点才可执行的动作放在此模式下。 2. 当系统处于③HomeRdy、⑤OrgRdy 的模式下，才可以执行。 3. 该模式下的自定义动作执行完，系统会自动进入 HomeRdy 状态模式 4. 例如：示教、手动定位



- OrgRdy 模式下的自定义动作：Org_SubActions

	说明
OrgRdy 模式下自定义动作 命令 VR(C_Cmd)的编号范围为:301~399	1. 需要回工作原点才可执行的动作放在此模式下。 2. 当系统处于⑤OrgRdy 的模式下，才可以执行。 3. 该模式下的自定义动作执行完，系统会自动进 OrgRdy 状态模式 4. 例如：改变加工速度



自动/单步/暂停

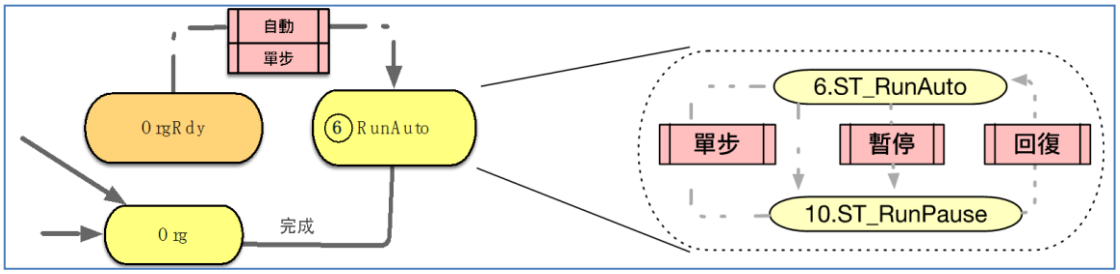
自动流程动作（Run）可以实现自动顺序执行、单步执行以及暂停操作。

- 自动流程顺序动作需写在 Sub ModeRun 的函数体里
- 一般情况下，用户定义的设备组件（如 XYZ Table）下的 WaitDone()方法作为一个单步节点
- 如果需要动作步骤间实现暂停操作，需在步骤节点间插入 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()
  XYZ.MOVE_XYZ 15000, 16000, 17000     'path 1
  XYZ.WaitDone()
  FC.ChecknPause()
END SUB
```

暫停節點

暫停節點



如何使用框架

框架已具备的功能

- 14 种设备系统状态自动转换
- 发生重大错误与一般错误时暂停动作
- 特定状态，拒绝不合法的操作(Ex: 机器还没有回零，不能操作 RUN)
- 9 种基本动作处理：

  a) 回原点，到工作点，

  b) 自动运行，暂停， 单步运行，停止

  c) 重大错误，一般错误，重置错误

两个步骤开发设备

步骤 1：添加基本动作处理

添加步骤	说明
1. 初始化	在 Init()，编写您的初始化动作...
2. 回机械原点	在 ModeHome ()，编写您的回机械原点动作...
3. 到工作原点	在 ModeOrg ()，编写您的回工作原点包含的动作...
4. 自动运行	在 ModeRun ()，编写您的自动运行动作...
5. 停止	1. 在 UserStop()，编写如何停止您的设备动作 2. 在 ModeStop()，编写停止后，您希望执行的动作...
6. 错误处理	**重大错误:** 1. 在 UserErrB(),编写发生错误时您要采取的动作(Ex: 停止) 2. 在 ModeErrB()，编写 Motion 错误停止后，您希望执行的动作... 3. 当发生错误，系统会暂停 Control02.bas 的执行，并等待清除错误命令 4. 请在 UserResetErr(),编写您的清除错误动作
	一般错误: 1. 在 UserErrS(),编写发生错误时您要采取的动作(Ex: 停止) 2. 在 ModeErrS()，编写 Motion 错误停止后，您希望执行的动作... 3. 当发生错误，系统会暂停 Control02.bas 的执行，并等待清除错误命令 4. 请在 UserResetErr(),编写您的清除错误动作

步骤 2：添加基本动作处理

- 步骤 2-1：在 DbgRdy 模式下添加可执行的自定义动作

添加步骤
1. 在 DoCmd()，定义您的命令编号 1xx ... 命令编号范围：101-199
2. 在 DoCmd() 设定子动作模式编号 VR(C_SubMD)= n
3. 在 ModeDbg_SubActions(), 添加当子动作模式 VR(C_SubMD) = n 时, 要执行的动作

范例：如需在 DbgRdy 模式下添加“对轴 0、1、2 三个轴的伺服使能”动作。

```
SUB DoCmd()
SELECT CASE INT(VR(C_Cmd))
CASE 101                        '指令编号(Debug Ready 模式下)
    VR(C_SubMD)=1               '设定子动作模式编号 VR(C_SubMD) = n
    Ret =Err_Success
END SELECT
```

```
SUB ModeDbg_SubActions()
SELECT CASE INT(VR(C_SubMD))
CASE 1                  '当子动作模式 VR(C_SubMD) = n 时
    BASE 0,1,2          '示范动作...对轴 0,1,2 使能伺服
    SVON
END SELECT
```

- 步骤 2-2：在 HomeRdy 模式下添加可执行的自定义动作

添加步骤
1. 在 DoCmd()，定义您的命令编号 2xx … 命令编号范围：201-299 2. 在 DoCmd() 设定子动作模式编号 VR(C_SubMD)= n 3. 在 ModeHome_SubActions(), 添加当子动作模式 VR(C_SubMD) = n 时, 要执行的动作

范例：如需在 HomeRdy 模式下添加“做轴 0、1、2 三轴的直线插补”动作。

```
SUB DoCmd()
SELECT CASE INT(VR(C_Cmd))
CASE 201                          '指令编号(Home Ready 模式下)
    VR(C_SubMD)=2                 '设定子动作模式编号 VR(C_SubMD) =  n
    Ret = Err_Success
END SELECT
```

```
SUB ModeHome_SubActions()
SELECT CASE INT(VR(C_SubMD))
CASE 2                        '当子动作模式 VR(C_SubMD) =  n 时
    BASE 0,1,2                '示范动作...到指定点 A
    LINEABS 1000, 1000, 1000
    WAIT DONE
END SELECT
```

- 步骤 2-3：在 OrgRdy 模式下添加可执行的自定义动作

添加步骤
1. 在 DoCmd()，定义您的命令编号 3xx … 命令编号范围：301-399 2. 在 DoCmd() 设定子动作模式编号 VR(C_SubMD)= n 3. 在 ModeOrg_SubActions()，添加当子动作模式 VR(C_SubMD) = n 时，要执行的动作

范例：如需在 OrgRdy 模式下添加“载入工单”动作。

```
SUB DoCmd()
SELECT CASE INT(VR(C_Cmd))
CASE 301                        '指令编号(Org Ready 模式下)
    VR(C_SubMD)=3               '设定子动作模式编号 VR(C_SubMD) =  n
    Ret = Err_Success
END SELECT
```

```
SUB ModeOrg_SubActions()
SELECT CASE INT(VR(C_SubMD))
CASE 3                          '当子动作模式 VR(C_SubMD) =  n 时
    LoadWorkOrder()             '示范动作...载入工单，准备加工。
END SELECT
```

系统函数说明

ScanMIO( )

格　式：　Function ScanMIO( ) AS INTEGER

描　述：　扫描 Motion I/O 的状态。默认框架下，扫描轴上 MIO.ALM 、MIO.PEL、MIO.NEL 、MIO.EMG 四个 Motion I/O 的状态。需检测其他 Motion I/O 的状态，需在 ScanMIO( )函数体中自行添加

返回值：

- ERR_AxisAlmError：发生伺服报警时，返回 ERR_AxisAlmError
- ERR_AxisPelError：发生正向硬件限位报警时，返回 ERR_AxisPelError
- ERR_AxisNelError：发生负向限位报警时，返回 ERR_AxisNelError
- ERR_AxisEmgError：发生负向限位报警时，返回 ERR_AxisEmgError
- ERR_Success：没有发生轴错误时返回 ERR_Success

FC.CheckCmd( )

格　式：　FUNCTION FlowCtrl.CheckCmd( ) AS INTEGER

描　述：　检测 VR(C_Cmd)命令是否合规。系统内部已做宏定义 ,VR(C_Cmd)即 VR(5)。VR(C_Cmd)中的命令值说明如下表：

命令	对应值	说明
N_NO_CMD	0	无命令
N_CMD_Home	1	回机械原点命令
N_CMD_Org	2	到工作原点命令
N_CMD_Run	3	自动流程运行命令
N_CMD_Stop	4	停止运动命令
N_CMD_Pause	5	暂停自动流程命令
N_CMD_Step	6	自动流程单步执行命令
N_CMD_Resume	7	暂停后恢复自动流程运行命令
N_CMD_OnErrB	8	重大错误后处理命令
N_CMD_OnErrS	9	一般错误后处理命令

N_CMD_ResetErr	10	清除轴错误状态命令
N_CMD_DbgSub	101~199	DebugReady 模式下的子程序执行命令
N_CMD_HomeSub	201~299	HomeReady 模式下的子程序执行命令
N_CMD_OrgSub	301~399	OrgReady 模式下的子程序执行命令
N_CMD_RunSub	401~499	Run 模式下的子程序执行命令
其他命令	以上值之外	不合规命令

返回值：

- ERR_NoCommand：检测到 VR(C_Cmd)的值为 0
- ERR_InvalidOperation 检测到 VR(C_Cmd)的值不符合机器当前状态允许的操作，比如当前机器正在执行自动流程动作，这时候 HMI 下发一个回原点命令，机器不会执行回原点命令。FC.CheckCmd( )函数的返回值则为 ERR_InvalidOperation
- ERR_UnknowCommand： 检测到 VR(C_Cmd)的值非（0~10）或（101~499）
- ERR_Success：检测到 VR(C_Cmd)的值非以上三种情况时，返回 ERR_Success

DoCmd( )

格　式： FUNCTION DoCmd( ) AS INTEGER

描　述： 根据检测到的 VR(C_Cmd)值，系统内部会改变 VR(C_RunMD)和 VR(C_SubMD)的值。系统内部已做宏定义，VR(C_RunMD)即 VR(0)，VR(C_SubMD)即 VR(4)。

根据 VR(C_Cmd)的值，产生的 VR(C_RunMD)和 VR(C_SubMD)的值如下表

VR(C_Cmd)值	产生的 VR(C_RunMD)值	产生的 VR(C_SubMD)值
N_NO_CMD	无变化	无变化
N_CMD_Home	N_MD_Home：2	无变化
N_CMD_Org	N_MD_Org：4	无变化
N_CMD_Run	N_MD_Run：6	无变化
N_CMD_Stop	N_MD_Stop：7	无变化
N_CMD_Pause	无变化，但程序会执行到 FC.ChecknPause( )这行停下	无变化
N_CMD_Step	无变化，但程序会执行到下一个	无变化

	FC.ChecknPause( )行停下	
N_CMD_Resume	无变化，但程序会恢复执行，继续未执行完的自动流程动作	无变化
N_CMD_OnErrB	N_MD_ErrB：8	无变化
N_CMD_OnErrS	N_MD_ErrS：9	无变化
N_CMD_ResetErr	● ErrB：N_MD_DbgRdy ● ErrS：曾回过机械原点，则 N_MD_HomeRdy ● ErrS：未回过机械原点，则 N_MD_DbgRdy	无变化
N_CMD_DbgSub	N_MD_Dbg_SubActions：13	依照用户在该状态模式下的子动作定义的编号
N_CMD_HomeSub	N_MD_Home_SubActions :14	依照用户在该状态模式下的子动作定义的编号
N_CMD_OrgSub	N_MD_Org_SubActions：15	依照用户在该状态模式下的子动作定义的编号
N_CMD_RunSub	N_MD_Run_SubActions：16	依照用户在该状态模式下的子动作定义的编号
其他命令	无变化	无变化

返回值：

ERR_UnknowCommand：检测到 VR(C_Cmd)的值非（0~10）或（101~499）

ERR_Success：检测到 VR(C_Cmd)的值合规

FC. SetPreMode( )

格　式：　FlowCtrl. SetPreMode( )

描　述：　根据机器得到的动作命令 VR(C_RunMD)，更新设置机器当前的系统状态 VR(CS_RunSTA)。

VR(C_RunMD)与 VR(CS_RunSTA)的对应关系如下：

命令 VR（C_RunMD）	设置的系统状态 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：　目前模式动作完成后，依据目前的模式，决定了下一个模式。即当前动作命令 VR(C_RunMD)的模式下动作完成后，用该函数设置更新下一个 VR(C_RunMD)，这个运行模式的管理是依据框架系统内部定义的状态机实现的，请参考“状态机与机器运行模式”章节。

当前 VR(C_RunMD)模式与下一个 VR(C_RunMD)模式的对应关系如下：

当前 VR（C_RunMD）模式	下一个 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)为 1，则 N_MD_Org：4 VR(CF_AutoOrg)为 0，则 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	回过机械原点：N_MD_HomeRdy：3 未回过机械原点：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	回过机械原点：N_MD_HomeRdy：3 未回过机械原点：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 是系统内部定义的一个类，这个类里面包含了一些流程控制用到的 SUB 和 Funtion。用户结合本文档的说明，无需了解该类里的方法细节即可完成开发。该类里的一些需要用户了解内部细节的方法如 FC.CheckCmd( )、FC.SetPreMoe( )、FC.SetNextMoe( )等已在本章节说明。

XYZ_TABLE

XYZ_TABLE 是框架下系统定义的一个示例类，这个类代表 XYZ 直角坐标平台这样的一个元件，该类包含了 XYZ 直角坐标平台下的一些方法：如三轴移动，回原点，WatiDone,停止移动，清除错误等。

该类定义在 ProjectDefine.bi 里面，如果用户的设备也是 XYZ 直角坐标平台，用户可以根据实际应用，在 TYPE XYZ_TABLE 里扩展方法、属性。如果用户的设备不是 XYZ 直角坐标平台，用户可以仿照 TYPE XYZ_TABLE 定义自己的类。