

# 变量命名规范

## 1.基本规范

- 禁止以数字起始命名，命名不超过 32 个字符。
- 推荐以英文单词命名，一个单词遵循完整字母规则。
- 多个词组合命名时，每个单词的第一个字母遵循大写规则。
- 多个词组合命名过长时，每个单词可以不超过 4 个字母缩写规则。
- 推荐以英语进行注释，内容遵循应简明扼要规则。

## 2.普通类型命名规范

类型	前缀	说明	示例
SINT, USINT, DINT, UDINT, BYTE, WORD, DWORD, LWORD....	n	整数	ncount
bool	x	位	xSwitch
REAL, LREAL	r	浮点数	rValue
STRING	s	字符串	sNetID
WSTRNG	ws	宽字符串	wsRouteName
TIME	t	时间	tDelay
DATE	d	日期	dMonday
DATA_AND_TIME	dt	时间+日期	dtNewyear
ARRAY[...] OF ...	ar	数组	aMessages
POINTER/PVOID	p	指针	pData
Nested Pointer	pp	嵌套指针	ppData
Pointer to Interface	pip	接口指针	pipCylinder
Reference to	ref	引用类型	refSize

## 3.对象命名规范

对象	前缀	说明	示例
FUNCTION_BLOCK	FB_	功能块	FB_GetData
METHOD	M_	方法	M_Init
ACTION	A_	动作	A_MoveStart
INTERFACE	I_	接口	I_Provider
PROPERTY	get/set	属性	GetValue、SetValue

对象	前缀	说明	示例
STRUCT	ST_	结构体	ST_Buffer
ENUM	E_	枚举	E_SignalStates
UNION	U_	联合体	U_VariableType
TYPE	T_	类型定义	T_Localiz
FUNCTION	F_	函数定义	F_Convert
PROGRAM	PRG_/无	静态程序类	PRG_Axis or Axis
GVL	GVL_	全局常量型文件	GVL_Subsystem
GVL CONSTANT	GCL_	全局参数型文件	GCL_Subsystem

## 4.实例命名规范

对象	前缀	说明	示例
Function Block	fb	功能块实例化	fbGetData
Interface	ip	接口	ipCylinde
Struct	st	结构体实例化	stBufferEntr
Union	u	联合体实例化	uCtr
Enum	e	枚举实例化	eSignalStat
Alias	Type	采用通用变量命名规则	sNetI
Constant	c	局部或全局常量类型	cMaxVelocity
Internal Variables	无/_	局部类型变量、实例前建议增加_前缀。	_bExecute/fbSetParam
IN/OUT Variables		所有 PRG、FB、Fuction 等具备输入/输出变量、实例参数的命名参考通用类型或对象命名规则	sNetID、nPort \stBufferEntry\fbSetParam 或者 fbSetParam_IN/OUT

## 5.其他命名规范

文件夹	说明	示例
DUTs	建议结构体类型都放在同一索引目录中且根目录为 DUTs	DUTs
GVLs	建议全局变量类型都放在同一索引目录中且根目录为 GVLs	GVLs
POUs	建议静态对象类型都放在同一索引目录中且根目录为 POU s	POUs
VISUs	建议 HMI 对象类型都放在同一索引目录中且根目录为 VISUs	VISUs

## 6.代码缩进规范

- 变量定义以统一的缩进（1 tab 键）规范，类型、初值、注释也遵循以 tab 键缩进来达到整体对其原则，确保变量定义清晰和整洁：

```
{attribute 'hide_all_locals'}
FUNCTION_BLOCK FB_SignatureWriter
VAR_INPUT
    sNetID           : T_AmsNetId      := '';
    nPort            : T_AmsPort       := 851;
    bExecute         : BOOL            := FALSE;
    tTimeout         : TIME            := T#10S;
END_VAR
VAR_OUTPUT
    bBusy            : BOOL;
    bError           : BOOL;
    nErrID           : UDINT;

    bError2          : BOOL;
    nErrID2          : UDINT;
    bBusy2           : BOOL;
    bSucceed         : BOOL;
END_VAR
(library PRIVATE)
VAR
    _fbAdsRead       : ADSREAD;
    _fbAdsWrite      : ADSWRITE;
    _fbAdsReadWrite  : ADSDWETEX;

    _eState          : (STATE_NULL, STATE_INIT, STATE_LOAD, STATE_MONITOR, STATE_ERROR) := STATE_NULL;

    _stListItem      : ST_SignItemList;
    _stListItemInfo  : ST_SignItemInfoList;

```

- 主代码以统一的缩排（1 tab 键缩进）规范，逻辑嵌套、注释等也以缩进对齐规范来确保变量定义清晰和整洁：

```
_rStartTriger(CLK:= bExecute);
IF _rStartTriger.Q THEN
    _eState := STATE_INIT;
    bSucceed := FALSE;
END_IF

_fbGetSystemId( bExecute:= bExecute, tTimeout:= T#5S, sNetId:= sNetID);
IF bExecute AND NOT _fbGetSystemId.bBusy AND NOT _fbGetSystemId.bError THEN
    _sDeviceID := GUID_TO_STRING(_fbGetSystemId.stSystemId);
    _bValid := TRUE;
END_IF

IF _bValid THEN
    M_Monitor();
    M_WriteToDB();
END_IF
```

## 7.硬件连接变量

前缀	说明	示例
di	Digital input	diStart
do	Digital output	doLed

前缀	说明	示例
ai	Analog input	aiFlow
ao	Analog output	aoTemperature

## 8.空格

为了增加可读性，在每个操作符的前后都加上一个空格:

```
xAxisPos = x0 + deltaX;  
if (machineState == STATE_RUN)
```