

# VAR\_IN\_OUT接口是什么

- 一个 VAR\_IN\_OUT variable 是一个输入/输出变量，它是Function Block接口的一部分，用作正式的按引用传递参数。

FB	FB_TestINOUT			
	uAddValue	USINT		VAR_INPUT
	uResultValue	USINT		VAR_OUTPUT
	guTest	USINT		VAR_IN_OUT

- VAR\_IN\_OUT能够实现的效果为，通过功能块直接操作外部的变量，对外部变量进行直接修改。
- 输入/输出变量 (VAR\_IN\_OUT) 将转变成为一个指针。因此，在调用时不能对它们赋值常数，而且不能进行外部的读或写操作。
- 在调用有VAR\_IN\_OUT类型的功能块时，必须如下图，VAR\_IN\_OUT变量赋值与函数调用同时执行。

```
//Function Block中IN OUT类型变量使用写法，与FB调用同时指向。  
FB_TestINOUT_O( guTest := guTestValue);
```

# 什么情况下建议使用VAR\_IN\_OUT类型接口

- 操作的数据类型是公用的数据类型，则使用VAR\_IN\_OUT可节省VAR\_IN与VAR\_OUT两次赋值操作
- 操作的数据类型比较大，使用VAR\_IN\_OUT可以减少整体数据的赋值操作
- 使用 VAR\_INPUT 和 VAR\_OUTPUT 会造成占用过多的内存，为此，在功能块编程时，可尽量使用 VAR\_IN\_OUT 替代，减少对存储区的占用

# 如何使用VAR\_IN\_OUT

## A.功能块操作

### A.1. 在库中定义一个VAR\_IN\_OUT接口

FB	FB_TestINOUT			
	uAddValue	USINT		VAR_INPUT
	uResultValue	USINT		VAR_OUTPUT
	guTest	USINT		VAR_IN_OUT

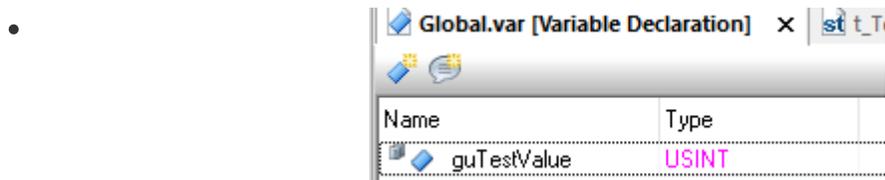
### A.2.在此库中编写代码

- 与常规变量一样使用
- 例如下图中代码案例即为，调用一次功能块，则guTest这个变量的数值将增加

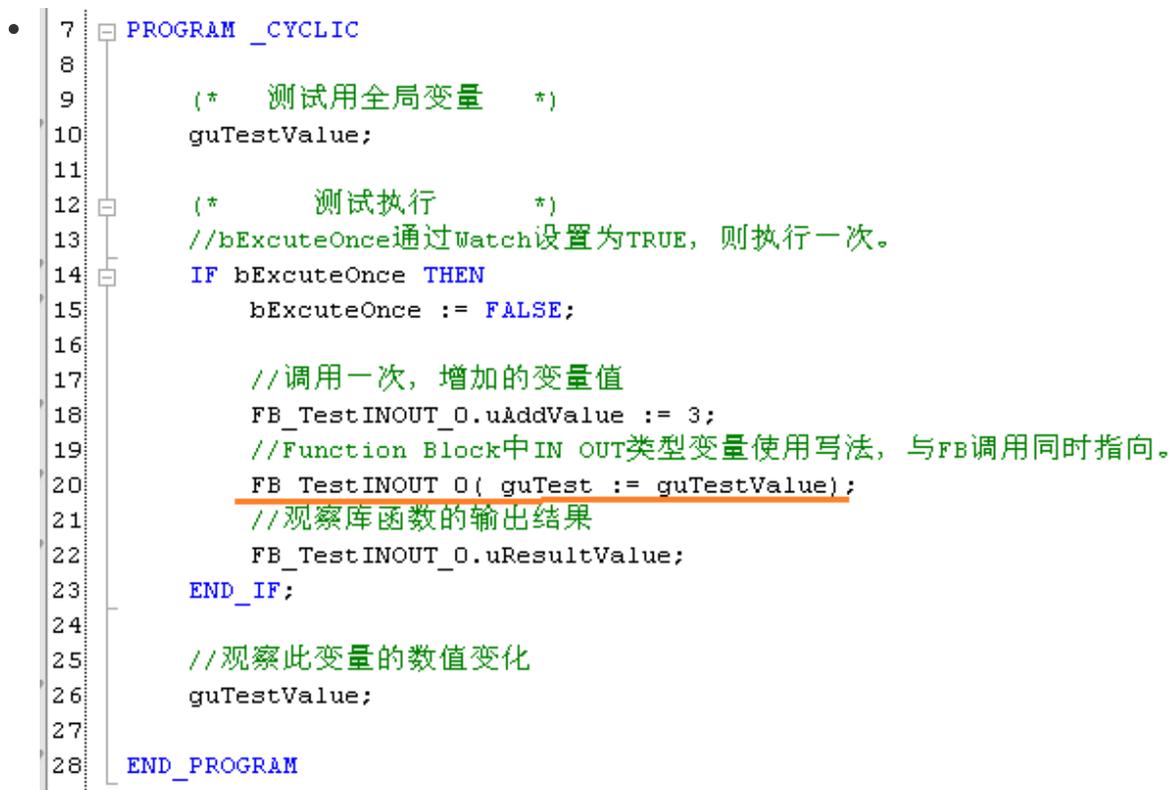
```
FUNCTION_BLOCK FB_TestINOUT  
  
//基于IN OUT 类型变量的写法，可便于直接对外部变量进行读写控制。  
guTest := guTest + uAddValue;  
uResultValue := guTest;  
  
END_FUNCTION_BLOCK
```

## B.代码调用

### B.1.声明与VAR\_IN\_OUT接口类型同类型的变量



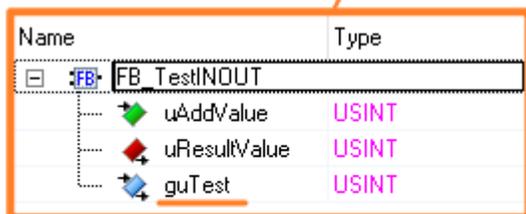
### B.2 调用方式如下，VAR\_IN\_OUT接口必须与函数同时调用



## C.注意事项

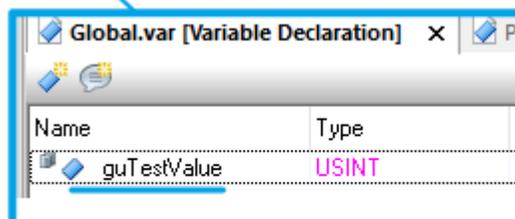
```
//调用一次, 增加的变量值
FB_TestINOUT_0.uAddValue := 3;
//Function Block中IN OUT类型变量使用写法, 与FB调用同时指向。
FB_TestINOUT_0( guTest := guTestValue);
```

**数据类型必须  
完全一致**



Name	Type
FB_TestINOUT	
uAddValue	USINT
uResultValue	USINT
guTest	USINT

**功能块中的接口**



Name	Type
guTestValue	USINT

**外部声明的变量**

## D.效果

The screenshot shows the SIMATIC Manager IDE with the following components:

- Editor:** Ladder logic program for `t_TestProg::Main.st`. It contains a `PROGRAM_INIT` block and a `PROGRAM_CYCLIC` block. The `PROGRAM_CYCLIC` block includes a test function block call `FB_TestINOUT_0` with the parameter `guTestValue`.
- Watch Window:** Titled "Watch [t\_TestProg::Main.st]", it displays the current values of variables: `bExcuteOnce` is `FALSE` and `guTestValue` is `0`.
- Annotations:** Orange text in the center of the editor reads: "通过功能块FB\_TestINOUT 执行一次，外部变量guTestValue + 3" (Execute once through function block FB\_TestINOUT, external variable guTestValue + 3).

## 测试Demo

- [Demo文件](#)

## 异常使用时报错信息

### 1177

- Short text
  - No external access to VAR\_IN\_OUT parameter `<Name>` of `<Function block>`.
- Error description
  - When calling a function block, the address of the current parameter is specified for VAR\_IN\_OUT parameters instead of the value. This allows the value of the current parameter in the function block to be changed.
  - A VAR\_IN\_OUT parameter is only permitted to be used when calling a function block; direct access using the instance variable is not possible.
  - In Automation Basic, an alias function block call cannot be assigned to a VAR\_IN\_OUT parameter. Assignment to a VAR\_IN\_OUT parameter of a function block is only possible with a normal function block call. In a normal function block call, all parameters are specified in the line in which the call is made.
- Suggestion for error correction
  - In Automation Basic, replace the alias function block call of a function block with a VAR\_IN\_OUT parameter with a normal function block call.