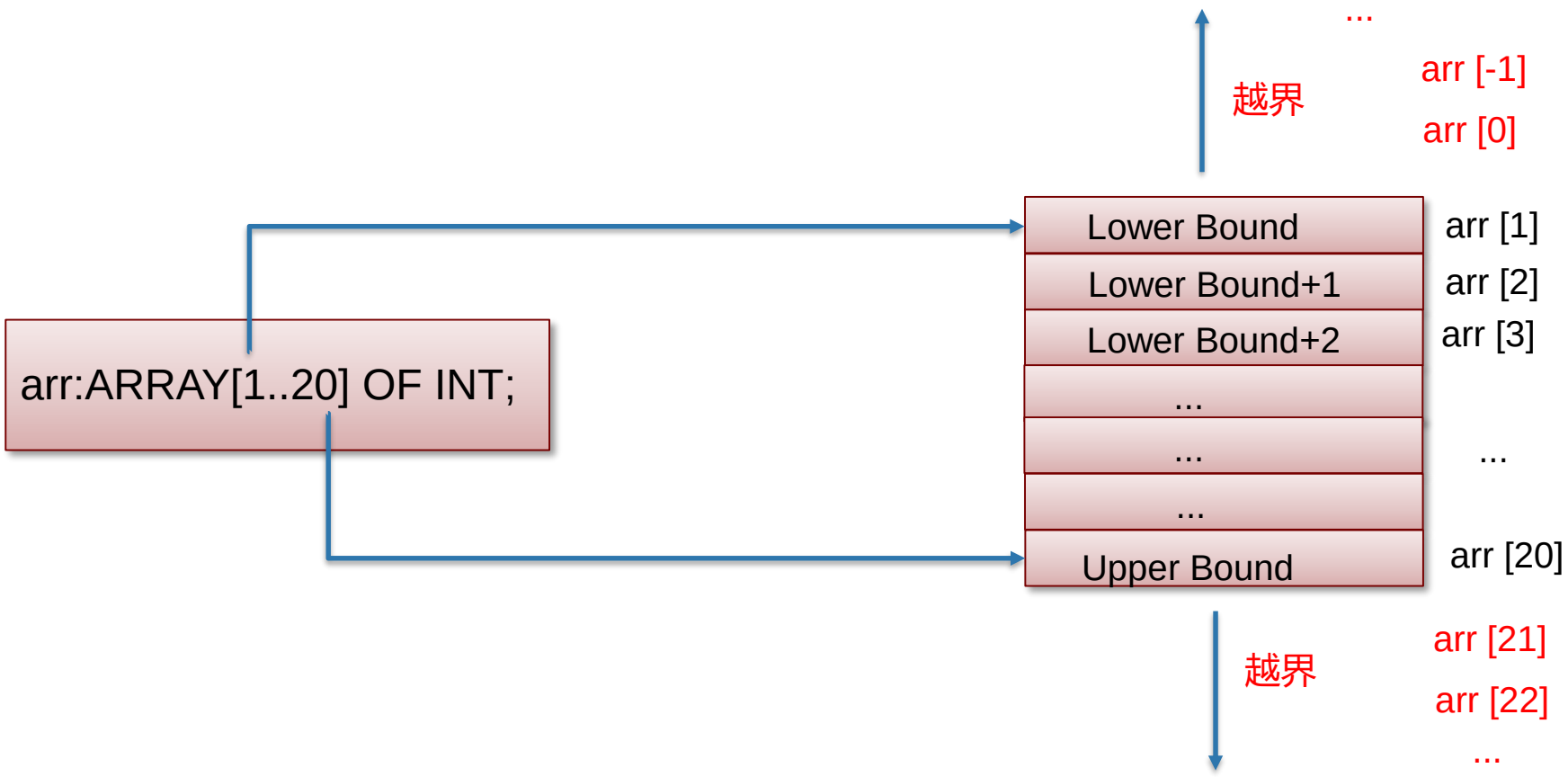
TC3数组越界与Plc Exception

BECKHOFF

1 数组边界和越界


...
越界
arr [-1]
arr [0]
Lower Bound
Lower Bound+1
Lower Bound+2
...
...
...
Upper Bound
arr [1]
arr [2]
arr [3]
...
arr [20]
arr:ARRAY[1..20] OF INT;
越界
arr [21]
arr [22]
...

2 常见数组越界现象

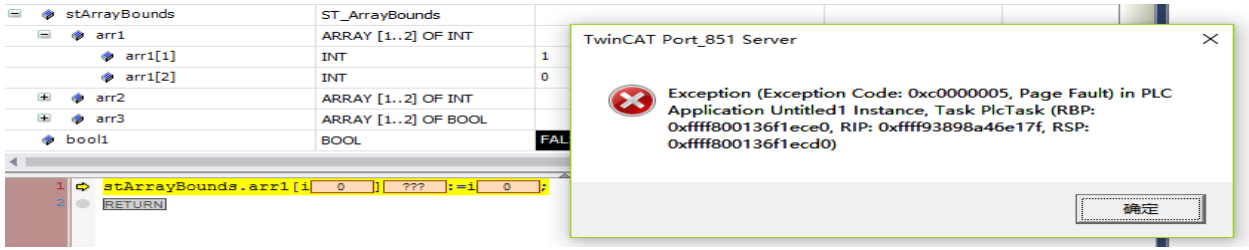
BOOL变量在线监控时显示<不能得到表达式的值>

arr3	ARRAY [1..2] OF BOOL	
arr3[1]	BOOL	<不能得到表达式的值。>
arr3[2]	BOOL	FALSE

程序中未对某变量赋值但该变量显示有值

arr2	ARRAY [1..2] OF INT	
arr2[1]	INT	3
arr2[2]	INT	0

数组索引小于数组Lower Bound导致PageFault



BECKHOFF

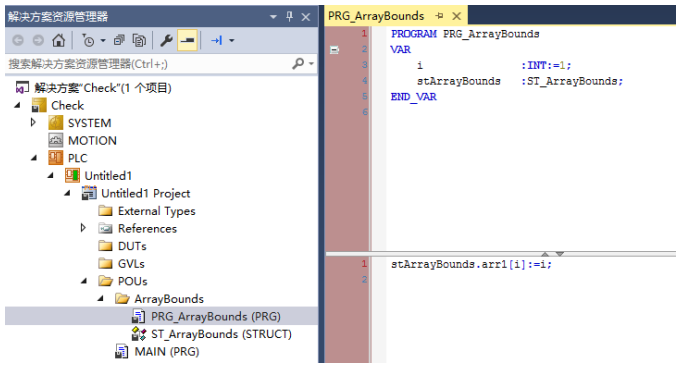
3 数组越界演示

Step1：
定义结构体

```
TYPE ST_ArrayBounds :
STRUCT
        arr1        :ARRAY[1..2] OF INT;
        arr2        :ARRAY[1..2] OF INT;
        arr3        :ARRAY[1..2] OF BOOL;
END_STRUCT
END_TYPE
```

说明：结构体中连续定义的变量的地址是连续的（忽略对齐造成的内存间隔），TC3程序或者全局变量中定义连续定义的变量的地址不一定是连续的，所以演示的时候以结构体举例

Step2：
程序编写

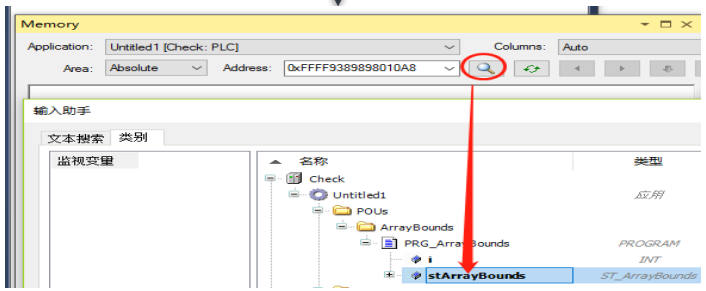


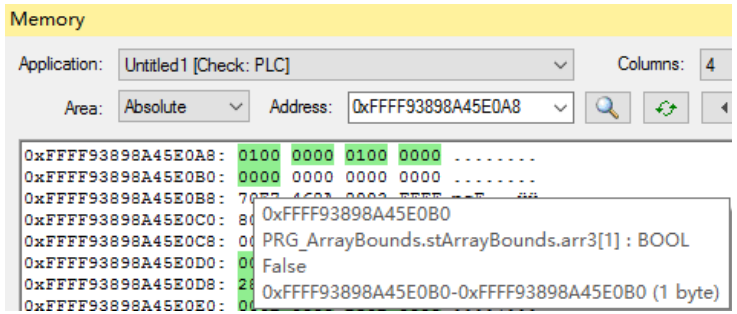
说明：变量i的初始值为1（如果初值为0，PLC运行后就会报Page Fault）

BECKHOFF

3 数组越界演示

Step3：
打开内存监视界面





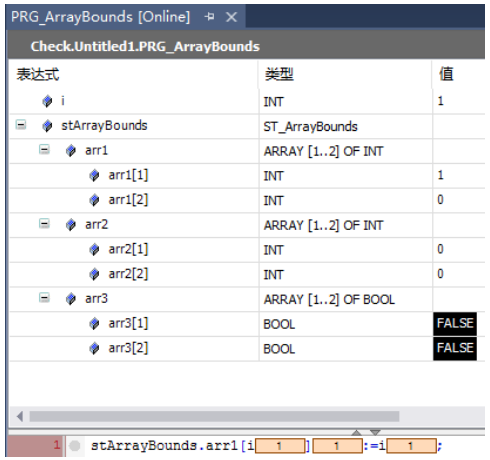
说明：

1.鼠标放置在绿色地址区上，可以显示内存对应的PLC变量名称

2. 通过内存监控界面，可以确认结构体中定义的三个数组变量的地址是连续，这样更方便演示数组越界

3 数组越界演示

Step4：
数组越界演示



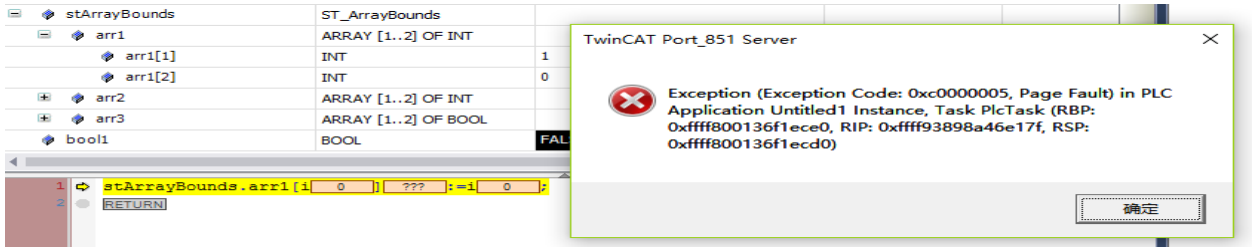
i赋值3

arr2	ARRAY [1..2] OF INT	
arr2[1]	INT	3
arr2[2]	INT	0

i赋值5

arr3	ARRAY [1..2] OF BOOL	
arr3[1]	BOOL	<不能得到表达式的值。>
arr3[2]	BOOL	FALSE

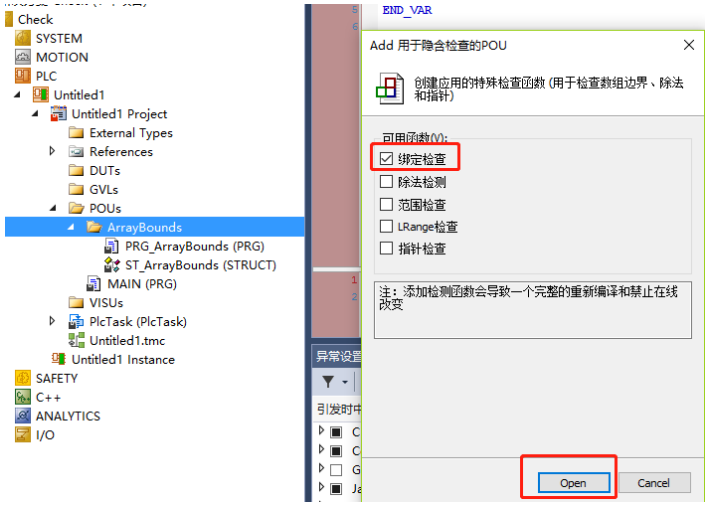
i赋值0

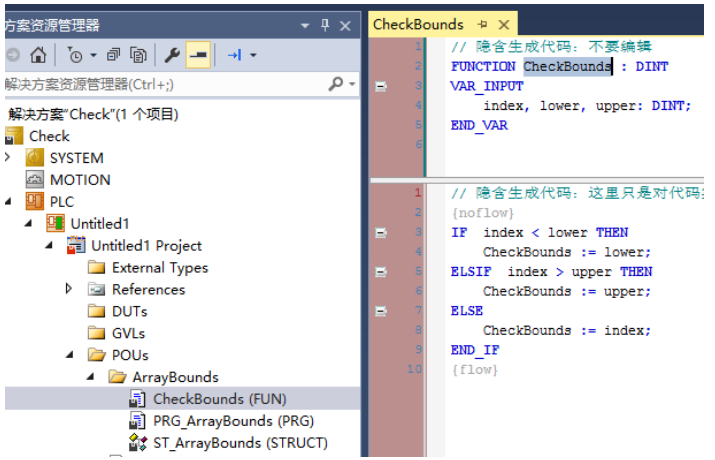


BECKHOFF

4 数组越界排查

Step1：
添加CheckBounds函数





说明：

添加CheckBounds函数后，数组发生越界时PLC会自动调用CheckBounds函数，不需要在PLC程序中手动调用。

发生越界时，如果数组索引小于LowerBound，PLC自动按照LowerBound处理；如果数组索引大于UpperBound，PLC自动按照UpperBound处理。

4 数组越界排查

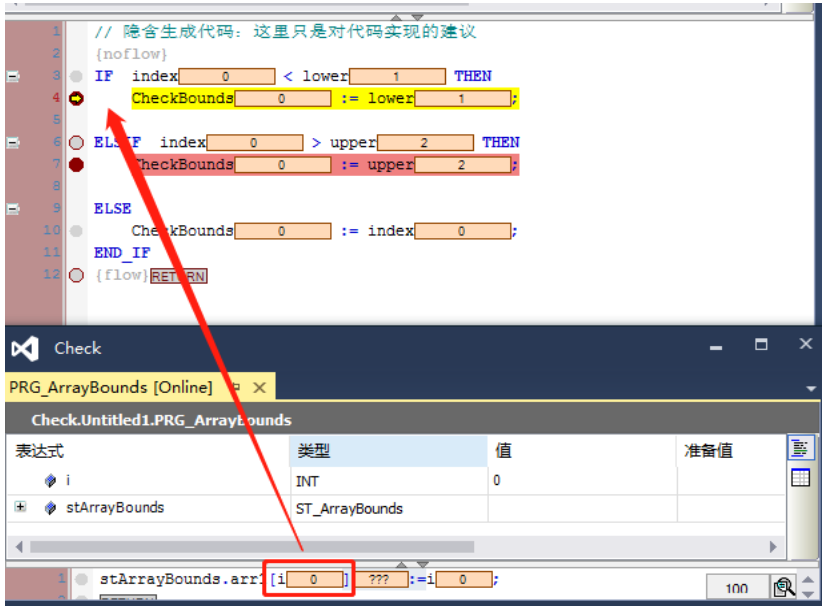
Step2：
CheckBounds函数中添加断点

```
// 隐含生成代码：这里只是对代码实现的建议
{noflow}
IF  index ??? < lower ??? THEN
    CheckBounds ??? := lower ???;

ELSIF  index ??? > upper ??? THEN
    CheckBounds ??? := upper ???;

ELSE
    CheckBounds ??? := index ???;
END_IF
```

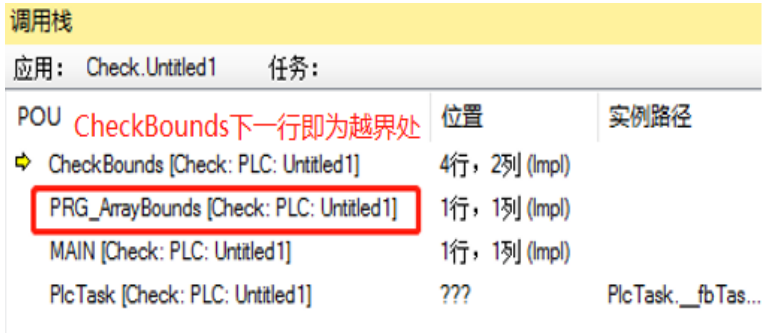
Step3：
变量i赋值0，触发越界



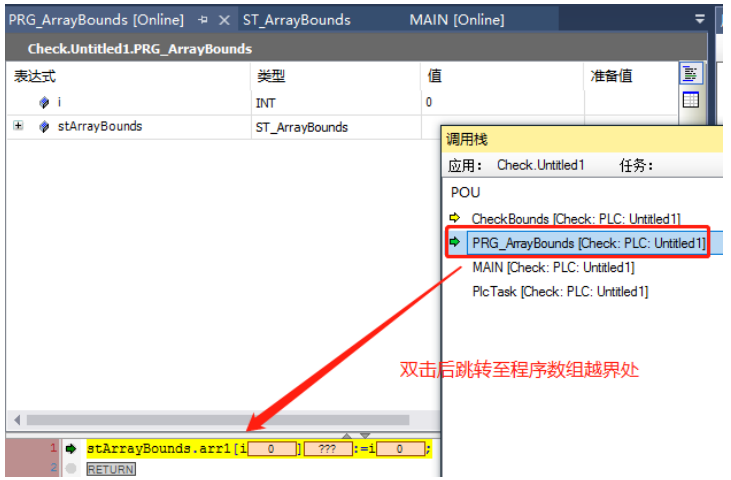
说明：
添加CheckBounds函数后，变量i赋值0，此时PLC并没有报Page Fault，而是运行至CheckBounds的lower处

4 数组越界排查

Step4：
查看调用栈



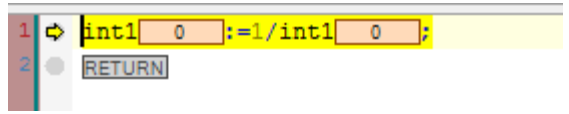
Step5：
定位至程序越界



BECKHOFF

1 Plc Exception（Plc 异常）简介

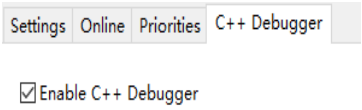
Twincat内核为了保证用户程序的强壮和稳定，提供了异常处理，方便编程人员定位和处理异常。



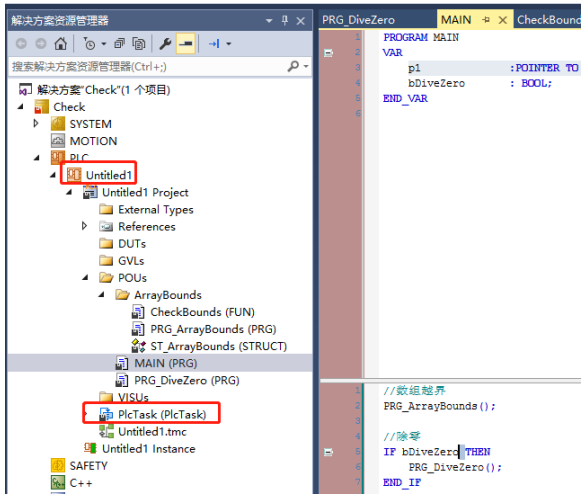
2 Plc Exception参数说明

异常输出源：
TC3 Plc 中的异常一般显示为为Port_85x Server
TC3 C++中的异常一般显示为为CTcDebugger Server

异常所在任务：Task名称

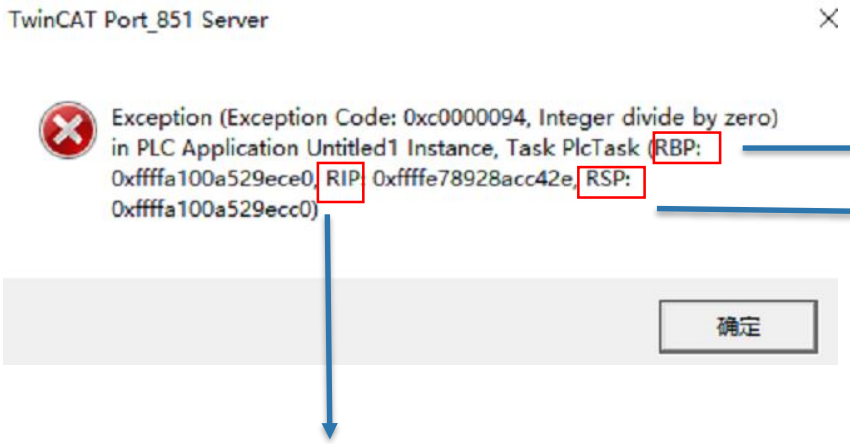


异常所在实例：Instance名称



异常码：
可以在微软网站查询异常码的更详细说明
https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-erref/596a1078-e883-4972-9bbc-49e60bebca55

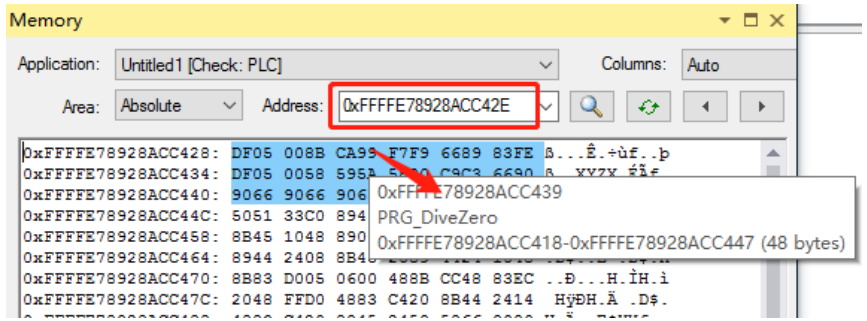
2 Plc Exception参数说明



RBP：栈基址寄存器,指向栈底

RSP：栈指针寄存器,指向栈顶

RIP：指令指针寄存器，用于保存下一条要执行的指令的地址



3 常规Plc Exception排查

Step1：
添加除零Check函数中添加断点

```
// 隐含生成代码：这里只是对代码实现的建议
{noflow}
IF divisor 0 = 0 THEN
    CheckDivDInt 0 :=1;
ELSE
    CheckDivDInt 0 :=divisor 0 ;
END_IF;
{flow}RETURN
```

Step2：
查看调用栈

```
int1 0 :=1/int1 0 ;
RETURN
```

调用栈

应用： Check.Untitled1　　任务：

POU	位置	实例路径
CheckDivDInt [Check: PLC: Untitled1]	4行，2列 (Impl)	
PRG_DiveZero [Check: PLC: Untitled1]	1行，1列 (Impl)	
MAIN [Check: PLC: Untitled1]	6行，1列 (Impl)	
PlcTask [Check: PLC: Untitled1]	???	PlcTask.__fbTask.CycleUpdate

4 排查Check函数不能处理的Plc Exception

Check函数能规避大部分Plc Exception，还有一些Plc Exception不能规避，下面列出部分不能规避的情况（更多的情况可以参考intel的FPU手册）

Table 31-20. Invalid Arithmetic Operations and the Masked Responses to Them

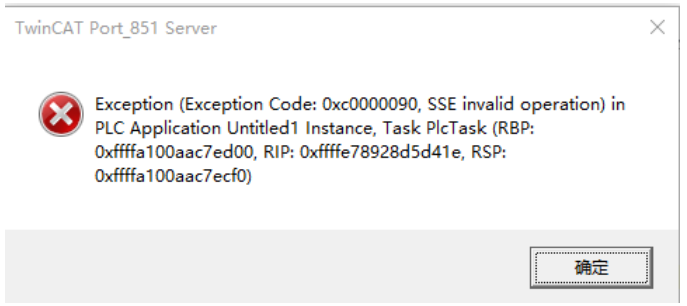
Condition	Masked Response
Any arithmetic operation on an operand that is in an unsupported format.	Return the real indefinite value to the destination operand.
Any arithmetic operation on a SNaN.	Return a QNaN to the destination operand (see "Operating on NaNs").
Compare and test operations: one or both operands are NaNs.	Set the condition code flags (C0, C2, and C3) in the FPU status word to 111B (not comparable).
Addition: operands are opposite-signed infinities. Subtraction: operands are like-signed infinities.	Return the real indefinite value to the destination operand.
Multiplication: ∞ by 0; 0 by ∞.	Return the real indefinite value to the destination operand.
Division: ∞ by ∞; 0 by 0.	Return the real indefinite value to the destination operand.
Remainder instructions FPREM, FPREM1: modulus (divisor) is 0 or dividend is ∞.	Return the real indefinite; clear condition code flag C2 to 0.
Trigonometric instructions FCOS, FPTAN, FSIN, FSINCOS: source operand is ∞.	Return the real indefinite; clear condition code flag C2 to 0.
FSQRT: negative operand (except FSQRT (–0) = –0); FYL2X: negative operand (except FYL2X (–0) = –∞); FYL2XP1: operand more negative than –1.	Return the real indefinite value to the destination operand.
FBSTP: source register is empty or it contains a NaN, ∞, or a value that cannot be represented in 18 decimal digits.	Store BCD integer indefinite value in the destination operand.
FXCH: one or both registers are tagged empty.	Load empty registers with the real indefinite value, then perform the exchange.
FIST/FISTP instruction when input operand <> MAXINT for destination operand size.	Return MAXNEG to destination operand.

4 排查Check函数不能处理的Plc Exception

1： SSE invalid operation

负数开平方操作

```
fVar1[ -1 ]:=SQRT(fVar1[ -1 ]);
```



2： floating point invalid operation

负数Ln的运算

```
fVar1[ -1 ]:=LN(fVar1[ -1 ]);
```

说明：

此类异常可以在运算前判断下被操作数的正负来规避

```
fVar1[ -1 ]:=-1.0;
IF fVar1[ -1 ]>=0 THEN
    fVar1[ -1 ]:=SQRT(fVar1[ -1 ]);
END_IF
```

4 排查Check函数不能处理的Plc Exception

3： SSE invalid operation

0乘以+∞

```
fInf:=1/0.0000000000000000000000000000000000000001;
//IF IsFinite(F_REAL( fInf ) ) THEN
    fVar1:=0*fInf;
//END_IF
```

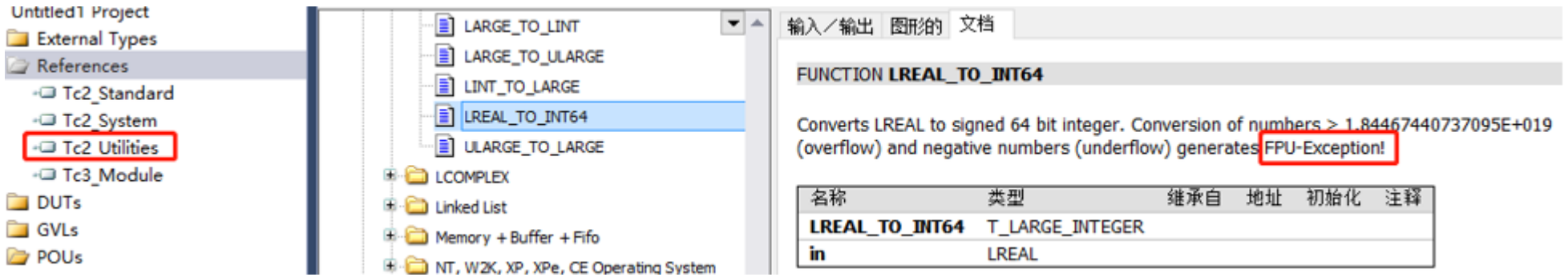
说明：

通过IsFinite函数可以规避此类异常

```
fInf:=1/0.0000000000000000000000000000000000000001;
IF IsFinite(F_REAL( fInf ) ) THEN
    fVar1:=0*fInf;
END_IF
```

4 排查Check函数不能处理的Plc Exception

4： Other operation



说明：

系统自带的函数，如果有关于FPU的提示，需要按照提示来使用。

TC3数组越界与Plc Exception

BECKHOFF

The End