
AN2004 在 GUTTA 中使用递归调用

COPYRIGHT © 2008 WWW.VISIBLECONTROL.COM

2008/02/15

概述.....	2
程序说明.....	2

概述

GUTTA PLC 支持若干子程序的调用。在每个子程序调用时，系统会为每个调用分配一段固定大小（取决于 L 变量区域的大小）的临时变量空间。分配顺序和子程序调用顺序一致，即子程序调用时分配，子程序返回时撤销。由于子程序的调用必然是先调用者后返回，故临时变量空间的分配和撤销也满足先分配者后撤销。正因为如此，临时变量空间的增长类似于栈空间的增长。当子程序嵌套达到最大时，临时变量空间达到最大。当所有子程序都返回时，临时变量空间回到最小。

需要注意的是，临时变量空间和子程序并不是对应的，而是和子程序调用对应的。若某个子程序是可以重入的，在不同的时候，临时变量对应的绝对地址是不同的。

临时变量主要有两方面的用途：

1. 用于子程序的参数传递，包括输入参数和输出参数的传递。
2. 用于存放子程序中无全局状态的值，使子程序尽量不污染全局地址空间。

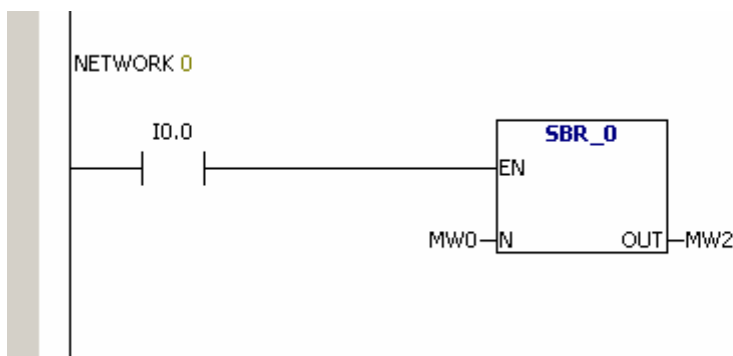
和 ICE61131 中 POU 的实例化概念不同，临时变量空间位于栈中，子程序返回时，临时变量就撤销了，因此临时变量不能存储全局状态。

利用子程序的递归调用，来进行 $N!$ 的数学运算，是本文档的内容。这个例子仅仅只是为了说明子程序的参数传递特性，在实际的工程应用中，应该避免这样的用法。

程序说明

<http://www.visiblecontrol.com/technologies/anindex/an2004/Recursion.vcw>

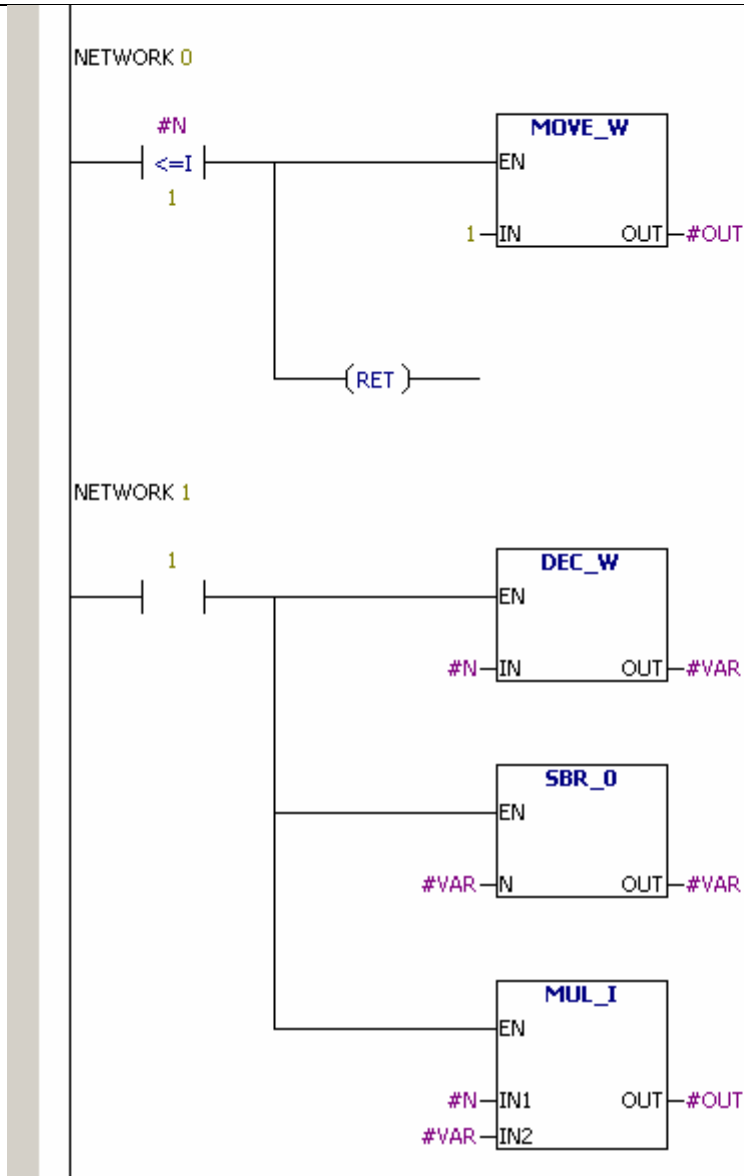
主程序 INT_0 (INT0)：



INT_0 (INT0) 只有 1 个梯级，即在 I0.0 按下时进行一次 $N!$ 的运算。MW0 表示 N ，MW0 为运算的结果。由于在 CPU-EC20 系统中，程序嵌套的层数不能超过 8 层，故 N 的值应该小于 8。

子程序 SBR_0 (SBR0)：

	符号	变量类型	数据类型	注释
<input checked="" type="checkbox"/> LW0	N	IN	WORD	
<input type="checkbox"/>		IN_OUT	BOOL	
<input checked="" type="checkbox"/> LW2	OUT	OUT	WORD	
<input checked="" type="checkbox"/> LW4	VAR	TEMP	WORD	



SBR_0 (SBR0) 有 3 个临时变量的定义。N 为输入参数，OUT 为输出参数，VAR 是临时变量。

NETWORK 0 用于递归的中止（否则会无限递归下去）。若 $N=0$ ，由于 $0! = 1$ ，故直接让输出 $OUT=1$ 。

NETWORK 1 用于递归的调用。若 $N! = 0$ ，先通过自己求出 $(N-1)!$ ，结果放入 VAR 中。然后令 $OUT=N*VAR$ ，即 $OUT=N!$ 。