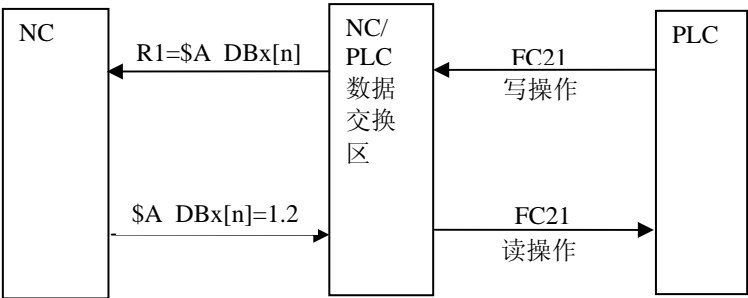


NC 读写 PLC 数据

系统内部有一块 NC/PLC 高速交换数据的独立的区域（双端口 RAM），该区域大小默认为 1024。这个数据交换区是在 DRAM 中，也就是说，系统下电后，交换区内数据不保持。



其中：“x”可以是 B/W/D/R
“n”是数据交换区的偏移量，注意与 PLC 侧数据类型要一致。

NC 侧：
使用系统变量\$A_DBB[n], \$A_DBW[n], \$A_DBD[n]和\$A_DBR[n]读/写数据交换区的数据，不支持位操作。

\$A_DBB[n]	字节（8 位）
\$A_DBW[n]	字（16 位）
\$A_DBD[n]	双字（32 位）
\$A_DBR[n]	实数（32 位）

使用\$A_DBR[n]读写时，数据会有一定误差，因为 NC 中使用 64 位表示实数，而 PLC 中使用 32 位表示实数。

如果只需要 NC 读取 PLC 的数据（例：R1=\$A_DBR[0]），不需要设置机床数据。
如果还需要 NC 写数据到 PLC（例：\$A_DBR[4]=123.456），则需要根据实际情况设置：
MD28150 MM_NUM_VDIVAR_ELEMENTS
该数据所占内存=MD28150*24字节。

PLC 侧：
使用 PLC 基本程序块 FC21 读/写数据交换区的数据。
FC21 有多种功能，其中功能 3 是读操作，功能 4 是写操作。

参数	参数类型	参数数据类型	取值范围	说明
Enable	输入	Bool		FC21 有效
Funct	输入	Byte	3, 4	3—读；4—写
S7Var	输入	Any	S7 数据	源/目的
IVAR1	输入	Int	0..1023	偏移
IVAR2	输入	Int	-1..1023	信号灯，-1 是无信号灯
Error	输出	Bool		
ErrCode	输出	Int		错误代码

其中 IVAR2（信号灯）用于 NC 与 PLC 的协调，告诉 NC/PLC 什么时候可以读数据，什么时候可以写数据。如果数据量不大，不会产生 NC/PLC 读写冲突可以把 IVAR2 设为 -1。（注：机床的设计者应保证 NC/PLC 之间交换数据的时序。）

例 1：

PLC 编程加入：
CALL FC21
Enable:=TRUE

Funct:=B#16#4 ←写操作
S7Var:=P#M100.0 BYTE 10 ←MB100-MB109
IVAR1:=0 ←偏移为 0
IVAR2:=99 ←信号灯字节为缓冲区分第 99 个字节
Error:=M200.0
Errcode:=MW210

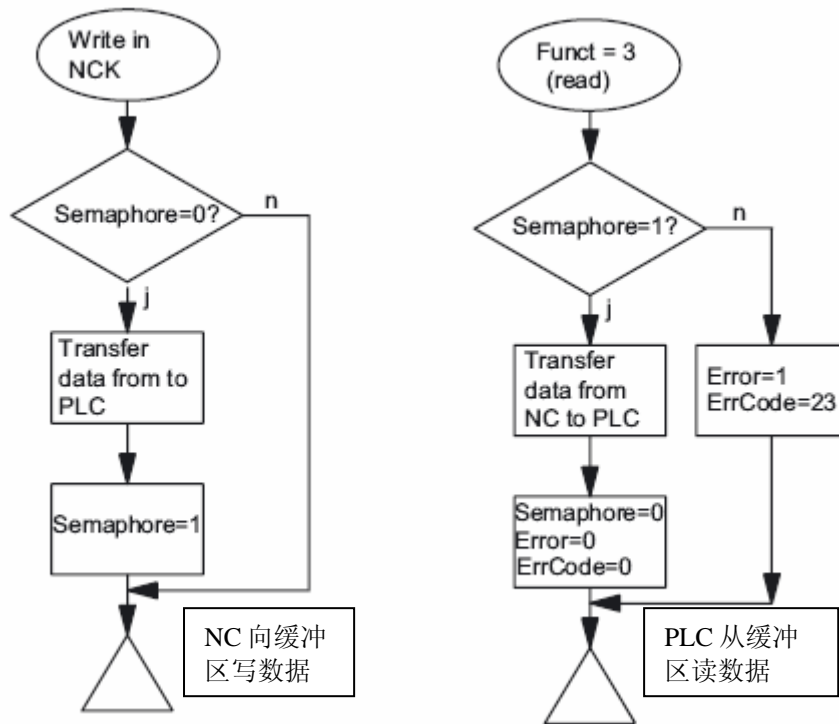
CALL FC21
Enable:=TRUE
Funct:=B#16#3 ←读操作
S7Var:=P#M150.0 BYTE 10 ←MB150-MB159
IVAR1:=21 ←偏移为 21
IVAR2:=98 ←信号灯字节为缓冲区分第 99 个字节
Error:=M200.1
Errcode:=MW212

加工程序例:

\$A_DBB[99]=0 ←信号灯=0 写操作
R50=0
AAA:
R[R50]=\$A_DBB[R50]
MSG("R"<<R50<<=" "<<R[R50])
G4F0.3
R50=R50+1
IF R50<10 GOTOB AAA ←循环将 MB100-MB109(通过中间缓冲区)的值读到 R0-R9
STOPRE
MSG("")
\$A_DBB[98]=1
\$A_DBB[21]=R10
G4F0.5
M02

例 2:

PLC-侧	DUAL-PORT-RAM		NC-侧
写信号灯	DPR_WRITE	Byte 0	
读信号灯	DPR_READ	Byte 1	
		Byte 2	
		Byte 3	
MB100, MB101, MB102, MB103	⇐	Byte 4	R1
	⇐	Byte 5	
	⇐	Byte 6	
	⇐	Byte 7	
MB 104,MB105, MB 106, MB107	⇐	Byte 8	R2
	⇐	Byte 9	
	⇐	Byte 10	
	⇐	Byte 11	
MB108, Mb109, MB110, MB111	⇐	Byte 12	R3
	⇐	Byte 13	
	⇐	Byte 14	
	⇐	Byte 15	
MB112,MB113, MB114,MB115	⇒	Byte 16	R4
	⇒	Byte 17	
	⇒	Byte 18	
	⇒	Byte 19	
MB116, MB117,MB118, MB119	⇒	Byte 20	R5
	⇒	Byte 21	
	⇒	Byte 22	
	⇒	Byte 23	
MB120, MB121, MB122, MB123	⇒	Byte 24	R6
	⇒	Byte 25	
	⇒	Byte 26	
	⇒	Byte 27	
		Byte 28	
		Byte 29	
		
		
	ENDE DPR	BYTE 1023	



```

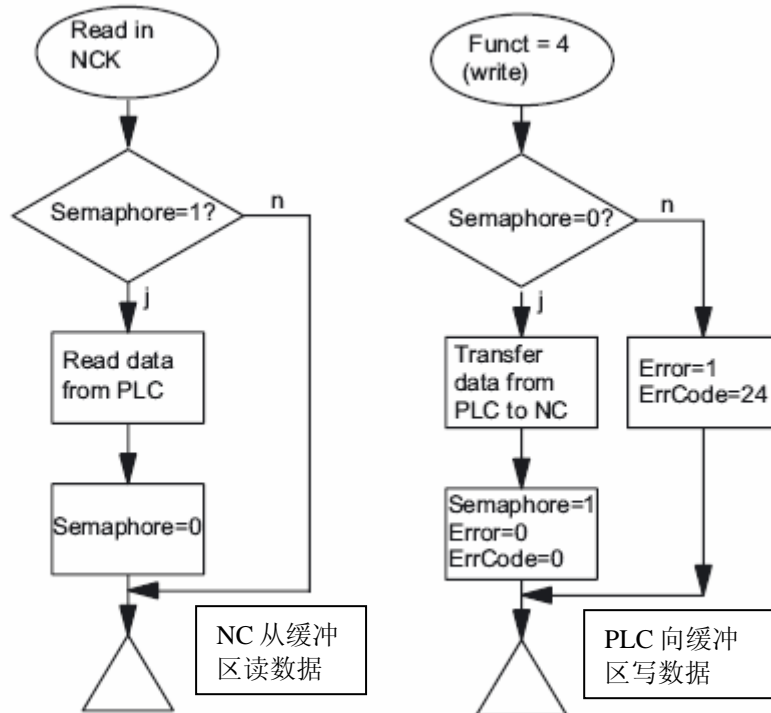
%_N_FC_21_WRITE_MPF
; $PATH=/_N_MPF_DIR
N10 MSG("NC_WRITE")
N20 if $A_DBB[0]<>0 GOTOF ENDE      ; 信号灯不允许 NC 向缓冲区写数据（即 PLC 正在读取数据还未完成），跳转到 ENDE

N30 $A_DBB[0]=0                  ; Semaphore wird auf Null gesetzt
N40 $A_DBR[4]=R1                 ; 把 R1 的值写到缓冲区（byte4-7）
STOPRE
N50 $A_DBR[8]=R2                 ; 把 R2 的值写到缓冲区（byte8-11）
STOPRE
N60 $A_DBR[12]=R3               ; 把 R3 的值写到缓冲区（byte12-15）
N70 STOPRE
N80 $A_DBB[0]=1                 ; 把信号灯 1 改为允许 PLC 读数据
N90 ENDE:                       ; Sprungmarke
N100 M30

CALL FC21
Enable:      =TRUE              ; FC21 启动条件
Funct:       =B#16#3            ; 3 = PLC 读缓冲区的数据
S7Var:       =P#M100.0 DWORD 3 ; 数据目的地，调用 FC21 后，PLC 会把缓冲区中
                                Byte4 开始的 3 个双字读出来并存放
                                到
                                MD100(MB100,MB101,MB102,MB103)
                                MD104(MB104,MB105,MB106,MB107)
                                MD108(MB108,MB109,MB110,MB111)

IVar1:       =4                 ; 偏移 4
Ivar2:       =0                 ; PLC 读数据的信号灯
Error:       =M79.4             ; 调用 FC21 错误指示位
Errorcode:   =MW77              ; 如出错，错误代码存放到 MW77

```



%_N_FC21_NCREAD_MPF

;\$PATH=/_N_MPF_DIR

N10 MSG("NC_READ")

N20 if \$A_DBB[1]==0 GOTOF ENDE

; 信号灯不允许 NC 读缓冲区数据（即 PLC 正在向缓冲区写数据还未完成），跳转到 ENDE

N40R4= \$A_DBR[16]

; 将缓冲区 byte16-19 的值读入到 R4

STOPRE

N50R5= \$A_DBR[20]

; 将缓冲区 byte20-23 的值读入到 R5

STOPRE

N60 R6= \$A_DBR[24]

; 将缓冲区 byte24-27 的值读入到 R6

N70 STOPRE

N80 \$A_DBB[1]=0

; 把信号灯改为 0 允许 PLC 写数据

N90 ENDE:

N100 M30

CALL FC21

Enable: =TRUE

; FC21 启动条件

Funct: =B#16#4

; 4 = PLC 向缓冲区写数据

S7Var: =P#M112.0 DWORD 3

; 数据源，调用 FC21 后，会把

MD112(MB112,MB113,MB114,MB115)

MD116(MB116,MB117,MB118,MB119)

MD120(MB120,MB121,MB122,MB123)

这 3 个双字写到缓冲区 Byte16-Byte27

IVar1: =16

; 偏移 16

Ivar2: =1

; PLC 写数据的信号灯

Error: =M79.5

; 调用 FC21 错误指示位

Errorcode: =MW88

; 如出错，错误代码存放到 MW88