

日常使用中偶尔会出现某个输出/输入点因某些原因需要定义到别的点位，而程序也因为各种原因需要修改使用的输入输出点，

本例程可用于程序提前布置好映射环境后，在后期维护中通过触摸屏 / 上位机修改偏移量来达成无需修改程序即可改动对应输入/输出点的目的。

内集成了数字量、模拟量输入输出的自定义目标寄存器功能：

	状态列	符号	地址	注释
1		输入输出映射	FC8	
2		AQ_switch_PLUS	FC7	模拟量输出自定义点功能块，自定义起始点
3		AI_switch_PLUS	FC6	模拟量输入自定义点功能块，自定义起始点
4		DQ_switch_PLUS	FC5	数字量输出自定义点功能块，自定义起始点
5		DI_switch_PLUS	FC4	数字量输入自定义点功能块，自定义起始点
6		AQ_switch	FC3	模拟量输出自定义点功能块
7		AI_switch	FC2	模拟量输入自定义点功能块
8		DQ_switch	FC1	数字量输出自定义点功能块
9		DI_switch	FC0	数字量输入自定义点功能块
10		主程序	OB1	

使用时只需在程序调用通过中间变量映射即可，其中后缀带 ‘\_PLUS’ 系列功能块多一个引脚可自定义偏移量为 0 时指向的位/字，不限制于 I、Q、AI、AQ，可指向 V、M 等能获取到指针的寄存器。

DI_switch .....	2
DI_switch_PLUS .....	2
DQ_switch .....	3
DQ_switch_PLUS .....	3
AI_switch .....	4
AI_switch_PLUS .....	4
AQ_switch .....	5
AQ_switch_PLUS .....	5
应用例程 .....	6
应用开发时输入部署例程 .....	7
应用开发时输出部署例程 .....	8
实际应用例程 .....	9

**DI\_switch:** 数字量输入自定义点功能块



引脚名称	引脚类型	数据类型	注释
DI_Offset	IN	Word	偏移量，基于 I0.0 为 0，I0.1 为 1.....I1.0 为 8，按位计算
DI_OUT	OUT	BOOL	偏移量指向点的当前状态

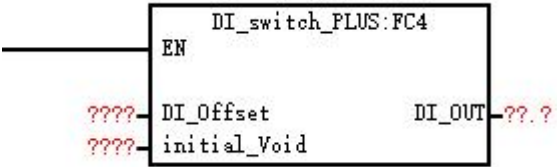
调用后将 ‘DI\_Offset’ 偏移量指定的 DI 输入状态每周期更新到 ‘DI\_OUT’ 中，用户使用时直接将 ‘DI\_OUT’ 作为输入在程序中应用。

偏移量规则：

$DI\_Offset = [实际 I 点字节数] * 8 + 位数$

例如： ‘I3.6’ = 3\*8 + 6 = 30。

**DI\_switch\_PLUS:** 数字量输入自定义点功能块，自定义起始点



引脚名称	引脚类型	数据类型	注释
DI_Offset	IN	Word	偏移量，基于 ‘Initial_Void’ 的 bit0 为 0， ‘Initial_Void’ 的 bit1 为 1 ..... ‘Initial_Void’ + 1 字节的 bit0 为 8，按位计算
Initial_Void	IN	Dowrd (指针)	指向起始地址的指针，偏移量为 0 时的指向寄存器 以字节为单位。
DI_OUT	OUT	BOOL	偏移量指向点的当前状态

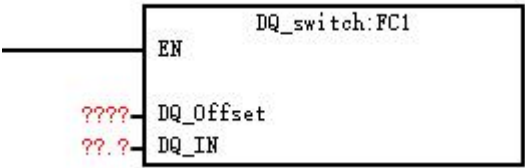
偏移量规则：

$DI\_Offset = ([实际 I 点字节数] - 'Initial\_Void' 字节数) * 8 + 位数$

例如（Initial\_Void = &IB1）： ‘I3.6’ = (3-1)\*8 + 6 = 22。



**DQ\_switch:** 数字量输出自定义点功能块



引脚名称	引脚类型	数据类型	注释
DQ_Offset	IN	Word	偏移量，Q0.0 为 0，Q0.1 为 1 ..... Q1.0 为 8，按位计算
DQ_OUT	IN	BOOL	要输出到偏移量指向 Q 点的状态，为 0 时指定 Q 点断开，为 1 时指定 Q 点接通。

调用后将 ‘DQ\_IN’ 的状态更新到 ‘DQ\_Offset’ 偏移量指定的 DQ 中，用户使用时直接控制 ‘DQ\_IN’ 的状态。

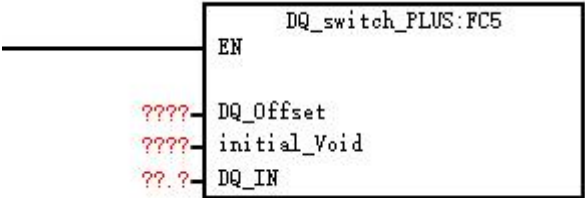
程序只修改当前‘DQ\_Offset’指定的 Q 点，修改偏移量时原先偏移量指定的 Q 点将保持最后状态直到下一次掉电上电或重新绑定到某偏移量 或者通过替他方式修改该位状态为止。

偏移量规则：

$$DQ\_Offset = [\text{实际 Q 点字节数}] * 8 + \text{位数}$$

例如： ‘Q3.6’ = 3\*8 + 6 = 30。

**DQ\_switch\_PLUS:** 数字量输出自定义点功能块，自定义起始点



引脚名称	引脚类型	数据类型	注释
DQ_Offset	IN	Word	偏移量，基于 ‘Initial_Void’ 的 bit0 为 0，‘Initial_Void’ 的 bit1 为 1 ..... ‘Initial_Void’ + 1 字节的 bit0 为 8，按位计算
Initial_Void	IN	Dowrd (指针)	指向起始地址的指针，偏移量为 0 时的指向寄存器以字节为单位。
DQ_OUT	IN	BOOL	要输出到偏移量指向 Q 点的状态，为 0 时指定 Q 点断开，为 1 时指定 Q 点接通。

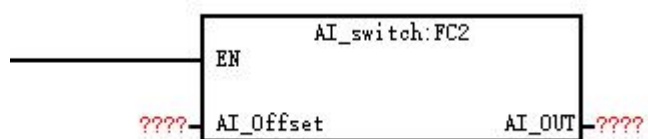
偏移量规则：

$$DI\_Offset = ([\text{实际 Q 点字节数}] - [\text{‘Initial_Void’ 字节数}]) * 8 + \text{位数}$$

例如 (Initial\_Void = &QB1): ‘Q3.6’ = (3-1)\*8 + 6 = 22。



**AI\_switch:** 模拟量输入自定义点功能块



引脚名称	引脚类型	数据类型	注释
AI_Offset	IN	Word	偏移量，基于 AIW0 为 0，AIW2 为 1.....AIW10 为 5，按字计算
AI_OUT	OUT	Word	偏移量指向 AIW 的当前状态

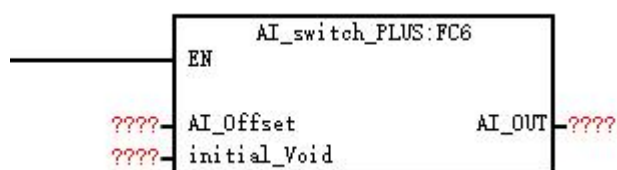
调用后会将 ‘AI\_Offset’ 偏移量指定的 AIW 输入数值每周期更新到 ‘AI\_OUT’ 中，用户使用时直接将 ‘AI\_OUT’ 作为模拟量输入在程序中应用。

偏移量规则：

$AI\_Offset = [实际用的\ aiw] / 2$

例如： ‘AIW24’ =  $24/2 = 12$ 。

**AI\_switch\_PLUS:** 模拟量输入自定义点功能块，自定义起始点



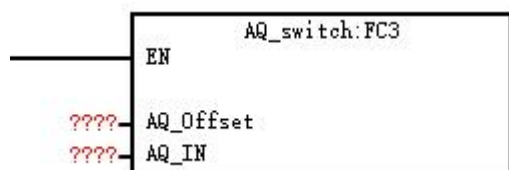
引脚名称	引脚类型	数据类型	注释
AI_Offset	IN	Word	偏移量，基于 ‘Initial_Void’ 指定寄存器为 0，‘Initial_Void’ 的下一个字为 1，按字计算
Initial_Void	IN	Dowrd (指针)	指向起始地址的指针，偏移量为 0 时的指向寄存器以字节为单位。
AI_OUT	OUT	Word	偏移量指向 AIW 的当前状态

偏移量规则：

$AI\_Offset = ([实际\ AIW] - 'Initial\_Void') / 2$

例如（Initial\_Void = &AIW6）： ‘AIW24’ =  $(24 - 6)/2 = 9$ 。

### AQ\_switch: 模拟量输出自定义点功能块



引脚名称	引脚类型	数据类型	注释
AQ_Offset	IN	Word	偏移量, 基于 AQW0 为 0, AQW2 为 1.....AQW10 为 5, 按字计算
AQ_IN	IN	Word	要输出到偏移量指向 AQW 的数值

调用后将 ‘AQ\_IN’ 的数值输出到 ‘AQ\_Offset’ 偏移量指定的 AQW 中, 用户使用时直接将 ‘AQ\_IN’ 作为模拟量输出在程序中应用。

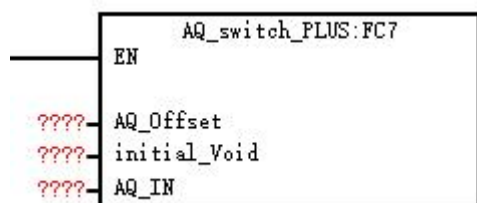
程序只修改当前 ‘AQ\_Offset’ 指定的寄存器, 修改偏移量时原先偏移量指定的寄存器将保持最后状态直到下一次掉电上电 或 重新绑定到某偏移量 或 者通过替他方式修改该寄存器状态为止。

偏移量规则:

$$AQ\_Offset = [\text{实际用的 AQW}] / 2$$

例如: ‘AQW24’ = 24/2 = 12。

### AQ\_switch\_PLUS: 模拟量输出自定义点功能块, 自定义起始点



引脚名称	引脚类型	数据类型	注释
AQ_Offset	IN	Word	偏移量, 基于 AQW0 为 0, AQW2 为 1.....AQW10 为 5, 按字计算
Initial_Void	IN	Dowrd (指针)	指向起始地址的指针, 偏移量为 0 时的指向寄存器以字节为单位。
AQ_IN	IN	Word	要输出到偏移量指向 AQW 的数值

偏移量规则:

$$AQ\_Offset = ([\text{实际 AQW}] - \text{'Initial\_Void'}) / 2$$

例如 (Initial\_Void = &AQW6): ‘AIW24’ = (24 - 6)/2 = 9。

## 简单应用例程

### DI 与 DQ 的自定义



LD SMO.0

CALL DI\_switch:FC0, VW2000, M20.0

CALL DQ\_switch:FC1, VW2002, M20.0

该例程检查实际的 I 点状态并实时输出到对应 Q 点

### AI 与 AQ 的自定义



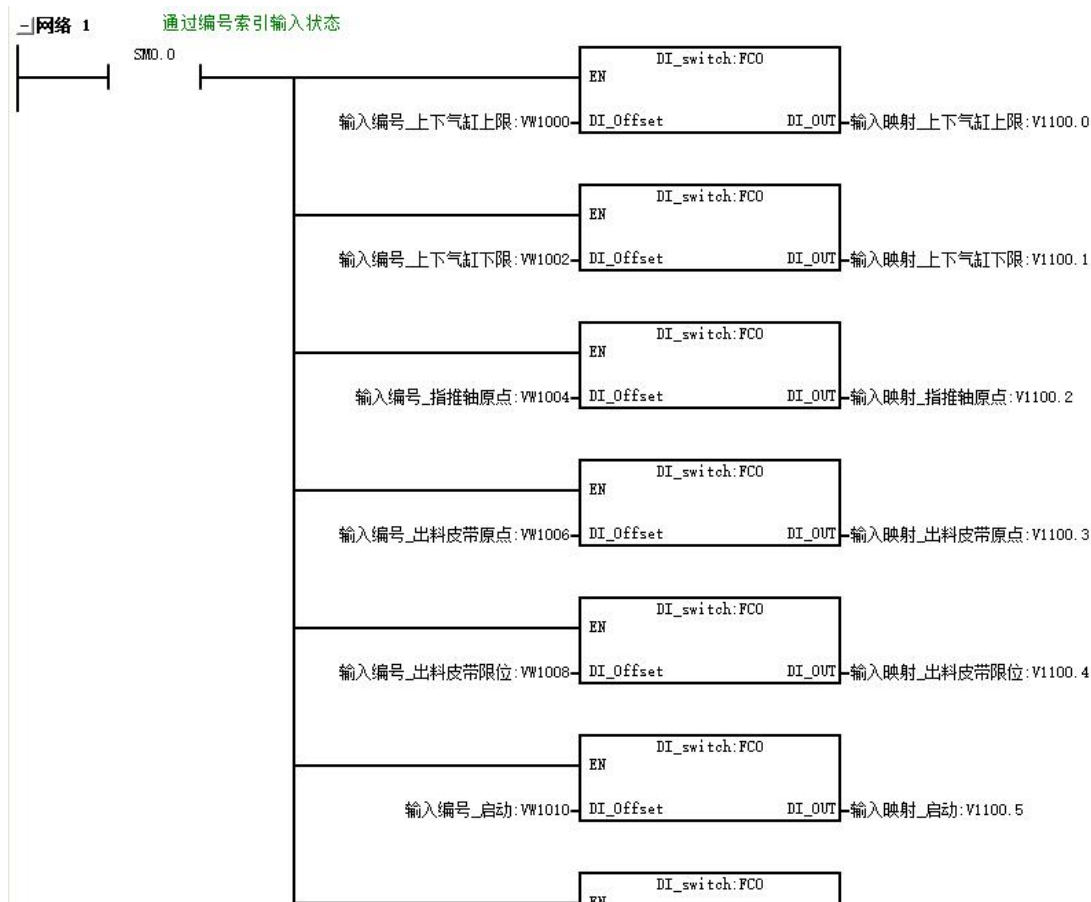
LD SMO.0

CALL AI\_switch:FC2, VW2000, MW20

CALL AQ\_switch:FC3, VW2002, MW20

该例程检查实际的 AIW 寄存器并输出到对应 AQW 中

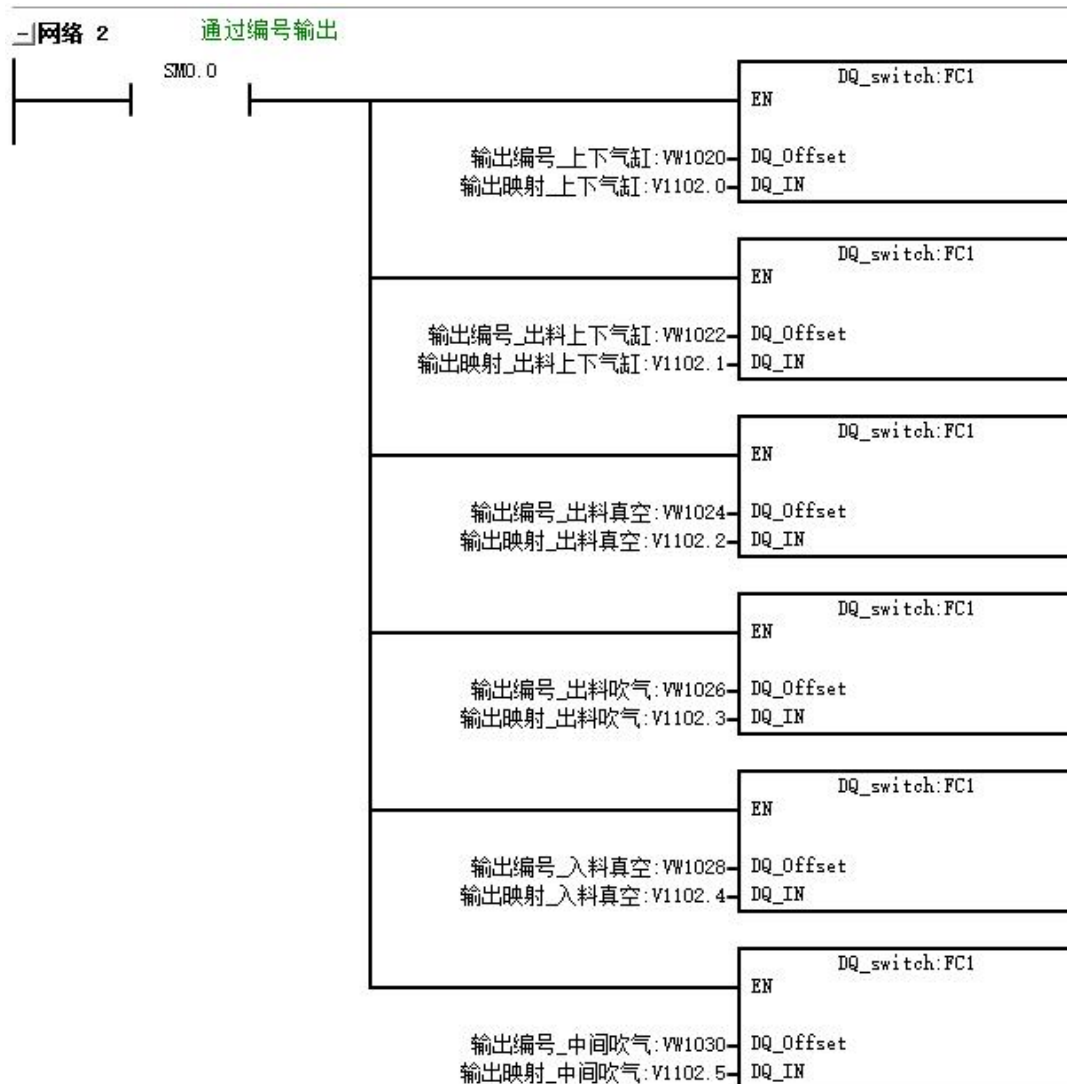
## 应用开发时输入部署例程：



部署完毕后程序中直接使用 ‘输入映射\_.....’ 变量当成 I 点使用，‘输入编号\_.....’ 可在触摸屏/上位机中更改，以此达成坏点直接换点的目的。

例如，VW1000 设置为 0，则将 I0.0 的状态绑定到 V1100.0 中，当 PLC 的 i0.0 接通，v1100.0 接通；设置为 1 则绑定到 i0.1

## 应用开发时输出部署例程：



部署完毕后程序中直接使用 ‘输出映射\_.....’ 变量当成 Q 点使用，‘输出编号\_.....’ 变量可在触摸屏/上位机中更改，以此达成坏点直接换点的目的。

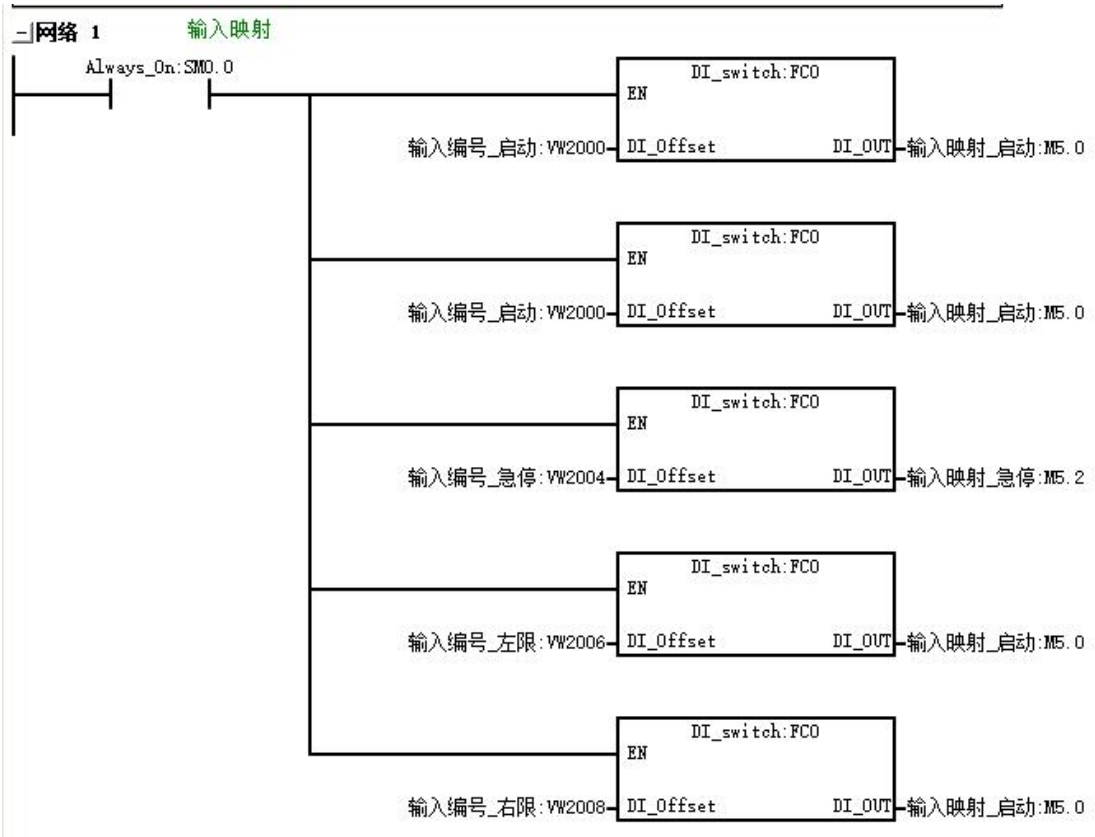
例如，VW1020 设置为 0，则将 V1101.0 的状态绑定到 Q0.0 中，当 PLC 的 i0.0 接通，v1100.0 接通；设置为 1 则绑定到 i0.1



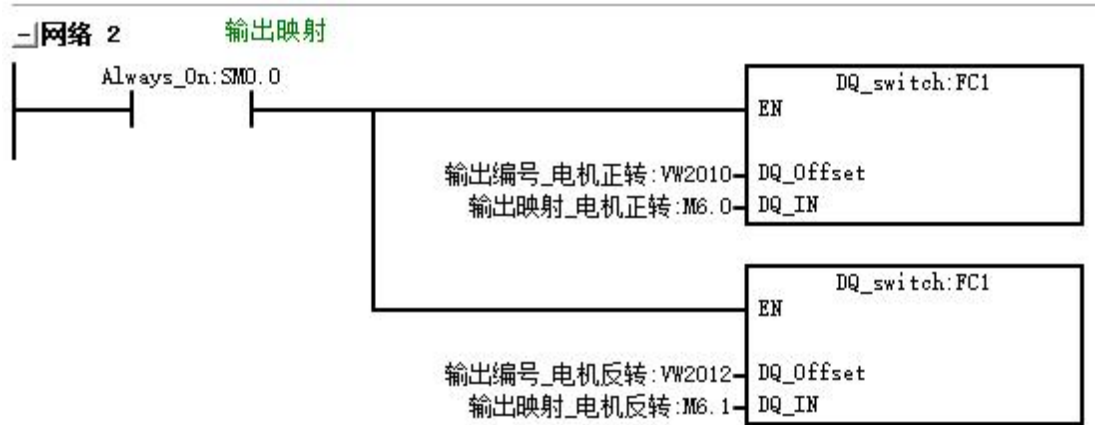
简单应用例程

输入		输出	
启动	I0.0	电机正转	Q0.0
停止	I0.1	电机反转	Q1.1
急停	I0.2		
左限	I1.0		
右限	I1.1		

程序搭建：



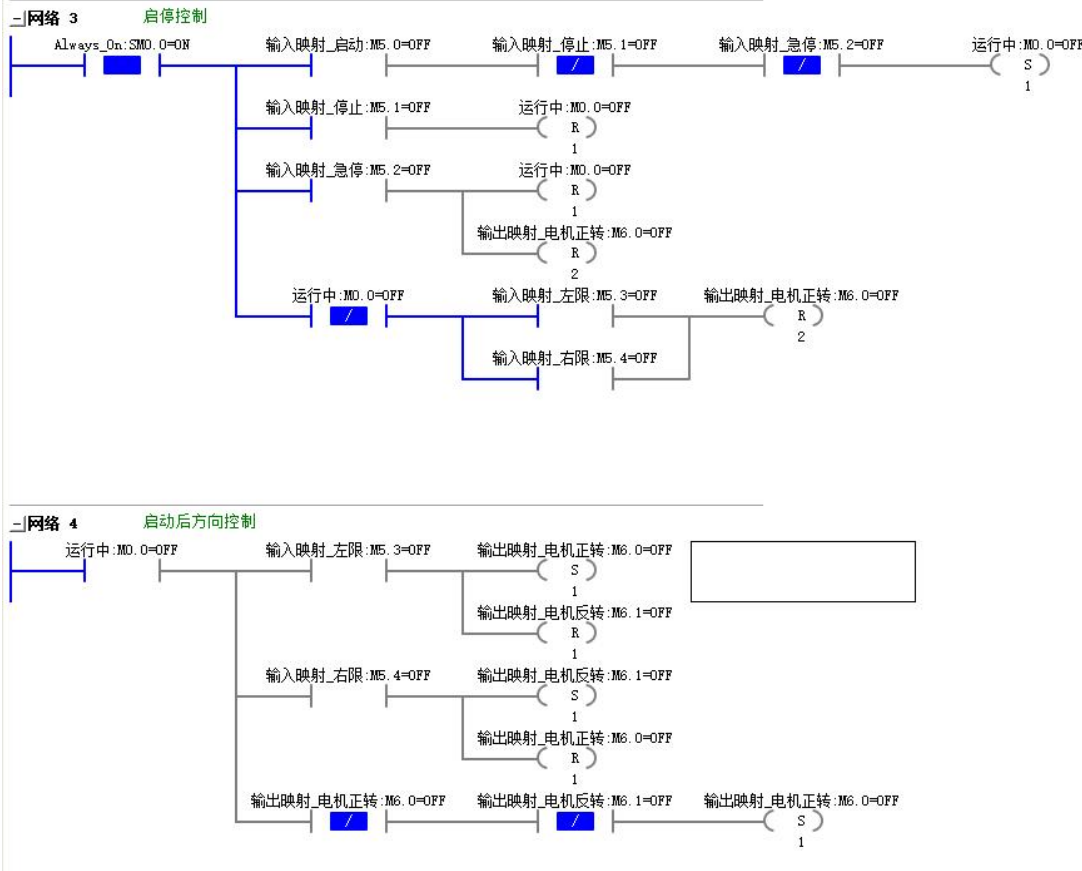
(输入映射)



(输出映射)

简单应用：

启动后电机在限位内往复运动，停止则运动到限位后停止，急停立即停止



触摸屏搭建：

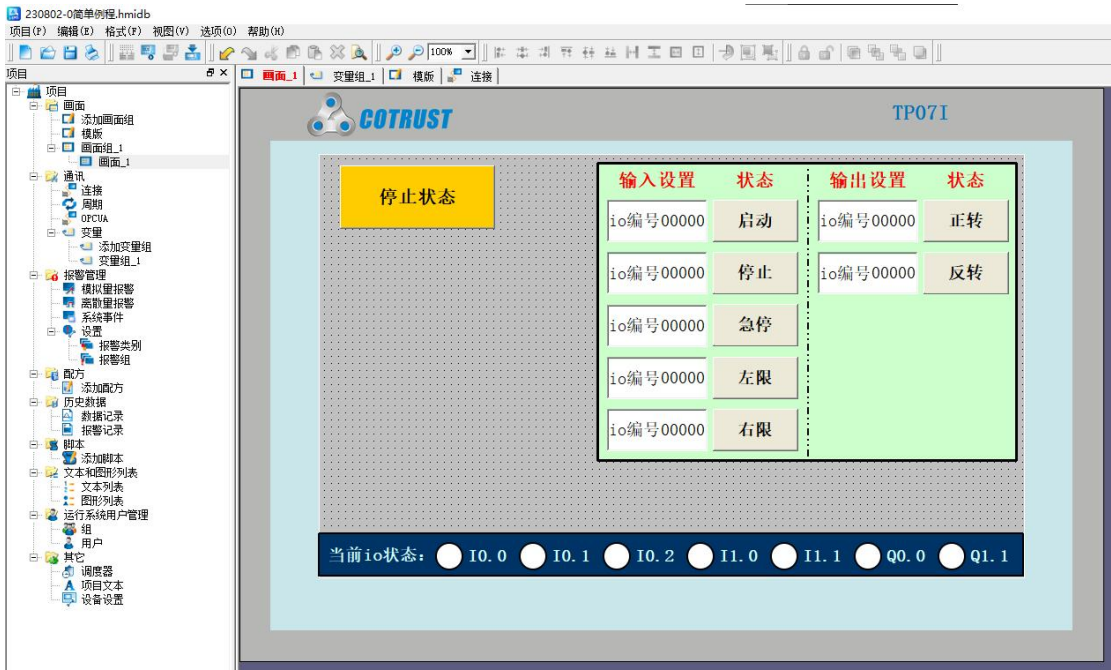
230802-0简单应用\_hmidx

项目(F) 编辑(E) 视图(V) 选项(O) 帮助(H)

项目 画面\_1 变量组\_1

Td	名称	连接	数据类型	长度	数组计数	地址	采集周期	数据记录	记录周期	记录采集模式	起始值	注释
1	1	输入映射_启动	连接_1	Bool	0	1	M 5.0	100ms	<未定义>	1s	循环连续	
2	2	输入映射_停止	连接_1	Bool	0	1	M 5.1	100ms	<未定义>	1s	循环连续	
3	3	输入映射_急停	连接_1	Bool	0	1	M 5.2	100ms	<未定义>	1s	循环连续	
4	4	输入映射_左限	连接_1	Bool	0	1	M 5.3	100ms	<未定义>	1s	循环连续	
5	5	输入映射_右限	连接_1	Bool	0	1	M 5.4	100ms	<未定义>	1s	循环连续	
6	6	输入编号_启动	连接_1	Int	2	1	VW 2000	100ms	<未定义>	1s	循环连续	
7	7	输入编号_停止	连接_1	Int	2	1	VW 2002	100ms	<未定义>	1s	循环连续	
8	8	输入编号_急停	连接_1	Int	2	1	VW 2004	100ms	<未定义>	1s	循环连续	
9	9	输入编号_左限	连接_1	Int	2	1	VW 2006	100ms	<未定义>	1s	循环连续	
10	10	输入编号_右限	连接_1	Int	2	1	VW 2008	100ms	<未定义>	1s	循环连续	
11	11	输出映射_电机正转	连接_1	Bool	0	1	M 6.0	100ms	<未定义>	1s	循环连续	
12	12	输出映射_电机反转	连接_1	Bool	0	1	M 6.1	100ms	<未定义>	1s	循环连续	
13	13	输出编号_电机正转	连接_1	Int	2	1	VW 2010	100ms	<未定义>	1s	循环连续	
14	14	输出编号_电机反转	连接_1	Int	2	1	VW 2012	100ms	<未定义>	1s	循环连续	
15	15	运行中	连接_1	Bool	0	1	M 0.0	100ms	<未定义>	1s	循环连续	
16	16	Q0.0	连接_1	Bool	0	1	Q 0.0	100ms	<未定义>	1s	循环连续	
17	17	Q1.1	连接_1	Bool	0	1	Q 1.1	100ms	<未定义>	1s	循环连续	
18	18	I0.0	连接_1	Bool	0	1	I 0.0	100ms	<未定义>	1s	循环连续	
19	19	I0.1	连接_1	Bool	0	1	I 0.1	100ms	<未定义>	1s	循环连续	
20	20	I0.2	连接_1	Bool	0	1	I 0.2	100ms	<未定义>	1s	循环连续	
21	22	I1.0	连接_1	Bool	0	1	I 1.0	100ms	<未定义>	1s	循环连续	
22	23	I1.1	连接_1	Bool	0	1	I 1.1	100ms	<未定义>	1s	循环连续	

(变量)



(画面组态)



(编号 8 的 io 接通，映射到 PLC 中用于左限位)



(模拟出现 io 损坏，将左限修改到 1 号点)



(多个 io 配置、触发不互相影响)