



H1U/H2U-XP系列可编程逻辑 控制器指令及编程手册



前 言

H1U/H2U系列PLC是汇川技术研发的高性价比控制产品，指令丰富，高速信号处理能力强，运算速度快，允许的用户程序容量可达16K步，且不需外扩存储设备。

控制器标配三个独立通讯口，可扩展为四个，方便现场接线；通讯端口支持多种通讯协议，包括MODBUS主站、从站协议，尤其方便了与变频器等设备的联机控制；提供了严密的用户程序保密功能，子程序单独加密功能，方便用户特有控制工艺的知识产权保护。支持USB下载，支持CANlink组网通讯。

控制器提供了多种编程语言，用户可选用梯形图、指令表、步进梯形图、SFC顺序功能图等编程方法。指令系统为广大工程技术人员所熟悉，而本公司提供的AutoShop编程环境，更是融合了众多PLC编程环境的优点，丰富的在线帮助信息，使得编程时无需查找说明资料，方便易用。

本《汇川小型PLC指令及编程手册》的知识产权属于汇川技术所有，我公司会根据情况不断更新升级，恕不另行通知，欢迎用户读者随时访问我公司网站，下载最新版本的手册与资料。

我们热忱欢迎读者以多种形式咨询和交流使用方法，反馈手册中的错误和遗漏。

公司网页：www.inovance.cn

信息交流：plc@inovance.cn

目 录

前 言.....	1
手册版本信息.....	6
指令详解索引	7
简单逻辑指令	7
手册快查引导.....	11
第一章 梯形图及梯形图程序.....	14
1.1 梯形图的编程特点.....	14
1.2 梯形图编程时使用的元件符号.....	14
1.3 PLC的执行原理	15
1.4 PLC数值的基本知识	16
第二章 H1U/H2U 系列PLC的使用方法.....	22
2.1 使用PLC的软件硬件需求.....	22
2.2 编程与用户程序下载.....	22
2.3 与HMI的配合使用	22
第三章 软元件说明.....	24
3.1 输入继电器X.....	24
3.2 输出继电器Y.....	24
3.3 辅助继电器M	25
3.4 状态继电器S	26
3.5 计时器T	26
3.6 计数器C.....	27
3.7 寄存器D.....	31
3.8 子程序与中断指针P、I.....	32
3.9 常数K、H.....	35
3.10 控制器软元件规格.....	35
第四章 指 令.....	38
4.1 基本指令	38
4.2 步进顺控指令.....	42
4.2.1 STL编程指令	42
4.2.2 SFC顺序功能图编程.....	43
4.2.3 SFC编程实例.....	49

4.3 应用指令	58
4.3.1 程序流程 (00~09)	60
4.3.2 传送与比较 (10~19)	69
4.3.3 四则逻辑运算 (20~29)	79
4.3.4 循环移位 (30~39)	89
4.3.5 数据处理 (40~49)	96
4.3.6 高速处理 (50~59)	106
4.3.7 方便指令 (60~69)	121
4.3.8 外围设备I/O (70~79)	130
4.3.9 外围SER设备 (80~93)	141
4.3.10 浮点数 (110~147)	158
4.3.11 定位控制 (150~159)	184
4.3.12 时钟运算 (160~169)	196
4.3.13 外围设备 (170~177)	203
4.3.14 接点比较 (224~246)	206
第五章 通信功能	212
5.1 概要	212
5.1.1 硬件及通信连线	212
5.1.2 H _{2U} 通信协议切换逻辑说明	213
5.2 HMI监控协议	217
5.3 并联协议 (1:1)	217
5.4 N:N协议	218
5.5 N:N增强协议	221
5.6 计算机链接协议	223
5.7 MODBUS协议说明	230
5.7.1 MODBUS主站通讯应用	230
5.7.2 MODBUS从站通讯应用	234
5.7.3 多台PLC之间采用MODBUS协议通讯的编程方法	238
5.7.4 MODBUS配置与编程方法	240
5.7.5 汇川QLink协议与编程方法	243
5.7.6 PLC与MD380系列变频器的通讯编程方法	245
5.7.7 MD380配合PLC应用的功能码设置	245
5.7.8 PLC修改MD380功能码的编程	250
5.9 CAN通讯指令说明	252
5.9.1 CANlink3.0应用简介	252
5.9.2 主从架构	253

5.9.3 心跳机制	255
5.9.4 掉站信息	255
5.9.5 CANlink3.0的使用	256
5.9.6 软件配置CANlink3.0主从站的通信配置方法	257
5.9.7 CAN网络配置表的填写方法	259
5.9.8 CANlink常见故障与排除	270
第六章 扩展模块及扩展卡的使用方法	272
6.1 扩展模块概要	272
6.1.1 命名规则	272
6.1.2 基本型号	272
6.1.3 H _{2U} -CAN-BD (或H _{1U} -CAN-BD) 扩展卡	273
6.1.4 扩展模块地址编号	274
6.1.5 模块的安装尺寸及用户接口表	275
6.2 数字量输入/输出扩展模块	276
6.2.1 数字量输入扩展模块	276
6.2.2 数字量输出扩展模块	277
6.2.3 数字量输入输出扩展模块	278
6.3 模拟量输入模块	280
6.4 模拟量输出模块	285
6.5 模拟量混合 (输入/输出) 模块	289
6.5.1 H _{2U} -4AM 2通道模拟电流输入/2通道模拟电压电流输出混合扩展模块	289
6.5.2 H _{2U} -6AM 4通道模拟电流输入/2通道模拟电压电流输出混合扩展模块	295
6.5.3 H _{2U} -6CM 4通道模拟电压输入/2通道模拟电压电流输出混合扩展模块	301
6.6 温度控制模块	308
6.7 扩展卡使用说明	319
6.7.1 H _{2U} 系列6A/6B扩展卡使用说明	319
6.7.2 H _{2U} 系列以太网通讯扩展卡使用说明	321
6.7.3 H _{1U} / H _{2U} 系列通讯扩展卡使用说明	325
6.7.4 H _{1U} 系列4A扩展卡使用说明	327
6.8 差分转集电极输出模块使用说明	329
第七章 特色功能	332
7.1 禁止上载	332
7.1.1 禁止上载功能介绍	332
7.1.2 PLC加密设置	332
7.2 工程授权保护使用 (24147及以上支持)	334

7.2.1 功能简介	334
7.2.2 工程授权保护	334
7.3 子程序加密/解密	336
7.4 PLC标识符	336
7.4.1 PLC标识符的设置界面	336
7.5 固件升级步骤指导	338
7.5.1 准备	338
7.5.2 升级操作	338
第八章 附 录	342
8.1 系统特殊软元件	342
8.2 系统错误记录功能	351
8.2.1 错误代码及内容	351
8.2.2 PLC运行时间/异常记录	354
8.2.3 使用Autoshop进行故障诊断和处理	355
8.3 错误代码寄存器	356
8.4 H2U系列MTQ型与MT型的软件差别	356
8.5 H2U系列MTP型软件补充说明	358
8.5.1 DRVI、DRVA特殊功能的描述	358
8.5.2 ZRN指令的描述	360
8.6 H2U系列MTQ/MTP机型不支持的指令一览	361
8.7 H2U系列高速处理指令的增强功能说明	362
8.8 程序流程控制指令的相互关系	366
8.9 部分特殊继电器和寄存器功能说明	368

手册版本信息

版本	发布时间	修订说明
V0.8	2009-03-18	首次发布
V0.88	2009-04-18	修订系统变量说明 增加PLC原理和基本知识介绍 增加MTQ版本的使用说明
V1.00	2009-10-09	增加IST指令说明 修订附录5.5节扩展卡的编程说明 增加附录5.5节的高速处理XP功能说明 增加附录5.9程序流程控制指令的相互关系 增加附录5.10部分特殊继电器和寄存器功能说明
V1.01	2010-7-12	对个别指令加入更详细的描述 增加对H _{1U} 的描述 增加附录5.11节扩展模块的详细使用 增加附录5.12 H _{1U} /H _{2U} 系列PLC通信说明 增加5.8 CAN通讯指令说明 增加附录5.14 PLC无线远程监控功能说明
V2.0	2011-6-1	增加XP型内容说明 增加4.1.3 SFC编程实例 附录5.5、5.6、5.11（扩展卡及扩展模块的内容）修改为第六章（扩展模块及扩展卡的使用方法） 附录5.12~5.14通信相关内容修改为第五章（通信功能） 增加5.7 PLC与MD320系列变频器的通讯编程方法 XP型PLC增加浮点数指令12条：EMOV、EXP、LOGE、LOG10、ASIN、ACOS、ATAN、RAD、DEG、SINH、COSH、TANH；定位指令1条：DSZR。
V2.1	2013-11	3.8节增加加密子程序说明 5.1节增加MiniUSB下载线的描述 5.5节增加N:N增强协议的说明 5.7.4节增加MODBUS配置功能的说明 5.7.5节增加Qlink通讯配置方法 5.8节增加PLC与MD380变频器通讯的内容； 5.9节增加CAN通讯举例与波特率、线缆要求的对应关系； 5.9.5节增加CANlink3.0通讯的介绍 5.11节增加PLC 3G无线扩展模块的使用 6.2.1节和6.2.2节新增部分型号数字量模块的使用说明； 6.2.3节新增数字量输入输出模块的使用说明； 6.7.4节新增H1U-4A-BD扩展卡的使用说明； 6.7.2节增加以太网扩展卡的调试和使用说明 6.8节新增对2DC模块的使用说明； 7.1节增加新的特殊用软元件的说明 7.2节增加系统错误记录功能的说明 7.2节新增后台故障诊断的调试说明； 软元件规格变更和增加 对上一版本的勘误 更新附表产品型号订货索引；
C02	2017-12	指令索引表中，删除FunNO信息；机型统一修改为“H1U-XP、H2U-XP”； 3.6节根据最新软件，优化M8166/M8169的定义； 第4章 指令：删除所有指令描述中的适用机型表信息； 4.3.6 高速处理相关指令中，删除Y03和Y04相关信息； PLSY指令增加频率为0的特殊处理说明； 第5章 通信：注销对CAN1.0协议的支持； 5.7小节关于Modbus从站说明，增加对02和04功能码的支持； 5.7.4小节下增加“配置加密/解密”功能说明； 删除5.10小节； 重新组织5.9小节； 第6章：分开描述4PT/4TC，新增4TC内容；删除H _{2U} -6AMR/6CMR/4AMR/2ADR/2DAR/0808ETDR等信息； 增加第7章 特色功能，含PLC标识符、禁止上载、固件升级、程序授权码等内容； 软元件规格变更和增加 对上一版本的勘误 删除附表产品型号订货索引

指令详解索引

简单逻辑指令

指令助记符	功能	适用机型【注】		页码
		H1U-XP	H2U-XP	
LD	加载常开接点	✓	✓	38
LDI	加载常闭接点	✓	✓	38
LDP	取脉冲上升沿	✓	✓	38
LDF	取脉冲下降沿	✓	✓	38
AND	串联常开接点	✓	✓	38
ANI	串联常闭接点	✓	✓	38
ANB	串联回路方块	✓	✓	39
ANDP	与脉冲上升沿检测串行连接	✓	✓	38
ANDF	与脉冲下降沿检测串行连接	✓	✓	38
OR	并联常开接点	✓	✓	39
ORI	并联常闭接点	✓	✓	39
ORB	并联回路方块	✓	✓	39
ORP	或脉冲上升沿检测并行连接	✓	✓	39
ORF	或脉冲(F)下降沿检测并行连接	✓	✓	39
OUT	驱动线圈	✓	✓	40
SET	置位动作保存线圈	✓	✓	40
RST	接点或缓寄存器清除	✓	✓	40
PLS	脉冲上升沿检测线圈	✓	✓	40
PLF	脉冲下降沿检测线圈	✓	✓	40
MC	主控公用串行接点用线圈指令	✓	✓	41
MCR	主控复位公用串行接点解除指令	✓	✓	41
MPS	存入堆栈	✓	✓	39
MRD	读出堆栈	✓	✓	39
MPP	读出堆栈	✓	✓	39
NOP	无动作	✓	✓	41
INV	运算结果取反	✓	✓	41
END	程序结束	✓	✓	41
P	指针	✓	✓	41
I	中断插入指针	✓	✓	41

【注】：

H1S-XP机型所支持的指令与H1U-XP所支持的指令一致；

H2E-XP机型所支持的指令与H2U-XP所支持的指令一致。

应用指令（以FNC NO为序）

分类	指令助记符	功能	适用机型【注】		页码	
			H1U-XP	H2U-XP		
程序流程	CJ	条件跳转	✓	✓	60	
	CALL	子程序调用	✓	✓	60	
	SRET	子程序完毕	✓	✓	61	
	IRET	中断返回	✓	✓	62	
	EI	中断允许	✓	✓	62	
	DI	中断禁止	✓	✓	62	
	FEND	主程序结束	✓	✓	65	
	WDT	监控定时器	✓	✓	66	
	FOR	循环范围开始	✓	✓	67	
	NEXT	循环范围结束	✓	✓	67	
	传送与比较	CMP	比较	✓	✓	69
		ZCP	区间比较	✓	✓	70
		MOV	传送	✓	✓	71
SMOV		移位传送	✓	✓	72	
CML		倒转传送	✓	✓	73	
BMOV		一并传送	✓	✓	74	
FMOV		多点传送	✓	✓	75	
XCH		交换	✓	✓	76	
BCD		BCD 转换	✓	✓	77	
BIN		BIN 转换	✓	✓	78	
四则逻辑运算		ADD	BIN 加法	✓	✓	79
	SUB	BIN 减法	✓	✓	80	
	MUL	BIN 乘法	✓	✓	81	
	DIV	BIN 除法	✓	✓	82	
	INC	BIN 加 1	✓	✓	83	
	DEC	BIN 减 1	✓	✓	84	
	WAND	逻辑与	✓	✓	85	
	WOR	逻辑或	✓	✓	85	
	WXOR	逻辑异或	✓	✓	85	
	NEG	求补码	✓	✓	87	
	循环移位	ROR	循环右移	✓	✓	89
		ROL	循环左移	✓	✓	89
		RCR	带进位循环右移	✓	✓	90
RCL		带进位循环左移	✓	✓	90	
SFTR		位右移	✓	✓	91	
SFTL		位左移	✓	✓	91	
WSFR		字右移	✓	✓	92	
WSFL		字左移	✓	✓	92	
SFWR		移位写入	✓	✓	94	
SFRD	“先进先出”读出	✓	✓	95		

分类	指令助记符	功能	适用机型【注】		页码
			H1U-XP	H2U-XP	
数据处理	ZRST	批次复位	✓	✓	96
	DECO	译码	✓	✓	97
	ENCO	编码	✓	✓	98
	SUM	ON位数	✓	✓	99
	BON	ON位数判定	✓	✓	100
	MEAN	平均值	✓	✓	101
	ANS	信号报警置位	✓	✓	102
	ANR	信号报警器复位	✓	✓	103
	SQR	BIN 开方	✓	✓	104
	FLT	整数→浮点数转换	✓	✓	105
高速处理	REF	输入输出刷新	✓	✓	106
	REFF	滤波器调整	✓	✓	107
	MTR	矩阵输入	✓	✓	108
	HSCS	比较置位 (高速计数器)	✓	✓	109
	HSCR	比较复位 (高速计数器)	✓	✓	111
	HSZ	比较区间 (高速计数器)	✓	✓	112
	SPD	脉冲密度	✓	✓	115/ 362
	PLSY	脉冲输出	✓	✓	117/ 363
	PWM	脉冲调制	✓	✓	118
	PLSR	带加减速脉冲输出	✓	✓	119
方便指令	SER	数据查找	✓	✓	121
	ABSD	凸轮控制(绝对方式)	✓	✓	122
	INCD	凸轮控制(增量方式)	✓	✓	123
	TTMR	示教定时器	✓	✓	124
	STMTR	特殊定时器	✓	✓	125
	ALT	交替输出	✓	✓	126
	RAMP	斜坡信号	✓	✓	127
	ROTC	旋转工作台控制	✓	✓	128
	SORT	数据排列	✓	✓	129
	外围设备I/O	TKY	数字键输入	✓	✓
HKY		16 键输入	✓	✓	131
DSW		数字式开关	✓	✓	132
SEGD		七段码译码	✓	✓	133
SEGL		七段码扫描显示	✓	✓	134
ARWS		方向开关	✓	✓	135
ASC		ASCII码转换	✓	✓	136
PR		ASCII码打印输出	✓	✓	137
FROM		BFM 读出	✓	✓	138
TO		BFM 写入	✓	✓	139

分类	指令助记符	功能	适用机型【注】		页码	
			H1U-XP	H2U-XP		
外设设备	RS	串行数据传送	✓	✓	141	
	PRUN	8进制位传送	✓	✓	143	
	ASCI	HEX-ASCII转换	✓	✓	144	
	HEX	ASCII-HEX转换	✓	✓	146	
	CCD	校验码	✓	✓	148	
	PID	PID 运算	✓	✓	149	
	MODBUS	数据传送	✓	✓	153	
	CANTX	CAN数据发送	✓	✓	154	
	CANRX	CAN数据接收	✓	✓	154	
	RS2	串行数据传送2	✓	✓	156	
	MODBUS2	数据传送2	✓	✓	157	
	浮点数	ECMP	2进制浮点数比较	✓	✓	158
		EZCP	2进制浮点数区间比较	✓	✓	159
		EMOV	2进制浮点数数据移动	✓	✓	160
EBCD		2进制-10进制浮点数转换	✓	✓	161	
EBIN		10进制-2进制浮点数转换	✓	✓	162	
EADD		2进制浮点数加法	✓	✓	163	
ESUB		2进制浮点数减法	✓	✓	164	
EMUL		2进制浮点数乘法	✓	✓	165	
EDIV		2进制浮点数除法	✓	✓	166	
EXP		2进制浮点数指数运算	✓	✓	167	
LOGE		2进制浮点数自然对数运算	✓	✓	168	
DLOG		2进制浮点数常用对数运算	✓	✓	169	
ESQR		2进制浮点数开方	✓	✓	170	
INT		2进制浮点数-BIN整数转换	✓	✓	171	
SIN		浮点数SIN运算	✓	✓	172	
COS		浮点数COS运算	✓	✓	173	
TAN		浮点数TAN运算	✓	✓	174	
ASIN		2进制浮点数ASIN运算	✓	✓	175	
ACOS		2进制浮点数ACOS运算	✓	✓	176	
ATAN		2进制浮点数ATAN运算	✓	✓	177	
RAD		2进制浮点数角度→弧度	✓	✓	178	
DEG		2进制浮点数弧度→角	✓	✓	179	
SINH	2进制浮点数SINH运算	✓	✓	180		
COSH	2进制浮点数COSH运算	✓	✓	181		
TANH	2进制浮点数TANH运算	✓	✓	182		
定位	SWAP	上下字节变换	✓	✓	183	
	DSZR	带DOG搜索的原点回归	✓	✓	184	
	ABS	ABS位置数读取	✓	✓	188	
	ZRN	原点回归	✓	✓	189	
	PLSV	可变脉冲输出	✓	✓	190	
	DRVI	相对定位	✓	✓	191	
DRVA	绝对定位	✓	✓	194		

分类	指令助记符	功能	适用机型【注】		页码
			H1U-XP	H2U-XP	
时钟运算	TCMP	时钟数据比较	✓	✓	196
	TZCP	时钟数据区间比较	✓	✓	197
	TADD	时钟数据加法	✓	✓	198
	TSUB	时钟数据减法	✓	✓	199
	TRD	时钟数据读出	✓	✓	200
	TWR	时钟数据写入	✓	✓	201
	HOUR	计时器	✓	✓	202
外围设备	GRY	格雷码变换	✓	✓	203
	GBIN	格雷码逆变换	✓	✓	204
	RD3A	模拟量模块读取	✓	✓	205
	WR3A	模拟量模块写入	✓	✓	206
接点比较	LD=	(S1)=(S2)	✓	✓	207
	LD>	(S1)>(S2)	✓	✓	207
	LD<	(S1)<(S2)	✓	✓	207
	LD<>	(S1)<>(S2)	✓	✓	207
	LD<=	(S1)<=(S2)	✓	✓	207
	LD>=	(S1)>=(S2)	✓	✓	207
	AND=	(S1)=(S2)	✓	✓	208
	AND >	(S1)>(S2)	✓	✓	208
	AND <	(S1)<(S2)	✓	✓	208
	AND <>	(S1)<>(S2)	✓	✓	208
	AND <=	(S1)<=(S2)	✓	✓	208
	AND >=	(S1)>=(S2)	✓	✓	208
	OR=	(S1)=(S2)	✓	✓	209
	OR >	(S1)>(S2)	✓	✓	209
	OR <	(S1)<(S2)	✓	✓	209
	OR <>	(S1)<>(S2)	✓	✓	209
	OR <=	(S1)<=(S2)	✓	✓	209
	OR >=	(S1)>=(S2)	✓	✓	209

【注】：

H1S-XP机型所支持的指令与H1U-XP所支持的指令一致；

H2E-XP机型所支持的指令与H2U-XP所支持的指令一致。

应用指令（以指令助记符为序，未含简单逻辑指令）

分类	指令助记符	功能	适用机型【注】		页码	
			H1U-XP	H2U-XP		
A	ABS	ABS现在值读出	✓	✓	188	
	ABSD	凸轮控制(绝对方式)	✓	✓	122	
	ACOS	2进制浮点数ACOS运算	✓	✓	176	
	ADD	BIN加法	✓	✓	79	
	ALT	交替输出	✓	✓	126	
	AND=	(S1)=(S2)	✓	✓	208	
	AND>	(S1)>(S2)	✓	✓	208	
	AND<	(S1)<(S2)	✓	✓	208	
	AND<>	(S1)<>(S2)	✓	✓	208	
	AND<=	(S1)<=(S2)	✓	✓	208	
	AND>=	(S1)>=(S2)	✓	✓	208	
	ANR	信号报警复位	✓	✓	103	
	ANS	信号报警置位	✓	✓	102	
	ARWS	箭形开关	✓	✓	135	
	ASC	ASCII码转换	✓	✓	136	
	ASCI	HEX→ASCII转换	✓	✓	144	
	ASIN	2进制浮点数ASIN运算	✓	✓	175	
	ATAN	2进制浮点数ATAN运算	✓	✓	177	
	B	BCD	BCD转换	✓	✓	77
		BIN	BIN转换	✓	✓	78
BMOV		成批转换	✓	✓	74	
BON		ON位数判定	✓	✓	100	
C	CALL	子程序调用	✓	✓	60	
	CANTX	CAN数据发送	✓	✓	154	
	CANRX	CAN数据接受	✓	✓	154	
	CCD	检验码	✓	✓	148	
	CJ	条件跳转	✓	✓	60	
	CML	取反传送	✓	✓	73	
	CMP	比较	✓	✓	69	
	COS	浮点数COS运算	✓	✓	173	
	COSH	2进制浮点数COSH运算	✓	✓	181	
	D	DEC	BIN减1	✓	✓	84
DECO		译码	✓	✓	97	
DEG		2进制浮点数弧度→角	✓	✓	179	
DI		中断禁止	✓	✓	62	
DIV		BIN除法	✓	✓	82	
DLOG		2进制浮点数常用对数运算	✓	✓	169	
DRVA		绝对定位	✓	✓	194	
DRVI		相对定位	✓	✓	191	
DSW		数字式开关	✓	✓	132	
DSZR		带DOG搜索的原点回归	✓	✓	184	
E	EADD	2进制浮点数加法	✓	✓	163	
	EBCD	2进制-10进制浮点数转换	✓	✓	161	
	EBIN	10进制-2进制浮点数转换	✓	✓	162	
	ECMP	2进制浮点数比较	✓	✓	158	
	EDIV	2进制浮点数除法	✓	✓	166	
	EI	中断允许	✓	✓	62	
	EMOV	2进制浮点数数据移动	✓	✓	160	
	EMUL	2进制浮点数乘法	✓	✓	165	
	ENCO	编码	✓	✓	98	
	ESQR	2进制浮点数开方	✓	✓	170	
ESUB	2进制浮点数减法	✓	✓	164		
EXP	2进制浮点数	✓	✓	167		
EZCP	2进制浮点数区间比较	✓	✓	159		

分类	指令助记符	功能	适用机型【注】		页码
			H1U-XP	H2U-XP	
F	FEND	主程序结束	✓	✓	65
	FLT	BIN整数→2进制浮点数转换	✓	✓	105
	FMOV	多点传送	✓	✓	75
	FOR	循环范围开始	✓	✓	67
	FROM	BFM读出	✓	✓	138
G	GBIN	格雷码逆变换	✓	✓	204
	GRY	格雷码变换	✓	✓	203
H	HEX	ASCII-HEX转换	✓	✓	146
	HKY	16键输入	✓	✓	131
	HOURL	计时器	✓	✓	202
	HSCR	比较复位 (高速计数器)	✓	✓	111
	HSCS	比较置位 (高速计数器)	✓	✓	109
	HSZ	区间比较 (高速计数器)	✓	✓	112
I	INC	BIN加1	✓	✓	83
	INCD	凸轮控制(增量方式)	✓	✓	123
	INT	浮点数-整数转换	✓	✓	171
	IRET	中断返回	✓	✓	62
L	LD=	(S1)=(S2)	✓	✓	207
	LD>	(S1)>(S2)	✓	✓	207
	LD<	(S1)<(S2)	✓	✓	207
	LD<>	(S1)<>(S2)	✓	✓	207
	LD<=	(S1)<=(S2)	✓	✓	207
	LD>=	(S1)>=(S2)	✓	✓	207
	LOGE	2进制浮点数自然对数运算	✓	✓	168
M	MEAN	平均值	✓	✓	101
	MODBUS	数据传送	✓	✓	153
	MODBUS2	数据传送2	✓	✓	157
	MOV	传送	✓	✓	71
	MTR	矩阵输入	✓	✓	108
	MUL	BIN乘法	✓	✓	81
N	NEG	求补码	✓	✓	87
	NEXT	循环范围结束	✓	✓	67
O	OR=	(S1)=(S2)	✓	✓	209
	OR>	(S1)>(S2)	✓	✓	209
	OR<	(S1)<(S2)	✓	✓	209
	OR<>	(S1)<>(S2)	✓	✓	209
	OR<=	(S1)<=(S2)	✓	✓	209
	OR>=	(S1)>=(S2)	✓	✓	209
P	PID	PID运算	✓	✓	149
	PLSV	可变速脉冲输出	✓	✓	190
	PLSY	脉冲输出	✓	✓	117/ 363
	PLSR	带加减速脉冲输出	✓	✓	119
	PR	ASCII键打印输出	✓	✓	137
	PRUN	8进制输送	✓	✓	143
	PWM	脉宽调制	✓	✓	118

分类	指令助记符	功能	适用机型【注】		页码	
			H1U-XP	H2U-XP		
R	RAD	2进制浮点数角度→弧度	✓	✓	178	
	RAMP	斜坡信号	✓	✓	127	
	RCL	带进位的循环左移	✓	✓	90	
	RCR	带进位的循环右移	✓	✓	90	
	RD3A	模拟量模块读取	✓	✓	205	
	REF	输入输出刷新	✓	✓	106	
	REFF	滤波器调整	✓	✓	107	
	ROL	循环左移	✓	✓	89	
	ROR	循环右移	✓	✓	89	
	ROTC	旋转工作台控制	✓	✓	128	
	RS	串行数据输送	✓	✓	141	
	RS2	串行数据输送2	✓	✓	156	
	SEGD	七段码译码	✓	✓	133	
	SEGL	七段码扫描显示	✓	✓	134	
	SER	数据查找	✓	✓	121	
	SFRD	“先进先出”读出	✓	✓	95	
	S	SFTL	位左移	✓	✓	91
SFTR		位右移	✓	✓	91	
SFWR		移位写入	✓	✓	94	
SIN		浮点数SIN运算	✓	✓	172	
SINH		2进制浮点数SINH运算	✓	✓	180	
SMOV		移位传送	✓	✓	72	
SORT		数据排列	✓	✓	129	
SPD		脉冲密度	✓	✓	115/ 362	
SQR		BIN开方	✓	✓	104	
SRET		子程序完毕	✓	✓	61	
STMR		特殊定时器	✓	✓	125	
SUB		BIN减法	✓	✓	80	
SUM		ON位数	✓	✓	99	
SWAP		上下字节交换	✓	✓	183	
T		TADD	时钟数据加法	✓	✓	198
		TAN	浮点数TAN运算	✓	✓	174
		TANH	2进制浮点数TANH运算	✓	✓	182
	TCMP	时钟数据比较	✓	✓	196	
	TKY	数字键输入	✓	✓	130	
	TO	BFM写入	✓	✓	139	
	TRD	时钟数据读出	✓	✓	200	
	TSUB	时钟数据减法	✓	✓	199	
	TTMR	示教定时器	✓	✓	124	
	TWR	时钟数据写入	✓	✓	201	
	TZCP	时钟数据区间比较	✓	✓	197	
W	WAND	逻辑与	✓	✓	85	
	WDT	监控定时器	✓	✓	66	
	WOR	逻辑或	✓	✓	85	
	WR3A	模拟量模块写入	✓	✓	206	
	WSFL	字左移	✓	✓	92	
	WSFR	字右移	✓	✓	92	
	WXOR	逻辑异或	✓	✓	85	
X	XCH	交换	✓	✓	76	
Z	ZCP	区间比较	✓	✓	70	
	ZRN	原点回归	✓	✓	189	
	ZRST	批次复位	✓	✓	96	

【注】：H1S-XP机型所支持的指令与H1U-XP所支持的指令一致；H2E-XP机型所支持的指令与H2U-XP所支持的指令一致。

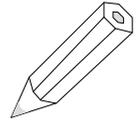
手册快查引导

您若有如下疑问，可参照指引：

序号	希望查阅的内容	请参阅页面
1	简单逻辑指令的解释	查阅P38~P41指令详解
2	应用指令的解释	先根据指令名查阅P7~P10，再详查指令详解
3	计时器的选用	查阅3.5节
4	高速计数器的选用	查阅3.6节
5	高速输出指令的使用	查阅指令 HSCS/HSCR/HSZ/PLSY/PLSV/PWM
6	输入中断的使用与设置	查阅3.8节
7	定时中断的使用与设置	查阅3.8节
8	高速计数中断的使用与设置	查阅3.8节
9	STL/SFC的编程方法	查阅4.2节
10	通讯格式的设置方法	查阅RS指令详解
11	各种通讯协议的设置方法	查阅RS指令详解
12	如何使用MODBUS主站指令	查阅RS指令详解
13	如何用MODBUS访问H _{1U} /H _{2U} 系列PLC	MODBUS内置从机协议定义
14	如何使用模拟量扩展卡	查阅6.7.1节
15	MTQ/MTP的高速功能与使用	参见8.4/8.5附录，查阅高速指令
16	脉冲捕捉功能	参见中断P说明、M8170~M8175变量说明
17	增强版的高速处理指令	参见8.5附录，增强高速指令
18	部分特殊继电器和寄存器功能说明	参见8.7附录

Memo NO. _____

Date / /



A series of horizontal lines providing a space for writing the memo content.



第一章 梯形图及梯形图程序

1.1 梯形图的编程特点.....	14
1.2 梯形图编程时使用的元件符号.....	14
1.3 PLC的执行原理.....	15
1.4 PLC数值的基本知识.....	16

第一章 梯形图及梯形图程序

梯形图及梯形图程序

1.1 梯形图的编程特点

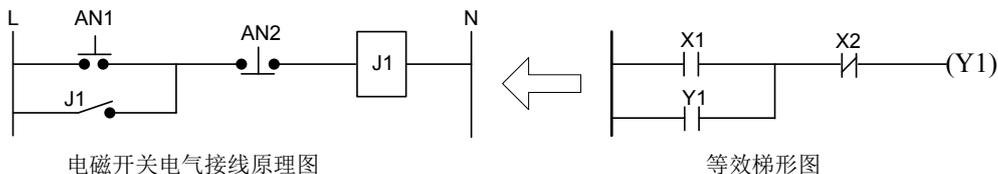
PLC中梯形图编程方法是仿传统继电器控制系统的电气原理设计的一种设计方法，设计中使用的元件如按钮开关X、中间继电器M、时间继电器T、计数器C、触点等，都和实际的电气元件的特性相似。

梯形图中常用“触点”和“线圈”元件，触点元件有“常开型”和“常闭型”，分别对应电工术语中的“A接点”和“B接点”，PLC中同一个继电器的“触点”可被无限次使用，我们可认为一个继电器（无论中间继电器M、时间继电器T、还是计数器C）元件，都具有无限个“A接点”和“B接点”。

对于时间继电器、计数器，具有线圈（信号触发端）和触点，部分元件还具有掉电保持特性，选择合适序号的元件，以得到所需特性的元件。

随着现代PLC的发展，PLC不仅可以完成顺序逻辑控制功能，还能完成数值计算功能，如数值比较、四则运算、函数运算等，数值宽度有16bit、32bit、浮点等，在H1U/H2U系列PLC中提供了大量的寄存器D元件，可在梯形图程序中用于数值运算。

梯形图的设计思想与传统继电器控制系统的设计方法基本相同，以常见的电磁开关的电气原理为例：

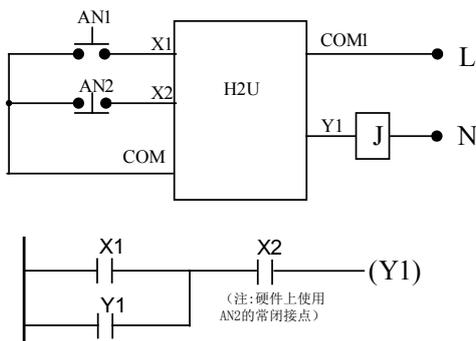


电磁开关电气接线原理图

等效梯形图

从图中可见，J1为继电器或接触器，AN1为启动J1的按钮，使用其常开接点；而AN2为断开J1的按钮，使用了其常闭接点；另外使用了J1的常开型辅助触点作为状态保持用。

若按右图设计PLC的信号输入连接和梯形图编程便可实现相同的起停控制功能了。（出于安全的考虑，停止按键一般用常闭型接点。）



用户程序梯形图

1.2 梯形图编程时使用的元件符号

梯形图设计中使用的元件符号及特性说明如下表，通过这些“触点”元件的“与”“或”逻辑组合，输出到元件“线圈”：

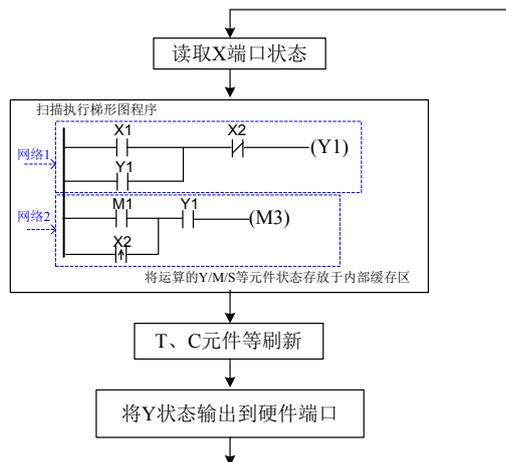
符号	说明	动作特性
	触点元件，代表元件的常开型触点，有输入X信号触点、输出Y的触点、中间继电器M、时间继电器T、计数器C的输出触点等。对于Y、M、T、C等元件，在未动作状态也为OFF。	X：当X端口信号接点闭合时，状态为ON；端口信号为断开状态时，触点状态为OFF
		Y：当Y继电器的“线圈”得电时为ON，否则为OFF。Y最后状态将对应于PLC的输出Y端口的状态。
		M：当M继电器的“线圈”得电时为ON，否则为OFF
		S：当S作为普通标志元件使用时，S继电器的“线圈”得电时为ON，否则为OFF
		T：当对应的时间继电器线圈得电，且计时时间达到设定的时间，状态为ON；否则为OFF
C：当对应的计数器的读数达到设定的时间，状态为ON；否则为OFF		
	触点元件，代表元件的常闭型触点，有输入X信号触点、输出Y的触点、中间继电器M、时间继电器T、计数器C的输出触点等。	逻辑与状态刚好与 的信号相反
	触点元件，仅在触点的上升沿有效	当触点元件（XYM）的状态由OFF→ON的上升沿变化时，该信号为有效，这个触点信号在一个扫描周期内有效，若下一状态不再变化，该信号恢复为“OFF”

符号	说明	动作特性
	触点元件，仅在触点的下降沿有效	当触点元件（XYM）的状态由ON→OFF的下降沿变化时，该信号为有效，这个触点信号在一个扫描周期内有效，若下一状态不再变化，该信号恢复为“OFF”
	状态取反	将当前信号点的状态进行取反
	步进梯形图中表示S状态信号	步进指令状态的转移
	线圈元件，在梯形图中是被激励的对象	Y、M元件的线圈“得电”时，其常开型触点动作闭合，其常闭型触点动作断开，“失电”时恢复原来状态
		T元件的线圈“得电”时，开始计时，“失电”时恢复为默认状态。当计时时间达到设定值时，其常开型触点动作闭合，其常闭型触点动作断开。
		C元件的线圈“得电”的瞬间，计数值增加1，当计数值达到设定值时，其常开型触点动作闭合，其常闭型触点动作断开。清除其“线圈”的操作，可使其计数值和触点恢复为默认状态。
		注意：X输入元件没有线圈，用户程序不能修改其状态，只由外部的用户线路决定其状态。
	操作指令，对元件或线圈、参数等进行操作	完成逻辑操作、数据处理等众多功能。如〔RST Y0〕、〔SET M2〕、〔MOV K5 D100〕、〔JC P1〕等指令。

1.3 PLC的执行原理

当编程人员将设计编译好的梯形图程序下载到PLC的内存后，PLC便可以对用户程序进行扫描执行了。

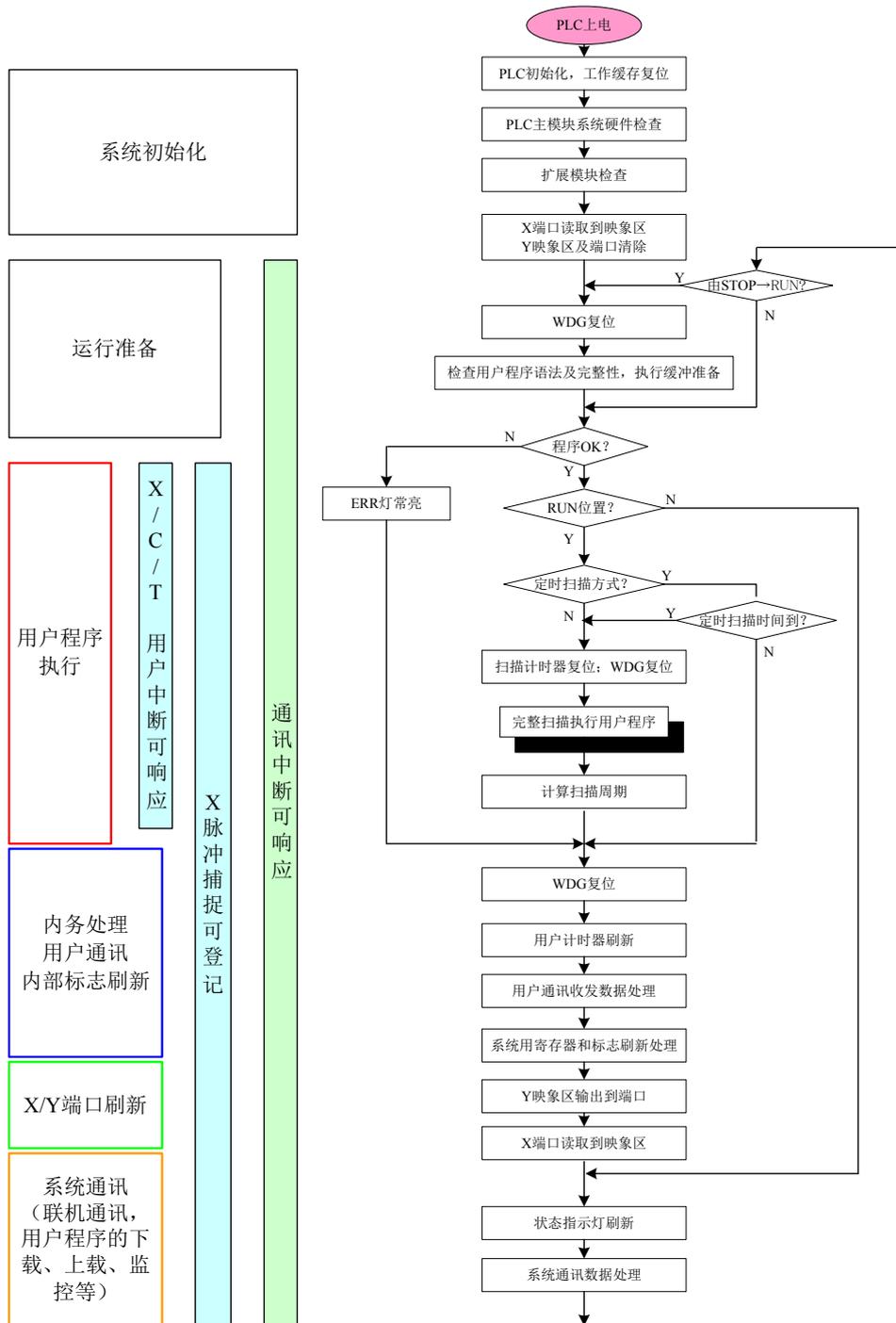
PLC运行时，主要进行执行X输入检测、用户程序扫描运算、其他元件的状态刷新、将Y状态缓存状态输出到PLC的Y硬件端口等，这些工作内容周而复始的进行，其中的扫描执行用户程序是PLC的核心工作，过程如下例图：



每次执行用户程序前，首先将X硬件端口的状态读取后存放到X变量缓存区。

用户程序的扫描执行，是以用户程序的网络块为单元进行逐步演算的，所谓“网络”是有联线关联的一组元件块，参见上图中的两个网络。执行演算从第一个网络开始，依次向下演算第二个、第三个……直到最后一个网络。而对每个网络进行演算方式是，则由左至右，逐个将元件的“触点”状态进行逻辑计算综合，直到最右边，输出到元件的“线圈”，或根据逻辑决定是否执行某个操作。

梯形图中，左侧目前相当于电源的“火线”，其默认的（电位）状态为ON，每经过一个元件后，逻辑运算结果暂存都被刷新，有时也称中间计算暂存状态为“能流”，中间逻辑计算结果为ON，即“能流”为有效，本网络的输出状态即为输出电的能流状态；若最右端为操作类型，若能流为有效，就进行操作，否则不进行操作。



由上至下，直到主程序的所有网络都扫描执行完毕，还有各定时器的刷新、例行的通讯等数据的处理后，PLC系统程序将Y寄存器缓存区的变量状态输出到Y硬件端口中。然后又开始新一轮的用户程序扫描，如此周而复始，直到控制用户执行的“RUN/STOP”开关被拨动到STOP位置为止。

对于整个PLC而言，其系统软件还需完成一些运行准备、系统通讯、中断处理等工作，系统软件运行流程如上图所示。对于复杂的用户程序，在系统扫描用户程序过程中，还可以采用“中断”处理的方法响应“用户中断”信号，对重要信号（也有称重要“事件”）作及时处理。

所谓“中断”处理，就是CPU检测到特定信号时，立即停下（或中断）当前的例行工作，去执行特定的子程序，子程序执行完毕，才返回到先前被停下的工作点，继续执行例行工作。中断信号的请求能得到及时的响应处理，是“中断”功能的主要特点。

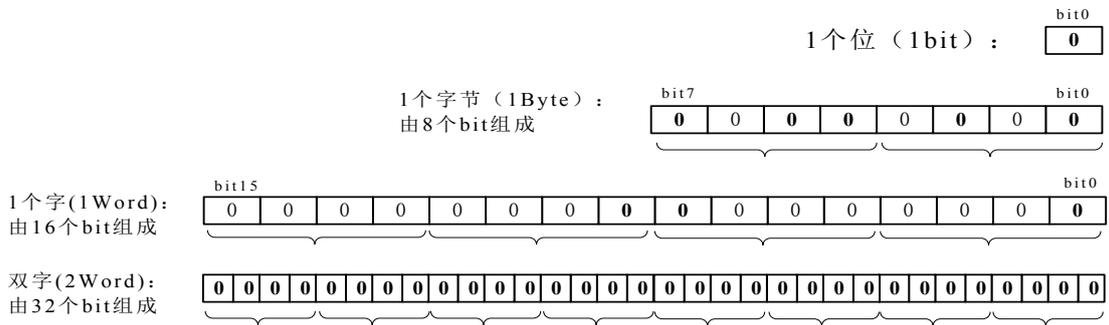
在PLC中，有高速信号输入（X0~X5）、高速计数、定时等中断（有时称为“用户中断”），还有通讯中断，包括系统通讯、用户程序发起的通讯等。在PLC中，各中断享有同一优先级，但不同中断类型，其允许区间稍有不同（参见前页插图）。

1.4 PLC数值的基本知识

H1U/H2U系列PLC内部采用高性能32bit作为核心处理器，其工作原理与其他的计算机设备是相似的。所有的CPU处理器采用的都是二进制码作为内部处理数据的格式，“数据”在计算机内部是以“信号电平”的方式进行处理，其中信号电平只有“低”或“高”两个状态，分别对应于二进制数的“0”或“1”，信号电路中不会出现电平误判，可确保处理结果的正确性。

二进制数

“二进制”用于计算机计算则是最简捷方便的进制，对于1位数的计算有：
 $0+0=0$ ； $0+1=1$ ； $1+0=1$ ； $1+1=10$ （有进位，此时需用2个位来表示）
 这些计算只需用典型的“与”、“或”、“非”逻辑电路就可组合完成运算了。



当需要处理的数值比较大时，就需用多个二进制位来表示，位数越多，可表示的数值越大，现在常用的CPU位(bit)数有：

位数	可一次处理的最大数值	应用说明
4bit	15	消费类简单产品中还有使用，已很少
8bit	255	如8051，常用于简单的控制系统中
16bit	65, 535	如808x，工业控制中有使用，使用较少
32bit	4, 294, 967, 295	如ARM，目前广泛应用于工控、消费类产品
64bit	18446744073709551615	通用计算机中使用

位数少的CPU，并非不能处理大的数值，只不过需要多次运算，有时还需要编程人员熟悉算法。就像大车一次可以搬运的货物，用小车就需要往返多次才能搬完，车越小，需要的次数越多，耗时也越多。

H1U/H2U系列PLC元件中，常用的数据宽度是1Word（即16bit）；部分计数器为2Word（32bit）。对于16bit的无符号数据，用2进制表示的最大值为1111, 1111, 1111, 1111，换算为十进制就是65, 535。

十六进制

当二进制数值小的时候，尚能阅读，当位数比较多时，就比较难读难写了，将每4位二进制数分成一组，用1个数来代表，就成了16进制数（HEX）；一个16bit二进制数用4位十六进制数来表示，易读性大为增加。在十六进制数中数值10~15（十进制）的数，分别以A~F的字符来代替。

八进制

由于传统习惯，在计算机中，以8bit宽度的数值、硬件端口数使用方式的为最多，8bit被定义为1Byte（即1个字节）；在PLC中也8个硬件端口作为分组，利于访问操作（读或写），如输入X端口、输出Y端口的编号就仍沿用八进制方式。

八进制数是由3位二进制数组成的，数字范围为000~111，即0~7，不可能存在8、9。由于CPU一般为8、16、32bit等，但用于数据计算时，一般还是用十六进制，而不用八进制。

十进制

我们生活中习惯使用的数据是采用“十进制”，基本数字为0~9共10个数，若“9”+“1”计数，便进位处理得到“10”。

日常生活中也有其他进制的，如星期日、星期一、...星期六分别以数字0、1、...6共7个数代表，就可理解为“七进制”，只不过“七进制”不便于计算，使用不多而已。

BCD码

最符合人们阅读习惯的数字格式是十进制，在人们监控或设置工作参数时，往往需要采用十进制格式进行数据显示，而计算机内部使用的是HEX格式，故需采用一种底层为每4个二进制位组成一个数字位，而每个数字位只能为十进制数的0~9，由此组成的数值，这种格式数字在存储器中的编码称为BCD码(Binary-Coded Decimal)。

在PLC内部，原理上用4位二进制数代表1位十进制数，在每一位BCD码中，不存在HEX格式中的A~F。对于一个8bit宽度的寄存器单元，能存储的最大BCD数只能是99，因此将HEX格式转换为BCD码后，会占用更大的存储空间。



PLC内部总是按HEX格式进行数据计算的，在驱动非智能的显示设备（如数码管）显示数据之前，往往需要将PLC内部的十六进制（HEX）格式数据先转换为BCD码，然后进行显示输出；将用户以十进制方式设置的参数存入PLC内存之前，则往往需要将该BCD码转换为十六进制（HEX）格式。

H1U/H2U系列PLC内部提供了HEX与BCD两种格式相互转换的命令，在需要进行显示输出，或设置开关读取的时候，执行该格式转换指令。

人们在电脑显示器上看到的十进制读数，都是经过了计算机自动作BCD转换后才显示的；监控时修改的参数，则是电脑软件作了HEX转换后写入的，无需人为干预而已。

各种进制数的对照举例：

二进制 BIN	八进制 OCT	十进制 DEC	十六进制 HEX	BCD码	二进制 BIN	八进制 OCT	十进制 DEC	十六进制 HEX	BCD码
0000	0	0	0	0000	1000	10	8	8	1000
0001	1	1	1	0001	1001	11	9	9	1001
0010	2	2	2	0010	1010	12	10	A	00010000
0011	3	3	3	0011	1011	13	11	B	00010001
0100	4	4	4	0100	1100	14	12	C	00010010
0101	5	5	5	0101	1101	15	13	D	00010011
0110	6	6	6	0110	1110	16	14	E	00010100
0111	7	7	7	0111	1111	17	15	F	00010101

进制的转换

二进制、八进制、十六进制等进制的转换非常简单，例如8位的二进制数“10110101”，写成十六进制时，从右向左按4位一组分为“1011，0101”，用十六进制表示为“B5”；

写成八进制时，从右向左按3位一组分为“10，110，101”，用八进制表示为“265”；

要将二进制数换算为十进制数，则计算要复杂很多，最通用的方法可采用权重累加法，从最右边一位开始计算：

第1位（bit0）为1时，权重为1，（即 2^0 ），否则为0；

第2位（bit1）为1时，权重为2，（即 2^1 ），否则为0；

第3位（bit2）为1时，权重为4，（即 2^2 ），否则为0；

第4位（bit3）为1时，权重为8，（即 2^3 ），否则为0；

第5位（bit4）为1时，权重为16，（即 2^4 ），否则为0；

第6位（bit5）为1时，权重为32，（即 2^5 ），否则为0；

第7位（bit6）为1时，权重为64，（即 2^6 ），否则为0；

第8位（bit7）为1时，权重为128，（即 2^7 ），否则为0； ……

对于本例子中，将“10110101”转换为十进制数即为（ $128+0+32+16+0+4+0+1$ ）=181。

对于16bit转换为十进制，如本例中的“B5”，也采用十六进制的权重累加法，从最右边一位开始计算：

第1位HEX数的权重为1，（即 16^0 ），即该位的实际值 $\times 1$ ；

第2位HEX数的权重为16，（即 16^1 ），即该位的实际值 $\times 16$ ；

第3位HEX数的权重为256，（即 16^2 ），即该位的实际值 $\times 256$ ；

第4位HEX数的权重为4096，（即 16^3 ），即该位的实际值 $\times 4096$ ；…..

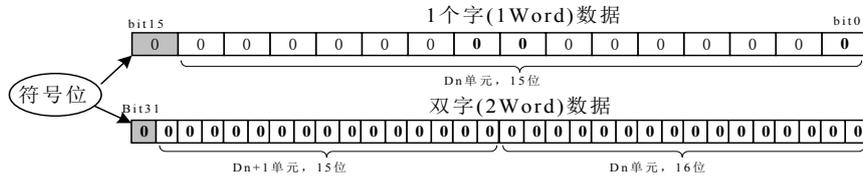
对于本例子中，将“B5”转换为十进制数即为（ $B \times 16 + 5 \times 1$ ）=（ $11 \times 16 + 5$ ）=181。

读者熟悉了HEX转换为十进制的方法，可先将二进制或八进制划分为十六进制（每4bit一组），然后再作十进制转换，计算比较简捷。

有符号数与无符号数

PLC内部的数据可以进行四则运算，运算结果可能产生负数，这样的计算结果就产生了“有符号数”，事实上H1U/H2U系列PLC内部的寄存器D、32bit计数器C的数据、所有四则和函数运算指令都可按“有符号数”进行运算操作。

16bit的D寄存器中最高位 (bit15) 便用于代表值的符号，因此D寄存器值的取值范围是-32,768~32,767。当用双字 (32bit, 2个连续的D寄存器) 表示一个数据时，用最高位 (bit31) 代表值的符号，因此D寄存器值的取值范围是-2,147,483,648~2,147,483,647。符号位如下图：



当符号位为0时，表示为正数，故1Word的正数是最大值为HEX格式的H7FFF，即32767；2Word的正数是最大值为HEX格式的H7FFFFFFF，即2,147,483,647。

当符号位为1时表示负数，是其数值的补码，其绝对值的计算方法是“先将符号数逐位取反，然后再加1”，例如HEX格式的HFFFF，其绝对值=H0000+1=1，即“HFFFF”代表-1；又例如HEX格式的H8000，其绝对值=H7FFF+1=32768，即有符号数H8000代表-32768，是1Word寄存器最小的负值。同理，2Word的最小负值为有符号数H80000000，即-2,147,483,648。

进行数值比较大的加减运算时，要注意符号的处理，尤其是出现进位或借位操作时，要进行“借位标志”、“进位标志”的判断及相应处理，否则可能导致计算结果出错。

无符号数，即没有符号位，默认都为正数，对于1Word的寄存器，其取值范围是0~65535，有些计时、计数的应用场合，就只有正数，需按无符号数处理，在作加减运算时，需要防止计算结果溢出，导致计算错误。

当进行逻辑运算时 (如“逻辑与”、“逻辑或”等运算指令)，操作数是当无符号数进行处理的，符号位 (bit15) 与其它位同等参与逻辑运算。

浮点数

浮点数在PLC (或计算机) 中用以近似表示任意实数，具体格式是由一个整数或定点数 (即尾数) 乘以某个基数 (计算机中通常是2) 的整数次幂，这种表示方法类似于基数为10的科学记数法。

一个浮点数可用 $m \times b^e$ 来表示。其中m为尾数，形如±d.ddd...dd；b为基数；e为指数。例如988436216用十进制浮点数表示就可为 9.8844×10^8 ，因尾数有四舍五入，精度有所下降。但可以看到，使用浮点数可表示更大范围的数值。

由此可以看出，在计算机中表示一个浮点数，其结构如下：

尾数部分 (定点小数)		阶码部分 (定点整数)	
数符±	尾数m	阶符±	阶码e

在PLC或计算机内使用的浮点数，都采用了国际标准的格式，为了计算的方便，仪表都采用二进制浮点数格式。一个浮点数占用32bit的存储器单元。实际使用浮点数时，并不需要用户对浮点格式有特别的了解，计算机会对输入的实数自动作标准格式化处理。

浮点计算是指浮点数参与的运算，这种运算通常伴随着因为无法精确表示而进行的近似或舍入。在PLC中有模拟量信号处理和运算时，可能用到浮点数。

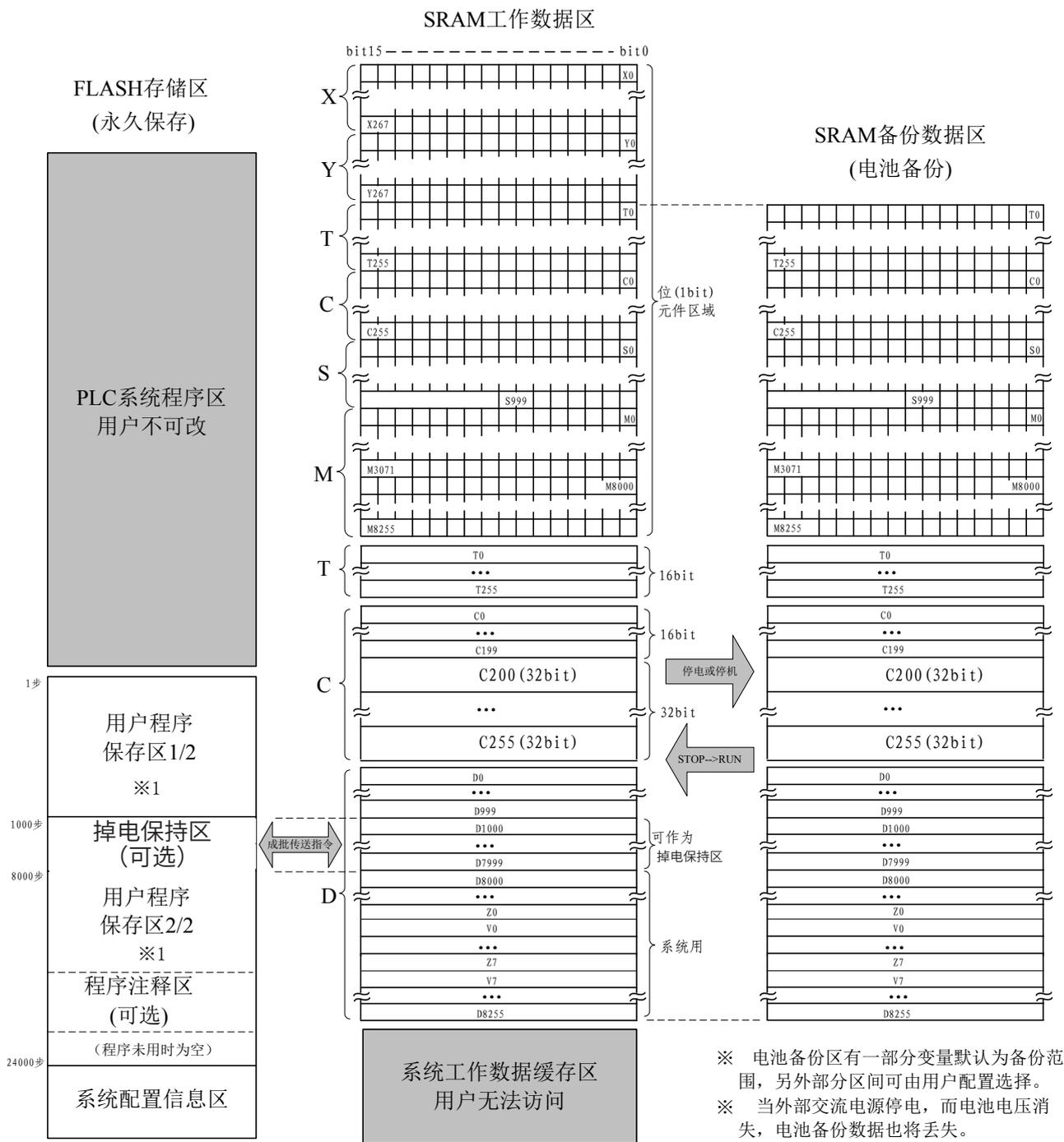
H1U/H2U系列PLC的内存结构

对于微机或单片机的系统，除了CPU内核以外，各种特性的内存是其主要配置。H1U/H2U系列PLC中有如下几种内存：

类型	用途	特性
FLASH	保存系统程序	永久保存
	保存用户程序	永久保存，除非人为删除，或下载刷新
SRAM	用于存放PLC的软件件、工作数据	有电池供电时，即使外部停电时数据也不会丢失

如前所述，PLC内的软件件有“位元件” (触点元件)，有16bit的“字元件” (寄存器D、计数器C、计时器T等)，还有32bit的“双字元件” (部分高速计数器C)，在PLC内部如下组织：

梯形图
及梯形
图程序



※1 用户程序保存区最大为16K步(Word), 存放用户程序时可自动回避文件寄存器区。



第二章 H1U/H2U系列PLC的 使用方法

2.1 使用PLC的软件硬件需求	22
2.2 编程与用户程序下载	22
2.3 与HMI的配合使用	22

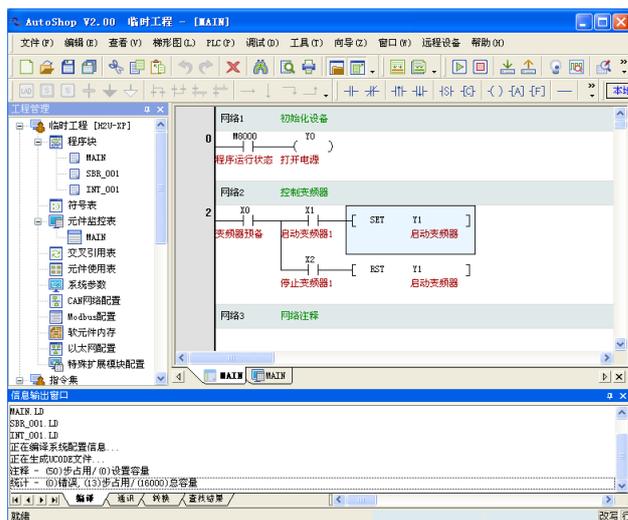
第二章 H1U/H2U 系列PLC的使用方法

2.1 使用PLC的软件硬件需求

项目	配置要求
电脑一台	PENTIUM 100MHz以上主频；内存256MB以上；鼠标等具有DB9型RS232串行通讯口，（否则需准备USB-RS232转换器；或USB-Mini DIN8型专用下载电缆），或Mini USB下载线缆。运行Windows 2000/XP操作系统；硬盘剩余空间不小于200MB；
AutoShop编程软件	汇川控制技术公司开发的AutoShop软件，安装于PC电脑中，用于用户程序的编写、下载、监控调试等。也可采用其他兼容机型的编程环境。
PLC主模块	H1U/H2U系列PLC主模块一只，可根据应用需要准备扩展模块。
下载电缆一条	市售RS232-Mini DIN8插头的PLC程序下载专用电缆，或Mini USB下载线缆，用于用户程序的下载、调试、监控等，还可用于HMI连接。对于没有配备DB9型RS232串口的电脑，也可准备USB-Mini DIN8型专用下载电缆。
电源联线和其他	用于PLC供电的电源线，根据需要可准备导线、拨码开关、螺丝批等常用工具。



其中AutoShop编程软件为汇川控制技术公司研发的编程后台软件，在该软件环境下，可进行H1U/H2U系列PLC用户程序的编写、下载和监控等功能。



2.2 编程与用户程序下载

AutoShop环境提供了梯形图、步进梯形图、SFC、指令表等编程语言，用户可选用自己熟悉的编程语言进行编程，根据PLC应用系统的控制工艺要求，设计程序。编程过程中，可随时进行按  编译，及时检查和修正编程错误。

程序设计完毕，在PLC和电脑正常连接，并已通电的情况下，按  即可下载用户程序，程序下载完毕，将PLC上RUN/STOP拨动开关拨至“RUN”位置，PLC即可开始运行用户程序。

在PLC运行用户程序时，按  键即可进行运行的停止和运行命令操作；按  可监控PLC内各种继电器和寄存器D的状态和读数，在当前编程画面上显示出来，方便了程序调试。

2.3 与HMI的配合使用

H1U/H2U系列PLC提供了MODBUS协议，也支持FX2N/3U系列PLC的监控协议，因此目前市售的HMI产品，基本上都可以与H1U/H2U系列PLC配合使用，包括连接电缆均可由市面购得。

关于H1U/H2U系列PLC所支持的协议的种类和使用的详细说明，可参见RS通讯指令的解释。



第三章 软元件说明

3.1 输入继电器X	24
3.2 输出继电器Y	24
3.3 辅助继电器M	25
3.4 状态继电器S	26
3.5 计时器T	26
3.6 计数器C	27
3.7 寄存器D	31
3.8 子程序与中断指针P、I.....	32
3.9 常数K、H.....	35
3.10 控制器软元件规格.....	35

第三章 软元件说明

系统支持的软元件种类：

序号	元件类型	功能与分类
1	输入继电器X	对应PLC的硬件开关量输入的位元件
2	输出继电器Y	对应PLC的控制输出的位元件
3	中间继电器M	普通中间继电器M位元件
		系统特殊继电器M位元件
4	状态继电器S	步进控制用状态标志位元件
5	计时器T	具有1ms、10ms、100ms步长的16bit计时器
6	计数器C	具有16bit/32bit增/减型计数器
		高速计数器、单/双相各种计数器
7	寄存器D	数据寄存器D
		数据间接寻址寄存器V、Z
8	指针P、I	跳转指针P
		子程序指针P
		中断子程序I，有高速输入、定时、计数等中断
9	常数K、H	二进制、十进制、十六进制、浮点数等

3

软元件
说明

3.1 输入继电器X

输入继电器X代表PLC外部输入信号状态的元件，通过X端口来检测外部信号状态，0代表外部信号开路，1代表外部信号闭合。

用程序指令方法不能修改输入继电器的状态，其接点信号（常开型、常闭型）在用户程序中都可无限次使用。

继电器信号以X0，X1，…X7，X10，X11，等符号标识，其序号是以8进制方式编号。

控制器的计数器信号、外部中断信号、脉冲捕捉等功能是通过X0~X7端口输入。部分产品型号输入输出点数如下表所示：

型号	输入	输出
H _{2U} -1616MR/T	X000-X017	Y000-Y017
H _{2U} -3624MR/T	X000-X043	Y000-Y027
H _{2U} -3232MR/T	X000-X037	Y000-Y037
H _{2U} -6464MR/T	X000-X077	Y000-Y077
H _{1U} -0806MR/T	X000-X007	Y000-Y005
H _{1U} -1614MR/T	X000-X017	Y000-Y015

当接入本地扩展模块后，扩展模块上X端口的编号按紧接主模块上X端口的编号，依次向后编号，例如当主模块为H_{2U}-1616MR，现在要接入H_{2U}-1600EX型扩展模块，因主模块最后的X端口编号为X17，则扩展模块的X在编程时的访问编号为X20~X37。

注意，扩展模块的编号总是从8进制个位为0开始的，例如，当主模块为H_{2U}-3624MR，其最后的X端口编号为X43，扩展模块的X在编程时的访问编号为X50~X67，即主模块上空缺的X44~X47的端口号被丢弃。扩展模块上Y端口也采取了同样的处理方法。

远程数字量扩展模块必须用FROM/TO指令来访问X点。具体请参考FROM/TO指令介绍。

3.2 输出继电器Y

输出继电器是直接关联到外部用户控制装置的硬件端口的软元件，在逻辑上与PLC的物理输出端口一一对应。PLC每次扫描完用户程序后，会将Y继电器的元件状态传送到PLC的硬件端口上，0表示输出端口开路；1表示输出端口闭合。

Y继电器编号以Y0，Y1，…Y7，Y10，Y11，…等符号标识，其序号是以8进制方式编号。Y继电器元件可在用户程序中无限次使用；

硬件上，根据输出元件的不同，可分为继电器型、晶体管型等；若有输出扩展模块端口，按照由主模块开始，依次序进行编号。

当接入本地扩展模块后，扩展模块上Y端口的编号按紧接主模块上Y端口的编号，依次向后编号，例如当主模块为H_{2U}-1616MR，现在要接入H_{2U}-0016EYR型扩展模块，因主模块最后的Y端口编号为Y17，则扩展模块的X在编程时的访问编号为Y20~Y37。

注意：扩展模块的端口编号总是从8进制个位为0开始的。

远程数字量扩展模块必须用FROM/TO指令来访问Y点。

3.3 辅助继电器M

辅助继电器M元件用作用户程序执行过程中的中间变量，如同实际电控系统中的辅助继电器，用于状态信息的传递，也可将多个M变量组成为字变量使用，M变量与外部端口没有直接的联系，但可通过程序语句将X复制到M，或将M复制到Y的方式与外界发生联系，一个M变量可无限次使用。

辅助继电器M以M0, M1, ……M8255等符号标识，其序号是以10进制方式编号。M8000以上的变量为系统专用变量，用于PLC用户程序与系统状态的交互；部分M变量还具有掉电保存特性。

H1u机型

M数量总计	一般用	停电保持用	特殊用
1792点	M0-M383 384点 ※1	M384-M3071 2688点 ※3	M8000-M8511 512点

H2u机型

M数量总计	一般用	停电保持用	停电保持专用	特殊用
3328点	M0-M499 500点 ※1	M500-M1023 524点 ※2	M1024-M3071 2048点 ※3	M8000-M8511 512点

3
软元件
说明

- ※1. 非停电保持区域。使用参数设定，可变更成停电保持区域。
- ※2. 停电保持区域。使用参数设定，可变更成非停电保持区域。
- ※3. 停电保持区域，无法用参数来改变。

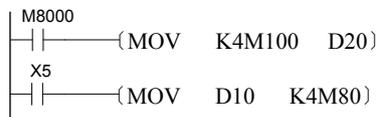
可编程控制器内的一般用辅助继电器、停电保持用辅助继电器的区域分配，可通过参数设定来进行调整。

可编程控制器内有大量的特殊辅助继电器。（参见系统特殊元件表）。这些特殊辅助继电器各有其特定的功能，可分为以下两类：

- 触点利用型的特殊辅助继电器，为PLC系统自动驱动线圈，用户程序只能读取使用，如：
 - M8000：运行监视器（在运行中接通），常用于需要驱动信号的指令之前。
 - M8002：初始脉冲（仅在运行开始时瞬间接通），常用于只需执行一次初始化指令。
 - M8012：100ms 时钟脉冲，用于产生固定间隔翻转的信号。
- 线圈驱动型特殊辅助继电器，为用户程序驱动线圈，用于控制PLC的工作状态和执行模式等，如：
 - M8030：电池发光两极管熄灯指令
 - M8033：停止时保持输出
 - M8034：输出全部禁止
 - M8039：恒定扫描

请注意，存在驱动时有效与 END 指令执行后有效两种情况；用户不可使用尚未定义的特殊辅助继电器。

- 可以将连续的M变量按字节或字来进行访问（读或写），例如：



其中K4M100表示将M100、M101、M102……M115共16个单元，组成一个字的单元进行读操作，（M100作为字的bit0……M115作为字的bit15），这样可提高编程效率。

3.4 状态继电器S

状态继电器S用于步进程序的设计和执处理，利用STL步进指令控制步进状态S的转移，简化编程设计。

若没有采用STL编程方式，S可当作普通的位元件，就如M变量一样来使用。状态S变量以S0, S1……S999等符号标识，其序号是以10进制方式编号。部分S变量具有掉电保存功能。如下表：

H_{1U}机型

保持用
S0-S999
1000点

H_{2U}机型

一般用		停电保持用	报警器用
S0-S499 500点 1	S0-S9 (10) 点	S500-S899 400点 2	S900-S999 100点 2

※1：非停电保持区域。通过参数的设定可变更成停电保持的区域。

※2：停电保持区域。通过参数的设定可变更成非停电保持的区域。

※3：停电保持区域，无法用参数来改变。

3.5 计时器T

计时器用于完成定时功能。每个计时器含有线圈、接点、计数值寄存器，当计时器线圈“得电”（能流有效）时，计时器开始计时，若计时值达到预设的时间值时，其接点动作，a接点（NO接点）闭合，b接点（NC接点）断开。若线圈“失电”（能流无效）时，计时器的接点恢复初始状态，计时值自动清除。也有部分计时器具有累计、掉电保持等特性，重新上电后仍维持掉电前的数值。

计时器T以T0, T1……T255等符号标识，其序号是以10进制方式编号。计时器有不同的计时步长，如有1ms、10ms、100ms等，部分具有掉电保持特性，H_{1U}、H_{2U}如下表说明：

100ms型0.1~3276.7s	10ms型 0.01~327.67s	1ms型0.001~32.767s	100ms累计型 0.1~3276.7s
T0~T199共200点；其中 T192~T199可用于中断/子程序	T200~T245 共46点	T246~T249 共4点 执行中断的保持用	T250~T255 共6点 保持用

- 没有用作定时器使用的定时器编号，也可用作数值存储用的数据寄存器。
- 定时器累计可编程控制器内的1ms, 10ms, 100ms 等的时钟脉冲，当计时的时间达到设定数值时，其触点只有在执行线圈指令或END指令时，输出触点才能动作。
- 采用程序存储器内的常数（K）作为设定值，也可用数据寄存器（D）的内容进行间接指定。注意，D的内容必须在开始计时前设定好，当开始计数后，D的数据变化只有在下一次启动计时的时候才能生效。
- 从驱动定时器的线圈开始到定时器的触点动作，可能的定时长度说明如下：

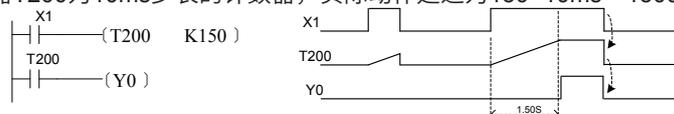
最长的情况为 $(T+T_0+a)$ ，其中：T为设定的定时时间； T_0 为程序扫描执行时间；a为定时器的计时步长。

最短的情况为 $(T-a)$ 。

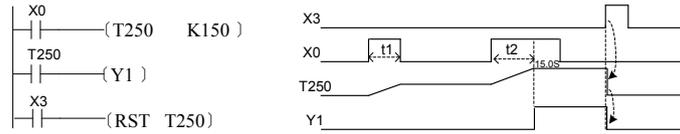
若计时器的触点指令位于线圈指令之前，最不理想的定时长度为 $(T+2T_0)$ 。

- 利用定时器的b触点，可以实现延时断开、自激振荡的输出信号等。
- PLC还提供了特殊定时器指令，如TTMR、STMR等，请参见相应指令的说明。

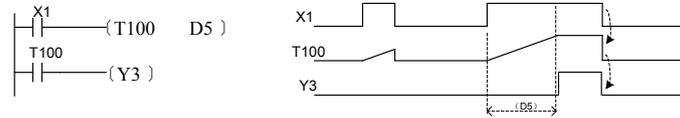
【使用举例1】：普通计时器T200为10ms步长的计数器，实际动作延迟为 $150 \times 10\text{ms} = 1500\text{ms}$ ，即1.50s，动作原理为：



【使用举例2】：对于有掉电保持的计时器T250，驱动信号为OFF，或PLC掉电时，其内部计数值维持不变，下次驱动信号为ON时，继续计时，直到满足计时到设定值时，输出触点闭合。当复位计时器线圈时，计时值清除，输出触点断开，如下图。因计数器T250为100ms步长的，实际动作延迟累计为 $150 \times 100\text{ms} = 15000\text{ms}$ ，即15.0s，即图中的 (t_1+t_2) 时间：



【使用举例3】：定时器的设定动作值可通过寄存器D来进行设定，如下图。（计数器计时过程中，若寄存器D内数值变化时，在下次计时器启动时生效。）



3.6 计数器C

计数器用于完成计数功能，每个计数器含有线圈、接点、计时数据值寄存器，每当计数器线圈的驱动信号由OFF→ON时，计数器读数增加1，若计时值达到预设的时间值时，其接点动作，a接点（NO接点）闭合，b接点（NC接点）断开；若清除计时值，输出a接点即断开，b接点（NC接点）闭合。部分计数器具有掉电保持、累计等特性，重新上电后仍维持掉电前的数值。

3
软元件
说明

计数器以C0，C1，……C255进行标识，顺序按10进制编号。

计数器中有16bit、32bit宽度；有单向计数型、增减计数型、双相计数型等，部分计数器的计数值还具有掉电保持特性，使用时根据需求选择合适的计数器。

H1U机型

16位顺/计数器 0~32,767计数		32位顺/计数器 -2,147,483,648~+2,147,483,647		
一般用	停电保持用	一般用	停电保持用	高速计数器
C0~C15 16点 1	C16~C199 184点 3	C200~C219 20点 1	C220~C234 15点 3	C235~C255 3

H2U机型

16位顺/计数器 0~32,767计数		32位顺/计数器 -2,147,483,648~+2,147,483,647		
一般用	停电保持用	一般用	停电保持用	高速计数器
C0~C99 100点 1	C100~C199 100点 2	C200~C219 20点 1	C220~C234 15点 2	C235~C255 2

※1非停电保持区域。通过设定参数可变更成停电保持区域。

※2停电保持区域。通过设定参数可变更成非停电保持区域。

※3停电保持区域。不可通过参数的设定变更。

不作为计数器使用的计数器编号，可以作为数据记忆用的数据寄存器使用。

对于32bit计数器C200~C234，由特殊辅助继电器M8200~M8234作为增计数/减计数器切换控制，见下表：

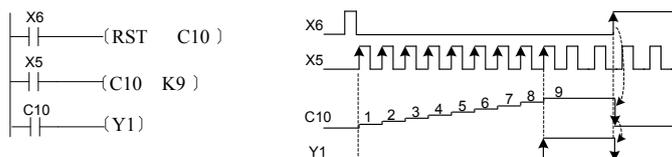
计数器NO.	方向切换	计数器NO.	方向切换	计数器NO.	方向切换	计数器NO.	方向切换
C200	M8200	C209	M8209	C218	M8218	C226	M8226
C201	M8201	C210	M8210	C219	M8219	C227	M8227
C202	M8202	C211	M8211	—	—	C228	M8228
C203	M8203	C212	M8212	C220	M8220	C229	M8229
C204	M8204	C213	M8213	C221	M8221	C230	M8230
C205	M8205	C214	M8214	C222	M8222	C231	M8231
C206	M8206	C215	M8215	C223	M8223	C232	M8232
C207	M8207	C216	M8216	C224	M8224	C233	M8233
C208	M8208	C217	M8217	C225	M8225	C234	M8234

16bit计数器与32bit计数器的特点如下表所示。可按计数方向的切换与计数范围的使用条件来分开使用。

项目	16位计数器	32位计数器
计数方向	顺数	顺/倒切换使用（见上表）
设定值	1~32,767	-2,147,483,648~+2,147,483,647
指定的设定值	常数K或数据寄存器	常数K，也可用2个D数据寄存器
当前值的变化	顺数后不变化	顺数后变化（循环计数器）
输出接点	顺数后保持动作	顺数保持动作，倒数复位
复位动作	执行RST命令时，计数器的当前值为零，输出接点复位	
当前值寄存器	16位	32位

16bit计数器

- 一般用计数器和停电保持用状态的分配，可通过系统参数配置进行变更设定。
- 对于16bit计数器，其有效设定值为 K1~K32,767（10进制常数）；设定值 K0和K1具有相同效果，即在第一次计数开始时输出触点就动作。如下例：

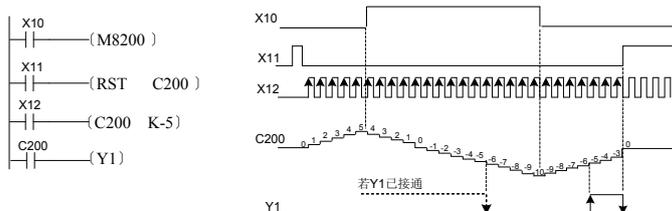


计数输入X5每驱动C10线圈一次，计数器的当前值就增加，在执行第9次的线圈指令时，输出触点动作。以后即使计数输入X5再动作，计数器的当前值不变。如果复位输入X6为ON，则执行RST指令，计数器的当前值清为0，输出触点复位。

- 计数器的设定值，除用上述常数K设定外，还可由数据寄存器编号指定。如上例中，指定D20，如果D20的内容为9，则与设定K9是一样的。
- 在以MOV等指令将设定值以上的数据写入当前值寄存器时，则在下次输入时，输出线圈接通，当前值寄存器变为设定值。
- 对于一般用计数器，如果切断可编程控制器的电源，则计数器的计数值被清除，而停电保持用的计数器则可存储停电前的计数值，因此计数器可接着上一次数值累计计数。

32bit计数器

- 对于32bit计数器，增计数/减计数的设定值有效范围为-2,147,483,648~+2,147,483,647（10进制常数），可用常数K或数据寄存器D的内容进行设定。利用特殊的辅助继电器 M8200~M8234 指定增计数/减计数的方向，如果对C△△△驱动M8△△△置1，则为减计数，不驱动时清零，则为增计数。



当前值的增减与输出触点的动作无关，但是如果从 2,147,483,647开始增计数，再输入一个脉冲后，则成为-2,147,483,648。同样，如果从-2,147,483,648开始减计数，再输入一个脉冲，则成为2,147,483,647。（这类动作被称为环形计数）；如果复位输入X11为ON，则执行RST指令，计数器的当前值变为0，输出触点也复位。

- 使用供停电保持用的计数器时，计数器的当前值、输出触点动作与复位状态停电保持。
- 32bit计数器也可作为32bit数据寄存器使用。但是，32bit计数器不能作为16位应用指令中的软元件。
- 在以DMOV指令等把设定值以上的数据写入当前值数据寄存器时，则在以后的计数输入时可继续计数，触点也不变化。
- 对于16bit计数器，最高位（bit15）为符号位，处理的数据为0~32767范围，即只能为正数；
- 对于32bit计数器，最高位（bit31，即高字节的最高位）为符号位，处理的数据范围为-2,147,483,648~2,147,483,647；

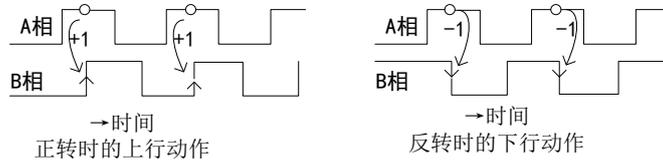
高速计数器

H_{1U}/H_{2U}系列PLC的内置高速计数器如下表所示，按计数器的编号（C）分配在输入 X000~X007。

而不作为高速计数器使用的X输入端口可在顺控程序内作为普通的输入继电器使用。此外，不作为高速计数器使用的高速计数器编号也可作为数值存储用的32位数据寄存器使用。

- 高速计数器有如下几种类型：

- 1) 1相1计数型，只需要1个计数脉冲信号输入端，由对应的特殊M寄存器决定为增计数或减计数；部分计数器还具有硬件复位、起停的信号输入端口；
- 2) 1相2计数型，有2个计数脉冲信号输入端，分别为增计数脉冲输入端和减计数脉冲输入端；部分计数器还具有硬件复位、起停的信号输入端口；
- 3) 2相2计数型，即AB两相计数脉冲计数器，是根据AB两相的相位决定计数的方向，计数方法是：当A脉冲为高电平时，B相的脉冲上升沿作加计数，B相的脉冲下降沿作减计数。通过读取M8251-M8255 的状态，可监控 C251-C255 的增计数/减计数状态。



双相式编码器输出的是有90度相位差的A相和B相，据此高速计数器自动地进行增计数/减计数动作。

- 通过特殊变量的设定，可以进行4倍频的AB相计数，可提供计数精度。
- 部分计数器还具有硬件复位、起停的信号输入端口。

项目	单相单计数输入	单相双计数输入	双相双计数输入
计数方向的指定	根据M8235-M8245的启动与否，C235-C245作增/减计数。	对应于增计数输入或减计数输入的动作，计数器自动地增/减计数。	A相输入处于ON同时，B相输入处于OFF→ON时增计数动作，ON→OFF时减计数动作。
计数方向监控	—	通过监控M8246-M8255，可以知道增（OFF）减（ON）情况	

[U]: 增计数输入; [D]: 减计数输入; [A]: A相输入;
[B]: B相输入; [R]: 复位输入; [S]: 启动输入

增计数/减计数切换用特殊辅助继电器

种类	计数器号	UP/DN指定
单相单计数输入	C235	M8235
	C236	M8236
	C237	M8237
	C238	M8238
	C239	M8239
	C240	M8240
	C241	M8241
	C242	M8242
	C243	M8243
	C244	M8244
	C245	M8245

计数方向监控用特殊辅助继电器

种类	计数器号	UP/DN监控
单相双计数输入	C246	M8246
	C247	M8247
	C248	M8248
	C249	M8249
	C250	M8250
双相双计数输入	C251	M8251
	C252	M8252
	C253	M8253
	C254	M8254
	C255	M8255

- 高速计数器编号与对应的X端口配套使用，即指定了高速计数器Cxxx后，对应的X输入端即被指定，故编程时不要让X端口有重复使用的情况，否则会出错。定义如下表：

分配输入	单相单计数输入										
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245
X000	U/D						U/D			U/D	
X001		U/D					R			R	
X002			U/D					U/D			U/D
X003				U/D				R			R
X004					U/D				U/D		
X005						U/D			R		
X006										S	
X007											S

3
软元件
说明

双计数及A/B相计数器如下表：

分配输入	单相双计数输入					A/B相计数				
	C246	C247	C248	C249	C250	C251	C252	C253	C254	C255
X000	U	U		U		A	A		A	
X001	D	D		D		B	B		B	
X002		R		R			R		R	
X003			U		U			A		A
X004			D		D			B		B
X005			R		R			R		R
X006				S					S	
X007					S					S

- 【注意】1) M8166为高速计数器禁止X端口电平复位（仅保留沿中断复位）的标志，默认不禁止。
 2) M8169为高速计数器禁止通过X端口进行复位（无论是电平复位还是沿中断复位）的标志，默认不禁止。

U：上升输入；D：下降输入；A：A相输入；B：B相输入；R：复位输入；S：开始输入

H_{1U}-XP机型1相输入的X0、X1输入点可达60kHz，X2~X5输入点可达10kHz，总频率不超过70kHz。

H_{2U}-XP机型1相输入的X0~X5输入点均可达100kHz。

AB相的最大输入频率要降半使用，如果是采用四倍频，则降为四分之一，例如H_{2U}机型的C251为AB相计数，占用X0、X1，在一倍频的时候最大输入频率是30 kHz，在四倍频时(M8195为ON)最大输入频率是15 kHz。

不作高速计数器使用的输入端子、可以作一般输入使用。

【表的阅读举例1】

表中C235为单相单输入计数，使用X0输入口，不需要中断复位与中断启动端口；

如果使用C235计数器，即默认使用了X0输入端口，便不可再使用C241，C244，C246，C247，C249，C251，C252，C254和中断I00口或者M8170（脉冲捕捉），因为这些计数器、中断、脉冲捕捉也需用到X0端口，形成了端口冲突。

【表的阅读举例2】

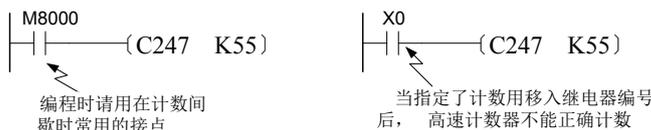
表中C254为2相2输入计数器，即AB相计数器，

X0口作为A相输入，X1口作为B相输入，X2口作为中断复位输入，X6口作为中断启动输入；

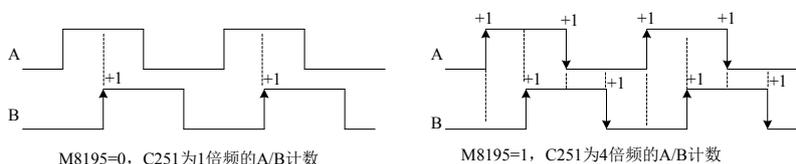
如果使用C254计数器，即默认使用了X0、X1、X2、X6输入端口，与这些端口相关的计数器、中断口或者脉冲捕捉等，便都不能再使用了。

计数器使用说明：

- 高速计数器根据特定的输入执行动作，在相关信号的跳变沿，采用中断方式处理进行高速动作，故与PLC的扫描时间无关。
- 高速计数器的当前值达到设定值时，如要立即进行输出处理，请使用高速脉冲比较指令HSCS、HSCR、HSZ等应用指令，具体参见指令解释。
- 高速计数器的当前值达到设定值时，如要立即进行一些逻辑处理，可使用高速计数中断，使用高速脉冲比较指令HSCS，将指令的操作指定为I0x0中断（其中x=1~6中断号），当然必须编写好对应中断号的子程序。
- 高速计数器的线圈驱动用触点，在高速计数时，请采用一直接通的触点。



- 如果对高速计数器的线圈编程，则与其对应的输入继电器的输入滤波器会自动变为20μs(X000，X001)，或50μs(X002-X005)（初始值为10ms）。此外，不作为高速计数器输入使用的输入继电器的输入滤波器维持初始值10ms。
- A/B相高速计数器C251~C255有1倍频和4倍频两种频率模式，分别由特殊寄存器M8195~M8199设定，见下列：



- 高速计数器均采用了硬件方式计数，对输入脉冲的总频率没有软件方面的限制；双相高速计数器的信号，占用两个脉冲输入口，对PLC的等效脉冲数影响按2倍计算，若C251~C255的A/B输入4倍频模式时，为软件计数模式，高速输入频率降为15kHz。
- 由于高速X计数、高速Y脉冲输出均采用中断方式进行处理，故信号路数较多时，可能会影响程序的执行速度，向高速计数器输入信号时，其所用频率要低于上述频率。如果输入超过这一频率的信号，可能会发生监视定时器（WDT）错误。
- 多个高速计数器同时或者高速计数器和（SPD）、（HSCS）、（HSCR）、（HSZ）、以及（PLSY）、（PLSR）等高速输出同时使用时，高速计数器需降额使用。可捕捉频率范围单相计数器最高10kHz、双相计数器5kHz。

3.7 寄存器D

数据寄存器D

寄存器用于数据的运算和存储，如对定时器、计数器、模拟量参数的运算和存储等，每个寄存器的宽度为16bit。若采用32bit指令，则自动将相邻的2个寄存器组成为32bit寄存器使用，地址较低的为低字节，而地址较高的为高字节。

H1U/H2U系列PLC多数指令中参与运算的数据是按有符号数进行处理的，对于16bit的寄存器，bit15为符号位，0表示正数，1表示负数（对于32bit的寄存器，高字节的bit15为符号位），数值范围为-32,768~+32,767。

当需要处理32bit的数据时，可将相邻的2个D寄存器组成为32bit双字，例如以32bit格式访问D100时，此时将高地址D101寄存器作为高字，同时将高字节的bit15作为双字的符号位，可处理-2,147,483,648~2,147,483,647的数值。

寄存器以D0, D1, ……D9, 999为标识，按10进制进行编号。

H1U机型

一般用	停电保持用	特殊用	变址用
D0~D127 128点 1	D128~D7999 7872点 3	D8000~D8511 512点	V0~V7 Z0~Z7

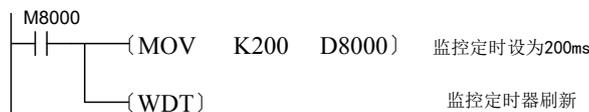
H2U机型

一般用	停电保持用	停电保持专用	特殊用	变址用
D0~D199 200点 1	D200~D511 312点 2	D512~D7999 7488点 3	D8000~D8511 512点	V0~V7 Z0~Z7

- ※1: 非停电保持区域。通过设定参数可变更成停电保持区域。
- ※2: 停电保持区域。通过设定参数可变更成非停电保持区域。
- ※3: 通过设定参数无法变更停电保持的特性。

⚠ 注：具有CANlink3.0功能的H1U-XP（26128版本及以上）和H2U-XP（24137版本及以上），因该功能占用D7000-D7999的D元件，D7000-D7999不允许用户使用！

- 以两个相邻的数据寄存器表现32位的数据。（高位为大的号码，低位为小的号码。在变址寄存器中，V为高位，Z为低位）。在指定32位时，如果指定了低位（例如：D0），则高位为继其之后的编号（例如，D1）被自动占用。低位可用奇数或偶数的任意一种软件编号指定，考虑到外围设备的监视功能，建议低位采用偶数软件编号。
- 一旦在数据寄存器中写入数据，只要不再写入其他数据，就不会变化。但是，在RUN→STOP时或停电时，所有数据被清除为0。（如果驱动特殊的辅助继电器M8033，则可以保持。）对此相对停电保持用的数据寄存器在RUN/STOP和停电时也可保持其内容。
- 利用系统参数配置功能，可改变D寄存器的一般用与停电保持用的分配；而且将停电保持专用的数据寄存器作为一般用途时，请在程序的起始步采用RST或ZRST指令，以清除其内容。
- 在使用PLC间简易链接或并联链接的情况下，一部分的数据寄存器作为默认区域被占用。
- 特殊用途的数据寄存器是指写入特定目的的数据，用于实现控制器的一些特殊功能，可理解为用户程序与PLC系统程序进行数据交互的特殊单元。例如，在D8000中，监视定时器的时间通过系统ROM进行初始设定，要将其改变时，利用MOV传送指令，在D8000中写入目标时间。



另外还有一些特殊D寄存器，用于系统工作状态参数缓存，查询这些寄存器，可用于判断运行参数。

- 关于特殊数据寄存器的停电保持特点请参照“特殊寄存器说明”。
- 数据寄存器可以处理各种数值数据，通过利用它，可以进行各种控制。如作为定时器与计数器的设定值被指定，用于数

3
软元件
说明

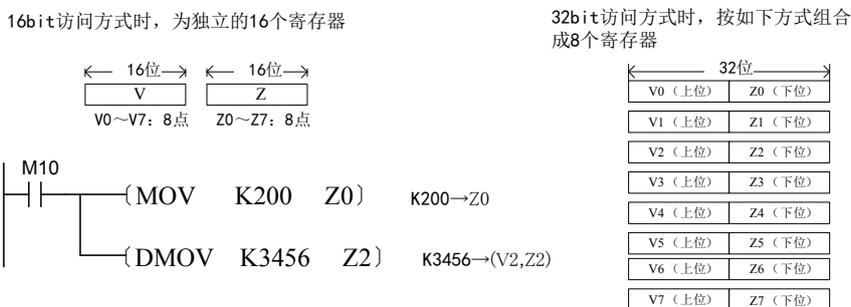
据的各种运算等，在后续的指令解释中，对支持使用D寄存器的指令有详细的说明。

数据寄存器V、Z

变址寄存器V与Z同普通的数据寄存器一样，是进行数值数据的读入、写出的16位数据寄存器。V0~V7，Z0~Z7共有16个。

变址寄存器除了和普通的数据寄存器有相同的使用方法外，在应用指令的操作数中，还可以同其他的软件编号或数值组合使用。但需注意LD，AND，OUT等基本顺控指令或步进梯形图指令的软件编号不能同变址寄存器组合使用。

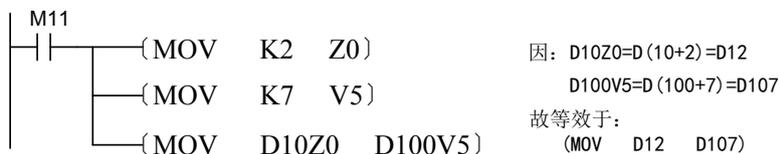
V、Z寄存器可采用16bit和32bit方式进行访问，如下图说明：



3
软件说明

按照惯例，在处理32位应用指令中的软件编号或处理超过16位范围的数值时，（为32bit寄存器方式），V（高位）、Z（低位）被同时访问，指定的寄存器名必须为Z0~Z7。即使指定了V0~V7的高位侧，也无法进行变址。

16bit变址应用举例：



32bit变址应用举例：



常数变址的特例：



当V、Z间接寻址方式用于循环指令中（V、Z随循环变量变化），进行成片数据区的操作，或用于查表操作等，简化编程，提高指令效率。

3.8 子程序与中断指针P、I

指针（P）用于跳转程序的入口地址和子程序起始地址的标识；指针（I）则用于中断程序的起始地址标识，其编号采用十进制数分配，如下表所示：

H_{1U}机型

分支用	结束跳转用	输入中断用	定时中断用	高速计数器中断用	多用户中断 (XP不具备此功能)
P0~P62; P64~P127: 共127点	P63	I00x(X0) I10x(X1) I20x(X2) I30x(X3) I40x(X4) I50x(X5)	I600 I700 I800	I010 I020 I030 I040 I050 I060	I507~I530 共24点
P128~P131: 共4点，加密程序	共1点	x=1上升沿中断 x=0下降沿中断 共12点（注）	共3点	共6点	

H2u机型

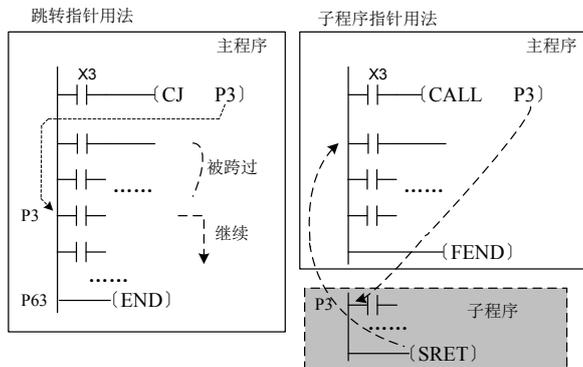
分支用	结束跳转用	输入中断用	定时中断用	高速计数器中断用	脉冲完成中断	多用户中断 (XP不具备此功能)
P0~P62; P64~P127: 共127点	P63	I00x(X0) I10x(X1) I20x(X2) I30x(X3) I40x(X4) I50x(X5)	1600 1700 1800	I010 I020 I030 I040 I050 I060	I502(Y0) I503(Y1) I504(Y2) I505(Y3) I506(Y4)	I507~I530 共24点
P128~P131: 共4点, 加密子程序	共1点	x=1上升沿中断 x=0下降沿中断 共12点 (注)	共3点	共6点	共5点	

※注1: 加强功能版本的允许输入中断数有扩展, 请参见附录8.6增强功能说明。

※注2: 支持4个加密子程序, 子程序可单独加密, 每个子程序最大480步。

因外部输入中断、高速计数、脉冲频率测量等功能都是通过X0~X7端口输入的, 故这几项功能所使用的X端口不能有重复使用的现象, 故使用输入中断指针时, 注意端口的功能安排, 检查高速计数器、脉冲密度指令所用的输入端口号情况。

跳转指针 (P) 和子程序指针 (P) 的使用及差别如下图所示。跳转指针 (P) 引导的指令语句仍在主程序内, 只是用于在满足条件是跨过一部分指令语句; 但子程序指针 (P) 则用于一段子程序, 若主程序中条件满足, 调用子程序, 在子程序执行完毕 (SRET) 后, 要返回原调用 (CALL) 指令的下一步继续执行。



- 两种P指针使用同一种编号体现, 在定义P指针时不要有重复;
- P63指针为专用指针编号, 指向程序的结束语句END, 注意不要再对P63编程。

指针 (I) 用于指定中断程序的起始地址, 而中断子程序是在“中断允许”的情况下, 当信号条件满足的瞬间, PLC系统暂停主程序的正常执行 (记住当前暂停点), 从指定的I指针所指定的地址入口, 开始执行中断子程序, 直到执行了IRET指令后, 返回主程序的暂停点, 继续执行。因PLC系统对中断信号采取了高优先的响应处理, 故不受扫描时间的影响。

PLC系统为H1u提供了两种类型的中断, 分别是X输入中断和计数器中断; 为H2u提供了五种类型的中断, 介绍如下 (前两个包含H1u) :

1) X输入中断: 控制器的X0~X5可分别设定为中断输入端口, 每个中断输入口又有上升沿中断、下降沿中断, 通过中断号来进行划分: 如“1100”中断号代表X1端口的下降沿中断, 而“1101”则代表X1端口的上升沿中断。

2) 计数器中断: 根据可编程控制器内置的高速计数器的比较结果 (HSCS), 执行中断子程序, 优先处理计数结果的控制。

当HSCS指令的输出目标设为I010~I060时, 便使用了高速计数器中断, 编程时需编制好相应的中断子程序, 开启响应的中断允许标志, 才能进行中断响应。

3) 定时器中断: 在各指定的中断循环时间 (1ms-99ms) 执行中断子程序。在需要有别于可编程控制器的运算周期的循环中断处理控制中使用。

系统提供了3个定时中断, 定时中断的周期可编程决定。定时中断使用系统内部的定时器, 不占用T0~T255。

4) 通过使用特殊位M8090~M8094(分别对应Y0~Y2)为ON, 可以实现脉冲输出完成后执行一次用户中断; 在指定脉冲个数发送完毕后, 立即执行用户中断I502~I506。

5) 为了满足在高速计速器运行时, 支持多高速自由任务, 实现了高速计速器多用户中断 (XP不具备此功能) (最大支持24个, 均为扩展的中断号), 设定和比较用数据表格的方式定义。

对应中断的“中断允许”标志如下表, 各标志可以独立设置:

中断允许/禁止设置			
M8050	驱动I00□中断禁止	X输入中断，共有12个中断，分别对应X0~X5端口的上升沿中断、下降沿中断。 □中： 1=上升沿中断； 0=下降沿中断	每个标志对应1个外部中断的控制； 当该M标志为OFF时，允许对应的X中断； 当该M标志为ON时，禁止对应的X中断；
M8051	驱动I10□中断禁止		
M8052	驱动I20□中断禁止		
M8053	驱动I30□中断禁止		
M8054	驱动I40□中断禁止		
M8055	驱动I50□中断禁止		
M8056	驱动I600中断禁止	定时中断0	为ON时，禁止I010~I060的中断
M8057	驱动I700中断禁止	定时中断1	
M8058	驱动I800中断禁止	定时中断2	
M8059	驱动计数器中断禁止	高速计数中断，共6个	

3
软元件
说明

端口号	使用特殊位	对应的用户中断
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504
Y003	M8093	I505
Y004	M8094	I506

标志位	使用描述
M8084	为ON使能高速计数器多用户中断（XP不具备此功能）
D8084	为高速计数器序号235~255
D8085	对应的用户中断个数，最大24个，从I507~I530
D8086	对应多个比较点数据的序号，只能为D元件，且为双字宽度，如200为D200开始的双字

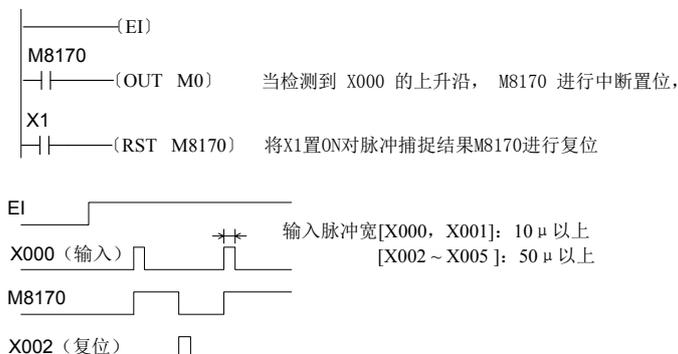
每个中断对应的“中断允许”标志开启后，还需要开启“全局中断允许”，即执行EI指令（FNC04）后才最后才能使能中断功能；若执行全局中断禁止DI指令（FNC05），则禁止所有的中断的响应。当启用了输入编号的中断允许设定标志，输入信号满足中断设定时，将执行对应的中断子程序。

每个中断子程序的末尾均要有IRET指令，以表示中断子程序完毕，PLC执行了该语句后，便会跳回本中断程序开始执行之前的位置。（AutoShop软件中中断程序不需要写IRET指令）

若需要对出现在X0~X5端口的瞬间脉冲信号作出反应，但对反应动作时间没有特别要求，就可以使用“脉冲捕捉”功能，PLC会将出现在X0~X5端口的上升沿信号保存在M8170~M8175单元，主程序中可作为判断处理的依据，响应处理完毕，可人为将之清除。

M8170~M8175的具体使用如下：

执行FNC04(EI)指令后，当输入继电器X000~X005 OFF→ON变化时，特殊辅助继电器M8170~M8175置位进行中断处理。为了再次获得输入，必须利用程序对设定的元件进行复位操作。脉冲捕捉动作同个别中断禁止用辅助继电器M8050~M8055的动作无关。



关于子程序和中断的描述请参考4.3.1章FNC00~FNC05和附录8.6的详细说明。

3.9 常数K、H

H1U/H2U系列可编程控制器根据不同的用途和目的，使用5种类型的数值。其作用和功能如下：

类型	编程中应用说明
十进制数，DEC	定时器和计数器的设定值（K常数） 辅助继电器（M），定时器（T），计数器（C），状态S等的编号（软元件编号） 指定应用指令操作数中的数值与指令动作（K常数）
十六进制数，HEX	同10进制数一样，用于指定应用指令中的操作数与指定动作（H常数）
二进制，BIN	以十进制数或十六进制数对定时器、计数器或数据寄存器进行数值指定，但在可编程控制器内部，这些数字都用二进制数处理。而且，在外围设备上进行检查时，这些软元件将如图所示自动变换为十进制数（也可切换为16进制）
八进制，OCT	输入继电器、输出继电器的软元件编号以8进制数值进行分配。因此，可进行 [0-7, 10-17……70-77, 100-107] 的进位，在8进制数中，不存在 [8, 9]
BCD	BCD是以4位二进制表示十进制数各位0-9数值的方法。各位的处理很容易，因此，可用于BCD输出形的数字式开关或七段码的显示器控制等方面
BIN浮点数	可编程控制器具有可进行高精度的浮点运算功能，内部用二进制（BIN）浮点数进行浮点运算
十进制浮点数	十进制浮点值只用于监视，便于阅读。

3
软元件
说明

常数K

[K]是表示10进制整数的符号。主要用于指定定时器或计数器的设定值或应用指令操作数中的数值。16bit指令中，常数K的取值范围为-32768~32767；32bit指令中，常数K的取值范围为-2, 47, 483, 648~2, 147, 483, 647。

常数H

[H]是16进制数的表示符号。主要用于指定应用指令的操作数的数值。常数H的取值范围为0000~FFFF；32bit指令中，常数K的取值范围为0000, 0000~FFFF, FFFF。

3.10 控制器软元件规格

H1U机型的停电和非停电保持区域不能更改，H2U机型有部分区域可进行更改。

H1U和H1U-XP机型

输入端口X	X0~X377, 最大可达256点; XY总和最大256点		X0~X5: 具有中断功能;		8进制命名规则		
输出端口Y	Y0~Y377, 最大可达256点; XY总和最大256点		晶体管型具3路高速脉冲输出功能, 具体规格请看用户手册		8进制命名规则		
辅助继电器M	[M0~M383] 384点 一般用		[M384~M3071] 2688点 保存用		M8000~M8511 512点 特殊用		
状态S	[S0-S999] 1000点 全部保持用						
定时器T	T0~T199 200点 100ms 一般用		T200~T245 46点 10ms 一般用	[T246~T249] 4点1ms累计 保持用	[T250~T255] 6点100ms累计 保持用		
计数器C	16位增计数器		32位增减计数器		高速计数器		
	C0~C15 16点一般用	[C16~C199] 168点 保持用	C200~C219 20点一般用	[C220~C234] 15点保持用	C235~C245单 相单向计数	C246~C250 单相双向计数	C251~C255 二相计数器
数据寄存器D, V, Z	D0~D127 128点 一般用	[D128~D7999] 7872点 保持用	[D1000~D7999] 最大7000点	[D8000~D8511] 512点 特殊用	V7~V0, Z7~Z0 16点 变址用		
嵌套指针	N0~N7 8点主控用	P0~P127共128点, 跳转子程序; P128~P131共4点, 加密子程序	I010~I060共6 点, 计数中断指针	I6**~I8**共3点, 定时中断指针	I00*~I50* 6点 输入中断用指针		
常数	K	16位 -32,768~32,767		32位 -2,147,483,648~2,147,483,647			
	H	16位 0~FFFFH		32位 0~FFFFFFFFH			
	E(浮点数)	-		32位 1175×10 ⁻⁴¹ ~3402×10 ³⁵			

H2U和H2U-XP机型:

输入端口X	X0~X377, 最大可达256点; XY总和最大256点		X0~X5: 具有中断功能; X0~X7: 滤波时间可设;		8进制命名规则
输出端口Y	Y0~Y377, 最大可达256点; XY总和最大256点		晶体管型具5路高速脉冲输出功能, 具体规格请看用户手册		8进制命名规则
辅助继电器M	M0~M499 500点, 通用I	[M500~M1023] 524点, 保存用2继电器	[M1024~M3071] 2048点, 保存用3	M8000~M8511 512点, 特殊用	
状态S	S0~S499共500点 1 初始用S0~S9		[S500~S899]共400点, 掉电保存用2		[S900~S999]共100点, 报警用2
定时器T	[T0~T199]共200点, 100ms。 子程序用: T192~T199	T200~T245 共46点, 10ms	[T246~T249]共4点, 1ms累计3	[T250~T255]共6点, 100ms累计3	
16位向上计数器C	[C0~C99]100点, 通用1			[C100~C199]100点, 保存用2	
32位计数器C	32位可逆			32位高速计数可逆	
	C200~C219 20点, 通用1	[C220~C234] 15点, 掉电保存用2	[C235~C245] 单相单计数输入2	[C246~C250]单相双计数输入2	[C251~C255] 2相计数输入2
数据寄存器D, V, Z	D0~D199共200点, 通用1	[D200~D511] 共312点, 保存用2	[D512~D7999] 共7488点, 保存用3	[D8000~D8511] 共512点, 特殊用	V7~V0, Z7~Z0 共16点, 变址用
嵌套指针	N0~N7 8点, 主控用	P0~P127共128点, 跳转子程序; P128~P131共4点, 加密子程序	100*~150*共6点, 输入中断指针	I6**~I8**共3点, 定时中断指针	I010~I060共6点, 计数中断指针
常数	K (十进制)	16位-32,768~32,767		32位-2,147,483,648~2,147,483,647	
	H (十六进制)	16位	0~FFFFH	32位	0~FFFFFFFFH
	E (浮点数)	-		32位	1175×10 ⁻⁴¹ ~3402×10 ³⁵

[]内的元件为电池保存区

※1: 非电池保存区。根据参数设定, 可以变更为电池保存区。

※2: 电池保存区。根据参数设定, 可以变更非电池保存区。

※3: 电池保存固定区, 区域特性不能变更。

㊦ 注1: H1U-XP (26128版本及以上) 和H2U-XP (24137版本及以上) 具有CANlink3.0功能, 因该功能占用D7000-D7999的D元件, 使用CANlink3.0功能后D7000-D7999不允许用户使用; 不使用CANlink3.0功能时D7000-D7999用户可正常使用!

㊦ 注2: 系统高速输入输出资源共56个单元, 使用情况如下;

项目	占用单元数
高速输出	7
单相单计数	6
单相双计数	13
AB相位计数	17
SPD测速	4
外部中断	1
脉冲捕捉	1

例如系统使用情况: 3路脉冲输出, 2路单相计数(剩下的4路都为脉冲捕捉); 所用的资源为 $3*7+2*6+4*1=37$ 个单元。



第四章 指令

4.1 基本指令	38
4.2 步进顺控指令	42
4.2.1 STL编程指令	42
4.2.2 SFC顺序功能图编程	43
4.2.3 SFC编程实例	49
4.3 应用指令	58
4.3.1 程序流程 (00~09)	60
4.3.2 传送与比较 (10~19)	69
4.3.3 四则逻辑运算 (20~29)	79
4.3.4 循环移位 (30~39)	89
4.3.5 数据处理 (40~49)	96
4.3.6 高速处理 (50~59)	106
4.3.7 方便指令 (60~69)	121
4.3.8 外围设备I/O (70~79)	130
4.3.9 外围SER设备 (80~93)	141
4.3.10 浮点数 (110~147)	158
4.3.11 定位控制 (150~159)	184
4.3.12 时钟运算 (160~169)	196
4.3.13 外围设备 (170~177)	203
4.3.14 接点比较 (224~246)	206

第四章 指令

4.1 基本指令

在基本指令当中，有部分指令采用“功能号”编码方式，若以手持编程器输入程序，输入方式可使用键盘中相对应的指令按键输入或使用功能编号方式输入。

指令符号	FUN NO	功能	操作数类型	指令步长	指令符号	FUN NO	功能	操作数类型	指令步长
LD		加载常开接点	S、X、Y、M、T、C	1	ORF	95	或脉冲(F)下降沿检测并行连接		
LDI		加载常闭接点	S、X、Y、M、T、C	1	OUT		驱动线圈	S、Y、M	1
LDP	90	取脉冲上升沿	S、X、Y、M、T、C	1	SET		置位动作保存线圈指令	S、Y、M	1
LDF	91	取脉冲下降沿	S、X、Y、M、T、C	1	RST		接点或缓存器清除	S、Y、M、T、C、D	3
AND		串联常开接点	S、X、Y、M、T、C	1	PLS		脉冲上升沿检测线圈指令		
ANI		串联常闭接点	S、X、Y、M、T、C	1	PLF		脉冲(F)下降沿检测线圈指令		
ANB		串联回路方块	无	1	MC		主控公用串行接点用线圈指令	N0~N7	3
ANDP	92	与脉冲上升沿检测串行连接		3	MCR		主控复位公用串行接点解除指令	N0~N7	3
ANDF	93	与脉冲(F)下降沿检测串行连接		3	MPS		存入堆栈	无	1
OR		并联常开接点	S、X、Y、M、T、C	1	MRD		读出堆栈(能流指针不变)	无	1
ORI		并联常闭接点	S、X、Y、M、T、C	1	MPP		读出堆栈	无	1
ORB		并联回路方块	无	1	NOP		无动作	无	1
ORP	94	或脉冲上升沿检测并行连接		3	INV	98	运算结果取反	无	1
					END		程序结束	无	1
					P		指针	0~127	1
					I		中断插入指针	I101/ I201/301等	1

基本指令解释

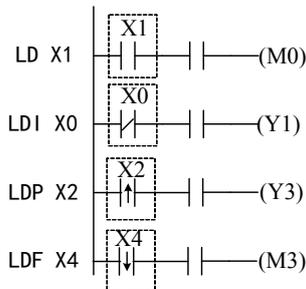
指令	操作数						
LD	X0~	Y0~	M0~M3071	S0~	T0~	C0~	D0~
LDI	X377	Y377	M8000~M8255	S999	T255	C255	D8255
LDP	✓	✓	✓	✓	✓	✓	
LDF							

LD/LDI/LDP/LDF指令用于左母线开始的接点，其中：

LD/LDI指令分别是把A接点和B接点的当前能流状态保存，同时把取来的接点状态存入累计缓存器内。

LDP指令用于取用接点信号的上升沿，若本次扫描中检测到对应信号的上升跳变，则触点有效，下一次扫描时，触点即变成无效。

LDF指令用于取用接点信号的下降沿，若本次扫描中检测到对应信号的下降跳变，则触点有效，下一次扫描时，触点即变成无效。

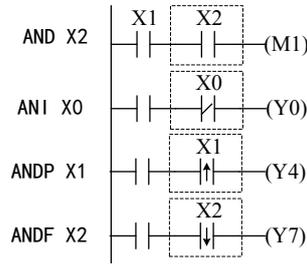


指令	操作数						
AND	X0~	Y0~	M0~M3071	S0~	T0~	C0~	D0~
ANI	X377	Y377	M8000~M8255	S999	T255	C255	D8255
ANDP	✓	✓	✓	✓	✓	✓	
ANDF							

4
指令

AND/ANI/ANDP/ANDF指令用于串联接点的状态运算，其操作是先读取目前所指定串联接点的状态再与接点之前逻辑运算结果作“与”（AND）的运算，并将结果存入累计缓存器内。

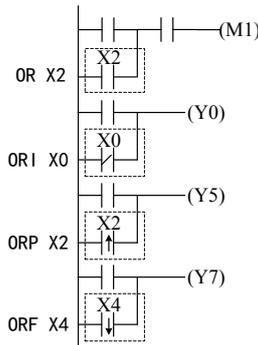
- AND/ANI指令分别是将A接点和/B接点的状态参与AND运算；
- ANDP指令是将接点的上升沿跳变状态参与AND运算；
- ANDF指令是将接点的下降沿跳变状态参与AND运算；



指令	操作数						
OR	X0~	Y0~	M0~M3071	S0~	T0~	C0~	D0~
ORI	X377	Y377	M8000~M8255	S999	T255	C255	D8255
ORP	✓	✓	✓	✓	✓	✓	
ORF							

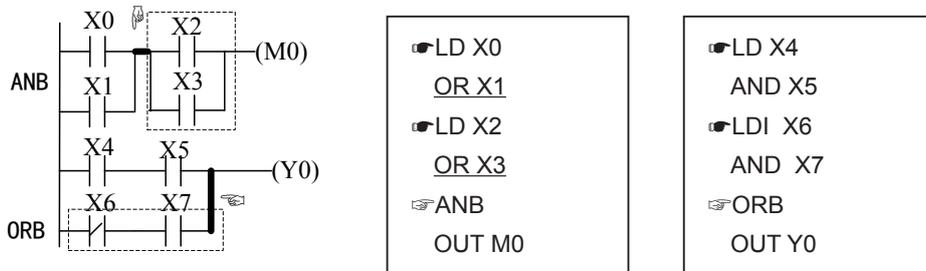
OR/ORI指令用于并联接点的状态运算，其操作是先读取目前所指定接点的状态，再与接点之前逻辑运算结果作“或”（OR）的运算，并将结果存入累计缓存器内。

- OR/ORI指令分别是将A接点和/B接点的状态参与OR运算；
- ORP指令是将接点的上升沿跳变状态参与OR运算；
- ORF指令是将接点的下降沿跳变状态参与OR运算。



指令	操作数
ANB	无。
ORB	参与块运算的是最近两次LD（或LDI/LDP/LDF）区间的计算能流。

ANB和ORB是将前一保存的逻辑结果与目前累计缓存器的内容作“与”和“或”的运算。

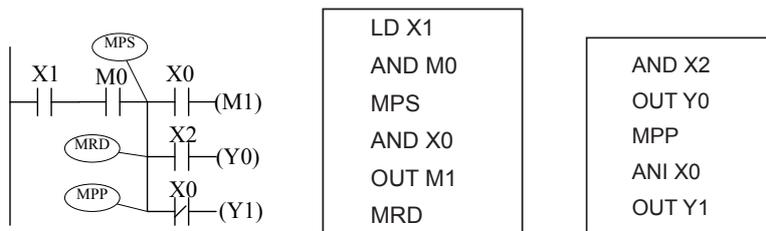


指令	操作数
MPS MRD MPP	无

- MPS: 将目前累计缓存器的内容存入堆栈。（堆栈指针加一）
- MRD: 读取堆栈内容存入累计缓存器。（堆栈指针不动）

4
指令

MPP: 自堆栈取回前一保存的逻辑运算结果, 存入累计寄存器。(堆栈指针减一)

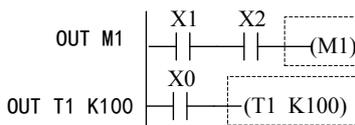


```
LD X1
AND M0
MPS
AND X0
OUT M1
MRD
```

```
AND X2
OUT Y0
MPP
ANI X0
OUT Y1
```

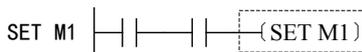
指令	操作数						
OUT	X0~ X377	Y0~ Y377	M0~M3071 M8000~M8255	S0~ S999	T0~ T255	C0~ C255	D0~ D8255
	□	✓	✓	✓	✓	✓	

将OUT 指令之前的逻辑运算结果输出至指定的元件。



指令	操作数						
SET	X0~ X377	Y0~ Y377	M0~M3071 M8000~M8255	S0~ S999	T0~ T255	C0~ C255	D0~ D8255
		✓	✓	✓			

当SET 指令被驱动, 其指定的组件被设定为ON, 且被设定的组件会维持ON, 不管SET指令是否仍被驱动。可利用RST指令将该组件设为OFF。



指令	操作数							
RST	X0~ X377	Y0~ Y377	M0~M3071 M8000~M8255	S0~ S999	T0~ T255	C0~ C255	D0~ D8255	V, Z
		✓	✓	✓	✓	✓	✓	✓

当RST 指令被驱动, 其指定的组件被设定为OFF, 且被设定的组件会维持OFF, 不管RST 指令是否仍被驱动。可利用SET 指令将该组件设为ON。

RST指令也可用于D、V、Z变量复位, 将指定D、V、Z元件的值清为0。



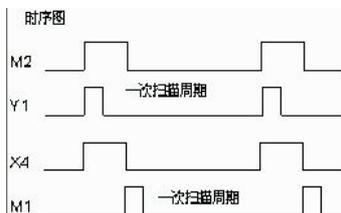
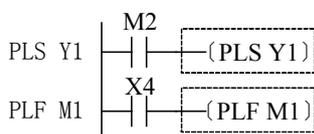
元件	操作结果
S, M, Y	线圈及接点被设定为OFF
T, C	目前计时或计数值会被设为0, 且线圈及接点被设定为OFF。
D, V, Z	元件的值清为0

指令	操作数						
PLS PLF	X0~ X377	Y0~ Y377	M0~M3071	S0~ S999	T0~ T255	C0~ C255	D0~ D8255
		✓	✓				

当PLS 指令被上升沿驱动时, 其指定的元件被设定为ON状态, 该ON状态仅持续1个扫描周期;

当PLF 指令被下降沿驱动时, 其指定的元件被设定为ON状态, 该ON状态仅持续1个扫描周期。

指令举例:



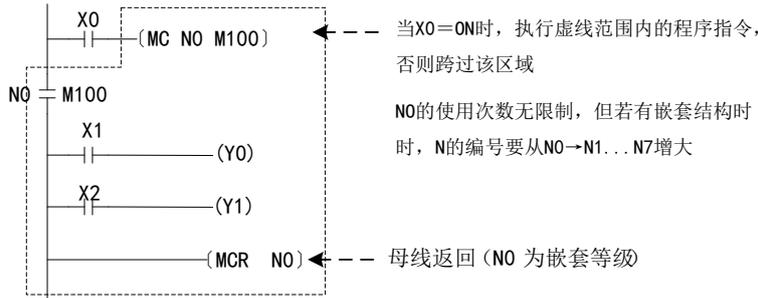
指令	操作数
MC MCR	N0~N7

MC为主控起始指令，当MC指令执行时，位于MC与MCR指令之间的指令照常执行。当MC指令OFF时，位于MC与MCR指令之间的指令动作如下所示：

定时器	计时值归零，线圈失电，接点不动作
计数器	线圈失电，计数值及接点保持目前状态
OUT指令驱动的线圈	全部不受电
SET, RST指令驱动的组件	保持目前状态
应用命令	全部不动作

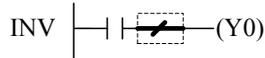
MCR为主控结束指令，置于主控程序最后，在MCR指令之前不可有接点指令。

MC-MCR主控程序指令支持巢状程序结构，最多可8层，使用时依N0~N7的顺序，



指令	操作数
INV	无

将INV指令之前的逻辑运算结果反相后存入累加寄存器内。当INV指令之前能流为ON，经过INV后能流变为OFF；反之，变为ON。



指令	操作数
NOP	无

指令NOP在程序不做任何运算，因此执行后仍会保持原逻辑运算结果，没有实际操作，在AutoShop编译时，会自动将之删除，减少程序空间的浪费，加快运行速度。

指令	操作数
FEND END	无

在主程序的末尾才加入FEND指令，以指明用户主程序的结束。

只在梯形图程序或指令程序最后才加入END指令。PLC执行时由用户程序的地址0扫描到END指令，执行之后，返回到地址0重新作扫描执行，对超过END指令之后的程序空间不再处理。在AutoShop编程环境中，无需用户输入FEND或END指令，系统在下载时会自动加入。

指令	操作数
P	P0~P127 用于标记主程序中跳转的地址起始，其中P63为指向END的专用地址。 用于标记子程序的起始地址，每个子程序都以SRET为结束。
I	I00*~I150*，6点，输入中断指针； I6**~I8**，3点，定时中断指针； I010~I060，6点，计数中断指针

指针(P)

指针P用于跳转指令CJ及子程序呼叫指令CALL，使用不须从编号0开始，但是编号不能重复使用，否则会发生不可预期的错误。使用时机如下所示：

- 1) 使用于指令CJ，指示程序执行跳跃的目的地址，并在目标程序的开头输入同编号的指针P。
- 2) 使用于指令CALL，指示子程序的目的地址，并在子程序的开头输入同编号的指针P。

4.2 步进顺控指令

4.2.1 STL 编程指令

STL	程序跳至副母线	S	1
RET	程序返回主母线	无	1

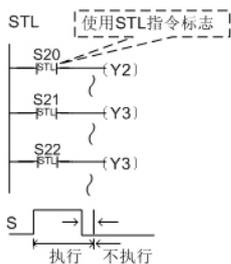
步进梯形图指令 (STL, RET)

步进梯形图是一种根据被控设备的运行过程，分解为若干个状态或工序，针对每一个状态进行逻辑编程的方式，再根据信号条件进行状态间的切换。编程时采用STL梯形图，这种编程方法思路清晰，简化了逻辑设计，方便调试和维护。

步进梯形图指令可用梯形图表示，在步进梯形图中，将状态 (S) 看作为一个控制工序，从中将输入条件与输出控制按顺序编程。这种控制最大的特点是在工序进行时，与前一工序不接通，以各道工序的简单顺序，即可控制设备。

步进梯形图有相应的编程规则，既包含了普通梯形图的编程方法，又与普通的梯形图编程有一定的差异，说明如下：

- 步进梯形图程序以STL指令开始（注意与普通梯形图中S不同），以RET指令结束，中间的程序以S状态引导，后续该S状态的所有操作逻辑，包括条件满足时切换为下一状态的操作。

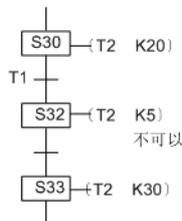


- 如果STL触点S接通，则与其相连的回路动作；若S触点断开，则与其相连的回路不动作。但是在一个扫描周期以后，不再执行指令（跳转状态）。
- 在不同的状态S，可对同样的输出软元件（如Y3）。此时，S21或S22接通时，Y3被输出。但在同一S状态中同样存在“双线圈”处理问题，请注意。
- 状态S号编号不可重复使用。



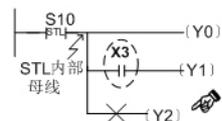
输出的互锁问题：

- 在状态的转移过程中，会存在两种状态同时接通瞬间（一个扫描周期）。因此，为了避免不能同时接通的一对输出同时接通，需要在可编程控制器外部设置互锁，同时要在相应的程序上设置互锁。



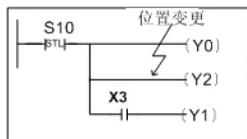
定时器重复使用的问题：

- 定时器线圈与输出线圈一样，也可在不同状态间对同一软元件编程。但是，在相邻状态中则不能编程。如果在相邻状态下编程，则工序转移时定时器线圈不断开，当前值不能复位。

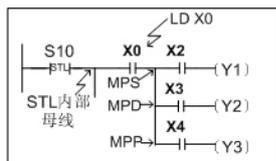
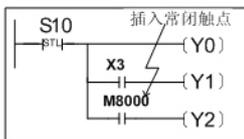


输出驱动方法

- 从状态内的母线，一旦写入LD或LDI指令后，不能再使用不需要触点的指令（如左图所示），需要按下图所示的方法改变这样的回路。

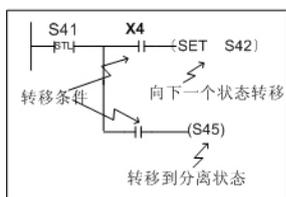


或



栈操作指令MPS/MRD//MPP的位置

- 在状态内，不能从 STL 内母线中直接使用 MPS / MRD / MPP 指令。而是须在 LD 或 LDI 指令以后编制程序。如左图所示。



状态的转移方法

- OUT 指令与 SET 指令对于 STL 指令后的状态 (S) 具有同样的功能，都将自动复位转移源。此外，还有自保持功能。
- 但使用 OUT 指令时，在 SFC 图中用于向分离的状态转移。

可在状态内处理的顺控指令一览表：

命令 状态		LD/LDI/LDP/LDF, AND/ANI/ ANDP/ANDF, OR/ORI/ORF, INV, OUT, SET/RST, PLS/PLF	ANB/ORB MPS/MRD/MPP	MC/MCR
		初始状态/一般状态	可使用	可使用
分支, 汇合状态	输出处理	可使用	可使用	不可使用
	转移处理	可使用	不可使用	不可使用

- 在中断程序与子程序内，不能使用STL指令。
- 在STL指令内不禁止使用跳转指令，但其动作复杂，建议不要使用。

4.2.2 SFC 顺序功能图编程

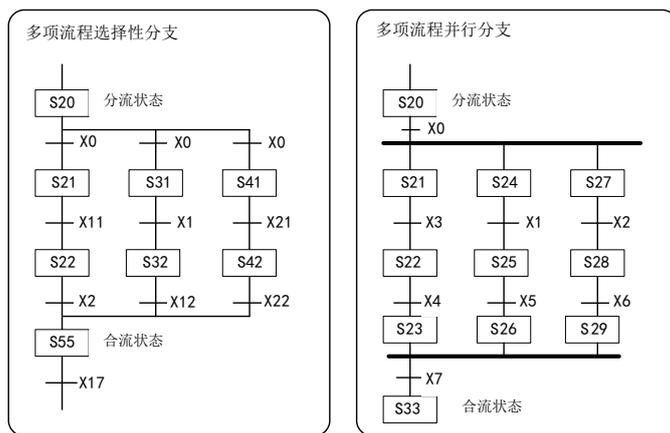
H1U/H2U系列可编程控制器内置有利用SFC图（顺序功能图）的顺控功能，SFC采用类似流程图的方式，将控制程序按流程图方式直观表述，使得编程调试、维护的大为简化。SFC图设计时使用的符号定义如下：

	起始步进点图形，用于S0~S9状态的起始编程点，一个用户程序中只有一个该起始符。
	梯形图块图形，表示内部为一般步行梯形图的程序。常带梯形图块编号，如LAD0、LAD1...等
	一般步进梯形图程序块图形，可使用S10~S889状态变量
	状态转移条件图形，用于标明上下相邻两状态转移的条件。
	状态分离图形，用于标明不相邻的两个状态的跳转。
	向上状态转移图形，用于标明向上转移的状态
	状态复位图形，将程序的状态复位到起始状态S0
	选择分支图形，由同一步进点按不同条件转移到相应步进点。
	选择汇合图形，由两个以上步进点状态，经相应的转移条件后，转移到相同的步进点。
	并行分支图形，由同一步进点将综合体以同一转移条件转移到两个以上步进点。
	并行分支汇合图形，由两个以上不同步进点状态同时成立时，以同一转移条件转移到相同的步进点。

4
指令

1) SFC的编程特点：

- 在该梯形图块 中，采用可编程控制器由STOP→RUN转换时，瞬间动作的辅助继电器M8002，使初始状态S0置位（ON）；
- 可编程控制器中S0-S9为初始状态软元件；
- 对各动作工序分配了S20-S889等状态。其中也有停电保持用的状态，即使在停电时也可保存其动作状态。此外，S10-S19可用于特殊目的；
- 可编程控制器内的定时器、计数器和辅助继电器等软元件，可随意使用；
- 当有多项工序的选择、或有多个需要同时进行的工序时，采用如下方法：



可见在SFC图中，每道工序中设备的动作清晰易懂，其顺控设计容易，方便调测维护。

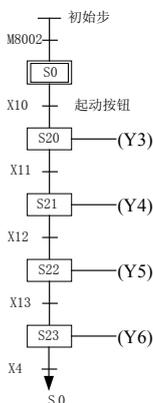
- SFC图与步进梯形图指令都按一定的规则编程，可相互转换，其内容是一样的。也可使用大家熟悉的继电器梯形图。

2) SFC的编程方法：

以下以举例的方式来逐项说明SFC编程的方法。

初始状态的作用

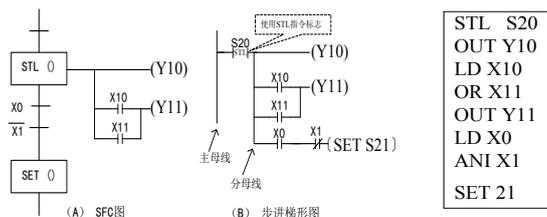
- 初始状态位于SFC图的最前面，可使用状态号S0-S9。
- 初始状态也要通过其他状态（如上图示例S23所示）来驱动时，需要在运行开始时，利用其他方法事先驱动。
- 下图所示例子是在可编程控制器由STOP→RUN切换时，利用只有瞬间动作的特殊辅助继电器M8002来驱动。
- 初始状态以外的一般状态一定要通过来自其他状态的STL指令驱动，不能从状态以外驱动。
- 将这种通过STL指令以外的触点驱动的状态称为初始状态。一定在流程的最前面表述。此外，对应初始状态的STL指令，必须在其之后的一系列STL指令之前编程。



没有分支与汇合的一般流程

下图A为典型的SFC图，每个状态具有驱动负载、指定转移目标以及指定转移条件三种功能。使用继电器顺控方式表示SFC图时，是下图（B）的步进梯形图。

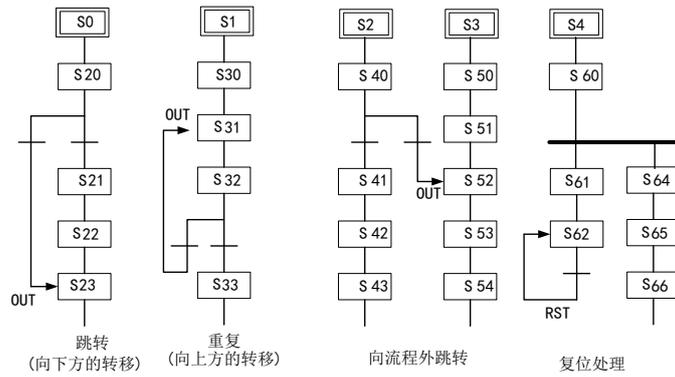
程序用SFC图或用步进梯形图均可编写。编程顺序为先进进行负载的驱动处理，接着进行转移处理。当然，如果是不需要驱动负载的状态，则不需要进行负载的驱动处理。



当然上述步进梯形图也可用指令表程序来等效描述，如右图所示。STL指令为与主母线连接的常开触点指令，接着就可在副母线上直接连接线圈，或者可以通过触点驱动线圈。

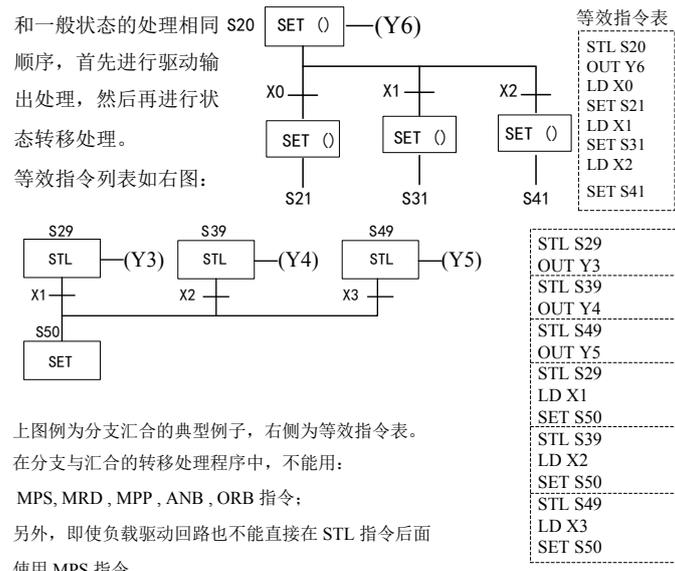
在一系列的STL指令前面要有初始状态，最后一定要写入RET指令。

带有跳转与重复的一般状态



如上图所示，向下方状态的转移（跳转）、向上方状态的转移（重复）、向流程外的转移等的分离状态转移，用OUT指令编程。

选择性分支与汇合状态

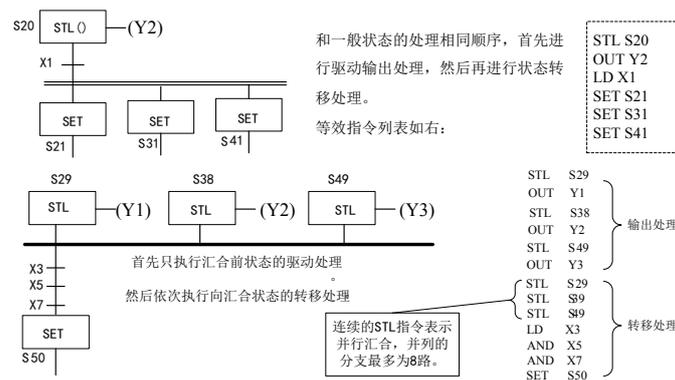


上图例为分支汇合的典型例子，右侧为等效指令表。

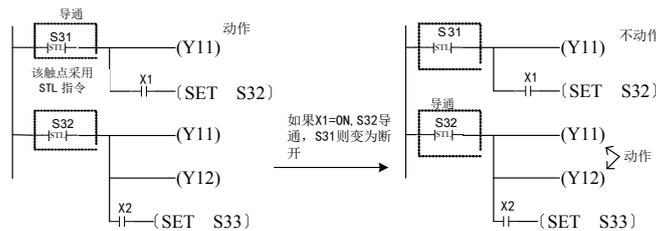
在分支与汇合的转移处理程序中，不能用：
MPS, MRD, MPP, ANB, ORB 指令；

另外，即使负载驱动回路也不能直接在 STL 指令后面使用 MPS 指令。

并行分支与汇合状态



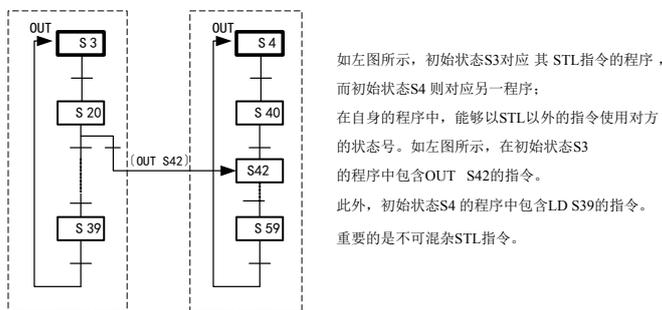
步进梯形图指令及其动作如下图所示：



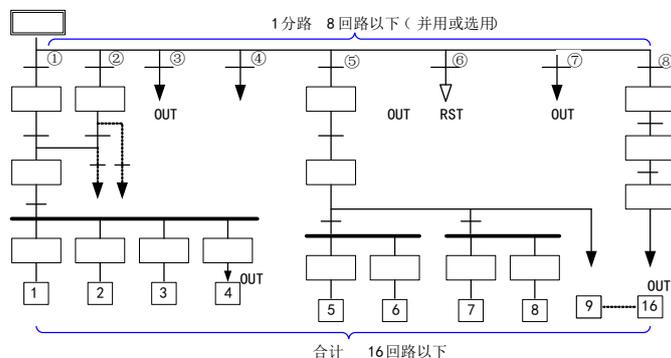
若以SFC图表示上图所示的步进梯形图回路，则其表示如下图所示：



具有多个初始状态的SFC图的程序将各初始状态分离编程。

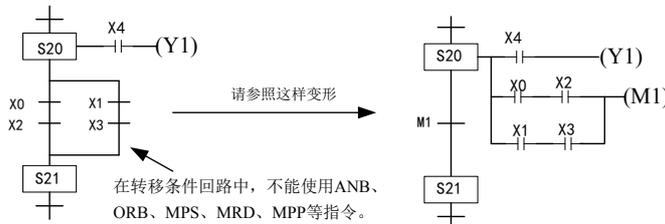


一条并行分支或选择性分支的回路数限定为8条以下; 但是, 有多条并行分支或选择性分支时, 每个初始状态的回路总数不超过16条。如下图:



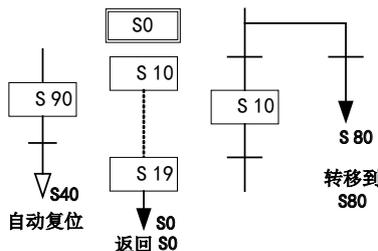
不能进行从汇合线或汇合前的状态开始向分离状态的转移处理或复位处理, 应设置空状态, 由分支线上向分离状态进行转移与复位处理。

对于状态转移条件比较复杂的情况, 建议作简化处理, 例如:



状态的转移与复位

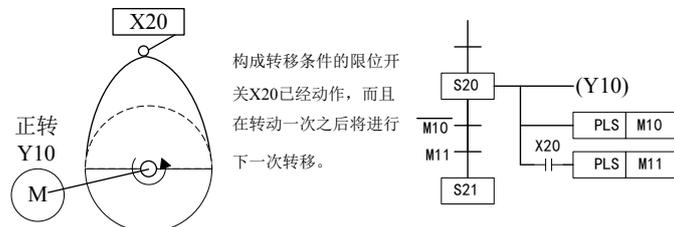
- 在流程中, 符号↓则表示向上的状态转移重复或向下面的状态转移 (跳转), 或者向分离的其他流程上的状态转移。符号▽则表示状态的复位处理。
- 状态标志S也可以采用ZRST指令对一个区间的标志进行批量复位。
- 编写 (SFC) 图程序时, 可使用如下特殊辅助继电器, 提高编程效率, 如下表所示说明。



4
指令

软件元件号	名称	功能和用途
M8000	RUN监视	可编程控制器在运行过程中，需要一直接通的继电器。可作为驱动的程序输入条件或作为可编程控制器运行状态的显示来使用。
M8002	初始脉冲	在可编程控制器由STOP→RUN时，仅在瞬间（1个扫描周期）接通的继电器。用于程序的初始设定或初始状态的置位。
M8040	禁止转移	驱动该继电器，则禁止在所有状态之间转移。然而，即使在禁止转移状态下，由于状态内的程序仍然动作，因此，输出线圈等不会自动断开。
M8046	STL动作	任一状态接通时，M8046自动接通。用于避免与其他流程同时启动或用作工序的动作标志。
M8047	STL监视有效	驱动该继电器，则编程功能则可自动读出正在动作中的状态并加以显示。详细事项请参照各外围设备的手册。

- 停电保持用状态，是用电池保持其动作状态。在机械动作中途发生停电之后，再通电时从这里继续运行的情况下使用这些状态。
- RET指令一定在一系列的STL指令的最后编写，没有编写RET指令时，会出现[程序出错]，可编程控制器不能运行。执行此指令，表明步进梯形图回路的结束。在希望中断一系列的工序而在主程序编程时，同样需要RET指令，RET指令可多次编程。
- 转移条件成立后状态的处理。

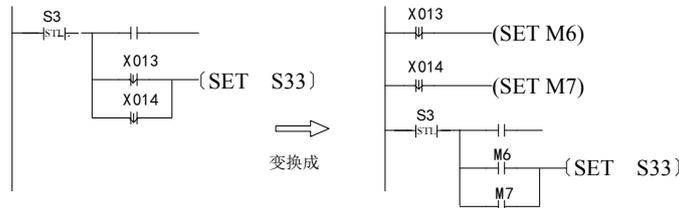


如上图所示的应用中，将转移条件脉冲化，S20首次动作，通过M10使不产生转移。

- 上升沿/下降沿检测触点使用时的注意事项：

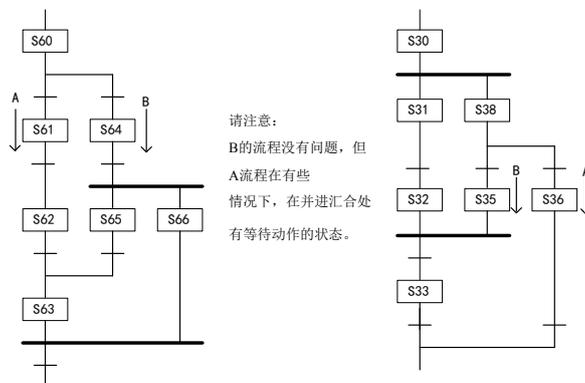
在状态内使用LDP、LDF、ANDP、ANF、ORP、ORF的上升沿/下降沿检测触点时，状态断开时变化的触点，在状态再次接通时被检出。

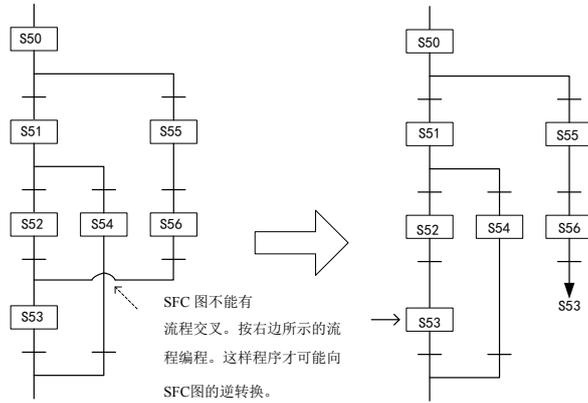
对于状态断开时变化的条件，必须上升沿/下降沿检测时，请按下图所示，修改程序：



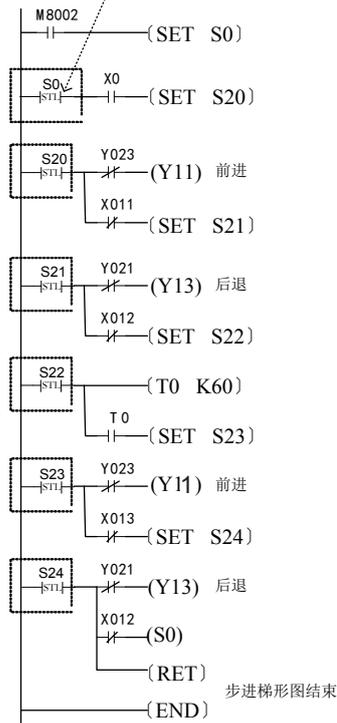
通过X013下降沿向S33转移后，若X014下降，此时因S3断开，X014的下降沿无法检出，S3再次接通时，被检测。因此，S3第2次动作时，立即向S33转移。

分支与汇合的组合流程

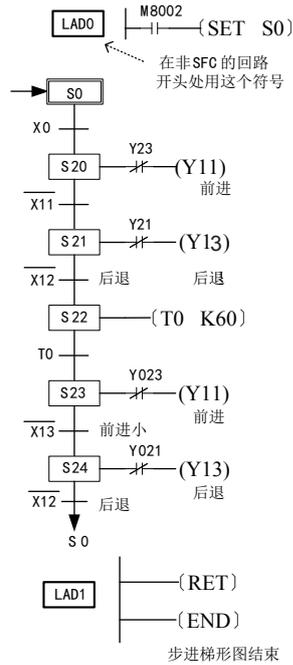




STL图设计方法



等效的SFC图设计方法



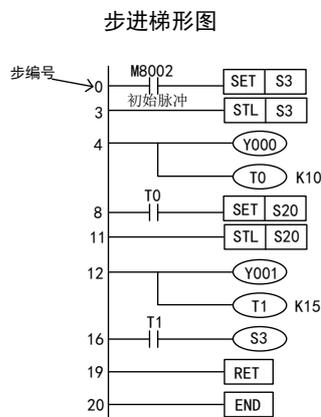
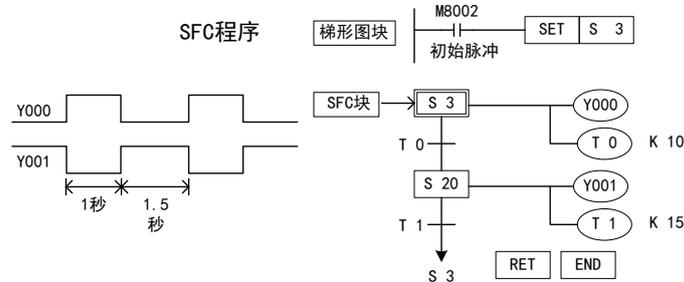
4
指令

4.2.3 SFC 编程实例

1) 单流程的例子

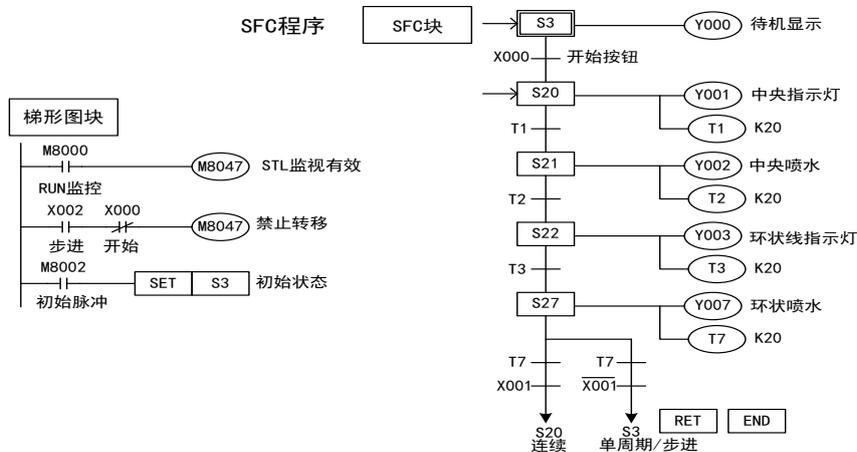
例子1: 闪烁回路:

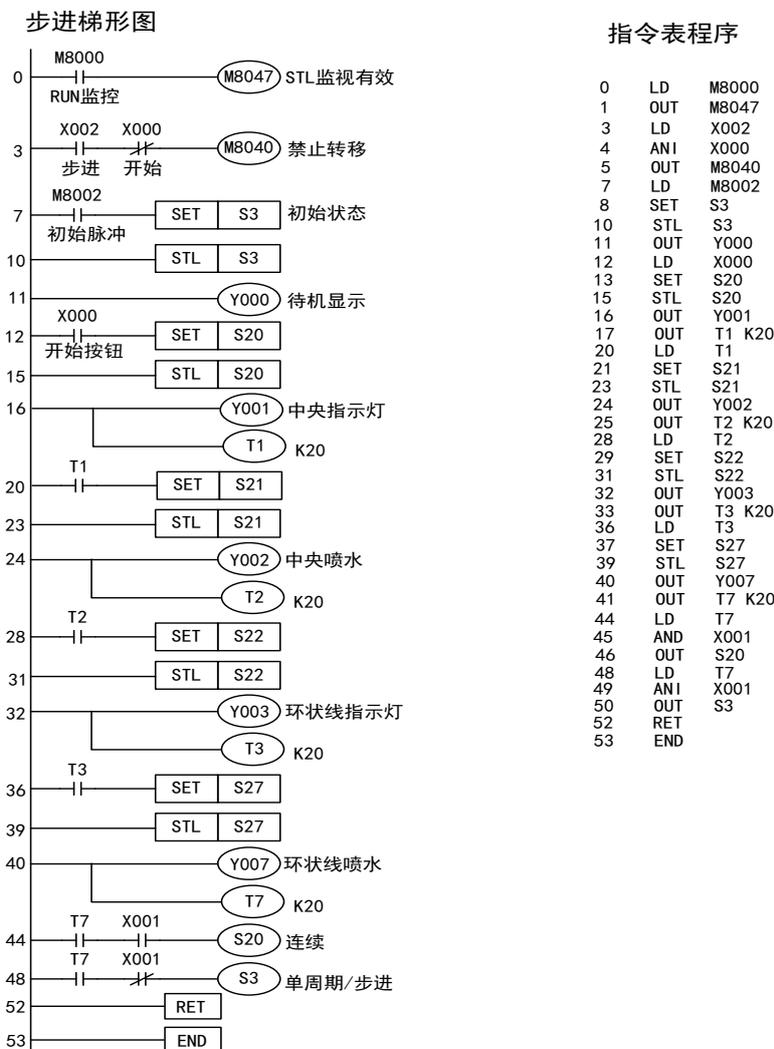
- a) 若可编程控制器运行, 则利用初始脉冲 (M8002) 驱动状态S3。
- b) 在状态S3中输出 Y000, 1 秒钟以后, 向状态S20 转移。
- c) 在状态 S20中输出 Y001, 1.5 秒钟后, 返回状态S3。



例子2: 喷水控制:

- d) 单周期运行 (X001=OFF, X002 = OFF)，若按下启动按钮X000后, 则按照 Y000 (待机显示) →Y001 (中央指示灯) →Y002 (中央喷水) →Y003 (环状线指示灯) →Y007 (环状线喷水) →Y000 (待机显示) 的顺序动作, 并返回到待机状态。用 2秒钟定时器依次切换各输出。
- e) 连续运行 (X001 = ON)
- 重复 Y001~Y007的动作。
- f) 步进运行 (X002 =ON)
- 每按一次启动按钮, 则依次各输出动作。





4
指令

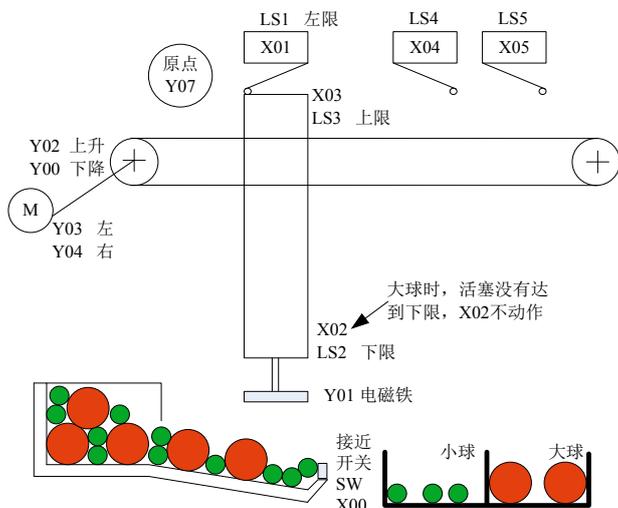
2) 选择分支与汇合流程的例子

例：大小球的选择传送：

下图为使用传送带，将大、小球分类选择传送的机械装置。

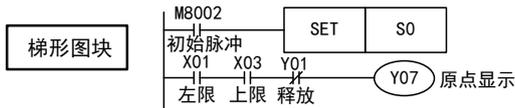
左上方为原点，其动作顺序为下降、吸住、上升、右行、下降、释放、上升、左行。

此外，机械手臂下降，当电磁铁压着大球时，下限限位开关 LS2断开；压着小球时，LS2 导通。

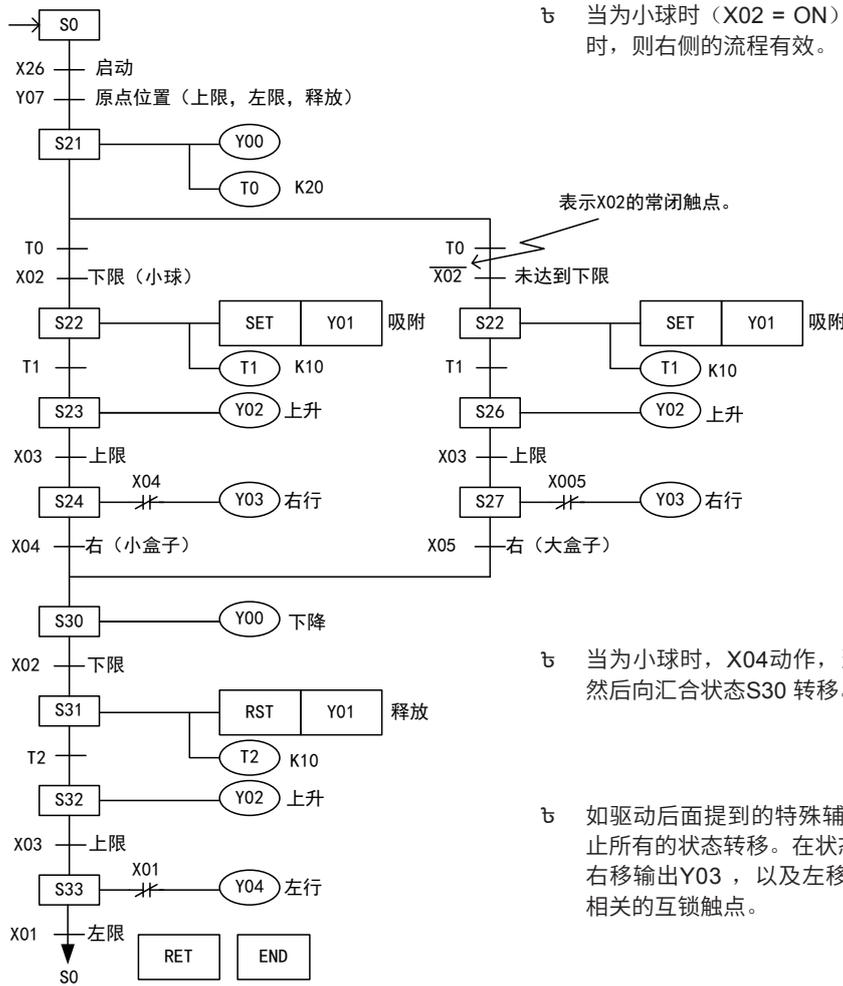


像这种大小分类选择或判别合格与否的SFC程序，就是如下图所示的选择性分支与汇合的SFC程序。

SFC程序



SFC块

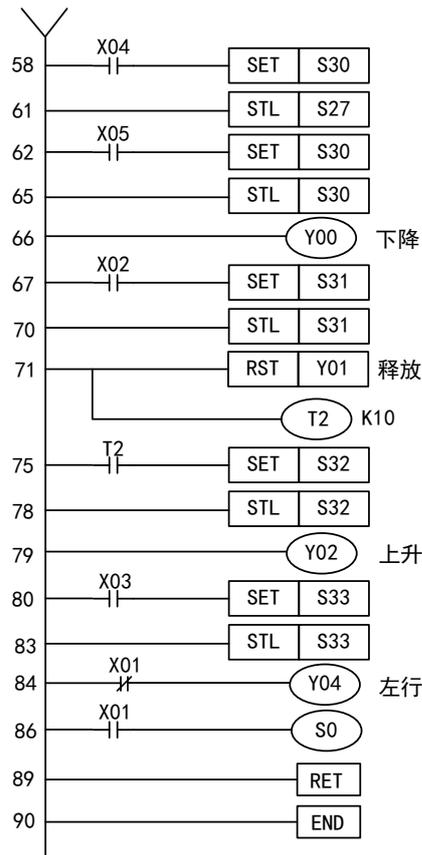
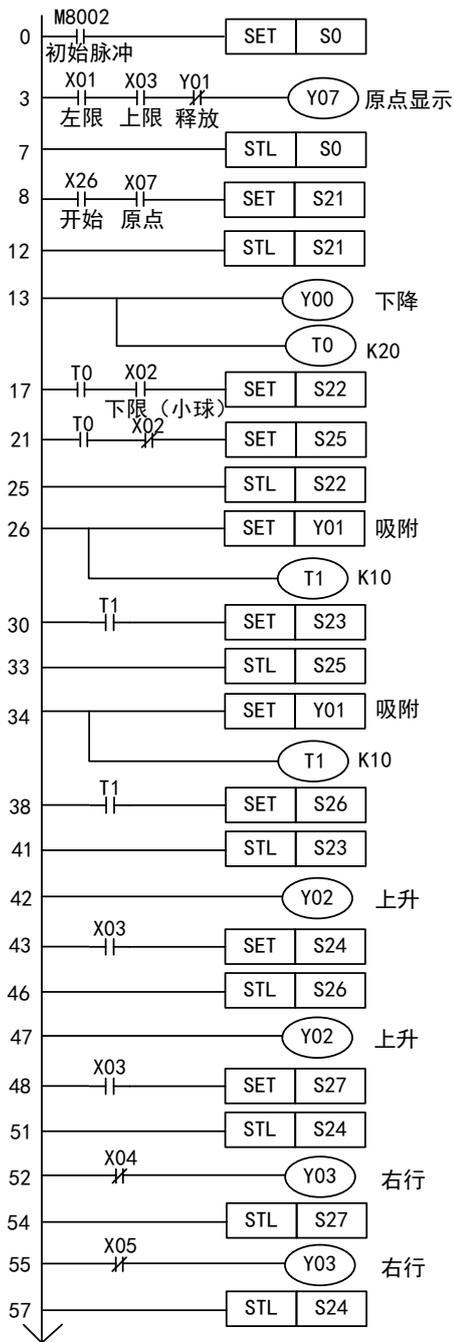


当为小球时 (X02 = ON) 左侧流程有效；当为大球时，则右侧的流程有效。

当为小球时，X04动作，当为大球时，X05动作，然后向汇合状态S30转移。

如驱动后面提到的特殊辅助继电器 M8040，则禁止所有的状态转移。在状态 S24、S27 与 S33 中，右移输出 Y03，以及左移输出 Y04中，各自串联有相关的互锁触点。

步进梯形图



4
指令

- 当为小球时，X04动作，当为大球时，X05动作，然后向汇合状态S30转移。
- 如驱动后面提到的特殊辅助继电器 M8040，则禁止所有的状态转移。在状态 S24、S27 与S33 中，右移输出Y03，以及左移输出Y04中，各自串联有相关的互锁触点

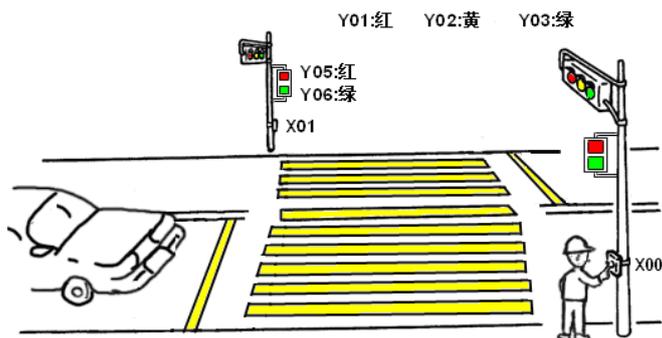
指令表程序

0	LD	M8002	47	OUT	Y02
1	SET	S0	48	LD	X03
3	LD	X01	49	SET	S27
4	AND	Y03	51	STL	S24
5	ANI	Y01	52	LD	X04
6	OUT	Y07	53	OUT	Y03
7	STL	S0	54	STL	S27
8	LD	X26	55	LDI	X05
9	AND	Y07	56	OUT	Y03
10	SET	S21	57	STL	S24
12	STL	S21	58	LD	X04
13	OUT	Y00	59	SET	S30
14	OUT	T0 K20	61	STL	S27
17	LD	T0	62	LD	X05
18	AND	X02	63	SET	S30
19	SET	S22	65	STL	S30
21	LD	T0	66	OUT	Y00
22	ANI	X02	67	LD	X02
23	SET	S25	68	SET	S31
25	STL	S22	70	STL	S31
26	SET	Y01	71	RST	Y01
27	OUT	T1 K10	72	OUT	T2 K10
30	LD	T1	75	LD	T2
31	SET	S23	76	SET	S32
33	STL	S25	79	OUT	Y02
34	SET	Y01	80	LD	X03
35	OUT	T1 K10	81	SET	S33
38	LD	T1	83	STL	S33
39	SET	S26	84	LDI	X01
41	STL	S23	85	OUT	Y04
42	OUT	Y02	86	LD	X01
43	LD	X03	87	OUT	S0
44	SET	S24	89	RET	
46	STL	S23	90	END	

3) 并行分支与汇合流程的例子

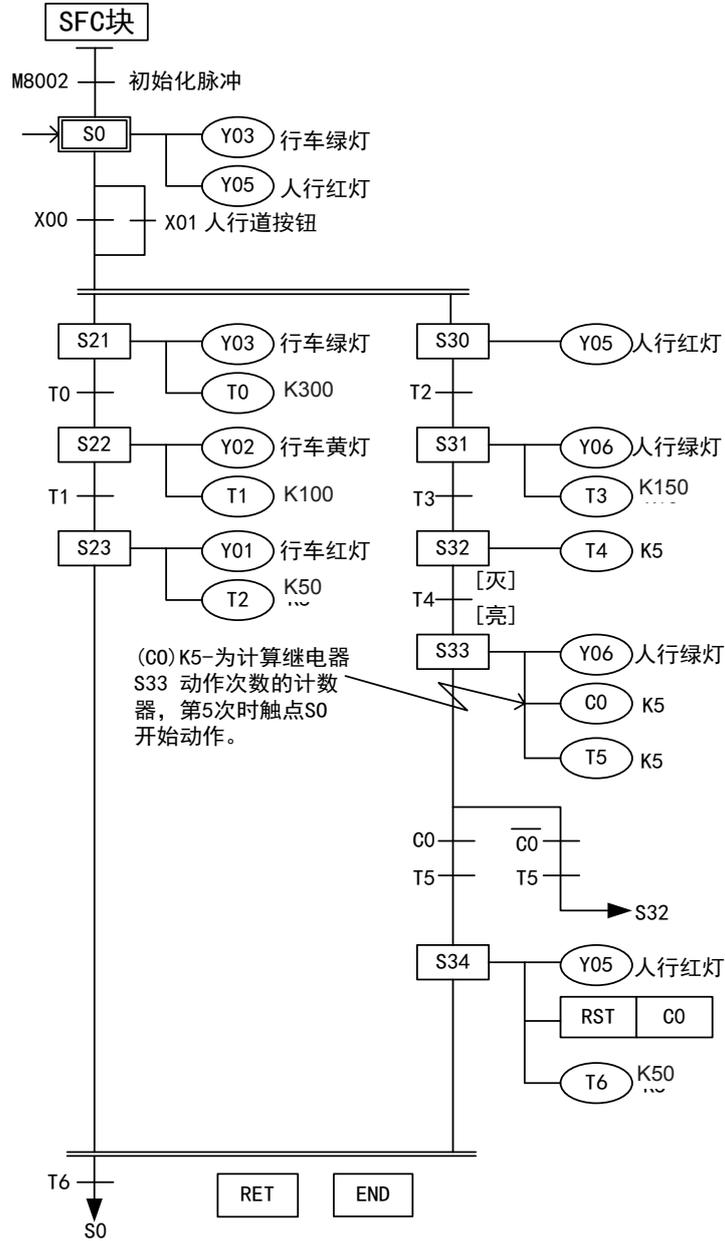
例：按钮式人行横道的例子：

下图以按钮式人行横道的例子，可以使用并行分支与汇合的流程表示。



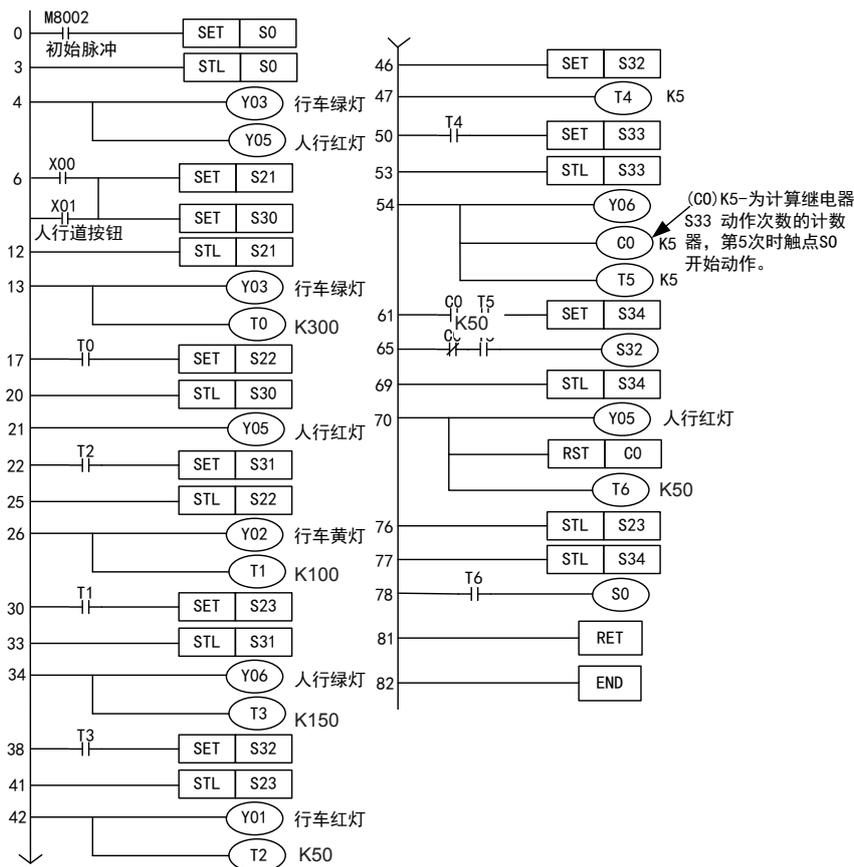
按钮式人行横道线的SFC程序如下所示。在下图中示例表示的是在人行道的绿灯闪灯时，部分流程重复的动作（跳转到上面的状态）。

- 可编程控制器从 STOP→RUN变换时，初始状态S0动作，通常车道信号灯为绿，而人行道信号灯为红。
- 按下人行道按钮X00或 X01,则状态S21车道为绿灯；状态S30中的人行道信号已经为红色，此时状态无变化。
- 30 秒钟后，车道信号为黄灯；再过10秒钟车道信号为红灯。
- 此后，定时器T2 (5秒钟) 启动，5秒后人行道变为绿灯。
- 15秒后，人行道绿灯开始闪烁。(S32为灭，S33为亮)。
- 闪烁中时S32、S33反复动作，计数器C0（设定值为 5 次）触点一接通，动作状态向S34 转移，人行道变为红灯，5秒后返回初始状态。
- 在动作过程中，即使按动人行道按钮X00、X01 也无效。



4
指令

步进梯形图



4
指令

指令表程序

```

0 LD M8002
1 SET S0
3 STL S0
4 OUT Y03
5 OUT Y05
6 LD X00
7 OR X01
8 SET S21
10 SET S30
12 STL S21
13 OUT Y03
14 OUT T0 K300
17 LD T0
18 SET S22
20 STL S30
22 LD T2
23 SET S31
25 STL S22
26 OUT Y02
27 OUT T1 K100
30 LD T1
31 SET S23
33 STL S31
34 OUT Y06
35 OUT T3 K150
38 LD T3
39 SET S32
41 STL S23
42 OUT Y01
43 OUT T2 K50
46 STL S32
47 OUT T4 K5
50 LD T4
51 SET S33
53 STL S33
54 OUT Y06
55 OUT C0 K5
58 OUT T5 K5
61 LD C0
62 AND T5
63 SET S34
65 LDI C0
66 AND T5
67 OUT S32
69 STL S34
70 OUT Y05
71 RST C0
73 OUT T6 K50
76 STL S23
77 STL S34
78 LD T6
79 OUT S0
81 RET
82 END
    
```

●	程序流程	FNC00 ~ FNC09
●	传送与比较	FNC10 ~ FNC19
●	四则逻辑运算	FNC20 ~ FNC29
●	循环移位	FNC30 ~ FNC39
●	数据处理	FNC40 ~ FNC49
●	高速处理	FNC50 ~ FNC59
●	方便指令	FNC60 ~ FNC69
●	外部设备I/O	FNC70 ~ FNC79
●	外部设备SER	FNC80 ~ FNC93
●	浮点数	FNC110 ~ FNC147
●	定位指令	FNC155 ~ FNC159
●	时钟运算	FNC160 ~ FNC169
●	外围设备	FNC170 ~ FNC171
●	接点比较	FNC224 ~ FNC246

4.3 应用指令

应用指令是指除逻辑处理指令之外的编程指令，指令功能涉及程序流程控制、数值计算、高速信号处理、通讯与扩展、特殊控制功能等多个方面。这些指令除了具有指令名之外，往

往还有“功能号”的编号，便于使用手持编程器设备进行编程。按功能分类列表如下。

分类	指令符号			指令功能
	16bit	32bit	P指令	
程序流程	CJ	-	✓	有条件跳转
	CALL	-	✓	子程序调用
	SRET	-	-	子程序返回
	IRET	-	-	中断返回
	EI	-	-	开中断
	DI	-	-	关中断
	FEND	-	-	主程序结束
	WDT	-	✓	监控定时器
	FOR	-	-	循环范围开始
	NEXT	-	-	循环范围终了
4 指令 传送与比较	CMP	✓	✓	比较
	ZCP	✓	✓	区域比较
	MOV	✓	✓	传送
	SMOV	-	✓	移位传送
	CML	✓	✓	倒转传送
	BMOV	-	✓	一并传送
	FMOV	✓	✓	多点传送
	XCH	✓	✓	交换
	BCD	✓	✓	BCD转换
	BIN	✓	✓	BIN转换
四则逻辑运算	ADD	✓	✓	BIN加法
	SUB	✓	✓	BIN减法
	MUL	✓	✓	BIN乘法
	DIV	✓	✓	BIN除法
	INC	✓	✓	BIN加1
	DEC	✓	✓	BIN减1
	WAND	✓	✓	逻辑字与
	WOR	✓	✓	逻辑字或
	WXOR	✓	✓	逻辑字异或
	NEG	✓	✓	求补码
循环移位	ROR	✓	✓	循环右移
	ROL	✓	✓	循环左移
	RCR	✓	✓	带进位循环右移
	RCL	✓	✓	带进位循环左移
	SFTR	✓	✓	位右移
	SFTL	-	✓	位左移
	WSFR	-	✓	字右移
	WSFL	-	✓	字左移
	SFWR	-	✓	“先进先出”写入
	SFRD	-	✓	“先进先出”读出

分类	指令符号			指令功能
	16bit	32bit	P指令	
数据处理	ZRST	-	✓	区间复位
	DECO	-	✓	解码
	ENCO	-	✓	编码
	SUM	✓	✓	ON位数
	BON	✓	✓	ON位数判定
	MEAN	✓	✓	平均值
	ANS	-	-	报警器置位
	ANR	-	✓	报警器复位
	SQR	✓	✓	BIN平方根
	FLT	✓	✓	浮点数与十进制数间转换
高速处理	REF	-	✓	输入输出刷新
	REFF	-	✓	滤波器调整
	MTR	-	-	距阵输入
	HSCS	✓	-	比较置位 (高速计数器)
	HSCR	✓	-	比较复位 (高速计数器)
	HSZ	✓	-	比较区间 (高速计数器)
	SPD	-	-	脉冲密度
	PLSY	✓	-	脉冲输出
	PWM	-	-	脉冲幅宽调制
	PLSR	✓	-	带加减速的脉冲输出
方便指令	IST	-	-	状态初始化
	SER	✓	✓	数据搜索
	ABSD	✓	-	绝对值式凸轮顺控
	INCD	-	-	增量式凸轮顺控
	TTMR	-	-	示教定时器
	STMR	-	-	特殊定时器
	ALT	-	-	交替输出
	RAMP	-	-	斜坡信号
	ROTC	-	-	旋转台控制
	SORT	-	-	列表数据排列
外部设备 I/O	TKY	✓	-	0-9数字键输入
	HKY	✓	-	16键输入
	DSW	-	-	数字开关
	SEGD	-	✓	7段编码
	SEGL	-	-	带锁存的7段显示
	ARWS	-	-	矢量开关
	ASC	-	-	ASCII码转换
	PR	-	-	ASCII码打印输出
	FROM	✓	✓	特殊功能块读出
	TO	✓	✓	特殊功能块写入

分类	指令符号			指令功能
	16bit	32bit	P指令	
外设设备 SER	RS	-	-	串行数据传送
	PRUN	✓	✓	并联运行
	ASCI	-	✓	HEX→ASCII转换
	HEX	-	✓	ASCII→HEX转换
	CCD	-	✓	校正代码
	PID	-	-	PID运算
	MODBUS	-	-	数据传送
	CANTX	-	-	CAN数据发送
	CANRX	-	-	CAN数据接收
	RS2	-	-	串行数据传送2
	MODBUS2	-	-	数据传送2
浮点数	ECMP	✓	✓	2进制浮点数比较
	EZCP	✓	✓	2进制浮点数区间比较
	EBCD	✓	✓	2进制→10进制浮点数转换
	EBIN	✓	✓	10进制→2进制浮点数转换
	EADD	✓	✓	2进制浮点数加
	ESUB	✓	✓	2进制浮点数减
	EMUL	✓	✓	2进制浮点数乘
	EDIV	✓	✓	2进制浮点数除
	ESOR	✓	✓	2进制浮点数开平方
	INT	✓	✓	2进制浮点数→BIN整数转换
	SIN	✓	✓	浮点数SIN运算
	COS	✓	✓	浮点数COS运算
	TAN	✓	✓	浮点数TAN运算
SWAP	✓	✓	上下字节变换	
定位指令	ABS	✓	-	ABS位置数据读取
	ZRN	✓	-	原点回归
	PLSV	✓	-	可变脉冲输出
	DRVI	✓	-	相对位置控制
	DRVA	✓	-	绝对位置控制
时钟运算	TCMP	-	✓	时钟数据的比较
	TZCP	-	✓	时钟数据区域比较
	TADD	-	✓	时钟数据加法
	TSUB	-	✓	时钟数据减法
	TRD	-	✓	时钟数据读出
	TWR	-	✓	时钟数据写入
HOUR	✓	-	计时表	
外围设备	GRY	✓	✓	格雷码转换
	GBIN	✓	✓	格雷码逆转换

分类	指令符号			指令功能
	16bit	32bit	P指令	
接点比较	LD=	✓	-	(S1)=(S2)
	LD>	✓	-	(S1)>(S2)
	LD<	✓	-	(S1)<(S2)
	LD<>	✓	-	(S1)<>(S2)
	LD<=	✓	-	(S1)<=(S2)
	LD>=	✓	-	(S1)>=(S2)
	AND=	✓	-	(S1)=(S2)
	AND>	✓	-	(S1)>(S2)
	AND<	✓	-	(S1)<(S2)
	AND<>	✓	-	(S1)<>(S2)
	AND<=	✓	-	(S1)<=(S2)
	AND>=	✓	-	(S1)>=(S2)
	OR=	✓	-	(S1)=(S2)
	OR>	✓	-	(S1)>(S2)
	OR<	✓	-	(S1)<(S2)
	OR<>	✓	-	(S1)<>(S2)
	OR<=	✓	-	(S1)<=(S2)
OR>=	✓	-	(S1)>=(S2)	

4.3.1 程序流程 (00~09)

FNC00	16bit	32bit	\boxed{P}	FNC00	CJ P*** CJP P***	条件跳转
	✓		✓			
CJ	3步		3步	P000~P127		

- 1) 当能流有效时，程序自动从CJ（或CJP）指令的地址跳转至由P***指定的地址后继续执行，中间地址的程序指令被跳过，不予执行；
- 2) 当能流无效时，程序继续往下执行，此时CJ（或CJP）指令不被执行。

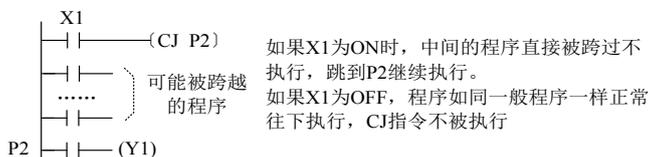
若被跨越的中间地址区的程序中有TMR定时器或计数器，且已被驱动，则动作反应为：

执行情况	CJ有跳转	CJ无跳转
T192~T199	正常执行	正常执行
其他定时器	停止计时	
C235~C255	正常执行	
其他计数器	停止计数	

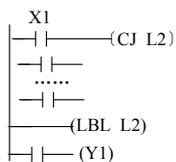
对P***地址指针的要求如下：

- 由CJ（或CJP）引用的地址指针，必须在主程序结束（FEND指令）之前的范围；
- P63特指END的地址，不要定义到其它程序步；
- 与子程序不同，P***开始的程序后不需要SRET语句作为结束；
- P***的定义地址不要有重复；
- 当使用者希望某一部份程序不需要执行时，以缩短扫描周期；或者想使用两个线圈输出时，为避免双线圈的出现。可使用此指令；
- CJ指令可重复指定同一指针P，但不可与CALL指定同一指针P，否则出错。

指令举例：



在AutoShop编程环境中，跳转指令用L，上例中格式如下：



且子程序和中断程序单独窗口写，故不必注意FEND等事项，CJP63在AutoShop编程环境中为CJEND。

16bit	32bit	\boxed{P}	FNC01	CALL P*** CALLP P***	子程序调用
3步			P000~P127		

当能流有效时，程序调用由P***指定的子程序。子程序执行完毕，会返回到该CALL（或CALLP）语句的下一指令，继续执行后续语句。

对P***地址指针的要求如下：

- 由P***开始的子程序，必须在主程序结束（FEND指令）之后的范围；
- 子程序必须以SRET语句结束；
- P***的子程序可被多处调用，也可被其他子程序调用，但嵌套层数不得超过5层；
- 在子程序内不得调用自身，防止死循环或程序运行超时。

- 在子程序中，可采用T192~T199或T246~T249作为定时器。

在AutoShop编程环境中子程序在单独窗口编写，故没有FEND、SRET等指令的问题存在，且子程序名可任意修改（包括中文）。

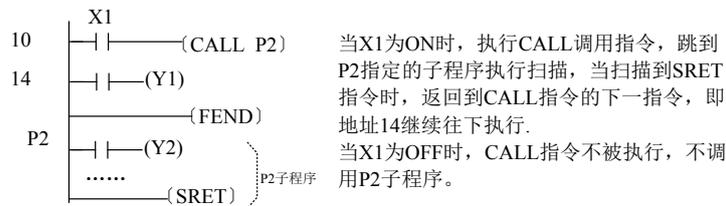
FNC01
FNC02
CALL
SRET

16bit	32bit		FNC02	SRET	子程序完毕
✓					
1步			无需触点驱动，无操作数的单独指令		

SRET语句位于子程序的结束处，执行了该指令后，会退回到调用该子程序的语句处，继续随后的程序执行。

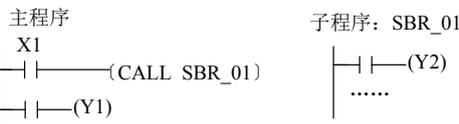
在AutoShop编程环境中，子程序最后无需编写SRET。

指令举例一：



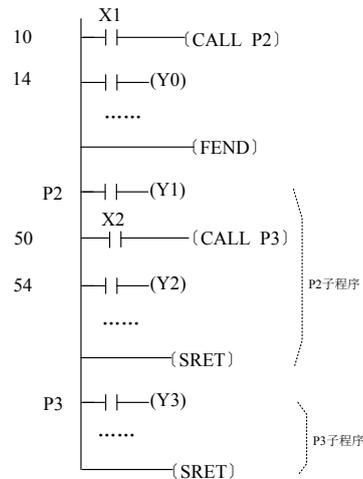
在AutoShop编程环境中“指令举例一”的格式如下：

右击程序块的子程序SBR_01，选择属性。可以修改01为你想要的名称，包括中文)



指令举例二：

当X1为ON时，执行CALL调用指令，跳到P2指定的子程序执行扫描，分2种情况：如果扫描到地址50时X2为OFF，则继续往下扫描直到SRET指令，返回到主程序的地址14继续往下执行；如果扫描到地址50时X2为ON，则执行CALL P3，转移到P3所指定的子程序进行扫描，当执行到SRET指令时，回到P2子程序地址54继续往下执行



16bit	32bit		FNC03	IRET	中断程序完毕	无需触点驱动，无操作数的单独指令
✓			FNC04	EI	中断允许	
1步			FNC05	DI	中断禁止	

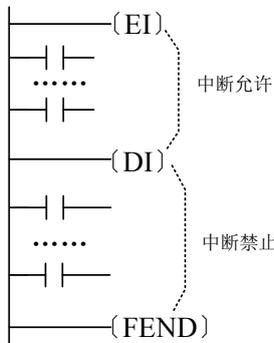
FNC03
FNC04
FNC05

IRET
EI
DI

IRET语句位于中断子程序的结束处，执行了该指令后，会返回到调用该中断子程序前的语句处，继续程序执行。在AutoShop编程环境中，中断程序在单独窗口编写，在中断程序最后无需写IRET指令

PLC程序开始运行时，默认为中断禁止状态；执行了EI语句后，中断功能允许；当中断为允许状态，执行了DI语句后，即进入中断禁止状态。在程序中如果没有中断插入禁止的区间时，可以不使用DI指令。

中断的种类与设置:



- 1) 外部信号输入中断：可定义X0~X5输入信号的上升沿或下降沿进行中断，对于不需要即时响应的X信号，还可以采用脉冲捕捉的功能；
- 2) 高速计数中断：与FNC53（DHSCS）比较置位指令并用，当高速计数的当前值达到设定值时，产生中断；
- 3) 定时器中断：以1ms~99ms的固定周期发生的中断；
- 4) 脉冲完成中断：在指定脉冲个数发送完毕后，立即执行中断；
- 5) 多用户中断（XP不具备此功能）：可任意取一个高速计数器进行最多24个中断。

外部信号输入中断指针与设置（H1U-XP、H2U-XP）：

输入编号	指针编号		禁止中断指令
	上升中断	下降中断	
X000	I001	I000	M8050
X001	I101	I100	M8051
X002	I201	I200	M8052
X003	I301	I300	M8053
X004	I401	I400	M8054
X005	I501	I500	M8055

定时中断指针与设置（H1U-XP、H2U-XP）：

输入编号	中断周期MS	中断禁止指令
I6□□	在指令的□□中输入1~99的数，如I605为每5MS执行一次定时中断	M8056
I7□□		M8057
I8□□		M8058

高速计数中断指针与设置（H1U-XP、H2U-XP）：

输入编号	中断禁止指令
I010	M8059
I020	
I030	
I040	
I050	
I060	

脉冲输出完成中断指针与设置：（此功能要启动M8090~M8094才能在脉冲输出完毕产生中断）（H1U-XP、H2U-XP）。

端口号	使用特殊位	对应的用户中断
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504
Y003	M8093	I505
Y004	M8094	I506

中断子程序选用不同的编号，即选择了不同的端口及中断触发沿；

外部输入中断中对于同一X输入，不能同时对上升中断和下降中断编号。对于一个X输入端口，只能使用一种触发沿，触发沿

通过指针编号来设定；

FNC03
FNC04
FNC05
IRET
EI
DI

外部输入中断：如果对M8050-M8055在程序执行过程中“ON”，则禁止了对应X端口的中断功能。

定时中断：如果对M8056-M8058在程序执行过程中“ON”，则禁止了对应的定时中断功能。

高速计数中断：如果对M8059在程序执行过程中“ON”，则禁止了所有的高速计数中断功能。

中断的编程规定与执行特性：

- 在DI-EI指令间（中断禁止区间）发生中断，亦能对其记忆并在EI指令后执行。
- 中断子程序必须写在FEND指令之后，子程序尾部必须以IRET结束，在AutoShop环境内，不要写在主程序中，子程序尾可省略IRET；
- 指针编号不能重复使用；
- 多个中断依次发生时，以先发生的为优先。完全同时发生时，优先级别高的为优先。优先级从高到低分别为高速计数器中断、外部中断、时间中断、脉冲输出完成中断。
- 在中断例行程序的执行过程中，禁止其它的中断。
- 在中断处理过程中控制输入继电器及输出继电器时，使用输入输出刷新指令(REFF)，可以通过读取最新的输入状态、或者立即输出运算结果，实现高速控制；
- 作为中断指针采用的输入继电器的编号，请不要与采用相同输入范围的[高速计数器]及 [脉冲密度 (FNC56)] 等的应用命令的编号相重复。
- 子程序及中断例行程序内的定时器，请采用例行程序用的定时器T192-T199；如果采用一般的定时器，除了不能进行计时外，在使用1ms累计定时器时亦需加注意；
- 如果指定输入中断指针I0口，则输入继电器的输入滤波特性自动关闭。因此，无需采用REFE(FNC51)指令及特殊数据寄存器D8020(输入滤波器调整)。另外，不作为输入中断指针用的输入继电器的输入滤波器能维持10ms(初始值)。

为了满足在高速计数器运行时，我们在H₂₀机型中另外新增加了24个高速计数中断，可以指定任意一个高速计数器产生24个中断响应，此功能命名为高速计数器多用户中断（XP不具备此功能）。使用设置有如下规律；

标志位	描述
M8084	为ON使能高速计数器多用户中断（XP不具备此功能）
D8084	为高速计数器序号C235~C255
D8085	对应的用户中断个数，最大24个，从I507~I530
D8086	对应多个比较点数据的序号，只能为D元件，如200为D200开始的双字

比较点数据存放例子：

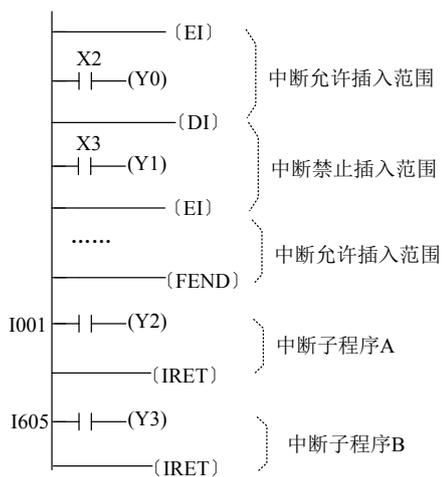
D8084=235； D8086=200； D8085=5； M8084=ON；

C235的数据	记录单元	存放单元数值	对应的用户中断	D8131的数值
100	D200, D201	=100	I507	0
200	D202, D203	=200	I508	1
300	D204, D205	=300	I509	2
400	D206, D207	=400	I510	3
500	D208, D209	=500	I511	4→0(M8133=ON)

每个中断可以由高速计数器的数值和记录单元的数值产生。

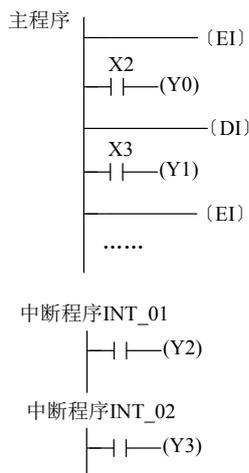
指令举例：

FNC03
FNC04
FNC05
IRET
EI
DI



PLC 执行时，当程序扫描到EI 指令到DI 指令间，X0=On 或者定时时间5MS到时，则执行中断插入子程序A或B，而当子程序执行至IRET时，则返回主程序并继续往下执行。

在AutoShop中左图的程序如下：（右击程序块的中断程序INT_01或者INT_02，选择属性。可以修改01或者02为你想要的名称，包括中文）



分别右击程序块的中断程序INT_01和INT_02，选择属性-中断事项：分别设为I001和I605

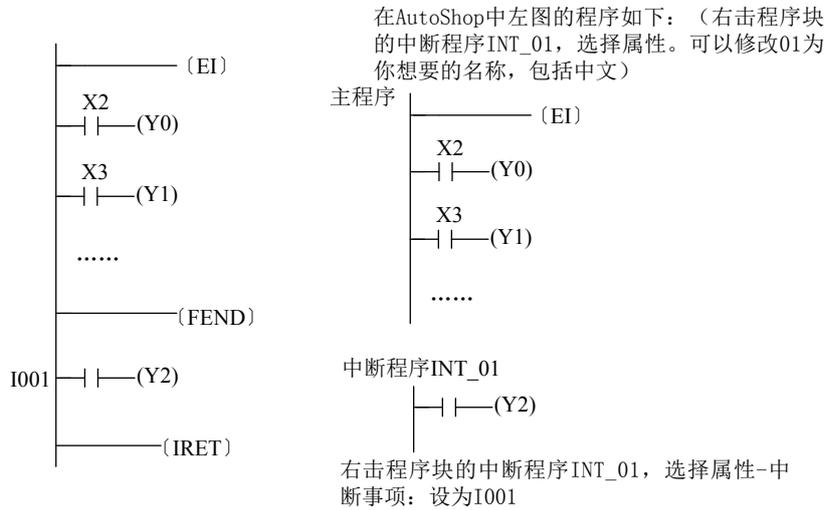
16bit	32bit		FNC06	FEND	主程序完毕。
✓					
1步			无需触点驱动，无操作数的单独指令		

FNC06
FEND

FEND语句位于主程序的结束处，执行了该指令后，即结束了本次用户程序的扫描，向0步程序返回，再次从头对程序进行扫描。

CALL命令调用的子程序必须写在FEND命令后，并且在该子程序结束加上SRET命令；中断子程序也必须写在FEND之后，并在该中断程序结束加上IRET指令，在AutoShop中，子程序或中断程序必须在单独窗口编写，在主程序末尾无需加FEND，在子程序或中断程序的末尾无需加SRET或IRET。

指令举例：



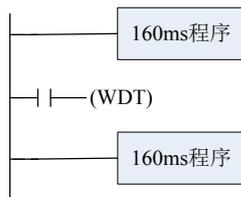
	16bit	32bit	\boxed{P}	FNC07	WDT	监视定时器复位。
FNC07	✓		✓			
WDT	1步		1步	无操作数的单独指令		

PLC系统内有用于监视用户程序执行一次的时间是否超时的定时器，若超时即会停止用户程序的执行并报警，执行WDT指令即可将该监视定时器复位，让监视定时器重新开始计时，避免超时错误。

$\overline{X2}$ (WDT) 若用户程序所执行的操作过于复杂（例如过多的循环计算），执行时有可能出现运行超时错误，编程时若必要，可用WDT指令（例如在FOR~NEXT指令之间中插入该指令）；

如果程序扫描时间大于D8000的值（默认200ms），可以在程序间插入WDT指令将每段程序分成扫描时间低于200ms或者根据需要修改D8000的设定值。

指令举例：



此程序扫描时间是320ms，用WDT指令将程序分割为2部分，使得每部分程序扫描时间都在200ms以下。

16bit	32bit		FNC 08	FOR	循环范围开始
✓					
3步			指令格式: FOR (S1)		

FNC08
FNC09

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

FOR
NEXT

FOR指令用于一个循环的起始，同时指明循环执行的次数，必须与NEXT指令配套使用。其中：

(S1) 为循环次数控制变量。

参见NEXT指令的解释与举例。

16bit	32bit		FNC09	NEXT	循环范围结束
✓					
1步			指令格式: NEXT(无操作数)		

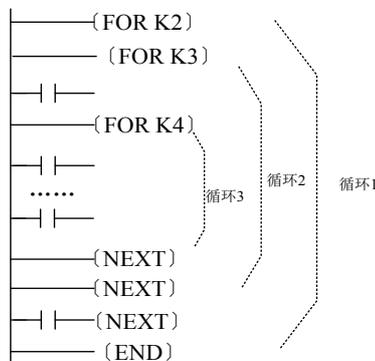
NEXT指令用于指示循环区域的尾部。由FOR指令指定FOR~NEXT循环来回执行N次后跳出FOR~NEXT循环往下继续执行。

在FOR~NEXT指令的循环区间，可以嵌入另一个FOR~NEXT循环，但规定：从最外层的FOR~NEXT计算，最多可内嵌6层FOR~NEXT循环。运行时PLC会以各FOR~NEXT层对应解析执行。但需要注意当循环次数过多时，会使PLC扫描周期延长，可能造成逾时监视定时器动作而导致错误产生。可在FOR~NEXT指令之间使用WDT指令来改善。

有下列情况者，都会出错：

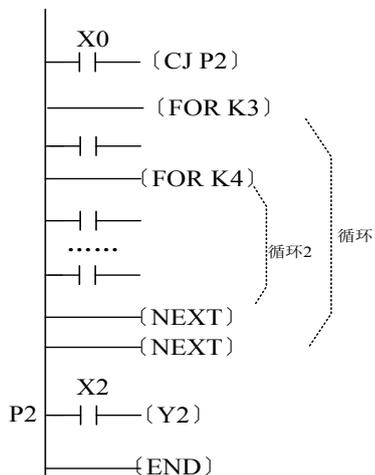
- NEXT指令在FOR指令之前；
- 有FOR指令而无NEXT指令；
- 在FEND，END指令以后有NEXT指令；
- FOR指令与NEXT指令个数不一致等。

指令举例一：



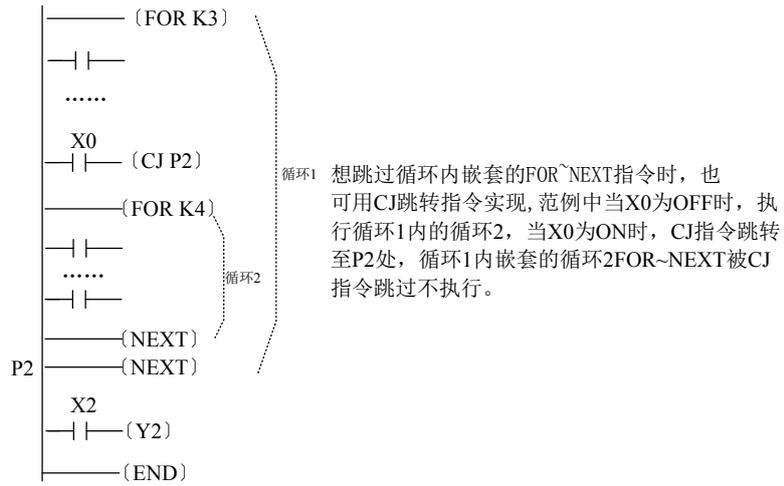
循环1执行2次后在到NEXT指令以后的程序继续执行，而循环1每执行一次循环2会执行3次，而循环2每执行一次循环3又执行4次，所以循环3共执行2*3*4=24次，循环2共执行2*3=6次

指令举例二：



想跳过FOR~NEXT指令时，可用CJ跳转指令实现，范例中当X0为OFF时，执行循环1和循环2，当X0为ON时，CJ指令跳转至P2处，循环1和循环2之间的程序不被执行

指令举例三



4.3.2 传送与比较 (10~19)

16bit	32bit	□	FNC 10	CMP	两个数比较
✓	✓	✓			
7步	13步		指令格式: CMP (S1) (S2) (D)		

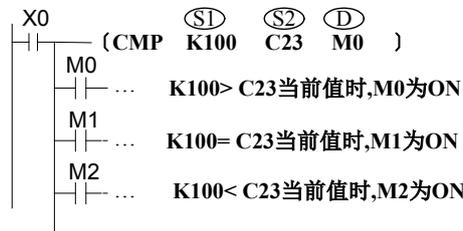
操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(S2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(D)		✓	✓	✓												

FNC10
CMP

本指令完成对两个操作变量的大小作比较，将比较结果输出给指定的位变量，操作数均按有符号数进行代数比较操作。

其中 (D) 会占用3个连续地址的位变量。

指令举例：



当X0=ON时，M0~M2其中之一会ON。
 X0由ON变OFF时，不执行CMP指令，M0~M2仍保持X0=OFF之前的状态，若要清除M0~M2的比较结果可用RST或者ZRST对M0~M2进行清除。
 若需要得到≥、≤、≠的结果时，可将M0~M2串并联即可取得。

16bit	32bit	\square	FNC 11	ZCP	区间比较
✓	✓	✓			
9步	17步		指令格式: ZCP (S1) (S2) (S) (D)		

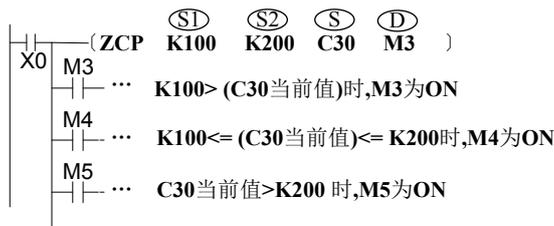
操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(S2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(S)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(D)		✓	✓	✓												

需要触点驱动，有4个操作变量。当控制能流有效时，按有符号数进行代数比较操作，以 (S1) (S2) 为区间，将 (S) 的值位于该区间的位置作为结果，存入 (D) 为起始地址的3个连续位变量中。

其中：

- (S1) 为比较区间的下限；
- (S2) 为比较区间的上限；
- (S) 为比较变量；
- (D) 为比较结果存储单元，会占用3个连续地址的位变量。

指令举例：



当X0=ON时，M3~M5其中之一会ON。
X0由ON变OFF时，不执行ZCP指令，M3~M5仍保持X0=OFF之前的状态，若要清除M3~M5的比较结果可用RST或者ZRST对M3~M5进行清除。

16bit	32bit		FNC 12	MOV	数据移动
✓	✓	✓			
5步	9步		指令格式: MOV (S) (D)		

操作数	位元件				字元件												FNC12
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z		
(S)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	MOV
(D)								✓	✓	✓	✓	✓	✓	✓	✓	✓	MOV

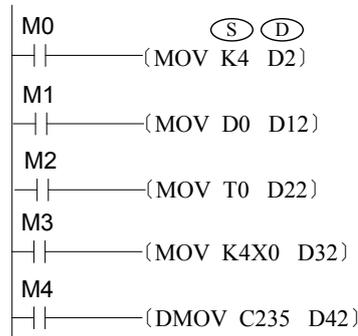
需要触点驱动，有2个操作变量，将 (S) 的值复制到 (D) 中。

当为32bit指令 (DMOV) 时，(S) 和 (D) 都会使用相邻高地址的变量单元参与运算。

例如语句：(DMOV D1 D5) 的操作结果是：D1→D5；D2→D6。

指令举例：

梯形图



指令列表

```

LD M0
MOV K4 D2
LD M1
MOV D0 D12
LD M2
MOV T0 D22
LD M3
MOV K4X0 D32
LD M4
DMOV C235 D42
    
```

当M0为ON时，将K4复制到D2中，当M0由ON ⇒ OFF时，D2保存K4的内容不变，除非用户程序再次将D2的值修改。或者PLC由STOP ⇒ RUN和PLC重新上电，D2的值才会变为0，停电保持寄存器上电或者由停止到运行保持原来的值不变。

16bit	32bit	\square	FNC 13	SMOV	移位传送
✓		✓			
11步		11步	指令格式: SMOV (S) (m1) (m2) (D) (n)		

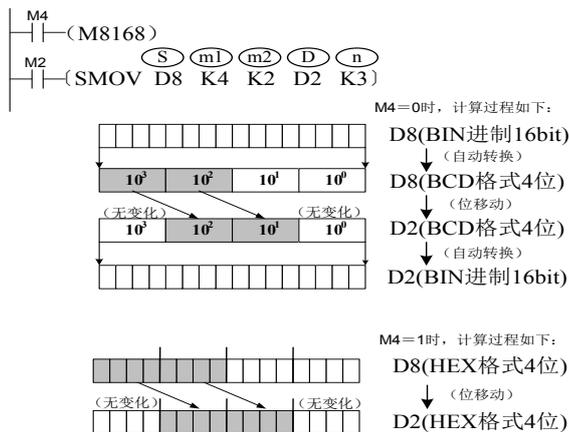
操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)							✓	✓	✓	✓	✓	✓	✓	✓	✓	
(m1)					✓	✓										
(m2)					✓	✓										
(D)								✓	✓	✓	✓	✓	✓	✓	✓	
(n)					✓	✓										

需要触点驱动，最多有5个操作变量，其中：

- (S) 为待复制的数据源变量；
- (m1) 为数据源传送的起始位号，(1~4)范围；
- (m2) 为数据源传送的位数，(1~m1)范围；
- (D) 为数据源传送的目的变量；
- (n) 为数据源传送的目的变量的起始位，(m2~4)范围。

数据位的传送过程与特殊标志M8168的状态有关，当M8168为OFF时是BCD模式（十进制的位），当M8168为ON时是BIN模式，在BIN模式下以4个位作为一个单位作传送（十六进制的位）。

指令举例：



假设D8=K1234，D2=K5678，则当M8168为OFF时（BCD模式），将M2置ON，则D2的值变为K5128；

当M8168为ON时（BIN模式），此时D8=H04D2=K1234，D2=H162E=K5678，将M2置ON，则D2=H104E=K4174。

16bit	32bit		FNC 14	CML	数据取反传送
✓	✓	✓			
5步	9步		指令格式: CML (S) (D)		

操作数	位元件				字元件												FNC14			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z					
(S)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
(D)									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	CML

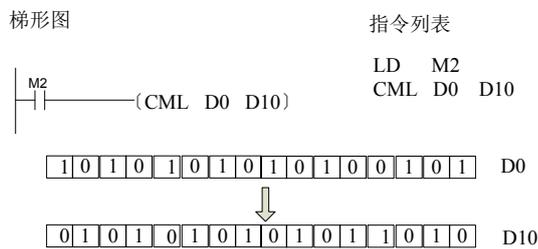
需要触点驱动，有2个操作变量，将 (S) 的BIN值逐位取反后复制到 (D) 中。

当 (D) 的位数不足16bit时，将 (S) 取反后按低位对齐传送到 (D) 变量中；

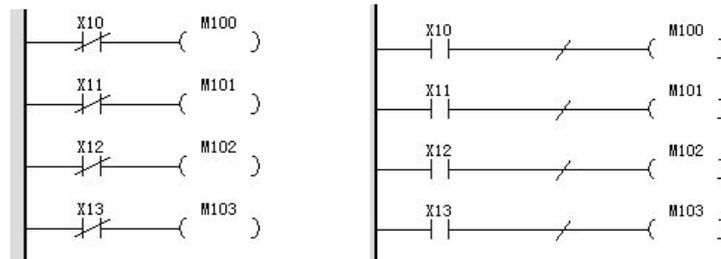
当为32bit指令（DCML）时，(S) 和 (D) 都会使用相邻高地址的变量单元参与运算。

例如语句：(DCML D1 D5) 的操作结果是：/D1→D5；/D2→D6

指令举例一：



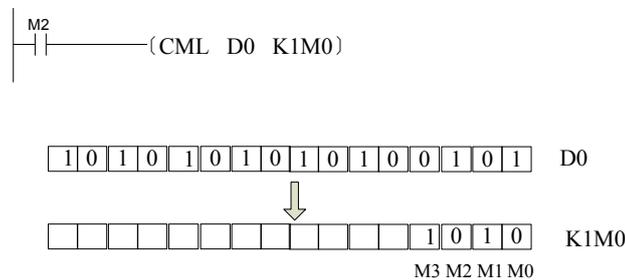
指令举例二：



上面2个程序可以用下面的CML指令来实现



指令举例三：

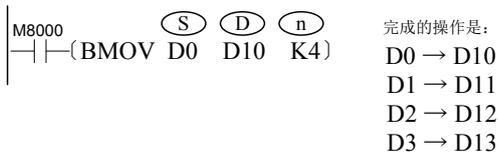


16bit	32bit	\square	FNC 15	BMOV	数据成批传送
✓		✓			
7步		7步	指令格式: BMOV (S) (D) (n)		

FNC15	操作数	位元件				字元件										
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
BMOV	(S)							✓	✓	✓	✓	✓	✓	✓		
	(D)								✓	✓	✓	✓	✓	✓		
	(n)	常数, n=1~512														

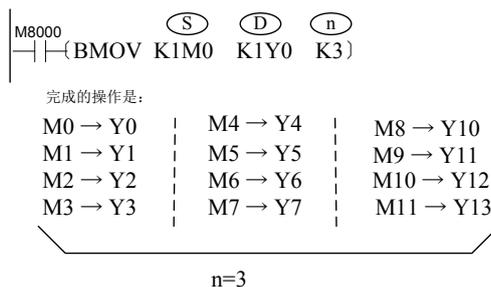
需要触点驱动, 有3个操作变量, 将由 (S) 指定起始地址的 (n) 个变量值复制到由 (D) 指定起始地址的 (n) 个单元中。其中 (n) 的取值范围是1~512。

当特殊变量M8024=1时, 成批传送的方向相反, 即将由 (D) 指定起始地址的 (n) 个变量值复制到由 (S) 指定起始地址的 (n) 个单元中。



当操作数为位元件时, (S) 和 (D) 位数必须相等。

指令举例:



16bit	32bit		FNC 16	FMOV	数据多点传送
✓	✓	✓			
7步	13步		指令格式: FMOV (S) (D) (n)		

操作数	位元件				字元件												FNC16	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z			
(S)							✓	✓	✓	✓	✓	✓	✓					
(D)								✓	✓	✓	✓	✓	✓					
(n)	常数, n=1~512																FMOV	

需要触点驱动，有3个操作变量，将由 (S) 的数据复制到由 (D) 指定起始地址的 (n) 个单元中。

其中 (n) 的取值范围是1~512。

FMOV是16位的多点传送指令，DFMOV是32位多点传送。

指令举例：



当M8置ON时完成的操作是：

- k100 → D100
- k100 → D101
- k100 → D102
- k100 → D103

16bit	32bit	\square	FNC 17	XCH	数据交换
✓	✓	✓			
5步	9步		指令格式: XCH \textcircled{S} \textcircled{D}		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
\textcircled{S}								✓	✓	✓	✓	✓	✓			
\textcircled{D}								✓	✓	✓	✓	✓	✓			

需要触点驱动，有2个操作变量，将 \textcircled{S} 和 \textcircled{D} 的值彼此交换。

指令举例一：



执行前
D110=K180
D120=K200

⇒

执行后
D110=K200
D120=K180

指令举例二：



执行前
D110=K180
D111=K150
D120=K200
D121=K100

⇒

执行后
D110=K200
D111=K100
D120=K180
D121=K150

当特殊变量M8160=1时，且 \textcircled{D} 与 \textcircled{S} 为同一地址，完成的操作将是高8位与低8位的交换，32位的指令也一样，完成的操作将是高8位与低8位的交换。相当于SWAP指令的操作。一般用SWAP指令来实现。

16bit	32bit		FNC 18	BCD	BCD交换
✓	✓	✓			
5步	9步		指令格式: BCD (S) (D)		

操作数	位元件				字元件												FNC18
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	BCD	
(S)									✓	✓	✓	✓	✓	✓	✓		✓
(D)									✓	✓	✓	✓	✓	✓	✓	✓	

需要触点驱动，有2个操作变量，将 (S) (BIN) 的值进行BCD变换后存入 (D) 中。该指令常用于将数据显示前的数据格式处理。

使用16bit指令，当转换结果超过9999时会出错；使用32bit指令，当转换结果超过99999999时会出错。M8067、M8068会置ON，D8067记录错误代码。

指令举例：



将D200的BIN值转换成BCD值后，将结果的个位数存于K1Y0中（Y0~Y3四个bit元件）。
 若D200=H000E(十六进制)=K14（十进制），则变换后Y0~Y3=0100(BIN)
 若D200=H0028(十六进制)=K40（十进制），则变换后Y0~Y3=0000(BIN)

16bit	32bit	\square	FNC 19	BIN	BIN交换
✓	✓	✓			
5步	9步		指令格式: BIN (S) (D)		

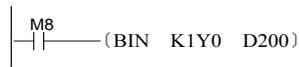
FNC19	操作数	位元件				字元件										
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
	(S)								✓	✓	✓	✓	✓	✓	✓	✓
	(D)								✓	✓	✓	✓	✓	✓	✓	✓

需要触点驱动，有2个操作变量，将 (S) (BCD) 的值进行BIN变换后存入 (D) 中。该指令常用于将外部端口读入数据（如编码盘设置）处理成能直接用于运算的BIN格式。

(S) (BCD) 的有效范围，16bit: 0~9999; 32bit: 0~99, 999, 999

(S) 的数据内容不是BCD值（以Hex表示有任一位数不在0~9的范围内）时将会产生运算错误，M8067、M8068会置位。

指令举例：



当M8置位时将K1Y0的BCD值作
BIN转换后存入D200中

4.3.3 四则逻辑运算 (20~29)

16bit	32bit	\square	FNC 20	ADD	BIN加法运算
✓	✓	✓			
7步	13步		指令格式: ADD (S1) (S2) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(S2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(D)								✓	✓	✓	✓	✓	✓	✓	✓	✓

需要触点驱动，有3个操作变量，将 (S1) 和 (S2) 的值进行BIN代数相加后存入 (D) 中，参与运算的变量都按有符号数处理，最高位为符号位，0为正数，1为负数。

若计算结果为0，则0标志(M8020)会置位；

若计算结果超过32,767 (16bit运算) 或2, 147, 483, 647 (32bit运算) 时，进位标志(M8022)会置位；

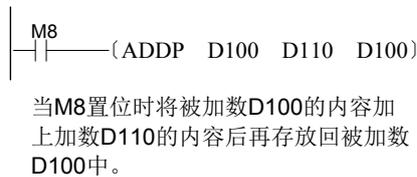
若计算结果不满-32,768 (16bit运算) 或-2,147,483,648 (32bit运算) 时，借位标志(M8021)会置位；

进行32bit运算时，指令中变量地址为为低16bit地址，相邻高编号地址单元为高16bit，编程时防止重复或误覆盖。

指令举例一：



指令举例二：



16bit	32bit	\square	FNC 21	SUB	BIN减法运算
✓	✓	✓			
7步	13步		指令格式: SUB (S1) (S2) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(S2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(D)								✓	✓	✓	✓	✓	✓	✓	✓	

FNC21

SUB

需要触点驱动，有3个操作变量，将(S1)和(S2)的值进行BIN代数相减后存入(D)中，参与运算的变量都按有符号数处理，最高位为符号位，0为正数，1为负数。

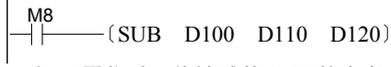
若计算结果为0，则0标志(M8020)会置位；

若计算结果超过32,767（16bit运算）或-2,147,483,647（32bit运算）时，进位标志(M8022)会置位；

若计算结果不满 -32,768（16bit运算）或-2,147,483,648（32bit运算）时，借位标志(M8021)会置位；

进行32bit运算时，指令中变量地址位为低16bit地址，相邻高编号地址单元为高16bit，编程时防止重复或误覆盖。

指令举例：



当M8置位时，将被减数D100的内容减去减数D110的内容后存放到D120中，假如D100=K10;D110=K8,则D120=10-8=K2

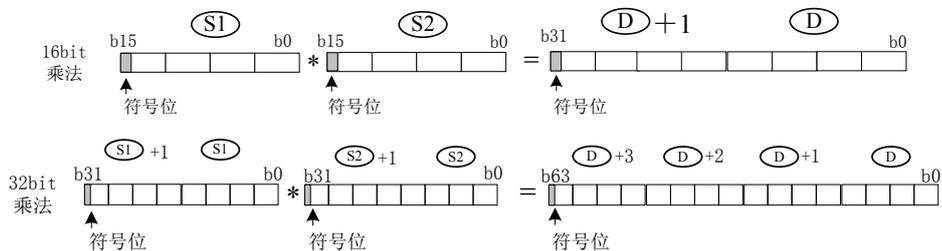
16bit	32bit		FNC 22	MUL	BIN乘法运算
✓	✓	✓			
7步	13步		指令格式: MUL (S1) (S2) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(S2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(D)								✓	✓	✓	✓	✓	✓			

FNC22
MUL

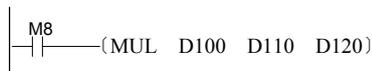
需要触点驱动，有3个操作变量，将(S1)和(S2)的值进行BIN代数相乘后存入(D)中，参与运算的变量都按有符号数处理，最高位为符号位，0为正数，1为负数。需注意32bit相乘的时候积占用4个寄存器。

进行32bit运算时，指令中变量地址为16bit地址，相邻高地址地址单元为高16bit，编程时防止重复或误覆盖；计算的结果只能为32bit，对于超出32bit范围的计算，最好采用浮点运算指令EMUL进行计算。



指令举例:

梯形图



指令列表

```
LD M8
MUL D100 D110 D120
```

当M8置位时将乘数D100的内容乘以乘数D110的内容后存放到D120中。
 假如D100=K5;D110=K9,则D120=5×9=K45
 假如D100=K1234;D110=K5678,则D120,d121=1234×5678=K7006652,需注意此时积大于16bit,需用到D的相邻高位D121,D120

16bit	32bit	\square	FNC 23	DIV	BIN除法运算
✓	✓	✓			
7步	13步		指令格式: DIV (S1) (S2) (D)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(S2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(D)								✓	✓	✓	✓	✓	✓		

FNC23

DIV

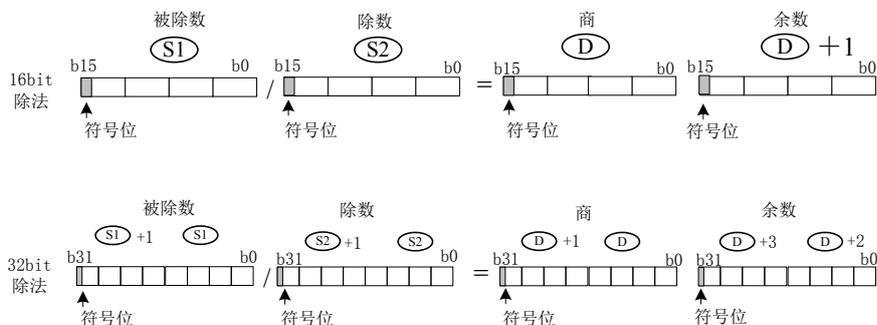
需要触点驱动，有3个操作变量，将被除数(S1)和除数(S2)的值进行BIN代数相除后存入(D)中，参与运算的变量都按有符号数处理，最高位为符号位，0为正数，1为负数。

表中Y、Z元件仅在16bit运算(S1)可用。(S2)变量地址为低16bit地址，相邻高编号地址单元为高16bit，编程时防止重复或误覆盖：计算所得的商存入(D)、(D)+1所指单元，余数存入(D)+2、(D)+3地址单元中。

若除数(S2)为0，会发生计算错误；

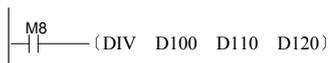
若将位元件(KnX/KnY/KnM/KnS)指定为(D)，不能得到余数；

若被除数为负数，余数即为负数。



指令举例：

梯形图



指令列表

```
LD M8
DIV D100 D110 D120
```

当M8置位时将 被除数D100的内容除以除数D110的内容后存放到D120中。
假如D100=K5；D110=K2，则D120=K2，商存放于D121，D121=K1。

16bit	32bit		FNC 24	INC	BIN加1运算
✓	✓	✓			
3步	5步		指令格式: INC (D)		

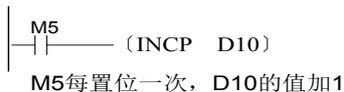
操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(D)								✓	✓	✓	✓	✓	✓	✓	✓	

指令每执行一次，**(D)** 中的数值增加1。

16位运算时，32,767再加1变为-32,768；32位运算时，2，147，483，647再加1变为-2,147,483,648。

本指令对0标志、进位、借位标志都不刷新

指令举例：



FNC24

INC

16bit	32bit	\square	FNC 25	DEC	BIN减1运算
✓	✓	✓			
3步	5步		指令格式: DEC \textcircled{D}		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
\textcircled{D}								✓	✓	✓	✓	✓	✓	✓	✓	

FNC25

指令每执行一次， \textcircled{D} 的数值减1。

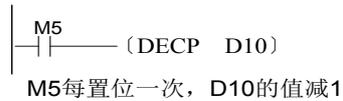
16位运算时，-32,768再减1变为32,767；32位运算时，-2,147,483,648再减1变为2, 147, 483, 647。

DEC

本指令对0标志、进位、借位标志都不刷新。

进行32bit运算时，指令中 \textcircled{D} 变量地址位为低16bit地址，相邻高编号地址单元为高16bit，编程时防止重复或误覆盖。

指令举例：



16bit	32bit	<input checked="" type="checkbox"/>	FNC 26	WAND	逻辑与
✓	✓	✓			
7步	13步		指令格式: WAND (S1) (S2) (D) (32bit指令符为DAND)		

本指令执行时, 将 (S1) 和 (S2) 中BIN值的各位对应作“逻辑与”运算, 将结果存入 (D) 变量。
逻辑的‘与’ (AND) 运算的规则为任一为0结果为0。

1·1=1 1·0=0 0·1=0 0·0=0

16bit	32bit	<input checked="" type="checkbox"/>	FNC 27	WOR	逻辑或
✓	✓	✓			
7步	13步		指令格式: WOR (S1) (S2) (D) (32bit指令符为DOR)		

FNC26
FNC27
FNC28
WAND
WOR
WXOR

本指令执行时, 将 (S1) 和 (S2) 中BIN值的各位对应作“逻辑或”运算, 将结果存入 (D) 变量。
逻辑的‘或’ (OR) 运算的规则为任一为1结果为1。

1+1=1 1+0=1 0+1=1 0+0=0

16bit	32bit	<input checked="" type="checkbox"/>	FNC 28	WXOR	逻辑异或
✓	✓	✓			
7步	13步		指令格式: WXOR (S1) (S2) (D) (32bit指令符为DXOR)		

本指令执行时, 将 (S1) 和 (S2) 中BIN值的各位对应作“逻辑异或”运算, 将结果存入 (D) 变量。
逻辑的‘异或’ (XOR) 运算的规则为两者相同结果为0, 两者不同结果为1。

1⊕1=0 1⊕0=1 0⊕1=1 0⊕0=0

上述三个指令的适用机型:

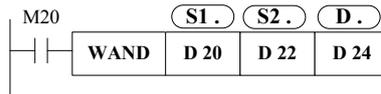
适用机型		
系列	通用	XP
H1U	✓	✓
H2U	✓	✓

这三个指令操作数适用变量类型如下表, 当为32bit指令时, 寄存器变量则占用后续相邻地址的共2个单元:

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(S2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(D)								✓	✓	✓	✓	✓	✓	✓	✓	

指令举例:

逻辑与



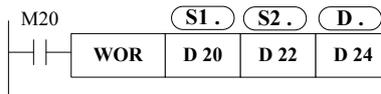
D20 0 0 1 0 0 0 1 1 0 0 1 0 1 1 0 1

D22 0 1 0 1 0 0 1 0 0 1 0 0 1 1 1 1



D24 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 1

逻辑或



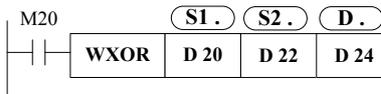
D20 0 0 1 0 0 0 1 1 0 0 1 0 1 1 0 1

D22 0 1 0 1 0 0 1 0 0 1 0 0 1 1 1 1



D24 0 1 1 1 0 0 1 1 0 1 1 0 1 1 1 1

逻辑异或



D20 0 0 1 0 0 0 1 1 0 0 1 0 1 1 0 1

D22 0 1 0 1 0 0 1 0 0 1 0 0 1 1 1 1



D24 0 1 1 1 0 0 0 1 0 1 1 0 0 0 1 0

FNC26
FNC27
FNC28WAND
WOR
WXOR

16bit	32bit		FNC 29	NEG	求补运算
✓	✓	✓			
3步	5步		指令格式: NEG (D)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(D)								✓	✓	✓	✓	✓	✓	✓	✓

FNC29
NEG

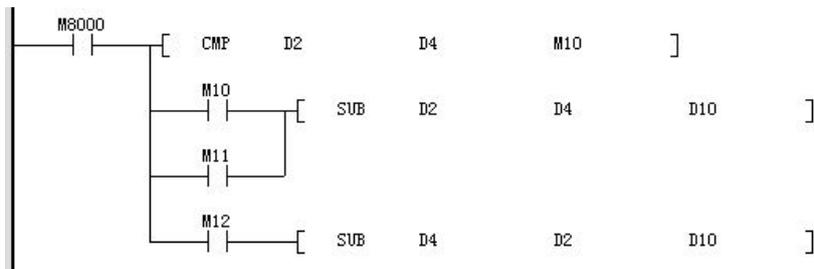
需要触点驱动，有1个操作变量。将 **(D)** 的数值逐位取反、再加1，存回 **(D)** 中。

此指令一般用脉冲执行型指令。

使用NEG指令，可得到与负的BIN值相对应的绝对值。

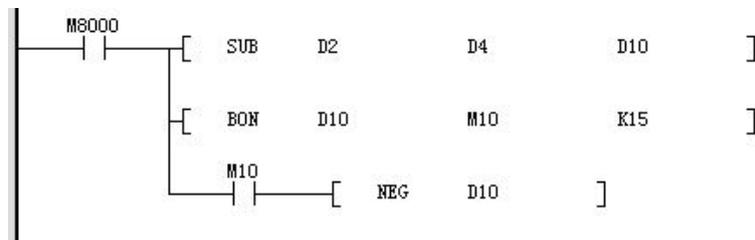
指令举例：

减法运算的差取绝对值



若D2>D4时，M10=On。若D2=D4时，M11=On。若D2<D4时，M12=On。由此可保证D10为正值。

此程序可用下列的程序来表示：



当D10的bit15为“1”时（表示D10为负数），M10=On，用NEG指令将D10取补码可得到D10的绝对值。

上述两例中假如D2=K4，D4=K8；或者D2=K8，D4=K4，D10的结果均为K4。

补充说明：负数的表现及绝对值：

- 1) 正负数是以寄存器最上位（最左边）的位内容来表现，为“0”时为正数、为“1”时为负数。
- 2) 最高位为1时，可使用NEG指令将它转成绝对值。

(D 10)=2

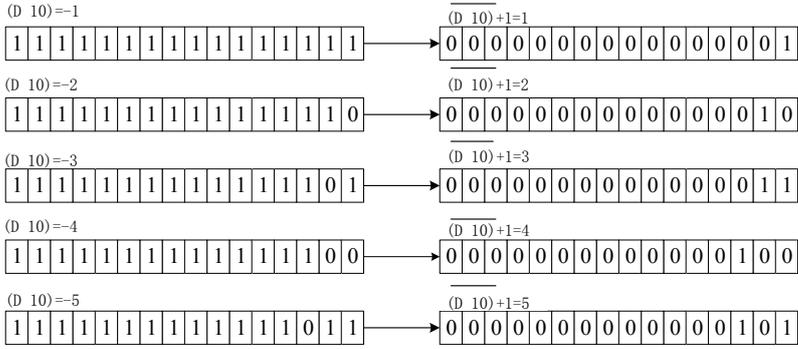
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(D 10)=1

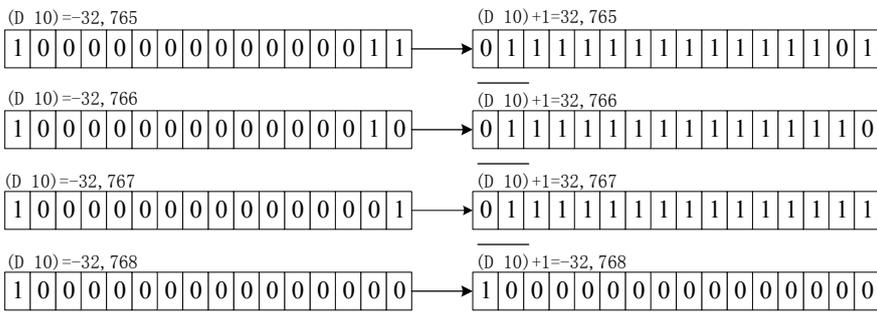
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(D 10)=0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



FNC29
NEG



↗
绝对值的最大范围只能达到**32,767**

4.3.4 循环移位 (30~39)

16bit	32bit		FNC 30	ROR	循环右移
✓	✓	✓			
5步	9步		指令格式: ROR (D) (n)		

将 (D) 的内容循环右移 (n) 位。
本指令一般使用脉冲执行型指令。

16bit	32bit		FNC 31	ROL	循环左移
✓	✓	✓			
5步	9步		指令格式: ROL (D) (n)		

将 (D) 的内容循环左移 (n) 位。
本指令一般使用脉冲执行型指令。

FNC30
FNC31

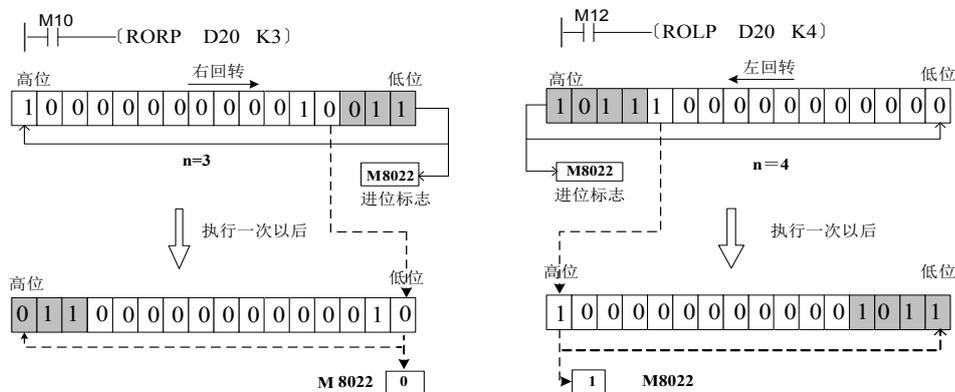
ROR
ROL

上述两个指令的操作数适用变量类型如下表，当为32bit指令时，寄存器变量则占用后续相邻地址的共2个单元：

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(D)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(n)	常数, n=1~16 (16bit) ; n=1~32 (32bit)														

若 (D) 中指定KnY、KnM、KnS时，只有K4 (16bit) 及K8 (32bit) 有效；
循环移动的最终位被存入进位标志中。

指令举例：



16bit	32bit	\square	FNC 32	RCR	带进位循环右移
✓	✓	✓			
5步	9步		指令格式: RCR (D) (n)		

将 (D) 的内容连同进位标志M8022循环右移 (n) 位。

本指令一般使用脉冲执行型指令。

16bit	32bit	\square	FNC 33	RCL	带进位循环左移
✓	✓	✓			
5步	9步		指令格式: RCL (D) (n)		

将 (D) 的内容连同进位标志M8022循环左移 (n) 位。

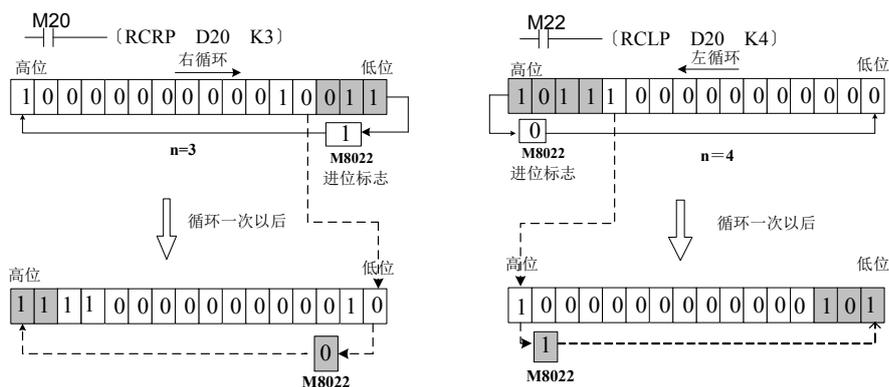
本指令一般使用脉冲执行型指令。

上述两个指令的操作数适用变量类型如下表，当为32bit指令时，寄存器变量则占用后续相邻地址的共2个单元：

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(D)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(n)	常数, n=1~16 (16bit) ; n=1~32 (32bit)														

若 (D) 中指定KnY、KnM、KnS时，只有K4 (16bit) 及K8 (32bit) 有效；

指令举例：



16bit	32bit		FNC 34	SFTR	位右移
✓		✓			
9步		9步	指令格式: SFTR (S) (D) (n1) (n2)		

对于位变量, 将 (S) 地址起始的 (n2) 位变量与 (D) 地址起始的 (n1) 变量, 按向右方向移动 (n2) 位后, 将结果保存在 (D) 中。

注: (n1): 移位数据的位数据长度; (n2): 右移的位点数。

本指令一般使用脉冲执行型指令。

16bit	32bit		FNC 35	SFTL	位左移
✓		✓			
9步		9步	指令格式: SFTL (S) (D) (n1) (n2)		

对于位变量, 将 (S) 地址起始的 (n2) 位变量与 (D) 地址起始的 (n1) 变量, 按向左方向移动 (n2) 位后, 将结果保存在 (D) 中。

注: (n1): 移位数据的位数据长度; (n2): 左移的位点数。

本指令一般使用脉冲执行型指令。

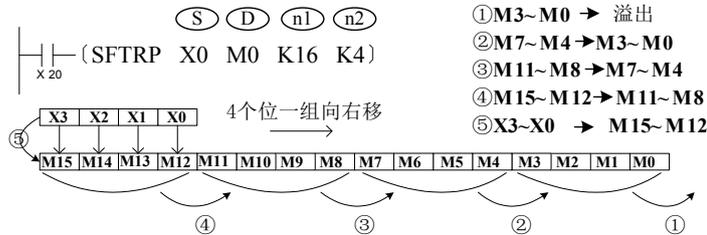
上述两个指令的操作数适用变量类型如下表:

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)	✓	✓	✓	✓												
(D)		✓	✓	✓												
(n1)	常数, H _{1U} 机型: n1≤256; H _{2U} 机型n1≤1024															
(n2)	常数, n2≤n1															

FNC34
FNC35
SFTR
SFTL

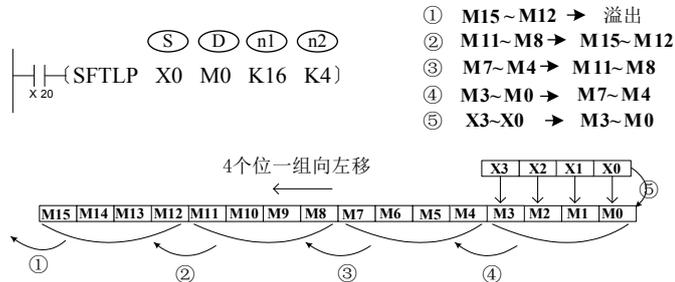
指令举例一:

SFTR命令:



指令举例二:

SFTL命令:



16bit	32bit	\square	FNC 36	WSFR	字右移
✓		✓			
9步		9步	指令格式: WSFR (S) (D) (n1) (n2)		

以字为单位, 将 (S) 地址起始的 (n2) 字变量与 (D) 地址起始的 (n1) 字变量, 按向右方向移动 (n2) 个字

注: (n1): 移位数据的字数据长度; (n2): 右移的字点数。

本指令一般使用脉冲执行型指令。

16bit	32bit	\square	FNC 37	WSFL	字左移
✓		✓			
9步		9步	指令格式: WSFL (S) (D) (n1) (n2)		

以字为单位, 将 (S) 地址起始的 (n2) 字变量与 (D) 地址起始的 (n1) 字变量, 按向左方向移动 (n2) 个字。

注: (n1): 移位数据的字数据长度; (n2): 左移的字点数。

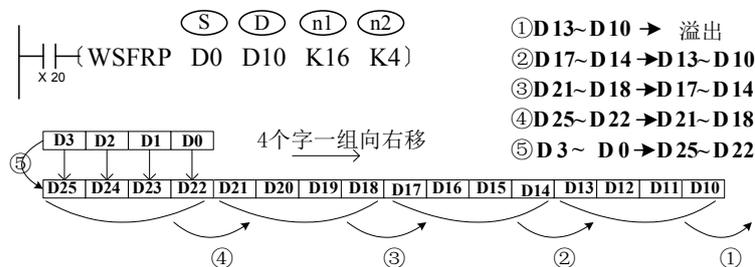
本指令一般使用脉冲执行型指令。

上述两个指令的操作数适用变量类型如下表:

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)							✓	✓	✓	✓	✓	✓	✓			
(D)								✓	✓	✓	✓	✓	✓			
(n1)	常数, H ₁₀ 机型: n1≤2048; H ₂₀ 机型n1≤2048															
(n2)	常数, n2≤n1															

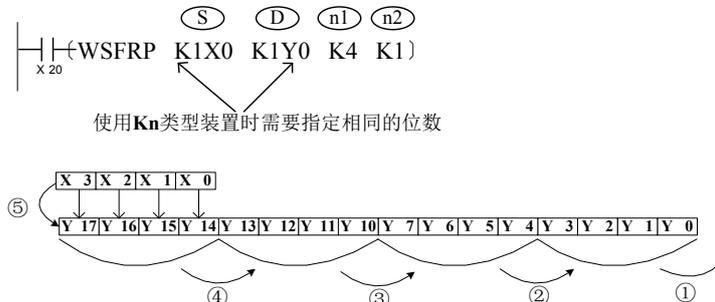
指令举例一:

WSFR命令:



指令举例二:

WSFR命令:

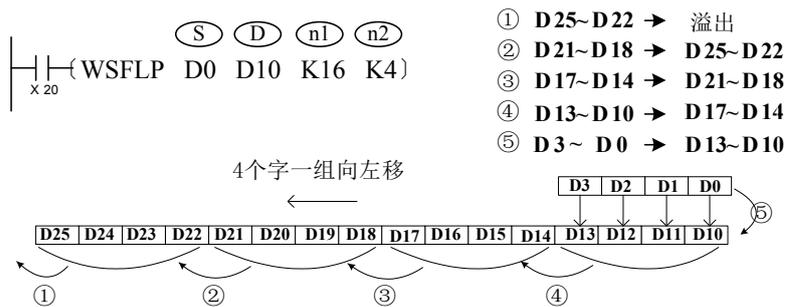


扫描一次的位数右移动作依照下列编号1~5动作。

- 1: Y3~Y0 → 进位
- 2: Y17~Y14 → Y13~Y10
- 3: Y13~Y10 → Y7~Y4
- 4: Y7~Y4 → Y3~Y0
- 5: X3~X0 → Y17~Y14 完成

指令举例三:

WSFL命令:



FNC36
 FNC37
 WSFR
 WSFL

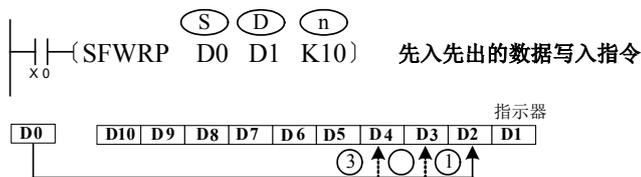
16bit	32bit	\square	FNC 38	SFWR	“先进先出”写入
✓		✓			
7步		7步	指令格式: SFWR (S) (D) (n)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)							✓	✓	✓	✓	✓	✓	✓	✓	✓	
(D)								✓	✓	✓	✓	✓	✓	✓	✓	
(n)	常数, 2≤n≤2048 (不能超出 (D) 的元件使用范围)															

将 (S) 的值写入由 (D) 地址起始, 个数为 (n) 的“先进先出”队列中, 以第一个编号装置作为指针, 当指令执行时, 指针内容值先加1, 之后S所指定的装置其内容值会写入先进先出 (D) 数据串列中由指针所指定的位置。

本指令一般使用脉冲执行型指令。

指令举例:



当X0=1时, D0的内容被存入D2, D1的内容变为1。当X0再次从OFF→ON时, D0的内容被存入D3, D1的内容变为2, 以此类推。若D1的内容超过n-1, 则指令不处理, 而进位标志M8022会置1。

FNC38

SFWR

16bit	32bit		FNC 39	SFRD	“先进先出”读出
✓		✓			
7步		7步	指令格式: SFRD (S) (D) (n)		

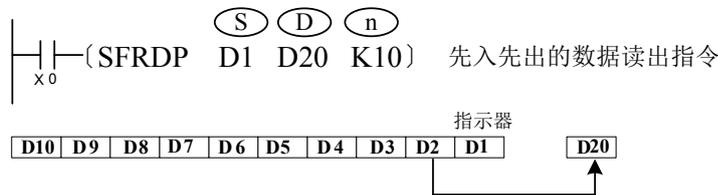
操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)								✓	✓	✓	✓	✓	✓	✓	✓	
(D)								✓	✓	✓	✓	✓	✓	✓	✓	
(n)	常数, 2≤n≤2048 (不能超出 (D) 的元件使用范围)															

从“先进先出”队列 (S) 的首项读出到 (D) 中，然后将队列 (S) 逐字右移1个字，将队列指针递减。以第一个编号装置作为指针，当指令执行时，指针内容值先减1，之后S所指定的装置其内容值会写入先进先出 (D) 数据串列中由指针所指定的位置。若指针已经为0，则指令不处理前述操作，而0标志M8020会置1

FNC39
SFRD

本指令一般使用脉冲执行型指令。

指令举例：



- X0由OFF到ON本指令动作依照下列编号1~3动作。(D10内容保持不变)，
- 1: D2的内容被读出传送至D20当中。
 - 2: D10~D3全部往右移位一个寄存器。
 - 3: 指针D1内容减1。

4.3.5 数据处理 (40~49)

16bit	32bit	\square	FNC 40	ZRST	区间复位
✓		✓			
5步		5步	指令格式: ZRST (D1) (D2)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(D1)		✓	✓	✓							✓	✓	✓		
(D2)		✓	✓	✓							✓	✓	✓		

将 (D1) 至 (D2) 区间的变量全部清0。(D1) 和 (D2) 可指定字变量, 也可为Y、M、S位变量。

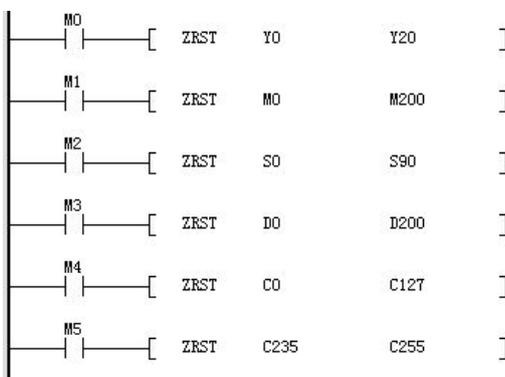
其中要求:

(D1) 和 (D2) 必须为同一类型的软元件;

编号 (D1) 应不大于 (D2), 若两者相同时, 仅复位指定的软元件;

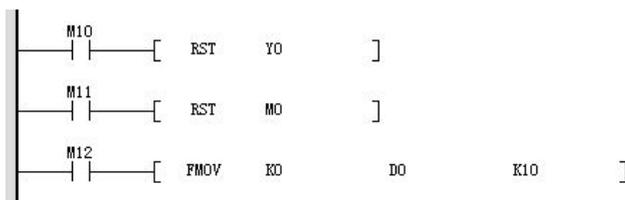
本指令为16bit, 但 (D1) 和 (D2) 可指定32bit的计数器, 此时应同为32bit型或同为16bit型;

指令举例:



补充说明:

位元件Y、M、S和字元件T、C、D也可使用RST指令来单独复位; 字元件T、C、D和位寄存器KnY、KnM、KnS也可以用FMOV来多点清除。例如:



FNC40

ZRST

16bit	32bit		FNC 41	DECO	解码
✓		✓			
7步		7步	指令格式: DECO (S) (D) (n)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)	✓	✓	✓	✓								✓	✓	✓	✓
(D)		✓	✓	✓								✓	✓	✓	
(n)	常数, n=1~8。若n=0, 指令不执行; 其他值则执行出错。														

计算 (S) 的最后n位的值, 作为bit位指针, 将 (D) 的对应位置1, 其他位清0。

源地址的低n位 (n≤4) 被解码至目标地址。n≤3时, 目标的高位都转为0;

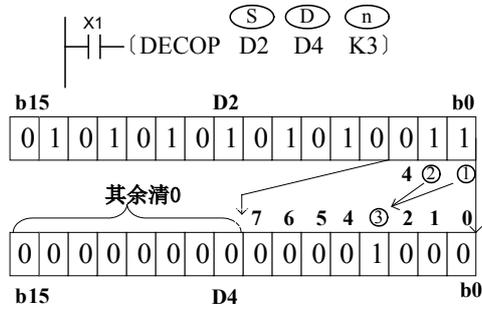
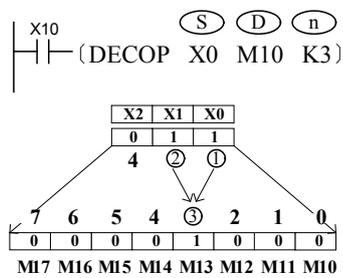
n=0时命令不执行, n=0~8以外时为运算错误;

n=8时, 如果译码命令 (D) 为位软元件时, 其点数是256点。

驱动输入为OFF时, 指令不执行, 正在动作的译码输出保持动作。

本指令一般使用脉冲执行型指令。

编程举例:



FNC41
DECO

16bit	32bit	\square	FNC 42	ENCO	编码
✓		✓			
7步		7步	指令格式: ENCO (S) (D) (n)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)	✓	✓	✓	✓								✓	✓	✓	✓	✓
(D)												✓	✓	✓	✓	✓
(n)	常数, n=1~8。若n=0, 指令不执行; 其他值则执行出错。															

计算 (S) 的最后 (n) 位的值, 作为bit位指针, 将 (D) 的对应位置1, 其他位清0。

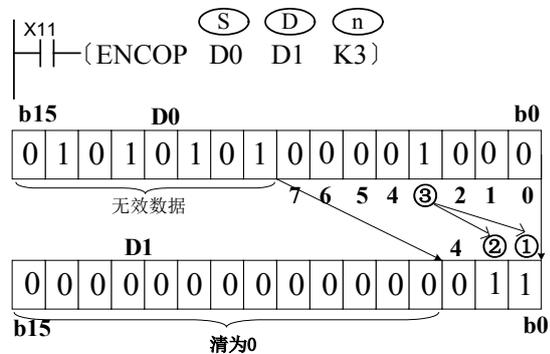
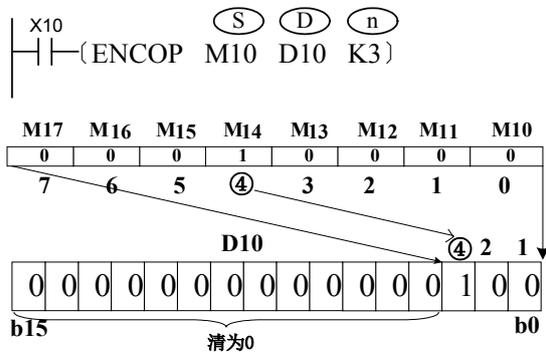
源地址内有多位是1时, 只计算高位侧的第一个为1的位; (S) 的所有位都为0时会出现运算错误;

驱动输入为OFF时, 指令不被执行, 编码输出不变化。

n=8时, 编码指令的 (S) 如果是位元件, 其点数是256点。

本指令一般使用脉冲执行型指令。

指令举例:



FNC42

ENCO

16bit	32bit	☐	FNC 43	SUM	ON位数
✓	✓	✓			
5步	9步		指令格式: SUM (S) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(D)								✓	✓	✓	✓	✓	✓	✓	✓	✓

计算 (S) 的BIN进制值中为1的位个数，存入 (D)。

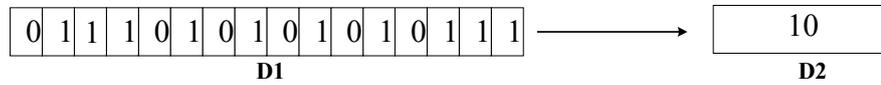
使用D SUM和D SUM P指令的情况下，((S)+1, (S))的32位中的1的个数写入 (D)，同时 (D)+1全部为0。

若 (S) 中的位全部为零，则零标志位M8020会置ON

指令举例：



将D1 中的位为1的个数总和存入D2中。



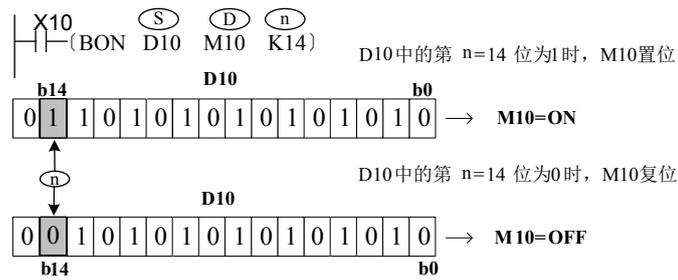
FNC43
SUM

16bit	32bit	\square	FNC 44	BON	ON位判断
✓	✓	✓			
5步	9步		指令格式: BON (S) (D) (n)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(D)		✓	✓	✓										✓	✓	
(n)	n=0~15 (16bit) ; n=0~31 (32bit)															

判断 (S) 的第 (n) 位的状态, 结果存入 (D)。

指令举例:



X10由ON变成Off 时, M10 仍保持之前的状态。

FNC44

BON

16bit	32bit		FNC 45	MEAN	平均值
✓	✓	✓			
7步	13步		指令格式: MEAN (S) (D) (n)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)							✓	✓	✓	✓	✓	✓	✓	✓	✓	
(D)								✓	✓	✓	✓	✓	✓	✓	✓	
(n)	常数, n=1~64, 其他值时, 计算会出错。															

求取由 (S) 开始的 (n) 个变量的平均值 (先求和, 再除以n), 存入 (D)。

若计算中有余数, 余数将被舍弃;

当n的值不在1~64的范围时, 会计算出错。

指令举例:



(D10+D11+D12+D13)/4=D20
 假如D10=K5,D11=K5,D12=K15,D13=K52;则D20=K19.余数1被舍去

FNC45
MEAN

M8490: Mean指令累加求和标志。

M8490=OFF时, 如上图所示。

M8490=ON时, Mean指令中 (D)+1, (D)+2存储上述变量的累加和。

如上例 (D22, D21) (32位) 存储 (D10+D11+D12+D13) 的和。

16bit	32bit	\square	FNC 46	ANS	报警器置位
✓					
7步			指令格式: ANS (S) (m) (D)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)											✓				
(D)				✓										✓	✓
(m)	常数, m=1~32767, (单位为100ms)。														

驱动信号报警器的方便指令。

其中 (S) 的范围为T0~T199, (D) 的范围为S900~S999。

指令举例:



如果X1和X2同时接通1秒以上, 则S900被置位, 以后即使X1或X2为OFF状态, S900仍保持动作状态 (但是T0会复位, 值变成0)。若不满1秒, X1或X2变为OFF时, 定时器复位。

如果预先将M8049 (信号报警器有效) 置ON, 则信号报警器S900~S999中最小ON状态编号被存入D8049 (ON状态最小编号) 且当S900~S999中任意一个为ON时, M8048 (报警器动作置ON)。

注: 相关软元件

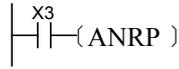
软元件	名称	内容
M8049	信号报警器有效	M8049置ON后, 下面的M8048和M8049工作。
M8048	信号报警器动作	M8049为ON, 状态S900~S999中任意一个动作的时候, M8048置ON。
D8049	ON状态最小编号	保存S900~S999中动作的最小编号。

FNC46

ANS

16bit	32bit		FNC 47	ANR	报警器复位
✓		✓			
1步		1步	指令格式: ANR(无操作数)		

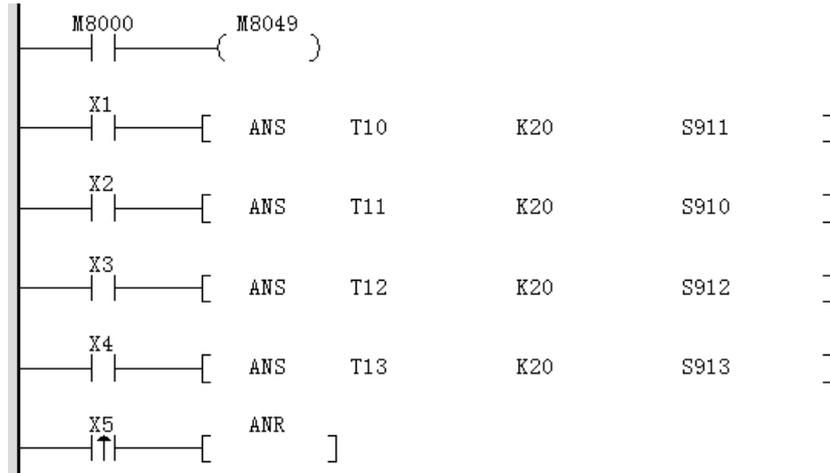
清除报警器信号的方便指令。例如:



如果X3接通, 则信号报警器S900~S999中正在动作的报警点被复位。如果同时有多个报警点动作时, 则复位最小编号为ON的报警点。

若将X3再次接通, 则下一编号的状态被复位。实际使用中多用ANRP指令。

指令举例:



M8049为ON, 当程序中S900~S999任意一个为ON时, M8048置ON。

假如程序中S910、S911、S912、S913都为ON, 则当X5第一次由OFF置ON时, S910被复位, 当X5第二次由OFF置ON时, S911被复位, 以此类推。

注: 相关软元件

软元件	名称	内容
M8049	信号报警器有效	M8049置ON后, 下面的M8048和M8049工作。
M8048	信号报警器动作	M8049为ON, 状态S900~S999中任意一个动作的时候, M8048置ON。
D8049	ON状态最小编号	保存S900~S999中动作的最小编号。

FNC47

ANR

16bit	32bit	\overline{P}	FNC 48	SQR	求平方根
✓	✓	✓			
5步	9步		指令格式: SQR (S) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)					✓	✓								✓	✓	✓
(D)														✓	✓	✓

将 (S) 按BIN值开平方运算，结果存入 (D)。

只能指定 (S) 为正数，如 (S) 为负数则运算错误标志M8067会置ON，指令不被执行；

运算结果 (D) 只取整数。舍去小数点，有小数点被舍去时借位标志M8021置ON；

运算结果是0时，零位标志M8020置ON。

指令举例：

FNC48

SQR

$$\begin{array}{|l} X2 \\ \hline (SQR D0 D12) \\ \hline \sqrt{D0} \rightarrow D12 \end{array}$$

假如D0=K100,则X2置ON的时候，D12=K10

假如D0=K110,则X2置ON的时候，D12=K10，小数被舍去

16bit	32bit		FNC 49	FLT	BIN整数至浮点数的转换指令
✓	✓	✓			
5步	9步		指令格式: FLT (S) (D)		

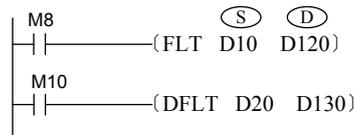
操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)														✓	✓	✓
(D)														✓	✓	✓

将整数 (S) 转换为浮点数，结果存入 (D) 和 (D) +1单元。

常数K、H在各浮点运算指令中自动转换，因此在本FLT指令中不能使用。

这个指令的逆变换指令是INT (将2进浮点数值变换成BIN整数)

指令举例一：



当M8=ON时，将16bit数D10中(16位BIN整数)转换为二进制浮点数后，存放到 (D121, D120)

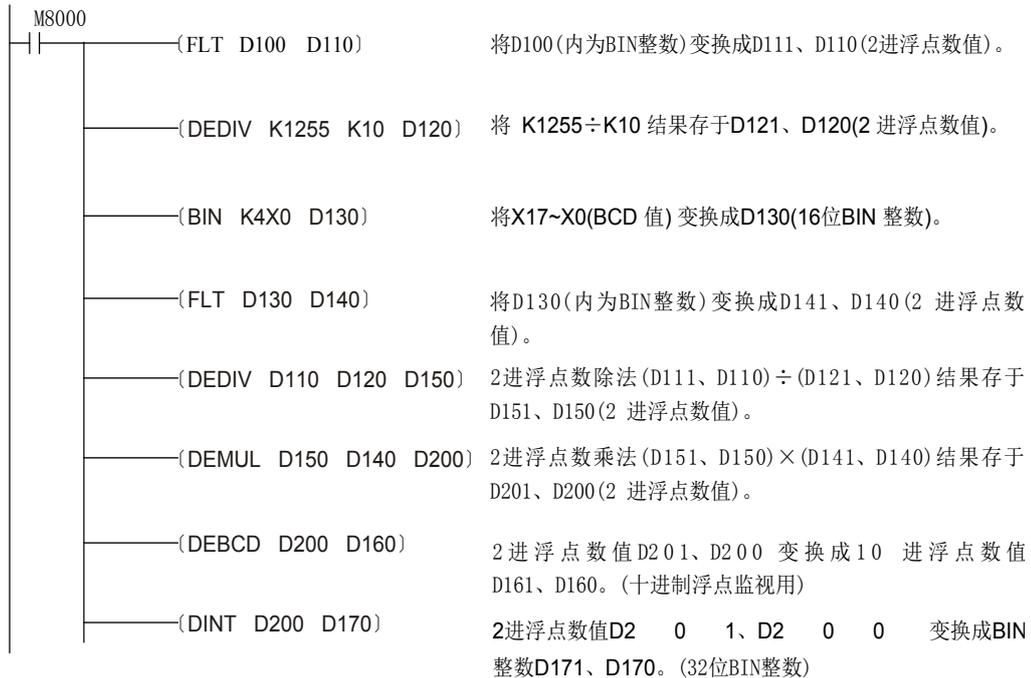
当M10=ON时，将32bit数 (D21, D20) 中(32位BIN整数)转换为二进制浮点数后，存放到 (D131, D130)

FNC49

FLT

指令举例二：

使用指令来完成下列的浮点运算。
D100/K125.5*(X17~X0)=D200



4.3.6 高速处理 (50~59)

16bit	32bit		FNC 50	REF	输入输出端口状态刷新
✓		✓			
5步		5步	指令格式: REF \textcircled{D} \textcircled{n}		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
\textcircled{D}	✓	✓													
\textcircled{n}	常数, n=8~256, 且为8的倍数														

将 \textcircled{D} 地址开始的 \textcircled{n} 个元件状态进行立即更新。

由于PLC访问端口是按字节访问的特性, 故要求:

\textcircled{D} 的地址应为X0、X10、……Y0、Y10、……等最低位为0的编号元件;

\textcircled{n} 的值必须是8的倍数 (n=8~256)。

正常情况下, 输入端口X的状态读取在每次程序开始执行扫描之前进行, 输出端口Y的状态刷新则在每次程序执行扫描完毕 (执行到END) 之后批次进行, 这样IO处理会有一定的延迟。若应用中需要最新的输入信息以及希望立即输出运算结果时, 可以使用立即刷新指令REF。

FNC50

REF

- 可用在FOR~NEXT指令之间、CJ指令之间等。
- 可用于中断子程序中进行输入输出刷新—获取最新的输入信息并及时输出运算结果。
- 实际的输入端口状态变化延迟决定于输入元件的滤波时间, X0~X7有数字滤波功能, 滤波时间在0~60ms范围内可设 (FNC51 (REFF指令)), 其余IO端口为硬件滤波, 滤波时间约10ms。具体参数请参考可编程控制器的用户手册。
- 实际的输出端口状态变化延迟决定于输出元件 (如继电器) 的响应时间。输出刷新中的输出接点将在输出继电器 (晶体管) 应答时间后动作。继电器输出型的应答滞后时间约为10ms (最大20ms), 晶体管输出型高速输出口约10 μ s、普通点输出口约0.5ms。具体参数请参考可编程控制器的用户手册。

指令举例一:

$$\begin{array}{|c} \text{X20} \\ \hline \text{---} \end{array} \text{---} (\text{REF X0 K16})$$

执行上述程序时, 若X20为ON状态, 会立即读取X0~X17的输入点状态, 更新输入信号, 没有产生输入延迟。

指令举例二:

$$\begin{array}{|c} \text{X0} \\ \hline \text{---} \end{array} \text{---} (\text{REF Y0 K16})$$

执行上述程序时, 若X0为ON状态, 会立即将Y0~Y17的状态进行刷新, 输出信号立即更新。

不必到END指令才输出。

16bit	32bit		FNC 51	REFF	输入滤波调整
✓		✓			
3步		3步	指令格式: REFF (n)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(n)	常数, n=0~60, 单位为ms														

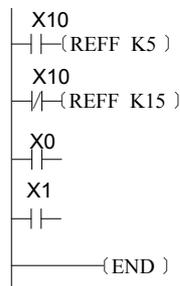
将X0~X7输入端口的滤波时间常数设定为 (n)。

可编程控制器中，X0~X7使用了数字滤波器，默认的滤波时间常数由D8020设定，通过REFF指令可将D8020改变为0~60ms。其余的X端口则只有硬件RC滤波，滤波时间常数约为10ms，不能修改；

当使用了高速计数器，或X输入端中断功能，则相关端口的滤波时间自动为最短时间，无关的端口的滤波时间仍为原设定值。

亦可用MOV指令直接对D8020进行赋值改变滤波时间。

指令举例：



FNC51

REFF

X10为ON时，将X0~X7的输入滤波时间设为5ms，X10为OFF时，将X0~X7的输入滤波时间设为15ms。

16bit	32bit	\square	FNC 52	MTR	矩阵输入
✓		✓			
9步		9步	指令格式: MTR (S) (D1) (D2) (n)		

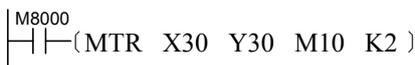
操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)	✓														
(D1)		✓													
(D2)		✓	✓	✓											
(n)	常数, n=2~8														

此指令只能用于晶体管输出型PLC，通过将8个X端口与若干个Y端口组成矩阵输入网络，以扩大输入信号的通道数。其中：

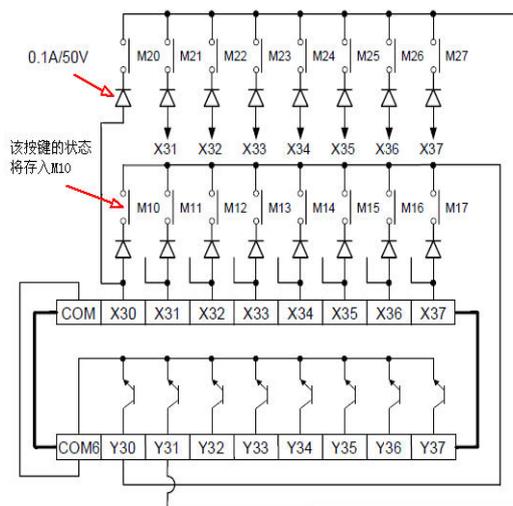
- (S) 为矩阵扫描输入的硬件X端口的起始地址，要求为X0、X10...等最低位为0的编号元件，占用连续8个；
- (D1) 为矩阵扫描输出的硬件Y端口的起始地址，要求为Y0、Y10...等最低位为0的编号元件，占用连续n个 (n=2~8)；
- (D2) 为矩阵扫描读取状态的存放单元的起始地址，要求为Y0、M0、S0等最低位为0的编号元件；
- (n) 为矩阵扫描的列数，即扫描用Y输出的个数。

本指令的条件接点一般都使用常On接点M8000。

指令举例：



适用如下接线：



考虑到X输入滤波应答延迟10ms，Y30、Y31输出按每20ms顺序中断，进行即时输入输出处理；

每次自动读取操作完成后，标志M8029置ON；

若通过8点X输入和8点晶体管Y输出，可获得最大64点的扫描输入，但是此时所有输入的读取需要20ms×8列=160ms时间，不适应高速输入操作，故一般使用X20以后的端口作扫描输入；

FNC52

MTR

16bit	32bit		FNC 53	HSCS	比较置位 (高速计数器)
	✓				
	13步		指令格式: HSCS (S1) (S2) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(S2)												✓				
(D)		✓	✓	✓												

当 (S2) 计数器的当前值等于设定值 (S1) 时, 立即置位 (D)。其中:

(S1) 为设定的比较值, 为32bit;

(S2) 变量必须为高速计数器C235~C255, 因涉及的计数器均为32bit计数器, 故必须采用32bit指令DHSCS;

(D) 为比较结果的存放单元, 也可以是调用计数中断子程序: 当为Y0~Y17范围端口时, 为立即输出; 当为Y20以后的端口时, 会等到本次用户程序扫描完毕才会输出; 当为M、S变量时, 也为立即刷新;

当 (D) 项为I010~I060时, 即为调用高速计数器中断0~5的子程序。当然必须编写好相应的中断子程序、开启相应中断允许标志和全局中断允许标志等, 才能正常响应定时器中断。M8059置ON则禁止了所有的高速计数器中断 (I010-I060)。

一般指令Y输出与DHSCS指令Y输出的差异: 以 (指令举例一) 为例:

- 1) 当C255的现在值由99→100变化时, C255接点立即导通, 但执行到OUT Y10时, Y10仍会受扫描周期影响, 在END后才输出。
- 2) 当C255的现在值由99→100及101→100变化时, DHSCS指令输出Y10是以中断方式立即输出到外部输出端, 与PLC 扫描周期无关。但仍会受输出模块继电器(10ms)或晶体管(10us)的输出延迟。

使用说明:

- 使用HSCS指令时, 应保证所使用的计数器已被启用 (见指令举例一), 否则该计数器的值将不会有变化;
- 计数器是以中断方式响应计数器的输入信号, 及时比较, 若本次比较时满足匹配关系, 比较输出立即置位。例如指令举例一中, 若C255的当前值变为99→100或101→100时, Y10立即置位, 且一直保持该状态, 之后即使C255与K100的比较结果变成不相等, Y10仍然保持On状态, 除非有另外的复位指令操作;
- 指令的比较输出只决定于脉冲输入时的比较结果动作, 即使采用DMOV、DADD等指令改写高速计数器C235~C255的内容, 若没有脉冲输入, 比较输出也不会变化; 单纯的指令驱动能流也不能改变比较结果;
- 指令输出若为Y端口, 必须为Y0~Y17范围, 这样才能保证输出得到立即响应; 多次驱动HSCS指令或与HSCR、HSZ指令同时驱动, 对象输出Y的高2位作为同一序号的软元件。例: 使用Y000时为Y000~Y007, Y010时为Y010~Y017等;
- 当HSCS指令的输出目标为中断I010~I060时, 每个中断号只能使用1次, 不可重复。
- HSCS、HSCR、HSZ与普通指令一样可以多次使用, 但这些指令同时驱动的个数限制在总计6个指令以下。

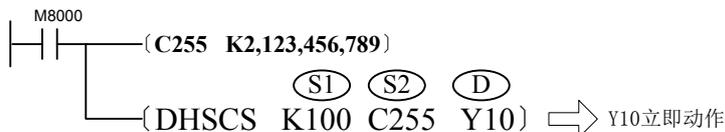
高速计数中断指针与设置:

输入编号	中断禁止指令
I010	M8059
I020	
I030	
I040	
I050	
I060	

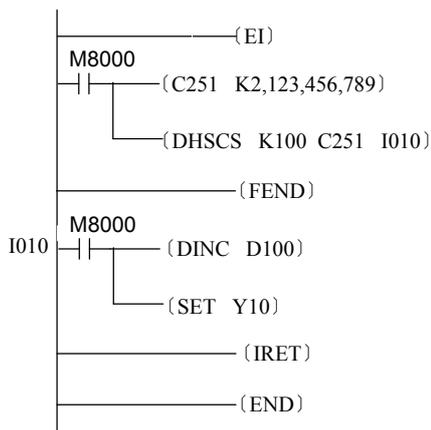
FNC53

HSCS

指令举例一:



指令举例二:



DHSCS指令的D操作数范围也可指定I0□0, □=1~6, 作为计数器计数到达时, 发生中断, 执行该中断服务程序。

如果M8059置ON则禁止了所有的高速计数器中断。

注意此时的D装置用I010和Y、M、S输出点的ON信号区别:

- 1、用Y输出点:若C251的当前值变为99→100或101→100时, Y立即置ON, 且一直保持ON状态, 之后即使C251与K100的比较结果变成不相等, Y仍然保持On 状态,除非有另外的复位指令操作;
- 2、用I010:若C251的当前值变为99→100或101→100时, I010只会产生一次中断, 不会常ON。

FNC53

HSCS

16bit	32bit	\square	FNC 54	HSCR	比较复位 (高速计数器)
	✓				
	13步		指令格式: HSCR (S1) (S2) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(S2)												✓				
(D2)		✓	✓	✓												

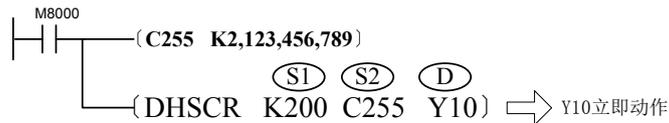
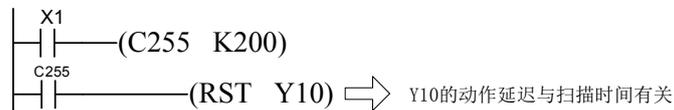
当 (S2) 计数器的当前值等于设定值 (S1) 时，立即复位 (D)。其中：

(S1) 为设定的比较值，为32bit；

(S2) 变量必须为高速计数器C235~C255，因涉及的计数器均为32bit计数器，故必须采用32bit指令DHSCR；

(D) 为比较结果的存放单元：当为Y0~Y17范围端口时，为立即输出；当为Y20以后的端口时，会等到本次用户程序扫描完毕才会输出；当为M、S变量时，也为立即刷新。

指令举例：



FNC54
HSCR

使用说明：

HSCR指令的动作原理和HSCS指令相似，只是HSCR的比较输出动作与HSCS指令刚好相反，即计数器的值达到相等时，指定的输出复位，因此使用中的一些规定可参考HSCS的说明。

16bit	32bit	\square	FNC 55	HSZ	高速计数器 (区间比较)
	✓				
	17步		指令格式: HSZ (S1) (S2) (S) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
(S2)					✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
(S)												✓				
(D)		✓	✓	✓												

根据计数器 (S) 的当前值, 与设定的比较区间 (S1) 和 (S2) 进行比较, 将比较结果立即输出到以 (D) 地址起始的3个单元中。其中:

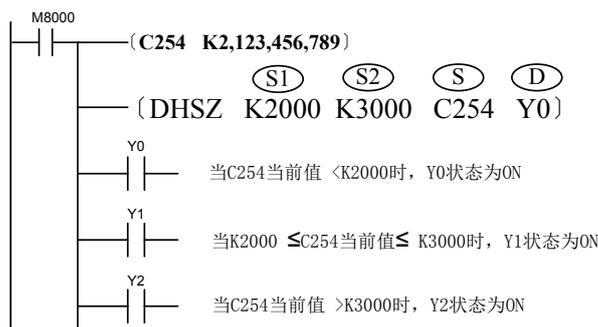
(S1) 为设定的比较区间区间下限, 其值的宽度 (bit位数) 决定于 (S) 计数器的位数, 其值必须不大于 (S2), 即 (S1) ≤ (S2);

(S2) 为设定的比较区间区间上限, 其值的宽度 (bit位数) 决定于 (S) 计数器的位数, 其值必须不小于 (S1), 即 (S1) ≤ (S2);

(S) 变量必须为高速计数器C235~C255, 因涉及的计数器均为32bit计数器, 故必须采用32bit指令DHSZ;

(D) 为比较结果的存放单元, 占用以 (D) 起始的3个连续地址的单元: 当为Y0~Y17范围端口时, 为立即输出; 当为Y20以后的端口时, 会等到本次用户程序扫描完毕才会输出; 当为M、S变量时, 也为立即刷新。

指令举例:



使用说明:

本指令的动作原理和HSCS、HSCR等指令相似, 差别是采用了两个比较值, 比较输出使用了3个连续的地址单元, 因此使用中的一些规定可参考HSCR的使用说明:

HSZ指令也是以中断方式进行工作的, 只有当计数器对应的输入端有计数脉冲时, 比较才会进行, 对应的输出才会被刷新:

表格高速比较模式

将指令 (DHSZ (S1) (S2) (S) (D)) 中的 (D) 指定为特殊辅助继电器M8130, 即表明为高速表格比较模式, 指令中的各变量将按表格方式进行解析, 说明如下:

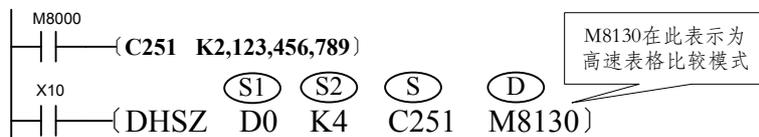
(S1) 只对应数据寄存器D变量, 用于表示比较表格的起始地址。

(S2) 只可用常数变量K、H, 用于表示表格的行数, 被限制为1≤(K, H)≤128, 可用Z, 指令启动后不再受Z的影响;

(S) 可以指定为高速计数器C235~C255;

(D) 为M8130, 指明为高速表格比较模式。

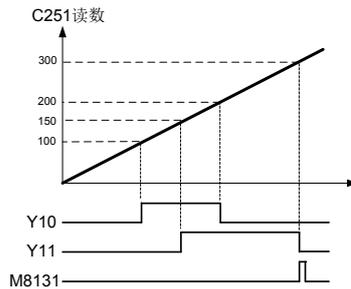
例如:



等效的比较表格为:

① 表格起始变量为D0	比较值 (32bit)		Y输出编号	ON/OFF状态	表格计数器D8130
	低字	高字			
② 表格行数为K4	D0	D1	D2	D3	0
	D4	D5	D6	D7	1
	D8	D9	D10	D11	2
	D12	D13	D14	D15	3
参数举例	K100	K0	H10	K1	执行时计数器 0→1→2→3→0依次循环
	K150	K0	H11	K1	
	K200	K0	H10	K0	
	K300	K0	H11	K0	
说明	在接收到第100脉冲时有动作 (表格中各行比较值应依次增大)		Y10端口动作 (若为H11则表示Y11)	动作是置为ON (若为K0则表示OFF动作)	

执行过程说明:



FNC55
HSZ

当①所指定的高速计数器C251的当前值等于(D1、D0)设定值的时候D2所指定的输出Y被复制成OFF (D3=K0) 或是ON (D3=K1) 并保持住。而输出Y的动作完全以中断方式来处理。

当C251的当前值与表格的第一组设定值相等时, D8130=K1、与第一组设定值相等时, D8130=K2, 如此的往下顺序执行比较操作, 直到最后一组比较动作完成时, M8131=ON一个扫描周期, 之后D8130清除为0, 再返回到第一组进行比较。

当指令的条件接点X10变成OFF时, 指令执行被中断, 表格计数器D8130被清0, 但指令相关的输出状态全部被保持。

本指令在被第一次扫描执行, 直到END指令后, 比较表格的各项设置即确定下来, 因此表格中的各参数设置需在本指令之前设置完成。

表格比较指令在用户程序中只能使用一次。此外, 与其他用途使用的HSCS/HSCR/HSZ指令结合, 可以同时驱动指令被限制在6点以下。

由指令HSZ和PLSY指令实现频率控制模式

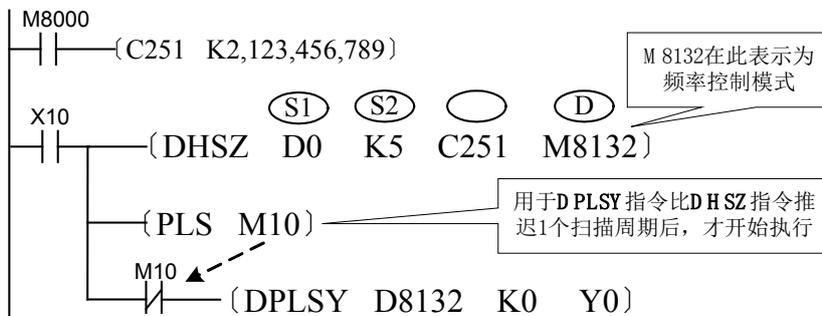
将指令(DHSZ①②③④)中的④指定为频率控制模式说明用特殊辅助继电器M8132, 通过与DPLSY指令的组合, 可实现一个高速计数器的当前值控制DPLSY输出频率的功能。

说明如下:

- ① 只对应数据寄存器D变量, 用于表示比较表格的起始地址。
- ② 只可用常数变量K、H, 用于表示表格的行数, 被限制为1≤(K, H)≤128, 可用带Z, 指令启动后不再受Z的影响;
- ③ 可以指定为高速计数器C235~C255;
- ④ 为M8132, 指明为根据高速计数值HSZ指令来控制PLSY输出频率模式。

本指令在用户程序中只能使用一次; 表格中的各个寄存器值需事先设定好;

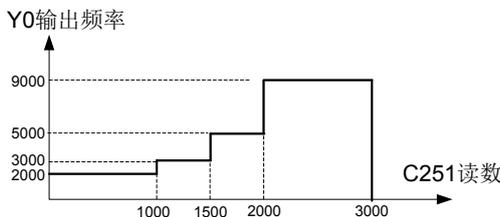
例如:



表示根据C251的当前频率，控制Y0输出频率的工作模式，等效的比较与输出频率表格为：

S1 表格起始 变量为D0	比较值 (32bit)		输出频率 (32bit)		表格计数器 D8131
	低字	高字	低字	高字	
S2 表格 行数为K5	D0	D1	D2	D3	0
	D4	D5	D6	D7	1
	D8	D9	D10	D11	2
	D12	D13	D14	D15	3
	D16	D17	D18	D19	4
参数举例	K1000	K0	K2000	K0	执行时计数器 0→1→2→3→4→0 依次循环
	K1500	K0	K3000	K0	
	K2000	K0	K5000	K0	
	K3000	K0	K9000	K0	
	K0	K0	K0	K0	
说明	接收脉冲后进行比较，（如第1000个脉冲）匹配时改变输出频率。 (表格中各行比较值应依次增大，最后一行可设为0。)		Y0端口的输出频率改变为对应表格 栏的设定值		

执行过程说明：



预先将所定的数据写入构成表格的数据寄存器，并有指令启动 S 指定的高速计数器C251，运行中请勿改变表格内容的设置；
当C251的当前值小于(D1, D0)时，PLSY指令的输出频率为 (D3, D2) 的值；

当C251的当前值等于(D1, D0)时，PLSY指令的输出频率变为 (D7, D6) 的值；当C251的当前值等于(D5, D4)时，PLSY指令的输出频率变为 (D11, D10) 的值；依此类推；

最后一行的操作完毕，完成标志M8133动作。并回到第一行重复运作；

若希望在最后一行停止动作时，将最后的表格的频率置为K0；

驱动输入X010为OFF时，脉冲输出变成OFF，表格计数D8131也复位；

该项指令在初次指令执行后的END指令完成表格制作，其后开始有效。因此，为了使PLSY指令，从驱动输入X10为ON后的第2个扫描周期开始动作，采用 (PLS M10) 的触点。

注意事项：

采用频率控制模式时，编程中使用其他的PLSY指令以及PLSR指令，无法同时得到2路脉冲输出。

16bit	32bit		FNC 56	SPD	脉冲密度, 转速, 线速度
✓					
7步			指令格式: SPD (S1) (S2) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)	✓															
(S2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(D)											✓	✓	✓	✓	✓	✓

将 (S1) 指定端口在 (S2) 时间内检测到的脉冲数, 保存到 (D) 地址单元。

其中 (D) 占用3个连续的单元, (D)+1为实时脉冲计数值; (D)+2为完成本次采样周期的剩余时间。

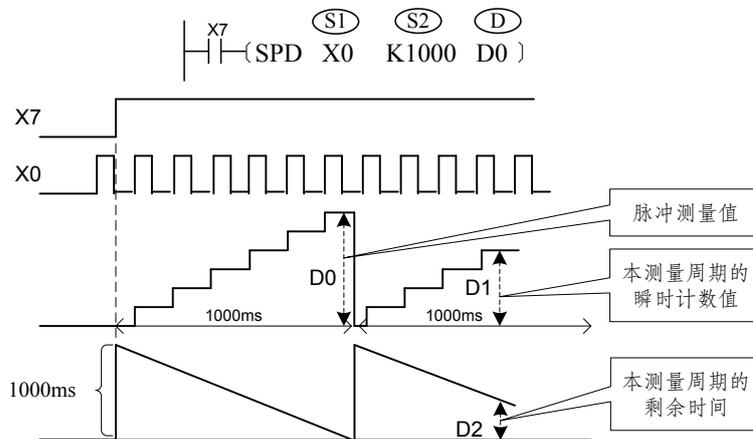
(S1) 脉冲信号输入端口, 只能为X0~X5;

(S2) 为设定的脉冲检测时间长度 (ms);

(D) 为设定时间长度 (S2) 内接收的脉冲个数;

被用于SPD指令的X0~X5端口, 可同时用于高速计数器或者中断输入中。

指令举例一:



FNC56

SPD

在图例中, X7置ON时, D1对X0的OFF→ON动作计数, 1000ms后将其结果存入D0, 随之D1复位, 再次对X0的动作计数。D2用于测定剩余时间。

因此根据D0、(S2) 的设定值就可以求得脉冲的频率; 若脉冲信号取自旋转编码器, 可求得转速等; SPD测速最高输入10K。

(S1) 脉冲信号输入端口, 只能为X0~X5;

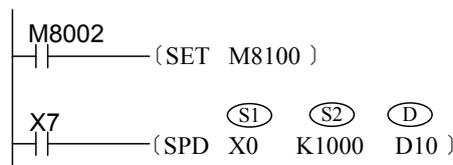
(S2) 为设定的脉冲检测时间长度(ms)1~32767;

(D)+0: 为在 时间内的脉冲个数, 为16位的数据;

(D)+1为实时脉冲计数值。

(D)+2为完成本次采样周期的剩余时间。

指令举例二:



D10为设定时间长度1000ms内接收的脉冲个数; D11为实时脉冲计数值; D12为完成本次采样周期的剩余时间。

注意: SPD指令的增强功能 (XP型不具备)

在新版本的H1U/H2U系列PLC中, SPD命令功能有增强, 通过置位[M8100~M8105]标志, 可改变计数功能, M8100~M8105标志分别对应X000~X005高速端。(M8100~M8105标志可分别置位, 即X0~X5端口的SPD指令功能可分别处理。)

如果功能使能标志 [M8100~M8105] 为OFF，SPD指令如前面介绍的；

如果功能使能标志M8100~M8105为ON，指令的新功能定义如下：

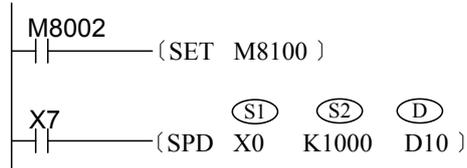
Ⓢ1：脉冲信号输入端口，只能为X0~X5；

Ⓢ2：为设定的脉冲检测时间长度(ms)1~32767；

Ⓤ+0：为在S2时间内的脉冲个数，为16位的数据；

Ⓤ+1，Ⓤ+2：为每分钟内的脉冲个数，为32位的数据；

指令举例三：



D10为设定时间长度1000ms内接收的脉冲个数；D11，D12为当前时刻1分钟所对应的脉冲个数放大10倍。

16bit	32bit		FNC 57	PLSY	脉冲输出
✓	✓				
7步	13步	指令格式: PLSY (S1) (S2) (D)			

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(S2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(D)		✓														

由于继电器不适合高频率动作，只有晶体管输出型PLC才适合使用该指令。指令功能是由：

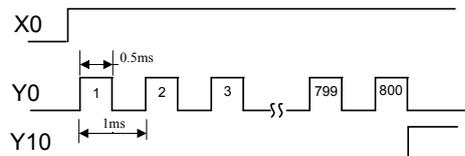
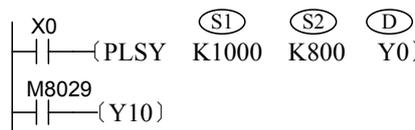
(D) 指定的端口，以 (S1) 的频率，输出 (S2) 个脉冲，脉冲发送完毕，M8029标志被置位。其中：

(D) 为脉冲输出端口，只能指定Y0/Y1/Y2。

(S1) 为设定的输出脉冲频率，对于16bit指令 (PLSY)，设定范围为0~32,767（注意，设定为0时无脉冲输出）；对于32bit指令 (DPLSY)，设定范围为1~100,000（即1Hz~100kHz）；在指令执行中可以改变 (S1) 的值；

(S2) 为设定的脉冲输出个数，对于16bit指令 (PLSY)，设定范围为1~32,767；对于32bit指令 (DPLSY)，设定范围为1~2,147,483,647；当 (S2) 等于零时为发送不间断的无限个脉冲。

指令举例：



FNC57
PLSY

使用PLSY(16bit指令)时，(S1)和(S2)都只能是16bit宽度；

使用DPLSY(32bit指令)时，(S1)和(S2)若为D、C、T变量，则按32bit宽度处理；

当执行到PLSY指令后，Y即开始输出脉冲；运行中若改变 (S2) 元件（为D、C、T变量）的参数值，对当前输出的脉冲数没有影响，将从下一次启动该指令时生效；（新版本的H2u系列PLC中，可以在运行中改变输出脉冲数，具体请参考附录8.6说明）

在PLSY输出脉冲过程中，若指令能流X0变为OFF，则输出脉冲被停止；若X0变为ON，PLSY指令将以当前的参数重新开始脉冲输出。

使用说明：

- PLSY所用的输出Y端口避免与PWM或PLSR指令所用的Y端口重复；
- 使用PLSY指令时，使用了如下特殊寄存器：

寄存器		定义	备注
D8140	低字	PLSY或PLSR指令中设定的输出至Y0口的脉冲总数	可用指令： DMOV K0 D81xx 进行清除操作
D8141	高字		
D8142	低字	PLSY或PLSR指令中设定的输出至Y1口的脉冲总数	
D8143	高字		
D8150	低字	PLSY或PLSR指令中设定的输出至Y2口的脉冲总数 (3624MT/2416MT没有此端口)	
D8151	高字		
D8136	低字	已向Y0及Y1输出的脉冲个数的累计值	
D8137	高字		

注意：

在新版本的H1u/H2u系列PLC中，PLSY指令的功能有增强，可在PLSY指令运行过程中修改脉冲个数、或立即启动下条脉冲输出指令、或实现脉冲输出完成中断等增强功能。

- 3) 通过使用特殊位M8135~M8137(分别对应Y0~Y2)为ON, 可以在运行中更改(S2)输出脉冲个数(可大可小);
- 4) 通过使用特殊位M8085~M8087(分别对应Y0~Y2)为ON, 可以实现马上启动下条脉冲输出指令, 不需要上条能流无效的处理;
- 5) 通过使用特殊位M8090~M8092(分别对应Y0~Y2)为ON, 可以实现脉冲输出完成中断; 具体对应如下:

端口号	使用特殊位	对应的用户中断
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504

具体可参考附录8.6中的说明。

16bit	32bit		FNC 58	PWM	脉宽调制
✓					
7步			指令格式: PWM (S1) (S2) (D)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(S2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(D)		✓													

由于继电器不适合高频率动作, 只有晶体管输出型PLC才适合使用该指令。指令功能是以

(S1) 指定的脉冲宽度, (S2) 指定的脉冲周期, 由 (D) 指定的端口持续输出脉冲。其中:

(S1) 为设定的输出脉冲宽度, 必须有 (S1) ≤ (S2), 设定范围为0~32,767ms;

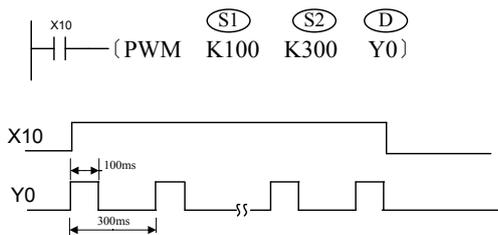
(S2) 为设定的脉冲输出周期, 必须有 (S1) ≤ (S2), 设定范围为1~32,767ms;

(D) 为脉冲输出端口, 只能指定Y0/Y1/Y2。不要与PLSY, PLSR指令的输出端口重复。

本指令是以中断方式执行, 当指令能流为OFF时, 输出停止。

(S1)、(S2) 可在PWM指令执行时更改。

指令举例:



16bit	32bit		FNC 59	PLSR	带加减速脉冲输出
✓	✓				
9步	17步		指令格式: PLSR (S1) (S2) (S3) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(S2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(S3)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(D)		✓														

由于继电器不适合高频率动作，只有晶体管输出型PLC才适合使用该指令。该功能是指带加减速功能的固定尺寸传送用脉冲输出指令。其中：

(S1) 为设定的输出脉冲的最高频率，设定范围为10~100,000Hz；

(S2) 为设定的输出脉冲数，16bit指令，设定范围为110~32,767；32bit指令，设定范围为110~2,147,483,647；设定的脉冲数小于110时，不能正常输出脉冲；

(S3) 为设定的加减速时间，范围50~5000(ms)，减速时间与加速时间相同，ms单位，设定时请注意：(H1U/H2U系列PLC中减速时间可单独设定，请参考后面的介绍)

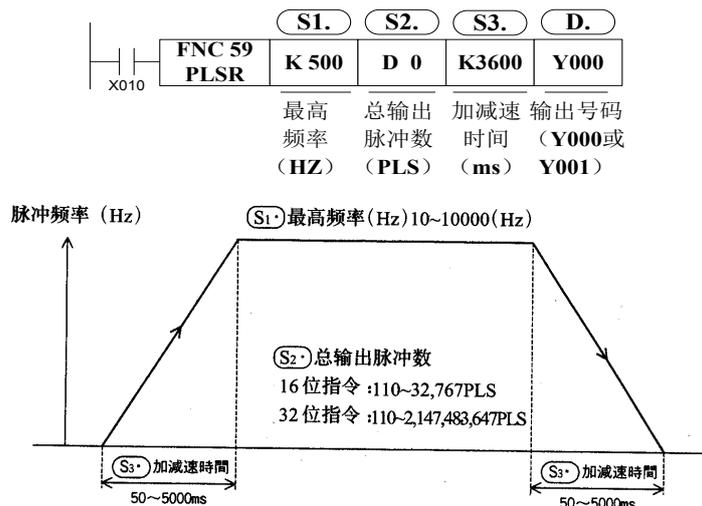
(D) 为脉冲输出端口，只能指定Y0/Y1/Y2。不要与PLSY指令的输出端口重复。

FNC58
PWM

使用说明：

- 本指令是以中断方式执行，不受扫描周期的影响；
- 当指令能流为OFF时，将减速停止；当能流由OFF→ON时，脉冲输出处理重新开始；
- 在脉冲输出过程中，改变操作数，对本次输出没有影响，修改的内容在指令下次执行的时候生效。指令执行完毕，M8029标志置为ON；
- 与PWM指令的输出端口号不能重复；
- 再次启动PLSR指令时，需在上次脉冲输出操作结束（Y0结束时M8147=0；Y1结束时M8148=0；Y2结束时M8149=0）后，再延迟1个扫描周期，方可再启动该指令（在新版本的H1U/H2U系列PLC中通过设置可以不受此限制，具体请参考附录8.6中的说明）；

指令举例：



各输出端口对应的特殊寄存器如下:

寄存器		定义	备注
D8140	低字	PLSY或PLSR指令中设定的输出至Y0口的脉冲总数	可用指令: DMOV K0 D81xx 进行清除操作
D8141	高字		
D8142	低字	PLSY或PLSR指令中设定的输出至Y1口的脉冲总数	
D8143	高字		
D8150	低字	PLSY或PLSR指令中设定的输出至Y2口的脉冲总数 (3624MT/2416MT没有此端口)	
D8151	高字		
D8136	低字	已向Y0及Y1输出的脉冲个数的累计值	
D8137	高字		

- 本指令的输出频率范围为10~100, 000Hz。最高速度或加减速的高速速度转换超出该范围时, 将自动转换(上升或下降)至范围内的数值后执行。但是, 实际能够输出的输出频率最低值取决于以下公式:

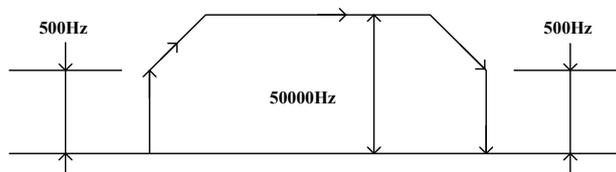
$$\sqrt{\text{最高频率 (S1) Hz} \div (2 \times (\text{加减速时间 (S3) Ms} \div 1000))} = \text{输出脉冲频率的最低频率数}$$

加速初期和减速末期的频率不得低于上述公式的计算结果。

[例] 最高速度: 50000Hz, 加减速时间: 100ms

$$\sqrt{50000 \div (2 \times (100 \div 1000))} = 500\text{Hz}$$

最高频率S1.指定为50000Hz时。加速初期和减速末期的实际输出频率为500Hz。



注意:

在新版本的H1U/H2U系列PLC中, PLSR, DRVI, DRVA等指令的功能有增强:

- 通过使用特殊位M8135~M8137(分别对应Y0~Y2)为ON, 可以在运行中更改输出脉冲个数(可大可小); 同时加速时间由(S3)决定, 减速时间分别由如下寄存器定义:

端口号	使用特殊位	修改减速时间用寄存器
Y000	M8135	D8165
Y001	M8136	D8166
Y002	M8137	D8167

- 通过使用特殊位M8085~M8087(分别对应Y0~Y2)为ON, 可以实现如下功能:

在驱动特殊位为ON, 可以马上启动下条脉冲输出指令, 不需要上条能流无效的处理;

- 通过使用特殊位M8090~M8092(分别对应Y0~Y2)为ON, 可以实现如下功能:

可以实现脉冲输出完成中断; 具体对应如下:

端口号	使用特殊位	对应的用户中断
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504

具体可参考附录8.6中的说明。

FNC59

PLSR

4.3.7 方便指令 (60~69)

16bit	32bit		FNC 61	SER	数据查找
✓	✓	✓			
9步	17步		指令格式: SER (S1) (S2) (D) (n)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)							✓	✓	✓	✓	✓	✓	✓	✓	✓	
(S2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(D)								✓	✓	✓	✓	✓	✓	✓	✓	
(n)					✓	✓							✓			

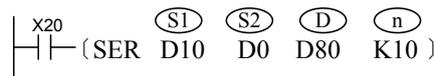
取值范围: 16bit指令: n=1~256; 32bit指令: n=1~128

该指令用于从一组数据中, 查找相同数据的单元、同时对最大值、最小值的检索。其中:

- (S1) 为数据组的起始地址;
- (S2) 为待检索的数据;
- (D) 为检索结果存放区的起始地址;
- (n) 为被检索数据区的长度。

当使用32bit指令时, (S1) (S2) (D) 均指向32bit变量, (n) 也按32bit变量宽度进行计算。

指令举例:



(S1)	被检索数据例	(S2)	元件序号	参数状况
D 10	(D 10)=K100	比较数据 (D0) =K100	0	相等
D 11	(D 11)=K123		1	
D 12	(D 12)=K100		2	相等
D 13	(D 13)=K98		3	
D 14	(D 14)=K111		4	
D 15	(D 15)=K66		5	最小
D 16	(D 16)=K100		6	相等
D 17	(D 17)=K100		7	相等
D 18	(D 18)=K210		8	最大
D 19	(D 19)=K88		9	

检索结果		
(D)	参数	定义
D80	4	相等参数的个数
D81	0	第一个相等参数的序号
D82	7	最后一个相等参数的序号
D83	5	最小参数的序号
D84	8	最大参数的序号

使用说明:

当指令能流X20为ON时, 方才进行比较;

比较的方法为有符号数的代数比较方法进行, 例如-8<2;

当最小值、最大值有多个时, 分别显示序号最大的元件;

存储检索结果的单元占用 (D) 开始的5个连续单元。若不存在相等数据时, 上例中的D80~D82均为0。

FNC59
PLSR

16bit	32bit	\square	FNC 62	ABSD	凸轮控制绝对方式
✓	✓				
9步	17步		指令格式: ABSD (S1) (S2) (D) (n)		

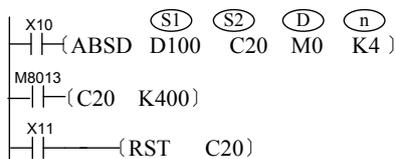
操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)							✓	✓	✓	✓	✓	✓	✓			
(S2)												✓				
(D)		✓	✓	✓												
(n)	常数, n=1~64;															
(S1)	操作数为KnX、KnY、KnM、KnS时, 若为16bit指令, 必须指定K4; 若为32bit指令, 必须指定K8且X、Y、M、S的元件编号必须是8的倍数;															
(S2)	操作数在16bit指令时只能指定C0~C199; 32bit指令时则只能指定C200~C254;															

该指令完成的操作是多区段比较, 用于实现凸轮控制, 比较用的表格、计数器等均按绝对方式设置。该指令是主程序中扫描执行, 比较结果受扫描时间的滞后影响。其中:

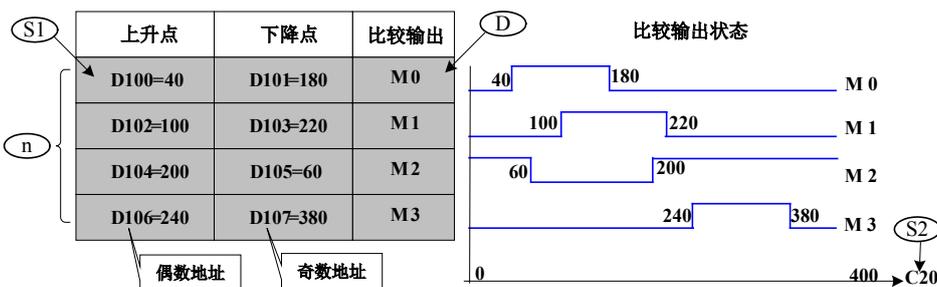
- (S1) 为比较表格的起始元件地址;
- (S2) 为计数器元件序号, 使用32bit指令时, 可为32bit计数器;
- (D) 为比较结果存放区的起始地址, 占用 (n) 个连续地址的bit变量单元;
- (n) 为多段比较数据的组数。

当使用32bit指令时, (S1) (S2) (D) 均指向32bit变量, (n) 也按32bit变量宽度进行计算。

指令举例:



若已给相关变量按如下赋值, 当X10=ON时, 执行结果如下图:



使用说明:

- ABSD指令执行前, 应给相关表格的各变量用MOV指令赋值;
- 即使DABSD指令中采用了高速指令, 比较输出 (D) 也受用户程序扫描时间的滞后影响, 对于需要及时响应的应用, 可采用HSZ高速比较指令;

16bit	32bit		FNC 63	INCD	凸轮控制增量方式
✓					
9步			指令格式: INCD (S1) (S2) (D) (n)		

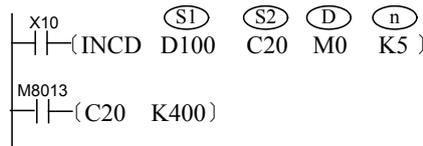
操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)							✓	✓	✓	✓	✓	✓	✓			
(S2)												✓				
(D)		✓	✓	✓												
(n)	常数, n=1~64;															
(S1) 操作数为KnX、KnY、KnM、KnS时, 若为16bit指令, 必须指定K4;																
(S2) 操作数在16bit指令时指定C0~C233。																

该指令完成的操作是多区段比较, 用于实现凸轮控制, 比较用的表格、计数器等均按增量方式设置。该指令是主程序中扫描执行, 比较结果受扫描时间的滞后影响。其中:

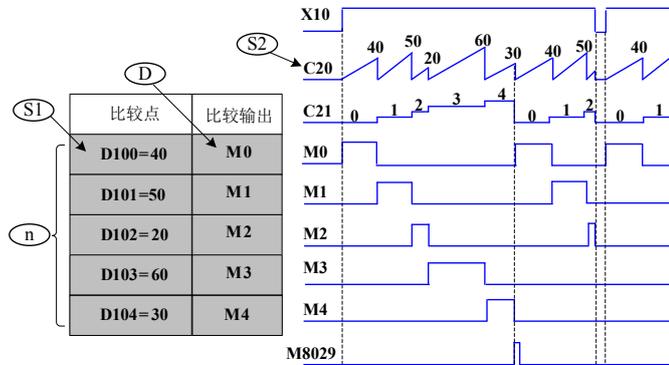
- (S1) 为比较表格的起始元件地址;
- (S2) 为计数器元件序号, 其相邻的 (S2) + 1 单元则被用于计算比较匹配后计数器复位的次数。
- (D) 为比较结果存放区的起始地址, 占用 (n) 个连续地址的bit变量单元;
- (n) 为多段比较数据的组数。

n 的组数比较完成时, 指令执行完毕标志M8029会置On。

指令举例:



若已给相关变量按如下赋值, 当X10=ON时, 执行结果如下图:



使用说明:

- INCD指令执行前, 应给相关表格的各变量用MOV指令赋值;
- 比较输出也受用户程序扫描时间的滞后影响, 对于需要及时响应的应用, 可采用HSZ高速比较指令;

FNC60
IST

16bit	32bit	\square	FNC 64	TTMR	演示定时器
✓					
5步			指令格式: TTMR (D) (n)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(D)													✓			
(n)					✓	✓										

该指令的功能是将指定输入端口的按键保持时间乘以 (n) 倍数后存入变量 (D)，一般用于参数设定。其中：

(D) 为按键保持时间以秒为单位乘以n倍数后的乘积，按键释放后 (D) 内容没有变化；而 (D) + 1单元则用于保存按键的按压时间，按键释放后 (D) + 1的内容被复位为0，(D) + 1的时间单位为100ms；

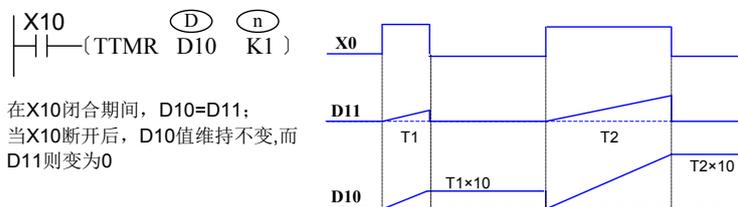
(n) 为倍数设定，注意实际倍数为10ⁿ计算方式，(n=0~2)

n=K0时，实际倍数为×1；

n=K1时，实际倍数为×10；

n=K2时，实际倍数为×100。

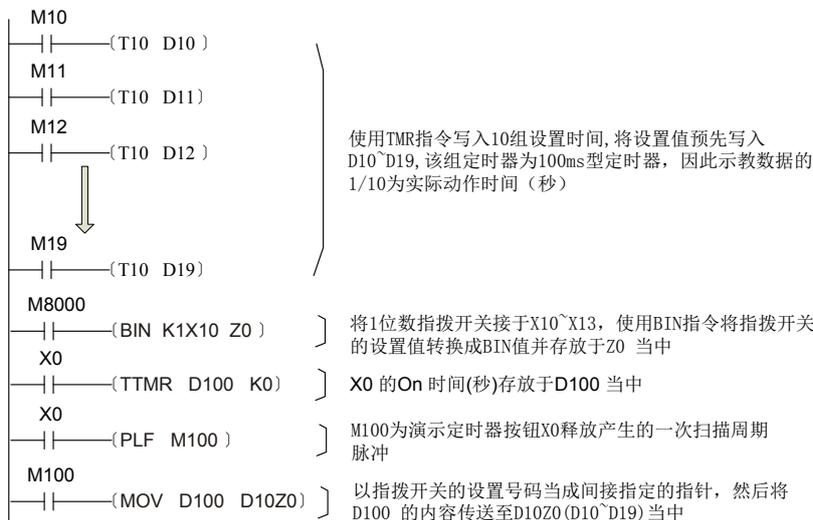
指令举例一：



假如X10的按键保持时间为T秒，D10、D11与n三者之间的关系如下：

n	D10	D11(单位: 100毫秒)
K0(单位: 秒)	1×T	D11=D10×10
K1(单位: 100毫秒)	10×T	D11=D10
K2(单位: 10毫秒)	100×T	D11=D10/10

指令举例二：



FNC64

TTMR

16bit	32bit		FNC 65	STMR	特殊定时器
✓					
7步			指令格式: STMR (S) (m) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)											✓					
(m)	常数, m=1~32767															
(D)		✓	✓	✓												

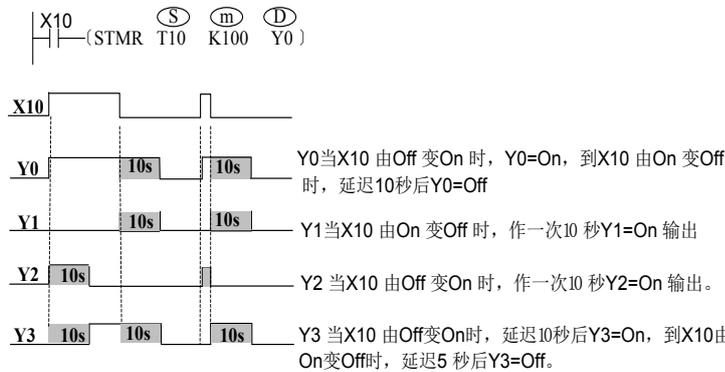
该指令的功能是根据指令能流, 产生4种延时动作的专用指令。其中:

- (S) 用于产生延迟动作的计时器序号, 可用T0~T199;
- (m) 为延时设定值, 单位为100ms, 设定值范围为K1~K32767;
- (D) 为延时动作输出元件的起始编号, 共占用4个连续单元。

使用注意事项:

在本指令中使用的计时器不得再用于其他指令中重复使用, 输出也不可重复使用。

指令举例一:

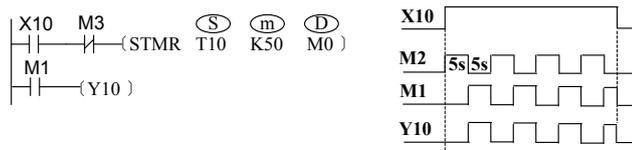


FNC65

STMR

指令举例二:

若在指令能流中引入 (D) 的元件, 可方便地实现振荡器输出 (此功能也可以用ALT指令来实现), 如下图:



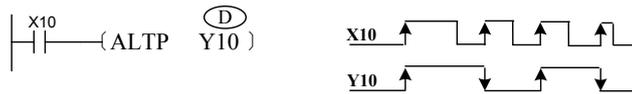
16bit	32bit	\square	FNC 66	ALT	ON/OFF交替
✓		✓			
3步		3步	指令格式: ALT \textcircled{D}		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
\textcircled{D}		✓	✓	✓												

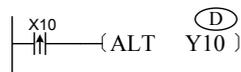
该指令的功能是当能流有效时，将 \textcircled{D} 元件的状态反转。其中 \textcircled{D} 为位变量元件。

一般使用脉冲执行型ALTP指令。

指令举例一：

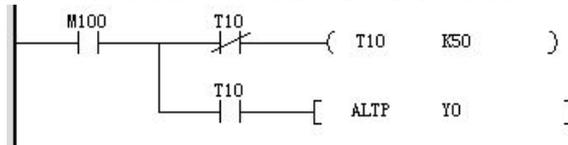


如下指令产生的动作与之相同：



指令举例二：

若在指令能流中引入定时器，可方便地实现振荡器输出（此功能也可以用特殊定时器STMR指令来实现），如下图：



FNC66

ALT

16bit	32bit		FNC 67	RAMP	斜坡指令
✓					
9步			指令格式: RAMP (S1) (S2) (D) (n)		

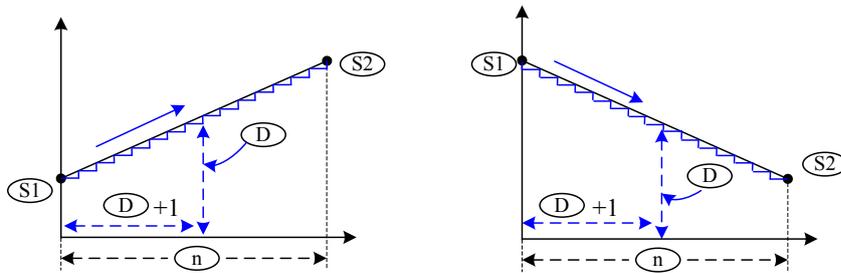
操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)														✓		
(S2)														✓		
(D)														✓		
(n)	常数, 1~32767															

该指令的功能是在给定的两个数据中间, 在指定的时间区间, 进行线性插值, 按扫描执行的时间依次输出过程值, 直到区间末端的终点值为止。其中:

- (S1) 斜坡信号的起始值单元;
- (S2) 斜坡信号的终点值单元;
- (D) 为线性插值信号的过程值存放单元, 而插值次数的计时器存放在 (D) + 1单元;

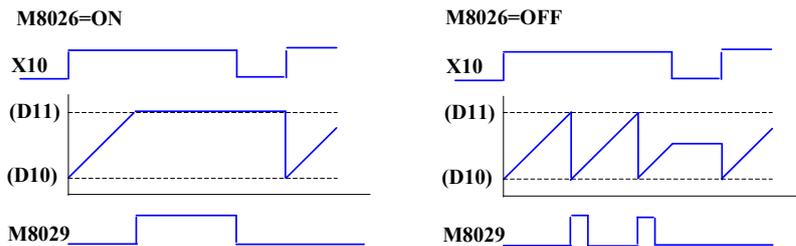
(n) 完成插补过程的程序扫描执行次数。由于插值输出是在正常主循环中进行的, 为了保证插值输出的线性, 需要将程序执行设置为固定扫描方式 (见M8039、D8039说明)。

插值计算按整型数计算, 丢弃了计算小数。指令功能如下图:



FNC67
RAMP

RAMP指令执行有2种模式, 由M8026标志进行选择; 每次插值运算完毕, M8029置ON。执行特点如下例:



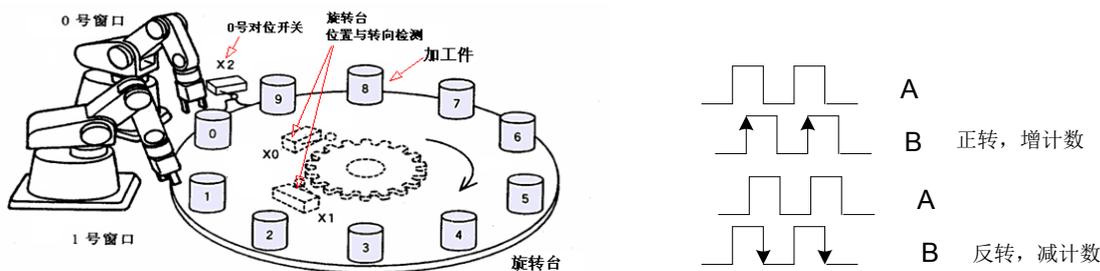
方便指令			FNC 68	ROTC	旋转台控制指令
16bit	32bit	P			
✓					
9步	指令格式: ROTC (S) (m1) (m2) (D)				

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)													✓			
(D)		✓	✓	✓												
(m1)	2~32767, m1 ≥ m2															
(m2)	0~32767, m1 ≥ m2															

该指令是用于旋转工作台上工件取放控制的简捷指令，旋转工作台的位置检测信号需按指定方式配置才能正常工作。其中：

- (S) 计数变量的起始单元；
- (m1) 旋转工作台上的工位数，必须 (m1) ≥ (m2)；
- (m2) 旋转工作台上的低速工位数，必须 (m1) ≥ (m2)；
- (D) 为旋转台位置检测信号存放的起始单元，占用随后的8个位变量单元。

信号配置方式如下图，图中X0、X1分别接AB正交编码器的A相和B相输出信号，也可采用机械开关得到正交相位的信号；X2接用于0号工位的检测输入（当旋转到0号工位时为ON状态），由此3个信号即可检测旋转工作台的当前转动速度和转向和工位。



应用举例：

```

X10 (ROTC (S) D200 (m1) K10 (m2) K2 (D) M0)
    
```

该代码实际使用的变量空间说明如下：

变量	功能定义	操作说明
D200	作为计数寄存器使用	由用户程序事先设定好该3个单元
D201	调用窗口号码设定	
D202	调用工件号码设定	
M0	A相信号	用户程序每次扫描本语句前执行： ┌── X0 ── (M0) ┌── X1 ── (M1) ┌── X2 ── (M2)
M1	B相信号	
M2	0点检测信号	

变量	功能定义	操作说明
M3	高速正转	当X10为ON时，可以自动得到M3~M7的结果； 当X10为OFF时，M3~M7均为OFF；
M4	低速正转	
M5	停止	
M6	低速反转	
M7	高速反转	

在后续的用户程序中，将M3~M7从Y输出口中输出，控制外部执行元件即可。

能流有效的情况下，当0点信号M2=ON，D200清0，需要先执行以上清0操作才开始运行。

程序中只能使用1次ROTC指令。

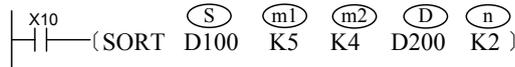
16bit	32bit	\square	FNC 69	SORT	数据排序
✓					
11步			指令格式: SORT (S) (m1) (m2) (D) (n)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)													✓		
(m1)	常数, 1~32														
(m2)	常数, 1~6														
(D)													✓		
(n)					✓	✓							✓		

该指令是将m1行×m2列的数组（由(S) (m1) (m2)描述），以第(n)列参数排序后，存放于由(D)单元起始的变量区域。其中：

- (S) 为第1行（或称第1条记录）的首个变量的起始单元；
 - (m1) 数组的行数，或称记录数；
 - (m2) 数组的列数，或称每条记录的栏目数；
 - (D) 为排序后存放的起始单元，占用随后的变量单元数目与排序前的数组变量数目相同；
 - (n) 为以排序为依据的数组列号。
- 其中(n)的值在1~(m2)范围。

指令举例：



FNC69

SORT

当X10=ON时，开始排序运算，指令执行完毕，M8029被置ON；

若需要再次排序，需将X10=OFF一次。

上述指令的等效表格及其数据举例：

(m1)	行号	(m2)				
		列号	1	2	3	4
			学号	语文	数学	物理
		1	D100 1	D105 85	D110 78	D115 83
		2	D101 2	D106 82	D111 91	D116 81
		3	D102 3	D107 77	D112 89	D117 88
4	D103 4	D108 90	D113 81	D118 75		
5	D104 5	D109 87	D114 95	D119 77		

按指令要求(n)=K2排序后的表格数据结果：

(D)	行号	(n)=K2				
		列号	1	2	3	4
			学号	语文	数学	物理
		1	D200 3	D205 77	D210 89	D215 88
		2	D201 2	D206 82	D211 91	D216 81
		3	D202 1	D207 85	D212 78	D217 83
4	D203 5	D208 87	D213 95	D218 77		
5	D204 4	D209 90	D214 81	D219 75		

若指令中(n)=K4，则排序后的表格数据结果如下：

(D)	行号	(n)=K4				
		列号	1	2	3	4
			学号	语文	数学	物理
		1	D200 4	D205 90	D210 81	D215 75
		2	D201 5	D206 87	D211 95	D216 77
		3	D202 2	D207 82	D212 91	D217 81
4	D203 1	D208 85	D213 78	D218 83		
5	D204 3	D209 77	D214 89	D219 88		

4.3.8 外围设备 I/O (70~79)

16bit	32bit	\square	FNC 70	TKY	0~9数字键输入
✓	✓				
7步	13步		指令格式: TKY (S) (D1) (D2)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)	✓	✓	✓	✓											
(D1)								✓	✓	✓	✓	✓	✓	✓	✓
(D2)		✓	✓	✓											

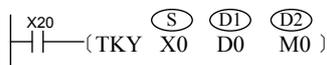
该指令是指定10个连续的位变量单元（如X输入端口），依次代表10进制的0~9按键，当有按键动作（状态为ON）时，以按键的动作顺序，可输入十进制0~9999的4位数值；若采用32bit指令，则可输入十进制0~99, 999, 999的8位数值。其中：

(S) 为按键的起始输入端口，使用随后的共10个位单元（如X端口）；

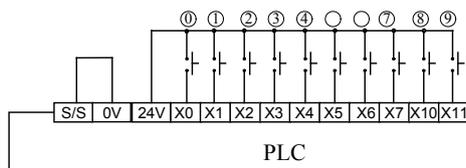
(D1) 为已输入数值的存放单元；

(D2) 为按键组当前状态的暂存起始单元，占用随后的共11个位单元。

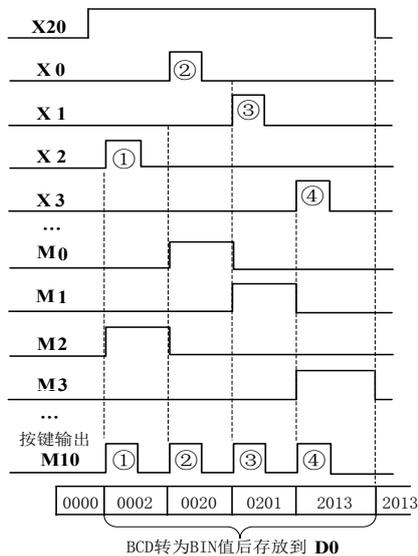
指令举例：



对应的硬件接线如下图：



若要输入数字“2013”，按顺序按键②①③④（X2、X0、X1、X3）即可，PLC内部变量的动作如下图。



- 由指令中参数的设置，X0~X11分别对应0~9数字键；M0~M9对应键的状态；当有任何按键按下时，按键输出单元M10都会置位；
- 按键值（如2013）被转换为BIN格式后存入指定的(D1)单元D0；(D0=0x7DD)，即使驱动的能量变为OFF，D0也不会改变；
- 当有多个按键按下时，先检测到的按键有效；当输入的数字超过4位后，最先输入的数字变化溢出，只留下输入的最后4个数字。

若采用32bit指令（DTKY），(D1) 占用32bit的变量，对上例即为D1、D0，分别为高字和低字。

FNC70

TKY

16bit	32bit		FNC 71	HKY	16键输入
✓	✓				
9步	17步		指令格式: HKY (S) (D1) (D2) (D3)		

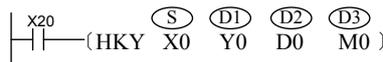
操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)	✓															
(D1)		✓														
(D2)										✓	✓	✓	✓	✓	✓	
(D3)		✓	✓	✓												

该指令是读取4×4矩阵型共16个按键，依次代表10进制的0~9按键，以及A~F的功能按键，当有按键动作（状态为ON）时，以按键的动作顺序，可输入十进制0~9999的4位数值，或A~F的功能键；若采用32bit指令，则可输入十进制0~99, 999, 999的8位数值，或A~F的功能键。其中：

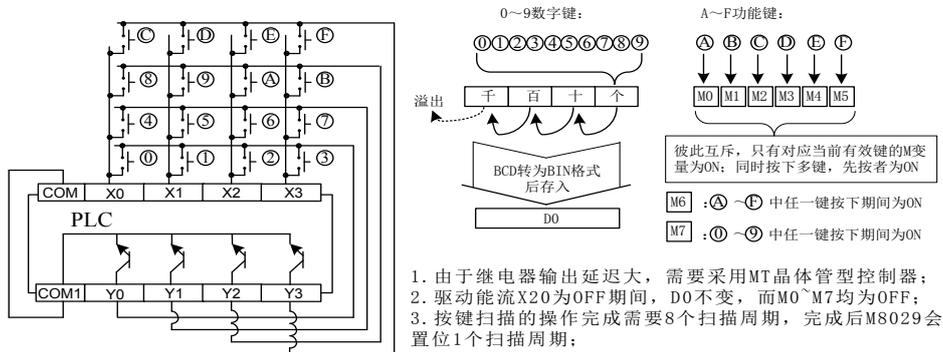
- (S) 为按键的扫描输入X端口的起始端口号，使用随后的共4个X端口；
- (D1) 为按键的扫描输出Y端口的起始端口号，使用随后的共4个Y端口；
- (D2) 为按键输入值存放单元，0~9999；当为32bit指令时可输入0~99, 999, 999范围内的数值；
- (D3) 为按键输入状态起始单元地址，共占用连续的8个位变量单元。

此指令只能用于晶体管输出型PLC。

指令举例：



对应的接线图及参数响应说明如下：



由于键扫描操作需要多个执行周期完成，为避免X端口滤波的影响，请采用“恒定扫描”模式，或定时中断处理。

扩展功能说明：

当将特殊变量M8167置为ON，本指令将①~④按键按16进制数据进行存储，保存到(D2)单元。

FNC71
HKY

16bit	32bit	\square	FNC 72	DSW	读取数字开关设定
✓					
9步			指令格式: DSW (S) (D1) (D2) (n)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)	✓															
(D1)		✓														
(D2)											✓	✓	✓	✓	✓	
(n)	常数, n=1~2															

该指令是读取矩阵型设置开关的状态，以4个BCD设定开关为1组，将设定值读取后存放指定单元，最多可读取2组设定开关。其中：

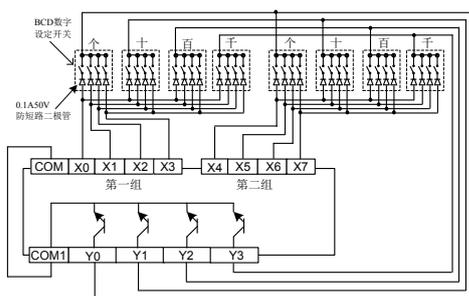
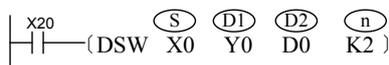
(S) 为按键的扫描输入X端口的起始端口号，若 (n) = 1，使用随后的共4个X端口；若 (n) = 2，则使用随后的共8个X端口；

(D1) 为按键的扫描输出Y端口的起始端口号，使用随后的共4个Y端口；

(D2) 为按键输入值存放单元，0~9999；

(n) 为数字设定开关的组数，只能取1~2。

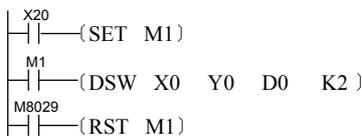
指令举例：



当X20=ON时，执行扫描读取数字开关设定的操作：
 1. 第一组数字开关的设定值，经BIN转换后存入D0；
 2. 第二组数字开关的设定值，经BIN转换后存入D1；
 3. 一次读取循环完毕，M8029会置位一个扫描周期。

使用说明：

- 需要使用晶体管输出型PLC才能正常检测数字开关；
- 一个数字开关设定的读取操作需要多个扫描周期才能完成，若采用按键启动读取操作，建议采用如下程序语句，保证可读取周期的完整：



FNC72
DSW

16bit	32bit		FNC 73	SEGD	七段码译码
✓		✓			
5步		5步	指令格式: SEGD (S) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(D)								✓	✓	✓	✓	✓	✓	✓	✓	✓

该指令是数据源的低4位，翻译成7段显示码，存放到目的变量的低8位中。其中：

(S) 为待译码的数据源（取BIN内容的最低四个位b0~b3）；

(D) 为译码后存放7段码的变量。

指令举例：



操作是，当X20为ON时，将D0内数据低4位译码后，输出到Y10~Y17端口。

翻译用的对应表如下表。该表格不需用户准备，PLC系统内部已具备该对照表。

数据		数码管组合	内部译码表值								译码后字符	
HEX数	BIN数		B7	B6	B5	B4	B3	B2	B1	B0		
0	0000	<p>每位对应一个笔段 1=笔段点亮 0=笔段熄灭</p>	0	1	1	1	1	1	1	0	0	
1	0001		0	0	0	0	0	1	1	0	0	0
2	0010		0	1	0	1	1	0	1	1	0	0
3	0011		0	1	0	0	1	1	1	1	0	0
4	0100		0	1	1	0	0	1	1	0	0	0
5	0101		0	1	1	0	1	1	0	1	0	0
6	0110		0	1	1	1	1	1	0	1	0	0
7	0111		0	0	1	0	0	1	1	1	0	0
8	1000		0	1	1	1	1	1	1	1	0	0
9	1001		0	1	1	0	1	1	1	1	0	0
A	1010		0	1	1	1	0	1	1	1	0	0
B	1011		0	1	1	1	1	1	0	0	0	0
C	1100		0	0	1	1	1	0	0	1	0	0
D	1101		0	1	0	1	1	1	1	0	0	0
E	1110		0	1	1	1	1	0	0	1	0	0
F	1111		0	1	1	1	0	0	0	1	0	0

FNC73
SEGD

16bit	32bit	\square	FNC 74	SEGL	七段码扫描显示
✓					
7步			指令格式: SEGL (S) (D) (n)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(D)		✓														
(n)	常数, n=0~7															

该指令是利用8个或12个Y端口，用于4位或8位七段锁存数码管的显示驱动，显示方式为扫描驱动方式。其中：

(S) 为待显示的数据，其值在BCD转换后才送到数码管进行显示；

(D) 为显示驱动用的Y端口起始地址号；

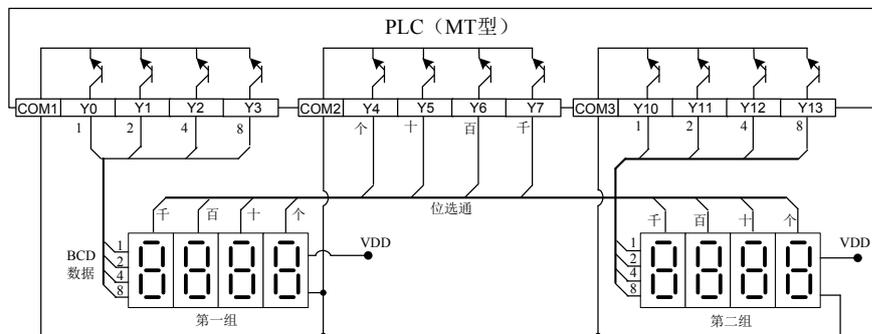
(n) 为根据显示数据的组数、信号正负逻辑等相关的设定值，见下面的详细描述。

指令举例：



对应的硬件接线如右图，在第一组数码管上显示D0的内容，在第二组数码管上显示D1的内容，若D0或D1的读数超过9999，程序运行就会出错：

接线图中所使用的数码管，自带显示数据锁存、7段译码和驱动、负逻辑型（输入端口为低电平时，表示输入的数据为1，或被选通）的7段数码显示管。显示处理中，PLC的Y4~Y7端口会自动循环扫描，每次只有1个端口为ON，作为位选通信号，此时Y0~Y3口上的数据即为送给对应位的BCD码



数据，当位选通信号由ON→OFF时，即被锁存到数码管内的锁存器中，经过内部的译码和驱动后，数码管将数字显示出来。PLC系统依次将Y4~Y7循环作同样处理，直到将4位都处理完毕。同样的道理，Y10~Y13是第二组4位数码管的数据输出端口，共用Y4~Y7的位选通线，处理方法相同，两组的显示处理是同时进行的。范例中若D0=K2468，D1=K9753，则第一组将会显示2 4 6 8，第二组显示9 7 5 3。

完成一次显示刷新需要12个扫描周期，处理完成后，M8029标志置ON。

(n) 的选择：根据可编程控制器的正负逻辑、七段码的正负逻辑等因素，按以下原则选择：

一组4位数的时候n=0~3。二组4位数的时候n=4~7

显示组数	1组				2组			
	PNP		NPN		PNP		NPN	
Y数据输出极性	相同	相反	相同	相反	相同	相反	相同	相反
选通与数据极性	相同	相反	相同	相反	相同	相反	相同	相反
(n) 的取值	0	1	2	3	4	5	6	7

PLC的晶体管输出极性与7段显示器的输入极性是否相同或者是不同时，可透过参数n的设置值来相互匹配。

H1U/H2U系列晶体管输出型的Y输出极性为NPN型。该指令在程序中最多只能使用2次。

使用说明：

由于继电器不适合较高频率的扫描输出动作，晶体管输出型PLC才能使用该指令。

FNC74

SEGL

16bit	32bit		FNC 75	ARWS	方向开关
✓					
9步			指令格式: ARWS (S) (D1) (D2) (n)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)	✓	✓	✓	✓												
(D1)												✓	✓	✓	✓	✓
(D2)		✓														
(n)	常数, n=0~3															

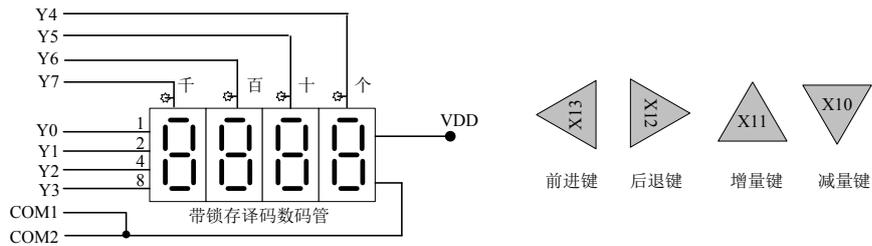
该指令可指定X作为编辑按键、Y端口驱动4位7段数码管，用作寄存器的参数编辑的简易界面。其中：

- (S) 为指定按键输入的起始地址，占用后续的共4个位单元；
- (D1) 用于被显示及修改的变量，只能显示一个16bit宽度的变量；
- (D2) 用作数码管显示驱动的Y端口起始地址，占用后续的共8个Y端口；
- (n) 为信号逻辑的设定值，参见前面SEGL指令中关于(n)的详细描述。

指令举例：



对应的硬件接线如下图所示，PLC应为晶体管输出型：



操作方法：

- 图中数码管显示D0的数值，按下X10~X13可修改其中的数值，D0的数值只能在0~9999之间；
- 当X20为ON时，光标位为千位。每次按后退键（X12）时，指定位按“千→百→十→个→千”的顺序切换；若按前进键（X13），则切换顺序相反；光标位由选通脉冲信号（Y004~Y007）连接的LED指示；
- 对于光标位，每次按增量键（X11）该内容按0→1→2→……8→9→0→1变化。按减量键（X10）时，则按0→9→8→7→……→1→0→9变化，修改的值立即生效。

指令使用说明：

当用户程序扫描时间短时，请使用恒定扫描模式，或在定时中断内按固定时间间隔运行。

FNC75

ARWS

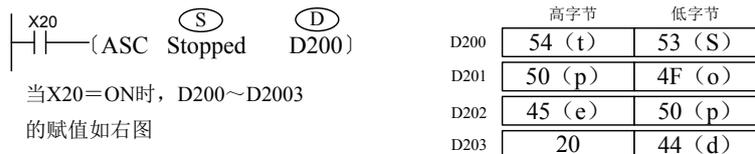
16bit	32bit	\square	FNC 76	ASC	ASCII转换
✓					
11步			指令格式: ASC \textcircled{S} \textcircled{D}		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
\textcircled{S}	在指令输入时, 由常数变量输入, 允许长度为8个字符。															
\textcircled{D}												✓	✓	✓		

\textcircled{S} 为欲执行ASCII码变换的英文字母, 由计算机输入, 最大允许长度为8个字符;

\textcircled{D} 为存放ASCII码的起始单元号, 占用随后的共4个 (M8161=0) 或8个变量单元 (M8161=1)。

指令举例:



若将特殊寄存器M8161置为ON, 则每个ASCII字符在转换后占用1个16bit变量, 如下图, 每个变量的高字节填0处理:

	高字节	低字节
D200	00	53 (S)
D201	00	54 (t)
D202	00	4F (o)
D203	00	50 (p)
D204	00	50 (p)
D205	00	45 (e)
D206	00	44 (d)
D207	00	20

附: 《ASCII代码对照表》

10进制位	ASCII (16进制数)
0	30
1	31
2	32
3	33
4	34
5	35
6	36
7	37
8	38
9	39

代码	ASCII (16进制数)
STX	02
ETX	03

英语字母	ASCII (16进制数)	英语字母	ASCII (16进制数)
A	41	N	4E
B	42	O	4F
C	43	P	50
D	44	Q	51
E	45	R	52
F	46	S	53
G	47	T	54
H	48	U	55
I	49	V	56
J	4A	W	57
K	4B	X	58
L	4C	Y	59
M	4D	Z	5A

FNC76

ASC

16bit	32bit	<input type="checkbox"/>	FNC 77	PR	ASCII打印输出
✓					
5步			指令格式: PR (S) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)												✓	✓	✓		
(D)		✓														

该指令将指定变量单元的数值，通过Y输出端口以同步方式逐个字节对外输出。其中：

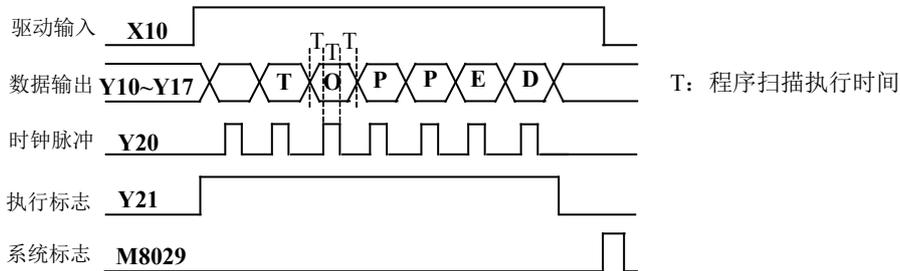
(S) 为待输出的变量单元起始地址；

(D) 为进行输出打印的Y端口起始号。

指令举例：



若D200~D203中的ASCII码为“STOPPED”，则对应的输出端口信号及其时序如下图：



当X020有效时，执行ASC指令，将字符串“STOPPED”转换成相应ASCII码存放在D200~D203中；

当X010有效时，执行PR打印指令，将D200~D203中存放的ASCII码数据按“S→T→O→P→P→E→D”的顺序一次经Y10~Y17发送到外部显示器进行打印。

使用说明：

- 必须使用晶体管输出型PLC，才能完成该指令功能；
- 打印过程中，驱动信号X10变为OFF时，打印输出即被中断。X10再次为ON时，打印动作重新开始；
- 打印输出过程中，遇到“00”的字符时，会自动结束打印操作，之后的文字不再处理；
- M8027=Off时，固定为8个字符的串行的输出；当M8027=On时，则可执行1~16个字符的串行输出。
- M8027为Off时：能流无效后M8029不动作；
- M8027为ON时：M8029完成标志在驱动能流信号无效后会置ON；

指令按扫描周期（图中T）执行，若扫描周期短时，请用恒定扫描模式；若过长则可以在定时中断程序中执行。

FNC77
PR

16bit	32bit	\square	FNC 78	FROM	BFM区读取
✓	✓	✓			
9步	17步		指令格式: FROM (m1) (m2) (D) (n)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(D)								✓	✓	✓	✓	✓	✓	✓	✓	

m1=0~7 (本地), 100+站号 (远程);
 m2=0~32767;
 n=1~32767;
 (D) 元件指定时, 16bit指令可用K1~K4; 32bit指令可用K1~K8; m1, m2, n不支持字符装置D寄存器

该指令用于读取特殊扩展模块的BFM区寄存器的数据读取操作。其中：

(m1) 为扩展模块的地址编号：

在本地扩展模块中，取值范围0~7，最靠近主模块的为0，依次编号，最多允许有8个本地扩展模块。适用于本地模拟量模块、温度模块等，不能用于本地数字量扩展模块。

在远程扩展模块中，取值范围为模块通信站号+100，最多允许有62个远程扩展模块。适用于所有远程扩展模块，包括远程数字量扩展模块，需注意本地数字量模块和远程数字量模块的使用区别；

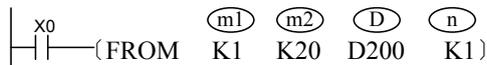
(m2) 为特殊模块内BFM的寄存器地址号，取值范围0~32767；

(D) 为读取参数在主模块中的存放地址，当读取的寄存器数多于1个时，占用随后的单元；

(n) 本次读取参数的个数（按Word计算），取值范围1~32767，为按寄存器地址依次读取。

注意H_{1U}只能使用远程扩展模块。

指令举例一：



FNC78

FROM

访问本地扩展模块，表示当X0为ON时，将#1号特殊模块中的第20号地址（16bit宽度）的内容读出到PLC的D200寄存器中，一次读取一笔（n=1）。当X0为OFF时，不执行操作。

指令举例二：



访问远程扩展模块，假设访问的是H_{2U}-4ADR

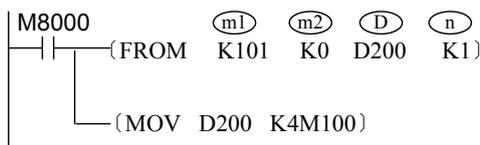
K101代表（100+站号1）；远程模块在使用FROM/TO指令时，增加站号偏移100。

K9表示4AD模块的BFM9。

D200表示读出的BFM9的存放地址。

K1表示读取1个16位BFM。

指令举例三：



本例子介绍远程数字量扩展模块H_{2U}-1600ENDR的访问方法，因为16点的输入模块用一个BFM区代表了所有的输入点，即BFM区的K0（一个寄存器）的每个位即为每个输入点，故可以将模块的状态读取回来后，用MOV D200 K4M100来分别取位，此时M100~M107为远程模块的X0~X7，M108~M115为远程模块的X10~X17，我们可以通过读取M100~M115就能读取X0~X17的状态，注意：为了保证能读取到数字量扩展模块输入点的最新状态，请使用M8000来驱动FROM 指令，在每个扫描周期都能更新。

当用32bit指令时，(D) 指定的地址为低16bit地址，(D) + 1为高16bit地址。

16bit	32bit		FNC 79	TO	BFM区写入
✓	✓	✓			
9步	17步		指令格式: TO (m1) (m2) (D) (n)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(D)								✓	✓	✓	✓	✓	✓	✓	✓	

m1=0~7 (本地), 100+站号 (远程);
 m2=0~32767;
 n=1~32767;
 (D) 元件指定时, 16bit指令可用K1~K4; 32bit指令可用K1~K8; m1, m2, n不支持字符装置D寄存器

该指令用于向特殊扩展模块的BFM区寄存器的作数据写入操作。其中：

(m1) 为特殊扩展模块的地址编号：

在本地扩展模块中，取值范围0~7，最靠近主模块的为0，依次编号，最多允许有8个本地扩展模块。适用于本地模拟量模块、温度模块等，不能用于本地数字量扩展模块。

在远程扩展模块中，取值范围为模块通信站号+100，最多允许有62个远程扩展模块。适用于所有远程扩展模块，包括远程数字量扩展模块，需注意本地数字量模块和远程数字量模块的使用区别；

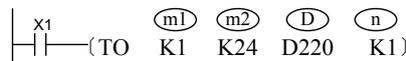
(m2) 为特殊模块内BFM的寄存器地址号，取值范围0~32767；

(D) 为主模块中参数寄存器地址，其参数作为写操作数据的来源。当写操作的寄存器数多于1个时，占用随后的单元；

(n) 本次写入参数的笔数，取值范围1~32767，为按寄存器地址依次写入。

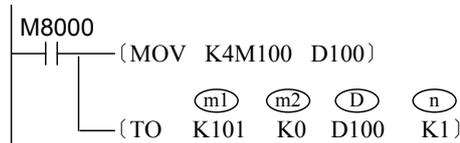
注意H10只能使用远程扩展模块。

指令举例一：



表示当X1为ON时，将PLC的D220寄存器中的数据，写入到#1号特殊模块中的第24号地址中，一次只写入一笔。当X1为OFF时，不执行操作。

指令举例二：



本例子介绍远程数字量扩展模块H2U-0016ERDR的访问方法，因为16点的输入模块用一个BFM区代表了所有的输出点，即BFM区的K0（一个寄存器）的每个位即为每个输出点，故可以将全部输出点的状态放到一个字里面后再传送到模块里面去，例子中M100~M107为远程模块的Y0~Y7，M108~M115为远程模块的Y10~Y17，我们可以通过改变M100~M115就可以改变远程模块中Y0~Y17的状态，在远程数字量输出模块中，可以在每次更新输出点后再驱动TO指令，也可以用M8000来驱动。

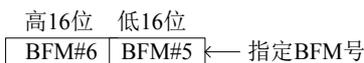
FNC79
TO

指令举例三:

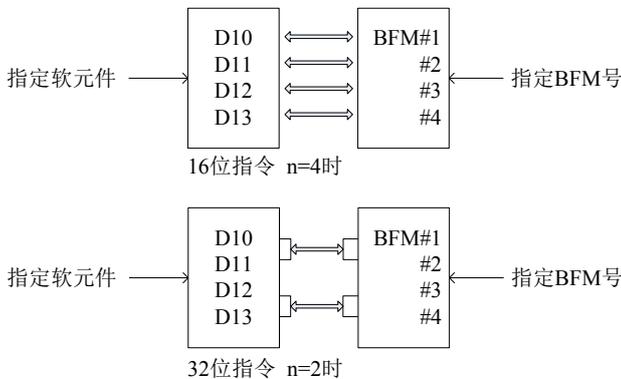
对一个#0编号地址的H2U-4AD模块的CH1通道作偏移/增益设置操作, 编程如下:



当用32bit指令时, (D) 指定的地址为低16bit地址, (D) + 1为高16bit地址。例如:



(n) 表示操作的参数个数, 16bit指令中, n=2表示2Word; 而32bit指令中, n=1表示2Word, 即16位指令的n=2与32位指令的n=1意义相同, 请注意。例如:



关于FROM/TO指令的使用说明:

与FROM/TO指令有关的特殊寄存器的说明:

1) M8164 (FROM/TO指令的传送点数可变模式)

若M8164=ON, 执行FROM/TO指令时, 特殊数据寄存器D8164 (FROM/TO指令的传送点数指定寄存器) 的内容作为传送点数n进行处理;

2) 用FROM/TO指令访问扩展模块是比较耗时的操作, 执行多个FROM/TO指令或传送多个缓冲存储器数据时, PLC的扫描周期会延长。为了防止运行超时, 可在FROM/TO前加入延长监视定时器时间的WDT指令, 或者错开FROM/TO指令的执行时间, 或者用脉冲执行型指令。

3) 关于特殊模块的连接方法和输入输出编号等使用方法, 请参阅各特殊模块附带的专用手册。

4) 远程模块的配置请参考《5.9 CAN通讯指令说明》。

4.3.9 外围 SER 设备 (80~93)

16bit	32bit		FNC 80	RS	串行数据传送
✓					
9步			指令格式: RS (S) (m) (D) (n)		

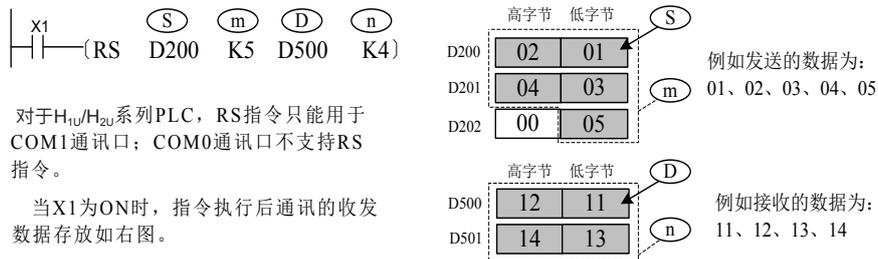
操作数	位元件					字元件										
	X	Y	M	S		K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)														✓		
(m)						✓	✓							✓		
(D)														✓		
(n)						✓	✓							✓		

该指令是一个通讯收发指令，将指定寄存器区域的数据，自动向串口依次发送，将串口接收到的数据存放指定区域，相当于用户程序直接访问通讯缓冲区，借助用户程序对通讯收发缓冲区的处理，实现自定义协议的通讯。其中：

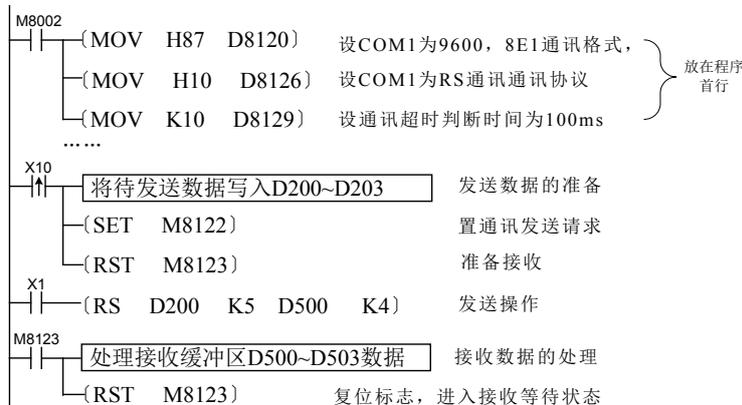
- (S) 为待发送数据存放的寄存器区的起始地址；
- (m) 为待发送数据的长度（字节数），取值范围0~256；
- (D) 为通讯接收数据的存放寄存器区的起始地址；
- (n) 本通讯接收的数据长度（字节数），取值范围0~256。

RS指令半双工/全双工模式由D8120的Bit10设定。用户程序可以写多条RS指令，但是同一时间只能有一条RS指令被驱动。在每次驱动RS指令前，必须将M8122置位；

指令举例一：



实际编程时，需要作一些串行通讯的配置和准备，如设定串口的收发模式、波特率、位数、校验位、软件协议的设定、超时判断条件、收发缓冲区的数据准备、收发标志处理等，才能按预期的要求进行通讯。仍以上述语句为例，一个比较完整的RS通讯设置程序如下：



FNC80
RS

FNC80

RS

16bit	32bit		FNC 81	PRUN	并联运行
✓	✓	✓			
5步	9步		指令格式: PRUN (S) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)							✓		✓							
(D)								✓	✓							

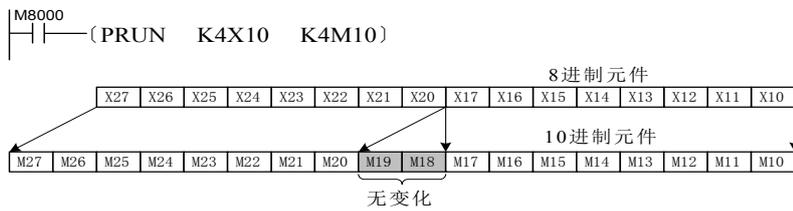
该指令是将 (S) 起始连续地址的位变量 (以8进制为宽度单位), 成批复制到 (D) 起始位变量组中。其中:

(S) 为待复制的位变量的起始地址, 要求地址的个位必须为0, 如X10, M20等;

(D) 为复制的目的位变量起始地址, 同样要求地址的个位必须为0, 如M30, Y10等。

其中Kn中, 允许n=1~8。

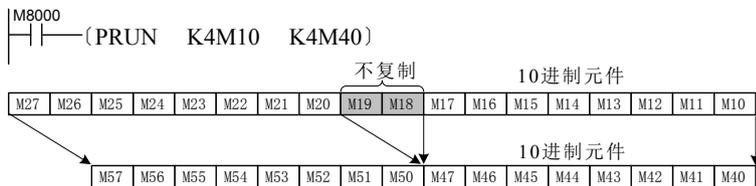
指令举例一:



指令举例二:



指令举例三:



FNC81
PRUN

16bit	32bit	\square	FNC 82	ASCII	HEX-ASCII转换
✓		✓			
7步		7步	指令格式: ASCII (S) (D) (n)		

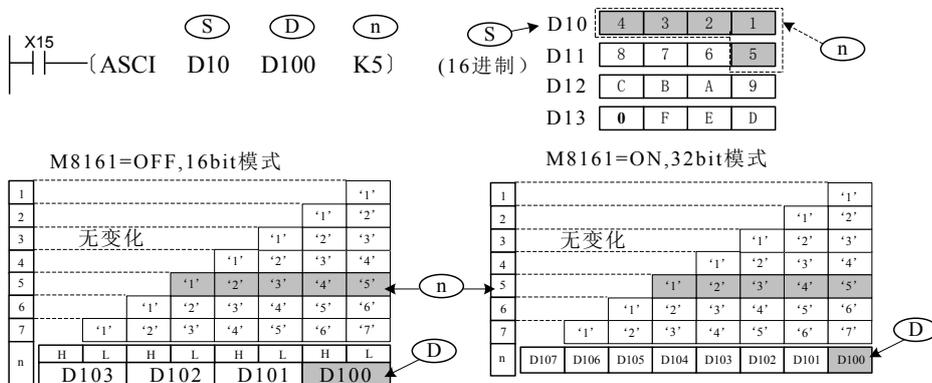
操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(D)								✓	✓	✓	✓	✓	✓			
(n)	常数, n=1~256															

该指令是将 (S) 的值转换成ASCII码后, 存储到 (D) 为起始地址的变量中。其中:

- (S) 为待转换的变量地址或常数数值;
- (D) 为转换后ASCII码的存放起始地址;
- (n) 为转换的字符位数。

ASCII数值转换遵照ASCII与HEX进制数值对照表, 如: ASCII '0' 对应HEX 'H30'; ASCII 'F' 对应HEX 'H46' 等。关于HEX和ASCII的对照关系请参考FNC76 (ASC) 指令后面的附录。

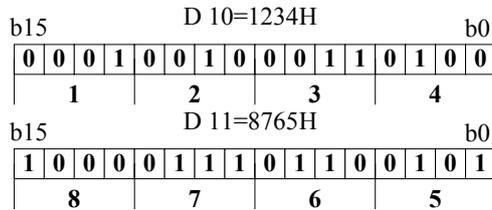
指令举例:



其中, M8161标志决定了计算结果存放目的变量的宽度模式, 当M8161=OFF时, 为16bit模式, 即变量的高字节和低字节分别存储; 当M8161=ON时, 为8bit模式, 只有变量的低字节存储结果, 因此实际使用变量区域的长度增加。

FNC82

ASCII

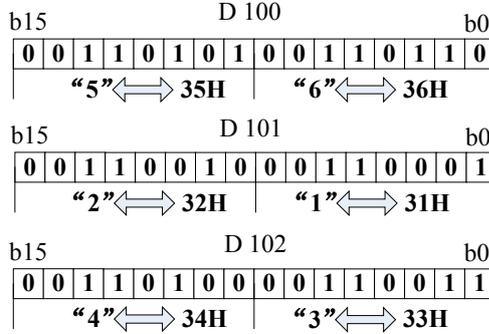
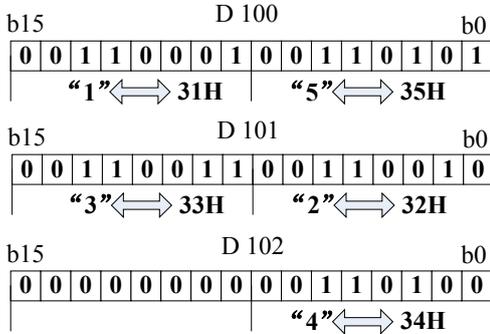


当M8161=0FF、n=5 时，位的组成

当M8161=0FF、n=6 时，位的组成

(D10~D11)转换

(D10~D11)转换

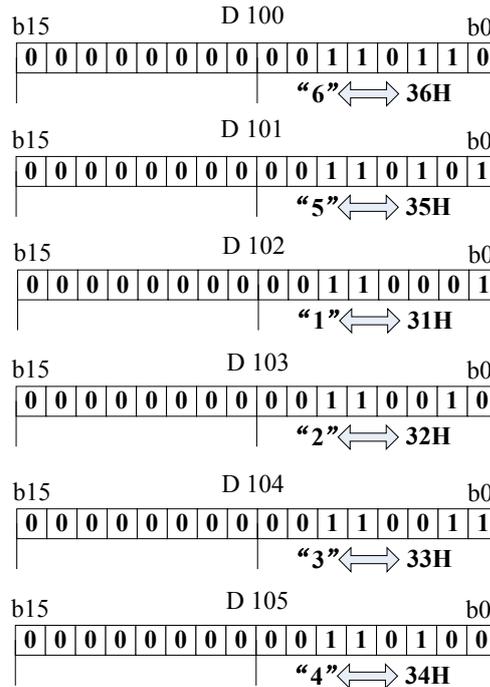
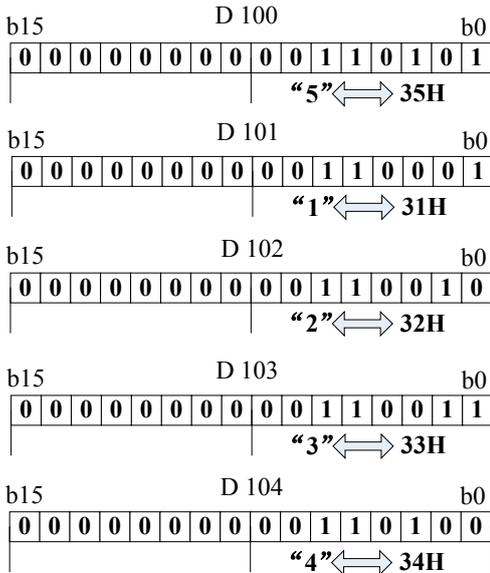


当M8161=0N、n=5 时，位的组成

当M8161=0N、n=6 时，位的组成

(D10~D11)转换

(D10~D11)转换



FNC82
ASCI

注：RS/HEX/ASCI/CCD等指令共用M8161模式标志，编程时注意。

16bit	32bit	\square	FNC 83	HEX	ASCII-HEX转换
✓		✓			
7步		7步	指令格式: HEX (S) (D) (n)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(D)								✓	✓	✓	✓	✓	✓	✓	✓	
(n)	常数, n=1~256															

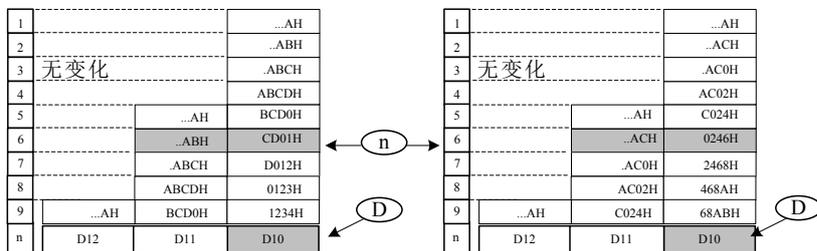
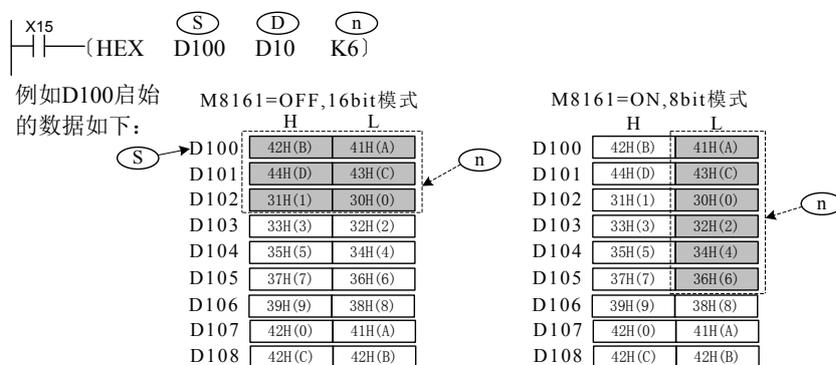
该指令是将 (S) 起始变量的值转换成HEX码后, 存储到 (D) 为起始地址的变量中, 转换的字符数、存储模式可以设定。其中:

(S) 为待转换的变量地址或常数数值, 若为寄存器变量, 以32bit变量宽度 (即4个ASCII字符) 为单位进行转换分隔;

(D) 为转换后HEX码的存放起始地址, 占用的变量空间与 (n) 有关;

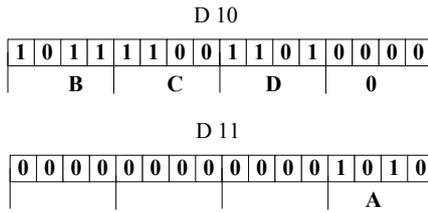
(n) 为转换的字符位数。

指令举例:

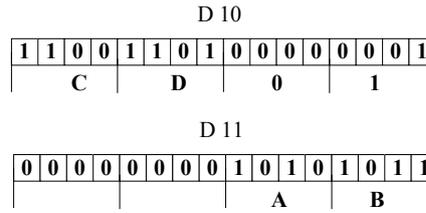


其中, M8161标志决定了变量宽度模式, 当M8161=OFF时, 为16bit模式, 即变量的高字节和低字节都参与运算; 当M8161=ON时, 为8bit模式, 只有变量的低字节参与运算, 高字节的内容丢弃, 因此实际使用变量区域 (S) 的长度增加。

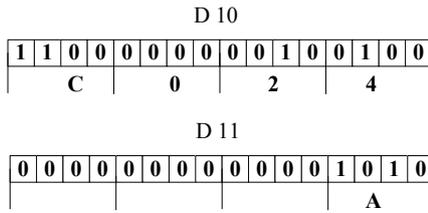
当M8161=0FF、n=5 时，位的组成
用了D100~D102(高低字节)转换



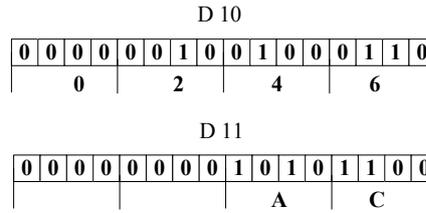
当M8161=0FF、n=6 时，位的组成
用了D100~D102(高低字节)转换



当M8161=0N、n=5 时，位的组成
用了D100~D104(低字节)转换



当M8161=0N、n=6 时，位的组成
用了D100~D105(低字节)转换



注:

RS/HEX/ASCII/CCD等指令共用M8161模式标志，编程时注意：

Ⓢ 数据区的源数据必须为ASCII码字符，否则转换出错；

若输出的数据为BCD格式，HEX转换后，需要进行BCD—BIN转换，才是正确的数值。

FNC83
HEX

16bit	32bit	\square	FNC 84	CCD	求校验和
✓		✓			
7步		7步	指令格式: CCD (S) (D) (n)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)							✓	✓	✓	✓	✓	✓	✓		
(D)								✓	✓	✓	✓	✓	✓		
(n)	常数, n=1~256														

该指令是对 (S) 起始的 (n) 个变量进行两种校验和运算, 将直接加法的求和运算结果存于 (D); 将逐个异或逻辑运算的结果存于 (D) + 1单元中。本指令用于作通信时, 为了确保数据传输时的正确性所做的字符串总和检查 (SumCheck)。其中:

- (S) 为待求校验和运算的变量起始地址, 使用后续地址的变量单元;
- (D) 用于存放校验和; (D) + 1单元用于存放逐项异或逻辑运算结果;
- (n) 为用于校验的变量字节数。

指令举例:

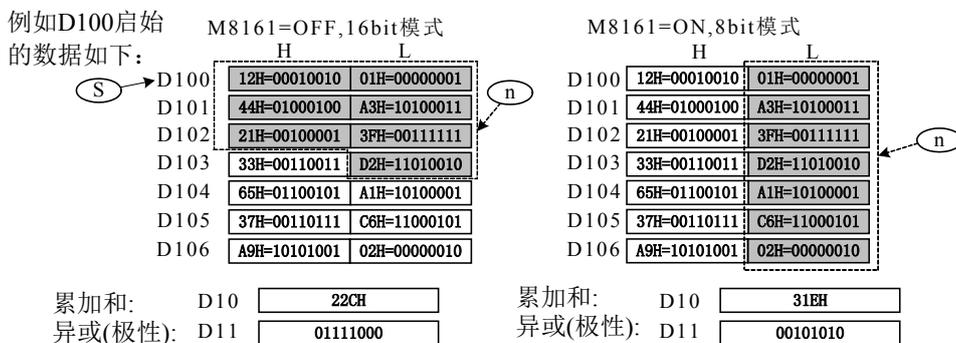


M8161标志决定了变量宽度模式, 当M8161=OFF时, 为16bit模式, 即变量的高字节和低字节都参与运算; 当M8161=ON时, 为8bit模式, 只有变量的低字节参与运算, 高字节的内容被丢弃, 因此实际使用变量区域的长度增加, 见下例图:

“累加和”就是将指定的n个变量直接相加的计算结果;

“异或”逻辑运算的方法则是:

- 1) 将参与运算的变量都换算为二进制数;
- 2) 先统计每个变量的bit0为1的个数, 若为偶数个, 则异或结果的bit0为0; 若为奇数个, 则异或结果的bit0为1;
- 3) 再统计每个变量的bit1为1的个数, 若为偶数个, 则异或结果的bit1为0; 若为奇数个, 则异或结果的bit1为1;
- 4) 依次类推, 逐个计算bit2~bit7, 所得的二进制数换算为HEX数值即为异或结果 (或称极性值)。



RS/HEX/ASCII/CCD等指令共用M8161模式标志, 编程时注意该标志的处理。

16bit	32bit		FNC 88	PID	PID运算
✓					
9步			指令格式: PID (S1) (S2) (S3) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)														✓		
(S2)														✓		
(S3)														✓		
(D)														✓		

该指令是完成PID运算，用于闭环控制系统参数的控制。PID控制在机械设备、气动设备、恒压供水和电子设备等等中有着广泛的应用。其中：

(S1) 为PID控制的目标值；

(S2) 为实测的反馈值；

(S3) 为PID运算所需设定参数、中间结果保存的缓存区起始地址，占用随后地址的共25个变量单元，取值范围是D0~D7975，最好指定停电保持区，在电源OFF后仍保持设定值，否则首次启动运算前需对缓存区作赋值处理。其中每个单元的功能和参数描述详见本节说明；

(D) 为PID计算结果的存放单元，请指定 (D) 为非电池保持区，否则首次启动运算前需对其作初始化清0处理。

指令举例：



其中参数说明如下：

D9中存放的是PID调节的目标值，D10为闭环反馈值，注意D9、D10必须为同一量纲，如都为0.01MPa单位，或都为1℃单位等等；

D200~D224共25个单元用于存放PID运算的设定值与过程值，这些值在首次PID计算前就必须逐项进行设定；

D130单元则用于存放计算后的控制输出值，用于控制执行的动作。

就 (S3) 起始的各单元参数值的功能和设定方法说明如下表：

单元	功能	设定说明
(S3)	采样时间 (TS)	设定范围1~32767 (ms)，但需大于PLC程序扫描周期
(S3)+1	动作方向 (ACT)	bit0: 0=正动作；1=逆动作 bit1: 0=输入变化量报警无效；1=输入变化量报警有效 bit2: 0=输出变化量报警无效；1=输出变化量报警有效 bit3: 不可使用 bit4: 0=自整定不动作；1=执行自整定（目前版本暂时不提供此项功能） bit5: 0=输出值上下限设定无效；1=输出值上下限设定有效 bit6~bit15不可使用 另外，请不要使bit5和bit2同时处于ON。
(S3)+2	输入滤波常数 (α)	0~99[%]，0=没有输入滤波
(S3)+3	比例增益 (Kp)	1~32767[%]
(S3)+4	积分时间 (T1)	0~32767 (×100ms)，0=作为∞处理（无积分）
(S3)+5	微分增益 (KD)	0~100[%]，0=无微分增益
(S3)+6	微分时间 (TD)	0~32767 (×10ms)，0=无微分处理
(S3)+ (7~19)	PID运算的内部处理占用，首次运行前应清为0	
在<ACT>的bit1=1，bit2=1或bit5=1时，(S3)+ (20~24) 被占用，定义如下：		
(S3)+20	输入变化量（增侧）报警设定值	0~32767，（<ACT>的bit1=1时有效）
(S3)+21	输入变化量（减侧）报警设定值	0~32767，（<ACT>的bit1=1时有效）

FNC88

PID

单元	功能	设定说明
(S3)+22	输出变化量（增侧）报警设定值	0~32767, (<ACT>的bit2=1, bit5=0时有效)
		输出上限设定值-32768~32767 (<ACT>的bit1=0, bit5=1时有效)
(S3)+23	输出变化量（减侧）报警设定值	0~32767 (S3+1<ACT>的bit2=1, bit5=0时有效)
		输出下限设定值-32768~32767 (<ACT>的bit1=0, bit5=1时有效)
(S3)+24	报警输出	bit0输入变化量（增侧）溢出 bit1输入变化量（减侧）溢出 Bit2输出变化量（增侧）溢出 Bit3输出变化量（减侧）溢出 (<ACT>的bit1=1或bit2=1时有效)

PID的理论运算公式如下:

正逻辑

$$\Delta MV = K_p | (EV_n - EV_{n-1}) + \frac{T_s}{T_i} EV_n + D_n |$$

$$EV_n = PV_{nf} - SV$$

$$D_n = \frac{T_d}{T_s + \alpha D} (-2PV_{nf-1} + PV_{nf} + PV_{nf-2}) + \frac{\alpha D * T_d}{T_s + \alpha D * T_d} * D_{n-1}$$

$$MV_n = \sum \Delta MV_n$$

逆逻辑

$$\Delta MV = K_p | (EV_n - EV_{n-1}) + \frac{T_s}{T_i} EV_n + D_n |$$

$$EV_n = SV - PV_{nf}$$

$$D_n = \frac{T_d}{T_s + \alpha D} (2PV_{nf-1} - PV_{nf} - PV_{nf-2}) + \frac{\alpha D * T_d}{T_s + \alpha D * T_d} * D_{n-1}$$

$$MV_n = \sum \Delta MV_n \Delta$$

EVn: 本次采样时的偏差

Dn: 本次的微分项

EVn-1: 1个周期前的偏差

Dn- 1: 1个周期前的微分项

SV: 目标值

Kp: 比例增益

PVnf: 本次采样时的测定值（滤波后）

Ts: 采样周期

PVnf-1: 1个周期前的测定值（滤波后）

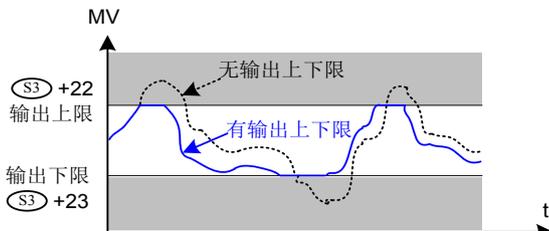
Ti: 积分常数

PVnf-2: 2个周期前的测定值（滤波后）

正逻辑也称正方向，如恒温控制系统的加热功率调节等，属于正逻辑PID控制；负逻辑也称反方向，如恒温控制系统的散热风扇运转速度控制，属于负逻辑PID控制。

输出上下限设定:

对于多数应用系统，PID的调节输出范围是有限的，如电平信号幅值范围、变频器调节频率范围等。根据实际情况设定PID运算中的输出上限、输出下限，让闭环系统的执行装置在正常运行参数下工作。



超限报警设定:

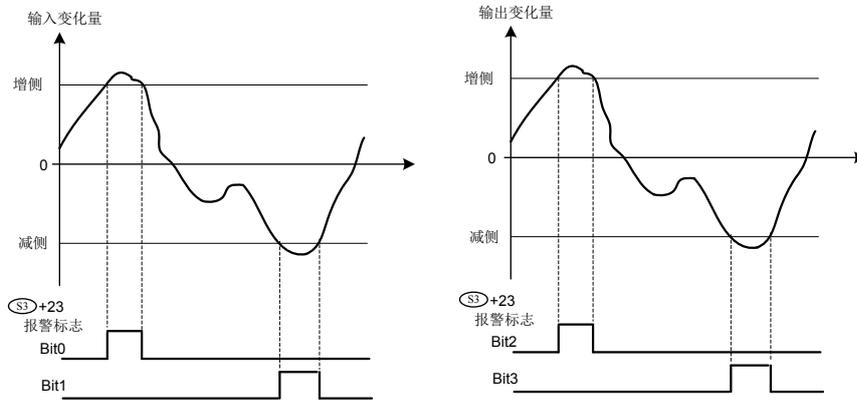
若需要检查控制器的输入量（反馈量）的变化量是否超限（上限或下限）、PID的运算输出的变化量是否过大（超过上限或下限），可在ACT ((S3)+1)单元中，将ACT的bit1=ON, bit2=ON, 启用报警功能；在(S3)+(20~23)单元中分别设定变化量报警限值，运行中就可在(S3)+24单元读取参数的超限状态了。这在一些需要判断调节状态的场合，简化了运算。

使用输出变化量的报警功能时，(S3)+1 (ACT)的bit5请必须被设置为OFF。

这里所述的“变化量” = (上次的数值) - (本次的数值)，对应动作示意图如下:

FNC88

PID



可见“减侧”的告警下限设定值为负值。为避免输入量波动过大，一般对输入的信号进行硬件滤波处理，还可在用户程序中进行软件滤波处理。

PID常数的设定:

PID的Kp、Ki、Kd、Ts等常数选择，对PID控制特性至关重要，这三个参数对稳定性的定性影响为：

- 5) 比例增益Kp过大，将出现过冲，甚至无法稳定；Kp过小，调节过程缓慢；设置恰当时，系统PID调节的响应速度效果好；
- 6) 积分时间Ti过大，调节过程缓慢；Ti过小，输出突变大，难以稳定；设置恰当时，稳定快，偏差小；
- 7) 微分增益Kd过大，系统极易自激振荡，无法稳定；Kd小对稳定影响小；设置恰当时利于快速逼近设定值。由于该参数对稳定的影响大，多数应用中常不使用微分调节，即将其增益设为0；
- 8) 采用周期Ts对应于执行PID运算的时间间隔，其设定值应大于PLC程序的扫描执行时间，为保证PID调节效果，PLC程序最好采用固定扫描周期的模式运行，或将PID运行放在定时中断中运行。

错误代码

控制参数的设定值或PID运算中的数据发生错误时，则运算错误M8067变为ON状态，根据其错误内容D8067中存入下述数据。

代码	错误内容	处理状态	处理方法
K6705	应用指令的操作数在对象软元件范围外	PID命令运算停止	请确认控制数据的内容
K6706	应用指令的操作数在对象软元件范围外		
K6730	采样时间(TS)在对象软元件范围外(TS<0)		
K6732	输入滤波常数在对象(α)范围外(α<0或100≤α)		
K6733	比例增益(KP)在对象范围外(KP<0)		
K6734	积分时间(TI)在对象范围外(TI<0)		
K6735	微分增益(KD)在对象范围外(KD<0或201≤KD)		
K6736	微分时间(TD)在对象范围外(KD<0)	PID命令运算继续	
K6740	采样时间(Ts)≤运算周期		
K6742	测定值变化量超过((PV<-32768或32767<PV)		
K6743	偏差超过(EV<-32768或32767<EV)		
K6744	积分计算值超过(-32768~32767以外)		
K6745	由于微分增益(舒)超过微分值超过		
K6746	微分计算值超过(-32768~32767以外)	自整定结束	
K6747	PID运算结果某超过(-32768~32767以外)		
K6750	自整定结果不良		自整定开始时的测定值和目标值的差为150以下或自整定开始时的测定值和目标值的差的1/3以上则结束确认测定值、目标值后，请再次进行自整定。
K6751	自整定动作方向不一致	自整定继续	从自整定开始时的的测定值预测的动作方向和自整定用输出时实际动作方向不一致。请使目标值、自整定用输出值、测定值的关系正确后，再次进行自整定。
K6752	自整定动作不良	自整定	自整定中测定值因上下变化不能正确动作。请使采样时间远远大于输出的变化周期，增大输入滤波常数。设定变更后请再次进行自整定。

FNC88
PID

要点

必须在PID运算执行前，将正确的测定值读入PID测定值（PV）中。特别对模拟量输入模块的输入值进行PID运算时，需注意其转换时间。

编程说明:

- 9) PID指令在程序中可多次使用，可同时执行，但各PID指令中使用的(S3)变量区域不要有重叠；在步进指令、跳转指令、定时中断、子程序中也可以使用，在此情况下执行PID指令时，需事先清除(S3)+7缓存单元。
- 10) 采样时间TS的最大误差为 $-(1\text{运算周期}+1\text{ms})\sim+(1\text{运算周期})$ 。如果采样时间 $TS\leq$ 可编程控制器的1个运算周期，则发生下述的PID运算错误（K6740），并以 $TS=\text{运算周期}$ 执行PID运算。在这种情况下，建议最好使用恒定扫描模式或在定时器中断（I6□□~I8□□）中使用PID指令。

FNC88

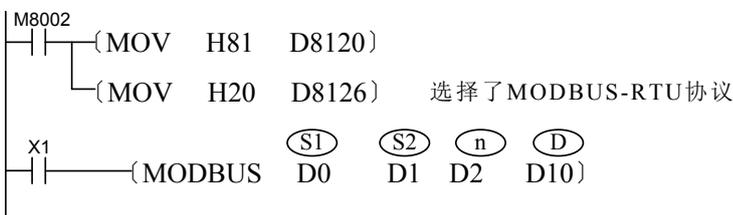
PID

16bit	32bit	P	FNC 89	MODBUS	数据传送
✓					
9步			指令格式: MODBUS (S1) (S2) (n) (D)		

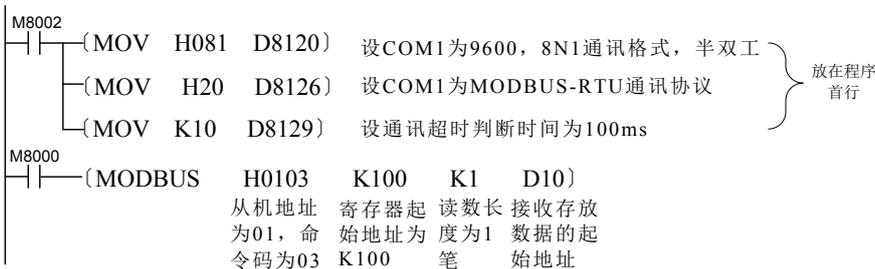
操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓							✓			
(S2)					✓	✓							✓			
(n)					✓	✓							✓			
(D)													✓			

为了直观方便，在标准MODBUS主站协议通讯中的RS指令我们可以用MODBUS来代替。各定义如下：

- (S1) 为从机地址（高字节）、通讯命令（低字节，按MODBUS协议定义）；
- (S2) 为访问从站的寄存器起始地址号；
- (n) 预读或写的数据长度；
- (D) 为读或写数据的存放单元起始地址，占用后续地址单元，长度由(n)决定。



指令举例一：



该指令用于具有MODBUS协议从站设备（如MD320/300/280等系列变频器）进行通讯，将非常方便，指令举例一中，PLC不断的读#1从机内，地址为100的寄存器，数据存于D10单元。

使用说明：

程序中可多次使用MODBUS指令，和标准RS不同，MODBUS协议中的MODBUS指令可有多处MODBUS指令被同时使能。和RS指令相比，可省略对M8122的置ON，可省略对M8123的复位。

FNC89
MODBUS

16bit	32bit	P	FNC 90	CANTX	CAN数据发送
✓					
13步			指令格式: CANTX (S1) (S2) (D) (n)		

16bit	32bit	P	FNC 91	CANRX	CAN数据接收
✓					
13步			指令格式: CANRX (S1) (S2) (D) (n)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✓	✓							✓		
(S2)					✓	✓							✓		
(D)													✓		
(n)					✓	✓							✓		

H1U系列主模块PLC和H2U的MRN(MTN)系列主模块PLC通过CAN扩展卡，可以组建CAN网络。

在CAN网络中，通过CANTX/CANRX自由指令，用户可编写任意CAN通信用户协议，各参数定义如下：

(S1)，(S2) 两个参数共同组成CAN地址：(S1)的bit15和bit14保留，(S1)的bit13为CAN地址位数设定，“0”表示标准CAN地址（11位），“1”表示扩展CAN地址（29位）；若为11位地址，则由(S2)的bit0~bit10位表示地址，此时(S1)为0，(S2)不能大于H7FF。若为29位地址，由(S2)（低16位地址）和(S1)的bit0~bit12（高13位地址）共同组成，此时(S1)的bit13为1。

(D) 在CANTX指令中为发送缓冲区，在CANRX指令中为接收缓冲区；从该D元件开始的最大4个D元件作为发送或者接收缓冲区。

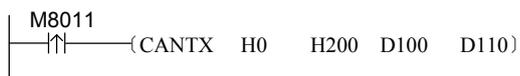
(n) 在CANTX指令中为发送数据个数，在CANRX指令中为接收数据个数；以字节为单位，最大为8。

一条CANTX/CANRX指令最多只能发送/接收8个字节，但是在程序里可以写多条CANTX/CANRX指令以便交换更多的数据，且不用分时驱动。

注意：可以同时用M8000同时驱动CANRX接收指令，但是CANRX接收只能是逐条接收，第一条接收完成或接收超时了，才能开始执行下一条指令接收，顺序执行，这个分时PLC后台自动处理，不用用户做程序做分时。若发送指令和接收指令的ID不能匹配，数据就会丢失。

指令举例一：

10ms 发送一组数据，缓冲区为D100~103，D110 存发送的字节数



若D110 = k8, D100 = h1234, D101 = h5678, D102 = h9ABC, D103 = hDEF0

发送的数据如下表：

第一字节	第二字节	第三字节	第四字节	第五字节	第六字节	第七字节	第八字节
h12	h34	h56	h78	h9A	hBC	hDE	hF0

若接收端字节从高到低，CAN接收数据寄存器MDL = h12345678, MDH = h9ABCDEF0

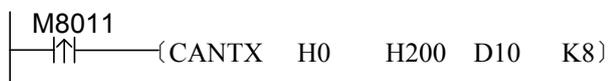
若D110 = k1, 只发送一个字节：h12。

若D110 = k3, 发送前三个字节：h12, h34, h56。

依次类推。

指令举例二：（(S1)的bit13为0）

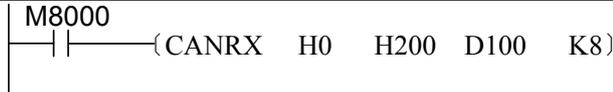
假设有8台PLC连接在CAN网络中，其中一台PLC程序里面写了如下发送指令



这台PLC往地址H200发送了D10~D13寄存器里面的数据，因为CAN协议不分主从站，所以这台PLC往地址H200发送的数据是开放的，在网络中的其他任意一台PLC想接收这台PLC里面D10~D13的数据，都可以在这任意一台PLC的程序里面写入如下接收指令：

FNC90
FNC91

CANTX
CANRX



只要执行了上面的语句，即可接收地址为H200的数据存放在D100~D103里面。可以在多台PLC里面分别写程序接收这一数据。

同样，这任意一台PLC想发送数据。亦可在程序里面写入CANTX指令，(S1) (S2)的地址可以由用户自由定义（注意需遵循11位标识符和29位标识符的定义规则），只要接收方在CANRX指令中写入同样地址就可以接收到这个用户定义地址的数据。

指令举例三：（(S1)的bit13为1）



此时(S1)为H334F（二进制的11, 0011, 0100, 1111），(S2)为H200（0000, 0010, 0000, 0000）。(S1)的bit13为1，表示29位地址，此时的地址由(S2)（低16位地址）和(S1)的bit0~bit12（高13位地址）共同组成，即(S1)的11, 0011, 0100, 1111和(S2)共同组成了地址11, 0011, 0100, 1111, 0000, 0010, 0000, 0000（十六进制的H334F0200）。即是PLC往地址H334F0200发送了D100~D103寄存器里面的数据

在满足CANlink协议或远程扩展模块访问协议的设备中，通过扩展模块FROM/TO指令，可读写通过CAN 连接的远程扩展模块（需要扩展模块支持）和远程PLC。

在自由CAN协议中，不需分配PLC的站号，在CANlink协议或远程扩展模块访问协议的设备中需要分配各PLC或远程扩展模块的站号，详细使用请参考《5.9 CAN通讯指令说明》。

FNC90 FNC91
CANTX CANRX

16bit	32bit	P	FNC 92	RS2	串行数据传送2
✓					
11步			指令格式: RS2 (S) (m) (D) (n) (n1)		

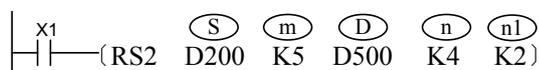
操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)														✓		
(m)					✓	✓								✓		
(D)														✓		
(n)					✓	✓								✓		
(n1)					✓	✓								✓		

该指令是一个通讯收发指令，将指定寄存器区域的数据，自动向串口依次发送，将串口接收到的数据存放到指定区域，相当于用户程序直接访问通讯缓冲区，借助用户程序对通讯收发缓冲区的处理，实现自定义协议的通讯。RS指令是固定COM1使用，而RS2指令可使用任何COM口，使用方法一样。其中：

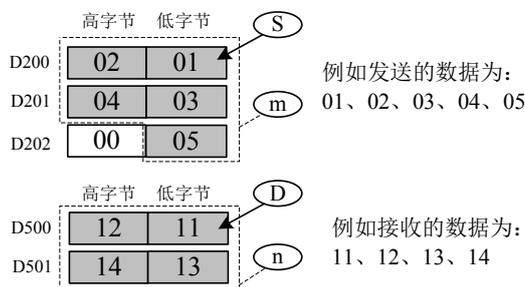
- (S) 为待发送数据存放的寄存器区的起始地址；
- (m) 为待发送数数据的长度（字节数），取值范围0~256；
- (D) 为通讯接收数据的存放寄存器区的起始地址；
- (n) 本通讯接收的数据长度（字节数），取值范围0~256。
- (n1) 为使用串口号：0：COM0；1：COM1；2：COM2；3：COM3；

RS2指令半双工/全双工模式由D8120的Bit10设定。用户程序可以写多条RS2指令，但是同一时间只能有一条RS2指令被驱动。在每次驱动RS2指令前，必须将M8122置位；

指令举例：



(n1) 设置为K2，RS2指令用于COM2通讯口
当X1为ON时，指令执行后通讯的收发数据存放如右图。



FNC92

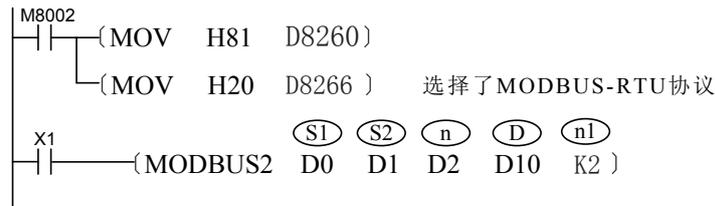
RS2

16bit	32bit	P	FNC 93	MODBUS2	数据传送2
✓					
11步			指令格式: MODBUS2 (S1) (S2) (n) (D) (n1)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓								✓		
(S2)					✓	✓								✓		
(n)					✓	✓								✓		
(D)														✓		
(n1)					✓	✓								✓		

为了直观方便，在标准MODBUS主站协议通讯中的RS2指令我们可以用MODBUS2来代替。MODBUS指令是固定COM1使用，而MODBUS 2指令可使用任何COM口，使用方法一样。各定义如下：

- (S1) 为从机地址（高字节）、通讯命令（低字节，按MODBUS协议定义）；
- (S2) 为访问从站的寄存器起始地址号；
- (n) 预读或写的数据长度；
- (D) 为读或写数据的存放单元起始地址，占用后续地址单元，长度由(n)决定。
- (n1) 为使用串口号：0: COM0；1: COM1；2: COM2；3: COM3；



(n1) 设置为K2，MODBUS2指令用于COM2通讯口。

FNC93
MODBUS2

4.3.10 浮点数 (110~147)

16bit	32bit		FNC 110	ECMP	二进制浮点数比较
	✓	✓			
	13步	13步	指令格式: ECMP (S1) (S2) (D)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✓	✓							✓		✓
(S2)					✓	✓							✓		✓
(D)		✓	✓	✓											

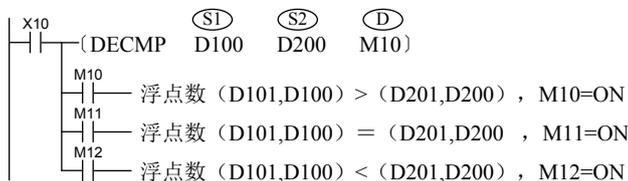
该指令是进行2个浮点数变量的比较，将比较的结果输出到 (D) 启始的3个变量中。其中：

(S1) 为待比较的二进制浮点数1；

(S2) 为待比较的二进制浮点数2；

(D) 为比较结果的存放单元，共占用3个（位）变量单元。

指令举例：



当X10=ON时，M10~M12其中之一会ON。

X10由ON变OFF时，不执行DECP指令，M10~M12仍保持X10=OFF之前的状态，要清除M10~M12的比较结果可用RST或者ZRST对M10~M12进行清除。

若需要得到≥、≤、≠的结果时，可将M10~M12串并联即可取得。

若 (S1) 或 (S2) 为K、H常数，系统会自动转换为浮点数参与运算。

FNC110

ECMP

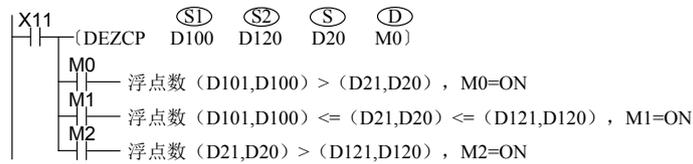
16bit	32bit	\square	FNC 111	EZCP	二进制浮点数区域比较
	✓	✓			
	17步	17步	指令格式: EZCP (S1) (S2) (S) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓								✓		
(S2)					✓	✓								✓		
(S)					✓	✓								✓		
(D)		✓	✓	✓												

该指令是进行二进制浮点数变量的区间比较，将比较的结果输出到 (D) 启始的3个变量中。其中：

- (S1) 为二进制浮点变量区间的下限；
- (S2) 为二进制浮点变量区间的上限；
- (S) 待比较的二进制浮点变量；
- (D) 为比较结果的存放单元，共占用3个（位）变量单元。

指令举例：



当X11=ON时，M0~M2其中之一会ON。
 当X11由ON变OFF时，不执行DEZCP指令，M0/M1/M2保持X11=OFF以前的状态不变。要清除M0~M2的比较结果可用RST或者ZRST对M0~M2进行清除。

FNC111

EZCP

16bit	32bit	P	FNC 112	EMOV	2进制浮点数数据移动
	✓	✓			
	9步	9步	指令格式: EMOV (S) (D)		

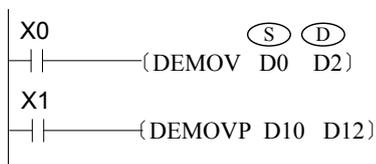
操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)													✓	✓	✓
(D)													✓	✓	✓

该指令是进行二进制浮点数数据的传送。需要触点驱动，当指令执行时，将 (S) 的2进制浮点数数据值复制到 (D) 中。其中：

(S) 为二进制浮点数数据的传送源；

(D) 为保存二进制浮点数数据的存放单元。

指令举例：



假设(D1、D0)的2进制浮点数数值为12.345，当X0为ON时，(D3、D2)的2进制浮点数数值变为12.345。当M0由ON \Rightarrow OFF时，(D3、D2)保持12.345的内容不变，除非用户程序再次将(D3、D2)的值修改。或者PLC由STOP \Rightarrow RUN和PLC重新上电，(D3、D2)的值才会变化，停电保持寄存器上电或者由停止到运行保持原来的值不变。

16bit	32bit		FNC 118	EBCD	二进制浮点数→十进制浮点转换
	✓	✓			
	9步	9步	指令格式: EBCD (S) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)														✓		
(D)														✓		

该指令是进行二进制浮点数转换为十进制浮点的运算。其中：

- (S) 为二进制浮点变量；
- (D) 为转换为十进制浮点数结果的存放单元。

指令举例：



将二进制浮点数 (D3,D2) 转换成十进制浮点数后, 存放于 (D11,D10) 单元。

其中2 进制浮点数 [D3, D2] 实数23 位, 指数 8 位, 符号位 1 位
10 进制浮点数 [D 11, D 10] 指数 (D3) 实数 (D2), 用科学计算式表示为 $D2 \times 10^{D3}$

PLC内部浮点数据计算均为二进制形式, 转换为十进制, 可方便监控。

16bit	32bit	\square	FNC 119	EBIN	十进制浮点数→二进制浮点转换
	✓	✓			
	9步	9步	指令格式: EBIN \textcircled{S} \textcircled{D}		

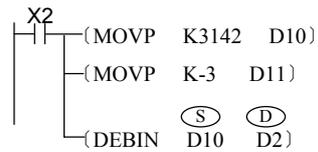
操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
\textcircled{S}														✓		
\textcircled{D}														✓		

该指令是进行十进制浮点数转换为二进制浮点的运算。其中：

\textcircled{S} 为十进制浮点变量；

\textcircled{D} 为转换为二进制浮点数结果的存放单元。

指令举例：



将十进制浮点数3.142（先放于D11，D10）转换成二进制浮点数后，存放于（D3，D2）单元。

16bit	32bit	\square	FNC 120	EADD	二进制浮点加法运算
	✓	✓			
	13步	13步	指令格式: EADD (S1) (S2) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓							✓			
(S2)					✓	✓							✓			
(D)													✓			

该指令是进行二进制浮点的加法运算。其中：

(S1) 和 (S2) 分别为二进制浮点的被加数和加数；

(D) 为二进制浮点加法和的存放单元。

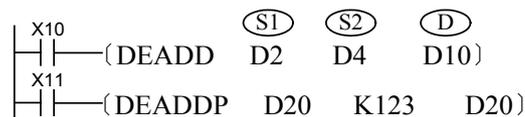
S1或S2来源操作数若是常数K或H，会自动将该常数转换成二进制浮点数值来作加法运算；

若计算结果为零，则0标志 (M8020) 会置位。

若运算结果的绝对值大于可表示的最大浮点值，则进位标志 (M8022) 会置位。

若运算结果的绝对值小于可表示的最小浮点值，则借位标志 (M8021) 会置位。

指令举例：



当X10=ON时，二进制浮点数 (D3, D2) 与二进制浮点数 (D5, D4) 相加后，二进制浮点数和存放于 (D11, D10) ；

当X11由OFF变为ON时，二进制浮点数 (D21, D20) 的值增大123。这里的常数K123在运算前已自动被调整为二进制浮点数；

和的存放单元可以与加数或被加数为同一单元，此时请使用脉冲执行型指令DEADDP，否则若采用连续执行指令，则程序每扫描一次，计算就会被执行一次。

16bit	32bit	\square	FNC 121	ESUB	二进制浮点减法运算
	✓	✓			
	13步	13步	指令格式: ESUB (S1) (S2) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓								✓		
(S2)					✓	✓								✓		
(D)														✓		

该指令是进行二进制浮点的减法运算。其中：

(S1) 和 (S2) 分别为二进制浮点的被减数和减数；

(D) 为二进制浮点减法差的存放单元。

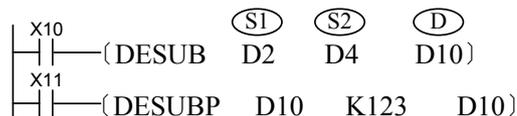
S1或S2来源操作数若是常数K或H，会自动将该常数变换成二进制浮点数值来作减法运算；

若计算结果为零，则0标志 (M8020) 会置位。

若运算结果的绝对值大于可表示的最大浮点值，则进位标志 (M8022) 会置位。

若运算结果的绝对值小于可表示的最小浮点值，则借位标志 (M8021) 会置位。

指令举例：



当X10=ON时，二进制浮点数 (D3, D2) 减去二进制浮点数 (D5, D4) 后，二进制浮点数差存放于 (D11, D10) ；

当X11由OFF变为ON时，二进制浮点数 (D11, D10) 的值减小123。这里的常数K123在运算前已自动被调整为二进制浮点数；

差的存放单元可以与减数或被减数为同一单元，此时请使用脉冲执行型指令DESUBP，否则若采用连续执行指令，则程序每扫描一次，计算就会被执行一次。

FNC121

ESUB

16bit	32bit		FNC 122	EMUL	二进制浮点乘法运算
	✓	✓			
	13步	13步	指令格式: EMUL (S1) (S2) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓								✓		
(S2)					✓	✓								✓		
(D)														✓		

该指令是进行二进制浮点的乘法运算。其中：

(S1) 和 (S2) 分别为二进制浮点的被乘数和乘数；

(D) 为二进制浮点乘法积的存放单元。

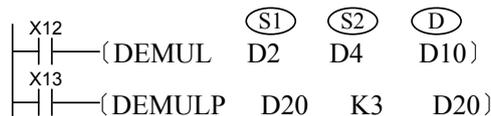
S1或S2来源操作数若是常数K或H，会自动将该常数转换成二进制浮点数值来作乘法运算；

若计算结果为零，则0标志（M8020）会置位。

若运算结果的绝对值大于可表示的最大浮点值，则进位标志（M8022）会置位。

若运算结果的绝对值小于可表示的最小浮点值，则借位标志（M8021）会置位。

指令举例：



当X12=ON时，二进制浮点数（D3，D2）乘以二进制浮点数（D5，D4）后，二进制浮点数积存放于（D11，D10）；

当X13由OFF变为ON时，二进制浮点数（D21，D20）的值乘以3倍后存回（D21，D20）。这里的常数K3在运算前已自动被调整为二进制浮点数；

积的存放单元可以与乘数或被乘数为同一单元，此时请使用脉冲执行型指令DEMULP，否则若采用连续执行指令，则程序每扫描一次，计算就会被执行一次。

FNC122

EMUL

16bit	32bit	\square	FNC 123	EDIV	二进制浮点除法运算
	✓	✓			
	13步	13步	指令格式: EDIV (S1) (S2) (D)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✓	✓							✓		
(S2)					✓	✓							✓		
(D)													✓		

该指令是进行二进制浮点的除法运算。其中:

(S1) 和 (S2) 分别为二进制浮点的被除数和除数;

(D) 为二进制浮点除法商的存放单元起始地址。

S1或S2来源操作数若是常数K或H, 会自动将该常数转换成二进制浮点数值来作除法运算;

若计算结果为零, 则0标志 (M8020) 会置位。

若运算结果的绝对值大于可表示的最大浮点值, 则进位标志 (M8022) 会置位。

若运算结果的绝对值小于可表示的最小浮点值, 则借位标志 (M8021) 会置位。

除数不得为0, 否则计算出错, M8067、M8068会置ON。

指令举例:

$\begin{array}{l} \text{X14} \\ | | \text{---} (\text{DEDIV } \text{D2} \text{ } \text{D4} \text{ } \text{D10}) \\ \text{X15} \\ | | \text{---} (\text{DEDIVP } \text{D10} \text{ } \text{K10} \text{ } \text{D10}) \end{array}$

当X14=ON时, 二进制浮点数 (D3, D2) 除以二进制浮点数 (D5, D4) 后, 二进制浮点数商存放于 (D11, D10);

当X15由OFF变为ON时, 二进制浮点数 (D11, D10) 的值除以10后存回 (D11, D10)。这里的常数K10在运算前已自动被调整为二进制浮点数;

商的存放单元可以与除数或被除数为同一单元, 此时请使用脉冲执行型指令DEDIVP, 否则若采用连续执行指令, 则程序每扫描一次, 计算就会被执行一次。

FNC123

EDIV

16bit	32bit	P	FNC 124	EXP	2进制浮点数指数运算
	✓	✓			
	9步	9步	指令格式: EXP (S) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)														✓	✓	✓
(D)														✓	✓	✓

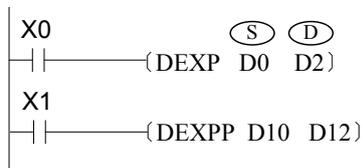
该指令是进行以e(2.71828)为底的二进制浮点数数据的指数运算。其中：

(S) 为待求二进制浮点数指数的2进制浮点数变量；

(D) 为指数运算后计算结果的存储单元。

注意：当运算结果不在 []时将发生运算出错。错误代码为K6706，K6706保存在D8067中，出错标志位M8067置ON。

指令举例： $2^{-126} \leq \text{运算结果} < 2^{128}$



当X0为ON时，以e为底对(D1、D0)中的2进制浮点数数值进行指数运算后保存到(D3、D2)中。 $e^{(D1、D0)} \Rightarrow (D3、D2)$

由于 $\log_e 2^{128} = 88.7$ ，因此当(D1、D0)大于88.7时。D8067为

K6706，M8067置ON

16bit	32bit	P	FNC 125	LOGE	2进制浮点数自然对数运算
	✓	✓			
	9步	9步	指令格式: LOGE (S) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)														✓	✓	✓
(D)														✓	✓	✓

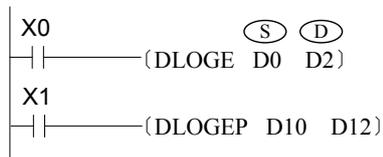
该指令是进行以e(2.71828)为底的二进制浮点数数据的自然对数运算。其中:

(S) 为待求二进制浮点数自然对数的2进制浮点数变量;

(D) 为自然对数运算后计算结果的存储单元。

注意: (S) 中的值只能为正数, 当 (S) 的内容值为0或者负数时将发生运算出错。错误代码为K6706, K6706会保存在D8067中, 出错标志位M8067置ON。

指令举例:



当X0为ON时, 以e为底对(D1、D0)中的2进制浮点数数值进行自然对数运算后保存到(D3、D2)中。

$$\log_e^{(D1, D0)} \Rightarrow (D3, D2)$$

从自然对数向常用对数的转换 (用0.4342945分割常用对数的值) 公式如下:

$$10^x = e^{\frac{x}{0.4342945}}$$

FNC125

LOGE

16bit	32bit	P	FNC 126	DLOG	2进制浮点数常用对数运算
	✓	✓			
	9步	9步	指令格式: DLOG (S) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)													✓	✓	✓	
(D)													✓	✓	✓	

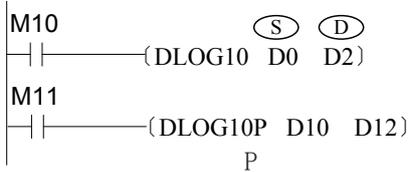
该指令是进行以10为底的二进制浮点数数据的常用对数运算。其中：

(S) 为待求二进制浮点数常用对数的2进制浮点数变量；

(D) 为常用对数运算后计算结果的存储单元。

注意：(S) 中的值只能为正数，当 (S) 的内容值为0或者负数时将发生运算出错。错误代码为K6706，K6706会保存在D8067中，出错标志位M8067置ON。

指令举例：



当M10为ON时，以10为底对(D1、D0)中的2进制浮点数数值进行常用对数运算后保存到(D3、D2)中。

$$\log_{10}(D1、D0) \Rightarrow (D3、D2)$$

16bit	32bit	\square	FNC 127	ESQR	二进制浮点开平方运算
	✓	✓			
	9步	9步	指令格式: ESQR (S) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)					✓	✓								✓		
(D)														✓		

该指令是进行二进制浮点的开平方运算，即求二进制浮点数的平方根。其中：

(S) 为待求平方根的二进制浮点数变量；

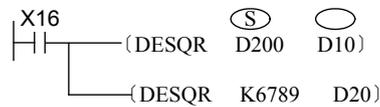
(D) 为二进制浮点平方根的存储单元。

操作数 (S) 若是常数K或H，会自动将该常数转换成二进制浮点数值来作开方运算；

若计算结果为零，则0标志 (M8020) 会置位。

(S) 只有正数有效，如果是负数则计算出错，M8067、M8068会置ON。

指令举例：



将二进制浮点数开方结果 $\sqrt{(D201,D200)}$ 存放到 $\rightarrow (D11,D10)$

将二进制浮点数K6789做开方，结果存放到 $\rightarrow (D21,D20)$ ，

这里的常数K6789在运算前已自动被调整为二进制浮点数；

16bit	32bit		FNC 129	INT	二进制浮点→BIN整数变换
✓	✓	✓			
5步	9步		指令格式: INT (S) (D)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)													✓		
(D)													✓		

该指令是进行二进制浮点的取整运算，丢弃小数部分，将二进制结果存于 (D) 中。其中：

(S) 为待取整变换的二进制浮点数变量；

(D) 为变换后BIN整数结果的存储单元。

若计算结果为零，则0标志 (M8020) 会置位。

若运算结果有浮点数被舍弃时，则借位标志 (M8021) 会置位。

若运算结果若超出下列范围时 (溢位)，则进位标志 (M8022) 会置位。

16位指令: -32,768~32,767

32位指令: -2,147,483,648~2, 147, 483, 647

指令举例：



将浮点数(D51,D50)取整后，存放到→ (D100)

将浮点数(D11,D10)取整后，存放到→ (D21,D20)

注意INT和DINT指令存放结果的区别

FNC129

INT

16bit	32bit	\square	FNC 130	SIN	浮点SIN运算
	✓	✓			
	9步	9步	指令格式: SIN (S) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)														✓		
(D)														✓		

该指令是求指定角度 (RAD, 弧度) 的SIN (正弦) 值, 变量为二进制浮点存储格式。其中:

(S) 为待求正弦值的角度变量, RAD单位, 以二进制浮点数表示。取值范围 $0 \leq \alpha \leq 2\pi$;

(D) 为变换后SIN计算结果的存储单元, 二进制浮点数格式。

指令举例一:

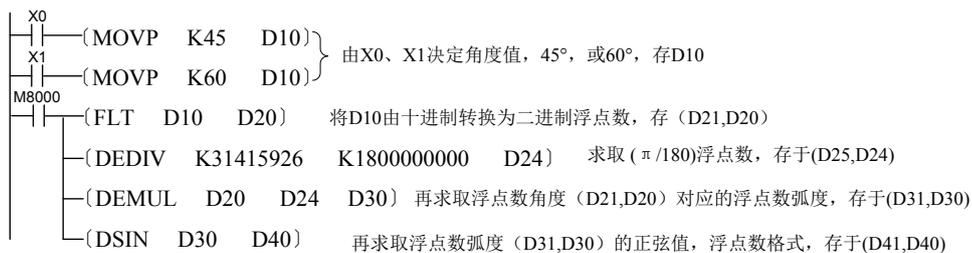


这里计算的源数据、SIN结果都为二进制浮点数格式。

RAD(弧度)值 = 角度 $\times \pi / 180^\circ$, 如角度 360° 对应的弧度 = $360^\circ \times \pi / 180^\circ = 2\pi$ 。

指令举例二:

根据角度值求对应SIN值的程序:



FNC130

SIN

16bit	32bit	\square	FNC 131	COS	浮点COS运算
	✓	✓			
	9步	9步	指令格式: COS (S) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)													✓			
(D)													✓			

该指令是求指定角度 (RAD, 弧度) 的COS (余弦) 值, 变量为二进制浮点存储格式。其中:

(S) 为待求余弦值的角度变量, RAD单位, 以二进制浮点数表示。取值范围 $0 \leq \alpha \leq 2\pi$;

(D) 为变换后COS计算结果的存储单元, 二进制浮点数格式。

指令举例:

$\begin{array}{|c|} \hline X2 \\ \hline \end{array}$ (DTAN (S) (D)) 将弧度(D21,D20)求TAN值, 存放到 \rightarrow (D31,D30)

这里计算的源数据、COS结果都为二进制浮点数格式。

RAD(弧度)值 = 角度 $\times \pi / 180^\circ$, 如角度 360° 对应的弧度 = $360^\circ \times \pi / 180^\circ = 2\pi$ 。

关于以角度求取COS值的编程语句, 可参考SIN指令中的举例。

16bit	32bit	\square	FNC 132	TAN	浮点TAN运算
	✓	✓			
	9步	9步	指令格式: TAN (S) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)														✓		
(D)														✓		

该指令是求指定角度 (RAD, 弧度) 的TAN (正切) 值, 变量为二进制浮点存储格式。其中:

(S) 为待求正切值的角度变量, RAD单位, 以二进制浮点数表示。取值范围 $0 \leq \alpha < 2\pi$;

(D) 为变换后TAN计算结果的存储单元, 二进制浮点数格式。

指令举例:

$\begin{array}{|c|} \hline X2 \\ \hline \end{array} \text{---} (\text{DTAN } \text{(S)} \text{ (D)})$ 将弧度(D21,D20)求TAN值, 存放到 \rightarrow (D31,D30)

这里计算的源数据、TAN结果都为二进制浮点数格式。

RAD(弧度)值 = 角度 $\times \pi / 180^\circ$, 如角度 360° 对应的弧度 = $360^\circ \times \pi / 180^\circ = 2\pi$ 。

关于以角度求取TAN值的编程语句, 可参考SIN指令中的举例。

FNC132

TAN

16bit	32bit	P	FNC 133	ASIN	2进制浮点数ASIN运算
	✓	✓			
	9步	9步	指令格式: ASIN (S) (D)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)													✓	✓	✓
(D)													✓	✓	✓

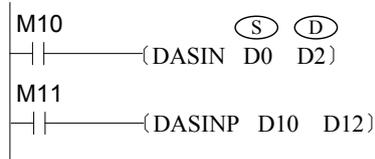
该指令是从SIN值求出对应弧度的运算。其中:

(S) 为存放待求 SIN^{-1} (反正弦) 的2进制浮点数变量;

(D) 为计算结果的存储单元(-π/2~π/2)。

注意: (S) 中的值不在-1.0~1.0的范围内时将发生运算出错。错误代码为K6706, K6706会保存在D8067中, 出错标志位M8067置ON。

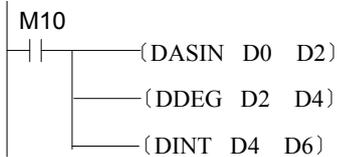
指令举例一:



当M10为ON时, 将(D1、D0)中的2进制浮点数数值进行 SIN^{-1} 运算后保存到(D3、D2)中。

$$\text{SIN}^{-1}(\text{D1、D0}) \Rightarrow (\text{D3、D2})$$

指令举例二:



假设(D1、D0)为0.707106781, 当M10由OFF到ON的时候, (D3、D2) 内容为0.78539815, (D5、D4) 的内容为45, (D7、D6) 的内容为45。

浮点数

16bit	32bit	P	FNC 134	ACOS	2进制浮点数ACOS运算
	✓	✓			
	9步	9步	指令格式: ACOS (S) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)														✓	✓	✓
(D)														✓	✓	✓

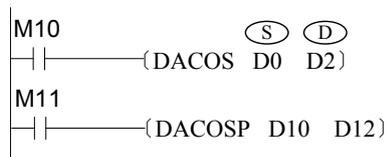
该指令是从COS值求出对应弧度的运算。其中:

(S) 为存放待求 COS^{-1} (反余弦) 的2进制浮点数变量;

(D) 为计算结果的存储单元(0~ π)。

注意: (S) 中的值不在-1.0~1.0的范围内时将发生运算出错。错误代码为K6706, K6706会保存在D8067中, 出错标志位M8067置ON。

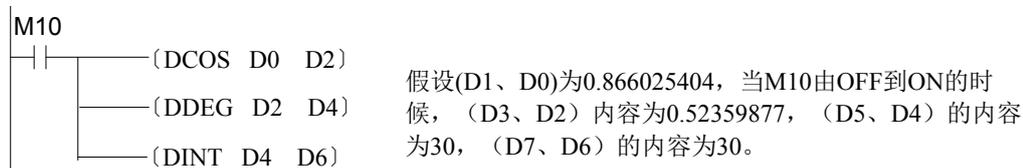
指令举例一:



当M10为ON时, 将(D1、D0)中的2进制浮点数数值进行 COS^{-1} 运算后保存到(D3、D2)中。

$$\text{COS}^{-1}(\text{D1、D0}) \Rightarrow (\text{D3、D2})$$

指令举例二:



FNC134

ACOS

16bit	32bit	P	FNC 135	ATAN	2进制浮点数ATAN运算
	✓	✓			
	9步	9步	指令格式: ATAN (S) (D)		

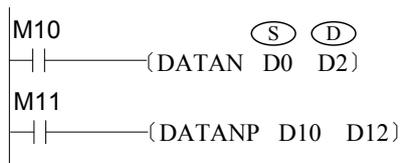
操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)													✓	✓	✓	
(D)													✓	✓	✓	

该指令是从TAN值求出对应弧度的运算。其中:

(S) 为存放待求 TAN^{-1} (反正切) 的2进制浮点数变量;

(D) 为计算结果的存储单元(- $\pi/2 \sim \pi/2$)。

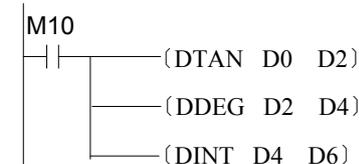
指令举例一:



当M10为ON时，将(D1、D0)中的2进制浮点数数值进行 TAN^{-1} 运算后保存到(D3、D2)中。

$$\text{TAN}^{-1}(D1, D0) \Rightarrow (D3, D2)$$

指令举例二:



假设(D1、D0)为1.732050808，当M10由OFF到ON的时候，(D3、D2)内容为1.04719753，(D5、D4)的内容为60，(D7、D6)的内容为60。

浮点数

16bit	32bit	P	FNC 136	RAD	2进制浮点数角度→弧度
	✓	✓			
	9步	9步	指令格式: ATAN (S) (D)		

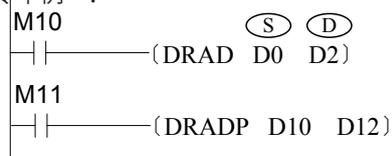
操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)														✓	✓	✓
(D)														✓	✓	✓

该指令是进行2进制浮点数角度转换成弧度的运算。其计算公式为[弧度单位=角度单位 $\times \pi/180$]，其中：

(S) 为存放待求弧度的2进制浮点数角度变量；

(D) 为计算结果的存储单元。

指令举例一：



当M10为ON时，将(D1、D0)中的2进制浮点数数值进行角度到弧度的运算后保存到(D3、D2)中。

指令举例二：



当M10由OFF到ON的时候，将90赋给D0，D0经过整数到浮点数的运算赋给(D3、D2)，(D3、D2)经过角度到弧度的运算赋给(D5、D4)，最后得(D3、D2)的值为 $\pi/2$ ，即为1.570796。

FNC136

RAD

16bit	32bit	P	FNC 137	DEG	2进制浮点数弧度→角度
	✓	✓			
	9步	9步	指令格式: DEG (S) (D)		

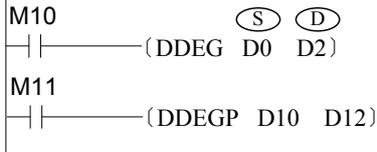
操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)													✓	✓	✓
(D)													✓	✓	✓

该指令是进行2进制浮点数弧度转换成角度的运算。其计算公式为[角度单位=弧度单位×π/180]，其中：

(S) 为存放待求角度的2进制浮点数弧度变量；

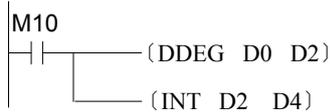
(D) 为计算结果的存储单元。

指令举例一：



当M10为ON时，将(D1、D0)中的2进制浮点数数值进行弧度到角度的运算后保存到(D3、D2)中。

指令举例二：



假设 (D1、D0) 为3.1415926，当M10由OFF到ON的时候，(D3、D2) 为180，经过浮点数变整数的运算后得到 (D5、D4) 的值为180。

FNC137
DEG

浮点数			FNC 438	SINH	2进制浮点数SINH运算
16bit	32bit	P			
	✓	✓			
	9步	9步	指令格式: SINH (S) (D)		

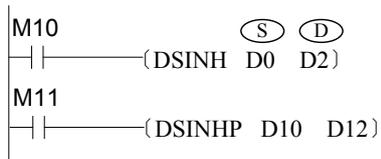
操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)														✓	✓	✓
(D)														✓	✓	✓

该指令是进行2进制浮点数取SINH值。其计算公式为 $\sinh值 = (e^S - e^{-S}) / 2$ ，其中：

(S) 为存放待求SINH值的2进制浮点数变量；

(D) 为计算结果的存储单元。

指令举例：



当M10为ON时，将(D1、D0)中的2进制浮点数数值求取SINH值后保存到(D3、D2)中。

FNC438

SINH

16bit	32bit	P	FNC 439	COSH	2进制浮点数COSH运算
	✓	✓			
	9步	9步	指令格式: COSH (S) (D)		

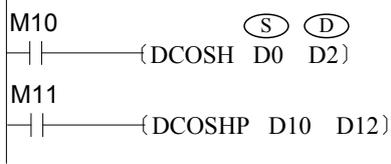
操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)													✓	✓	✓	
(D)													✓	✓	✓	

该指令是进行2进制浮点数取COSH值。其计算公式为 $\cosh值 = (e^S + e^{-S}) / 2$ ，其中：

(S) 为存放待求COSH值的2进制浮点数变量；

(D) 为计算结果的存储单元。

指令举例：



当M10为ON时，将(D1、D0)中的2进制浮点数数值求取COSH值后保存到(D3、D2)中。

16bit	32bit	P	FNC 440	TANH	2进制浮点数TANH运算
	✓	✓			
	9步	9步	指令格式: TANH (S) (D)		

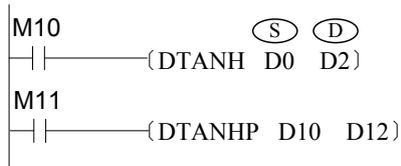
操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)														✓	✓	✓
(D)														✓	✓	✓

该指令是进行2进制浮点数取COSH值。其计算公式为 \tanh 值= $(e^S - e^{-S}) / (e^S + e^{-S})$ ，其中：

(S) 为存放待求TANH值的2进制浮点数变量；

(D) 为计算结果的存储单元。

指令举例：



当M10为ON时，将(D1、D0)中的2进制浮点数数值求取TANH值后保存到(D3、D2)中。

16bit	32bit		FNC 147	SWAP	高低字节交换
✓	✓	✓			
3步	5步		指令格式: SWAP (S)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)								✓	✓	✓	✓	✓	✓	✓	✓

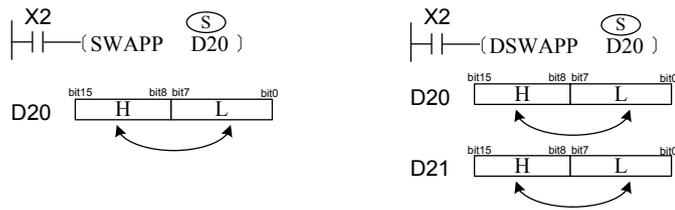
该指令是将指定变量 (S) 的高低字节的值进行互相交换。

16 位指令时，高8 位与低8 位的值进行互相交换。

32 位指令时，两个寄存器的高8 位与低8 位的值各自进行互相交换

注意此指令一般使用脉冲执行型指令，否则若采用连续执行指令，则程序每扫描一次，就会进行一次交换。

指令举例：



左图中将D20 的高8 位与低8 位的值进行互相交换
 右图中将D20 的高8 位与低8 位的值进行互相交换，
 D21 的高8 位与低8 位的值进行互相交换，

FNC147
SWAP

4.3.11 定位控制 (150~159)

16bit	32bit	P	FNC 150	DSZR	带DOG搜索的原点回归
✓	✓				
9步	17步		指令格式: DSZR (S1) (S2) (D1) (D2)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)	✓	✓	✓	✓											
(S2)	✓														
(D1)		✓													
(D2)		✓	✓												

该指令是PLC与伺服驱动器配合工作时，以特殊数据寄存器指定的脉冲速度从脉冲输出端口输出脉冲，使执行机构按照预先设定的动作顺序向动作原点移动，运行中近点信号（DOG）ON到OFF后，检测到零点信号从OFF到ON，则立即停止脉冲输出。

回归运动过程中输出旋转方向信号，回归结束输出清零信号。

在设定有正反转限位的系统中，可以通过启用DOG搜索模式回归原点；在未设定正反转限位和不使用正反转限位进行原点回归的系统中，可以通过指定原点回归方向进行原点回归。其中：

(S1) 为近点信号（DOG）输入。虽然XYMS信号都可以，但只有X信号的及时性最好；

(S2) 为零点信号输入。表示动作原点的确切位置，只能指定X信号；

(D1) 为脉冲输出端口。只能指定Y0/Y1/Y2；

(D2) 为旋转方向输出端口。指定YM信号。ON：正转（脉冲输出使当前值增加）；OFF：反转（脉冲输出使当前值减少）。

注：上述位软元件所使用M信号，范围为M0~M1023。

相关软元件一览表：

1) 特殊辅助继电器：

软元件			名称	属性
Y0	Y1	Y2		
M8029			指令执行结束标志位	读
M8429			指令执行异常结束标志位	读
M8440※1	M8450	M8460	脉冲输出中监控	读
M8441※1	M8451	M8461	清零信号输出功能有效※2	读写
M8442	M8452	M8462	原点回归方向指定※2	读写
M8443	M8453	M8463	正转极限	读写
M8444	M8454	M8464	反转极限	读写
M8445	M8455	M8465	近点信号逻辑反转※2	读写
M8446	M8456	M8466	零点信号逻辑反转※2	读写
M8448	M8458	M8468	定位指令驱动中	读
M8449	M8459	M8469	脉冲停止指令※2	读写
M8484	M8485	M8486	清零信号软元件指定功能有效※2	读写

注：※1：本端口和相关软元件保留。

※2：RUN→STOP，清除。

2) 特殊数据寄存器：

软元件						名称	默认值	取值范围※2
Y0		Y1		Y2				
D8440※1	低位	D8450	低位	D8460	低位	当前值寄存器（Hz）（32位）	0	/
D8441※1	高位	D8451	高位	D8461	高位			
D8442		D8452		D8462		基底速度（Hz）（16位）	200	10~32767
D8443	低位	D8453	低位	D8463	低位	最高速度（Hz）（32位）	100,000	715~100,000
D8444	高位	D8454	高位	D8464	高位			
D8445		D8455		D8465		爬行速度（Hz）（16位）	1000	10~32767

FNC150

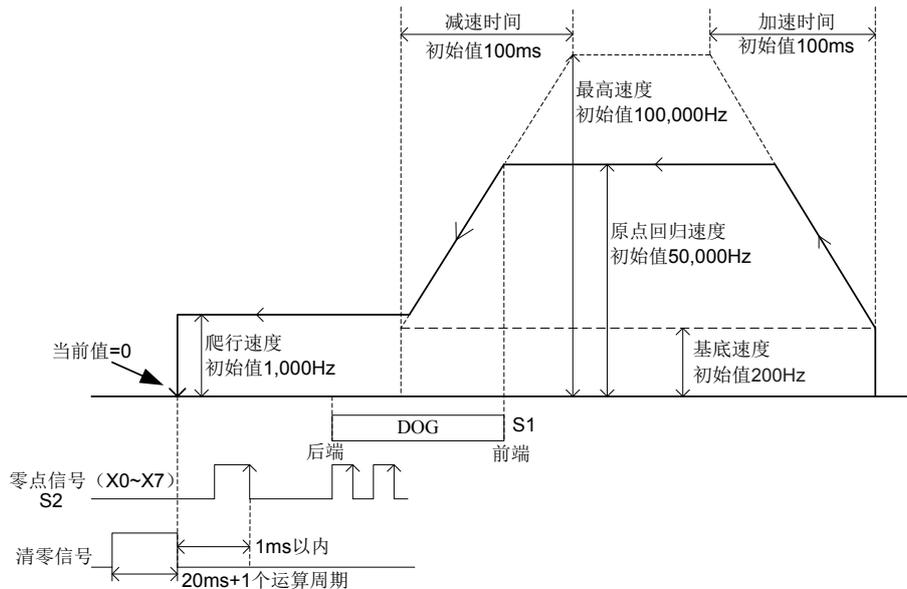
DSZR

软元件						名称	默认值	取值范围※2
Y0		Y1		Y2				
D8446	低位	D8456	低位	D8466	低位	原点回归速度 (Hz) (32位)	50,000	715~100,000
D8447	高位	D8457	高位	D8467	高位			
D8448		D8458		D8468		加速时间 (ms) (16位)	100	50~5000
D8449		D8459		D8469		减速时间 (ms) (16位)	100	50~5000
D8484		D8485		D8486		清零信号软元件指定	/	/

注：※1：本端口和相关软元件保留。

※2：一般要求“爬行速度≥基底速度”，出现爬行速度<0、爬行速度<基底速度、回归速度<基底速度、最大速度<爬行速度，均会报错【报元件超限 (6706)】；其他速度大小关系时，系统会自动调整至其靠近的边界值。

参数详析



3) 近点和零点信号逻辑反转:

(D1) 脉冲输出端	(S1) 指定的近点信号 (DOG) 逻辑 (近点信号逻辑反转标志位)	(S2) 指定的零点信号逻辑 (零点信号逻辑反转标志位)	说明
Y0	M8445	M8446	OFF: 正逻辑 (输入为ON时, 近点/零点信号为有效); ON: 负逻辑 (输入为OFF时, 近点/零点信号为有效)。
Y1	M8455	M8456	
Y2	M8465	M8466	

4) 清零信号的输出:

脉冲输出端	清零信号输出有效标志位	清零信号软元件指定功能有效标志位	清零软元件指定用软元件	清零软元件指定用软元件默认值	FNC150
Y0	M8441	M8484	D8484	D8484=4 (Y4)	DSZR
Y1	M8451	M8485	D8485	D8485=5 (Y5)	
Y2	M8461	M8486	D8486	D8486=6 (Y6)	

注:

- a) 原点回归结束后, 欲输出清零信号, 请使清零信号输出有效标志位=ON;
- b) 如果清零信号软元件指定功能有效标志位=OFF, 清零信号使用默认的Y输出口, 否则, 使用清零软元件指定用软元件指定的Y端口输出清零信号

5) 速度量的控制:

最高速度、原点回归速度、爬行速度、基底速度, 请参考特殊数据寄存器中的定义, 请遵循:

基底速度≤原点回归速度≤最高速度;

基底速度≤爬行速度≤最高速度;

请参照设定的参数范围选择合理的参数, 建议爬行速度≤原点回归速度!

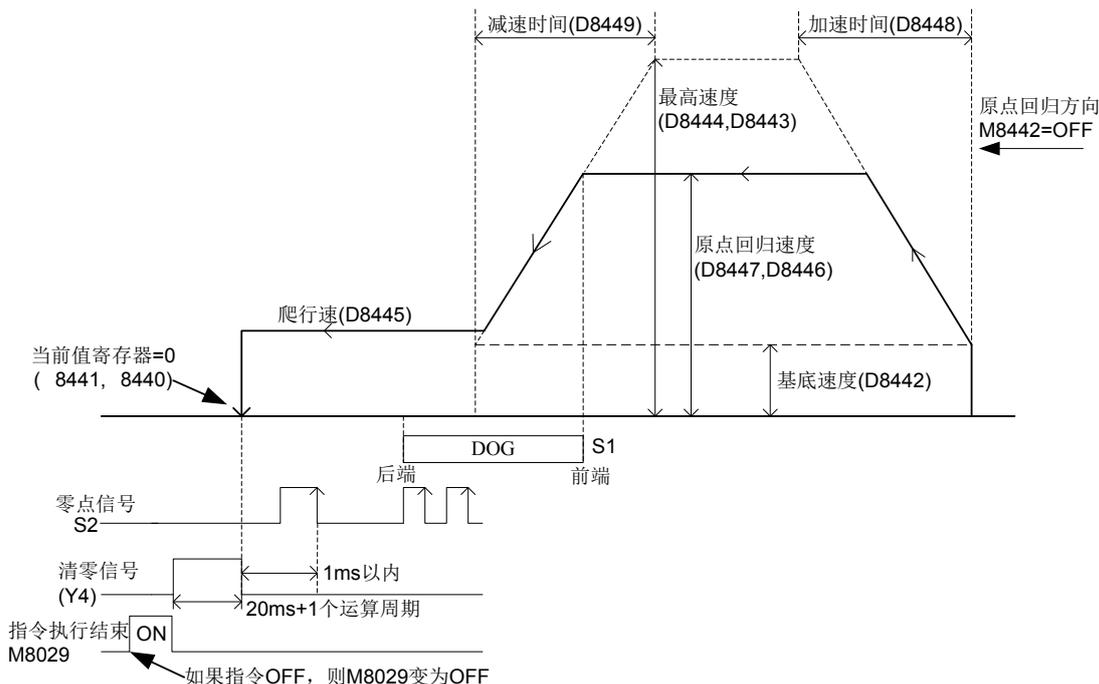
注: 基底速度设置过低, 在启动加减速初期 (前几个输出脉冲) 输出频率跳变剧烈, 且基底速度越低, 跳变越剧烈, 建议基底速度设置>50 (加速过程平滑)。

原点回归

指令举例:

```
——| |——(DSZR X7 X5 Y0 Y12)
```

以脉冲输出端口Y0为例说明, 使用其他端口注意修改相应的软元件。



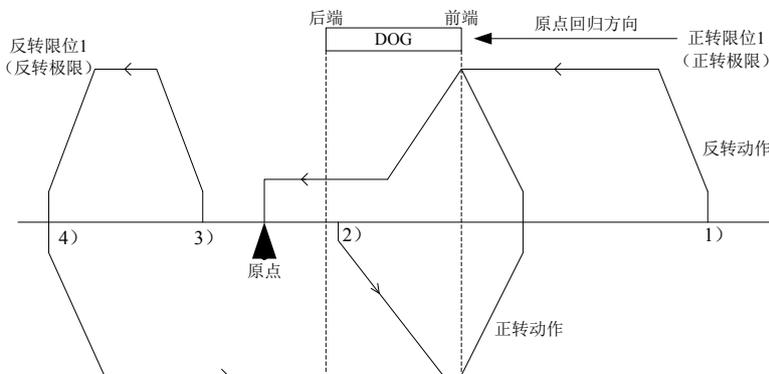
- 6) 指定原点回归方向: 通过原点回归方向指定标志位 (M8442) 的ON/OFF来指定回归原点方向。
- 7) 基底速度 (D8442) 加速, 直到原点回归速度 (D8447, D8446), 向原点回归方向指定标志位 (M8442) 指定的方向移动。
- 8) 测到 (S1) 指定的近点信号 (DOG) 为ON, 开始减速, 直到爬行速度 (D8445)。
- 9) (S1) 指定的近点信号 (DOG) 从ON到OFF后, 如果检测到 (S2) 指定的零点信号从OFF到ON, 立即停止脉冲的输出。
注: 如果近点信号和零点信号指定了相同的输入, 那么和ZRN指令一样, 不使用零点信号, 在近点信号从ON到OFF后, 立即停止脉冲的输出; 如果近点信号和零点信号指定了相同的输入, 那么如果有逻辑反转标志位 (M8445 或者M8446) 为ON, 以近点信号的逻辑 (M8445) 为准。
- 10) 清零信号输出功能 (M8441) 有效 (ON) 时, 在脉冲输出停止后 (1ms以内), 清零信号 (Y4) 在[20ms+1个运算周期]的时间内保持为ON。
- 11) 指令执行结束标志位 (M8029) 为ON, 结束原点回归动作。

注: 这是近点信号逻辑反转标志位 (M8445) 和零点信号逻辑反转标志位 (M8446) 为OFF的说明。如果该逻辑标志位为ON, 对应的近点和零点信号的ON改成OFF, OFF改成ON。

DOG搜索功能

设计有正转限位、反转限位时, 执行使用了DOG搜索功能的原点回归。

此时, 因原点回归的开始位置不同, 原定回归动作也各不同。



- 1) 开始位置在通过DOG前的时候, 包括正转限位1为ON的时候:
 - a) 通过执行原点回归用指令, 开始原点回归动作。
 - b) 以原点回归速度, 向原点回归方向开始移动。
 - c) 一旦检测到DOG的前端, 就开始减速到爬行速度。

FNC150
DSZR

- d) 检测出DOG的后端后，在检测出第一个零点信号时停止。
- 2) 开始位置在DOG内的时候：
 - a) 通过执行原点回归用指令，开始原点回归动作。
 - b) 以原点回归速度，向与原点回归方向相反的方向开始移动。
 - c) 检测出DOG的前端后减速停止。（离开DOG）
 - d) 以原点回归速度，向原点回归方向开始移动。（再次进入DOG）
 - e) 一旦检测出DOG的前端，就开始减速到爬行速度。
 - f) 检测出DOG的后端后，在检测出第一个零点信号时停止。
- 3) 开始位置在近点信号OFF（通过DOG后）的时候：
 - a) 通过执行原点回归用指令，开始原点回归动作。
 - b) 以原点回归速度，向原点回归方向开始移动。
 - c) 检测出反转限位1（反转极限）时减速停止。
 - d) 以原点回归速度，向与原点回归方向相反的方向开始移动。
 - e) 检测出DOG的前端后减速停止。（检出（离开）DOG）
 - f) 以原点回归速度，向原点回归方向开始移动。（再次进入DOG）
 - g) 一旦检测出DOG的前端，就开始减速到爬行速度。
 - h) 检测出DOG的后端后，在检测出第一个零点信号时停止。
- 4) 原点回归方向的限位开关（反转限位1）为ON的时候：
 - a) 通过执行原点回归用指令，开始原点回归动作。
 - b) 以原点回归速度，向与原点回归方向相反的方向开始移动。
 - c) 检测出DOG的前端后减速停止。（检出（离开）DOG）
 - d) 以原点回归速度，向原点回归方向开始移动。（再次进入DOG）
 - e) 一旦检测出DOG的前端，就开始减速到爬行速度。
 - f) 检测出DOG的后端后，在检测出第一个零点信号时停止。

注：设计近点信号（DOG）时，要考虑有足够的ON时间减速到爬行速度；请使爬行速度尽可能的慢，因为其停止没有减速，爬行速度过快，可能会导致位置偏移。

注意事项：

定位指令（ZRN/PLSV/DRVI/DRVA）在程序中可多次使用，但不要对同一高速Y输出口同时进行输出操作；

当指令的驱动能流OFF之后，再次驱动ON时，必需在状态位（Y000: [M8147], Y001: [M8148]）OFF后，经过一个运算周期后，方能再次驱动。

定位指令再次驱动时，必须有 1 个周期以上的 OFF 时间，若以比上述条件更短的时间内执行再驱动时，将在最初指令执行（运算）时发生「运算错误」。

FNC150

DSZR

16bit	32bit	\square	FNC 155	ABS	ABS位置数据读取
	✓				
	13步		指令格式: ABS (S) (D1) (D2)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)	✓	✓	✓	✓												
(D1)		✓	✓	✓												
(D2)								✓	✓	✓	✓	✓	✓	✓	✓	✓

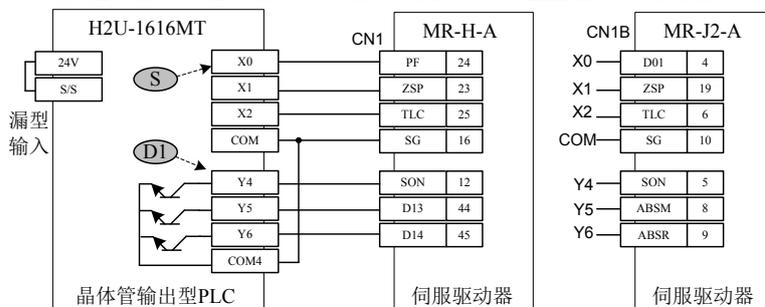
该指令是通过高速输入端口读取伺服驱动器中电机的绝对位置 (ABS) 数据。其中:

- (S) 为读取伺服装置的输入信号, 占用后续共3个单元;
- (D1) 为传送至伺服装置的控制信号, 占用后续共3个单元;
- (D2) 则是从伺服读取数据的存储单元, 32bit宽度, 占用 (D2) + 1、(D2) 单元, 通常指定D8140。

指令举例:



对应的接线方式如下图, 其中伺服驱动器为市售三菱公司产品, 配合有绝对位置检测的编码器的伺服马达:



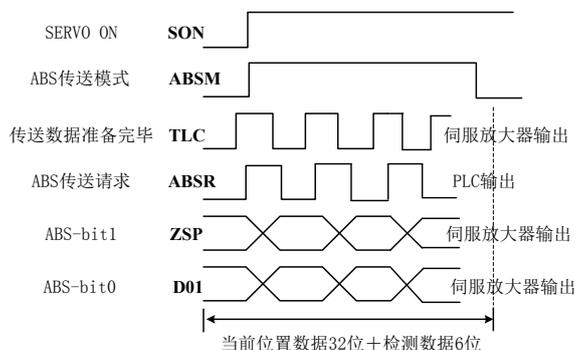
当指令驱动M10变为ON时开始读取, 读取完毕, M8029标志置为ON;

当指令执行过程中, 驱动标志变为OFF, 读取操作即被中断;

读取ABS数据的编程举例如下, 当X6端子闭合时才读取ABS数据, 若5s内没有完成读取操作, 超时标志M21被置位, 编程如下:



ABS读操作的信号时序如下图, 指令执行时, PLC会按该实现自动完成与伺服驱动器的访问操作:



该指令只能以32bit方式执行, 即指令为DABS。

FNC155
ABS

16bit	32bit		FNC 156	ZRN	原点回归
✓	✓				
9步	17步		指令格式: ZRN (S1) (S2) (S3) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(S2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(S3)	✓		✓	✓												
(D)		✓														

该指令是PLC与伺服驱动器配合工作时，用指定脉冲速度和脉冲输出口，让执行机构向动作原点（DOG）移动，直到遇到原点信号满足条件为止。其中：

- (S1) 为原点回归动作的启始速度。16bit指令时，范围是10~32,767Hz；32bit指令时，范围是10~100,000Hz；
- (S2) 为原点信号变为ON后的爬行速度，范围是10~32,767Hz；
- (S3) 原点信号（DOG）输入，虽然XYMS信号都可以，但只有X信号的及时性最好；
- (D) 为脉冲输出启始地址。只能指定Y0/Y1/Y2；

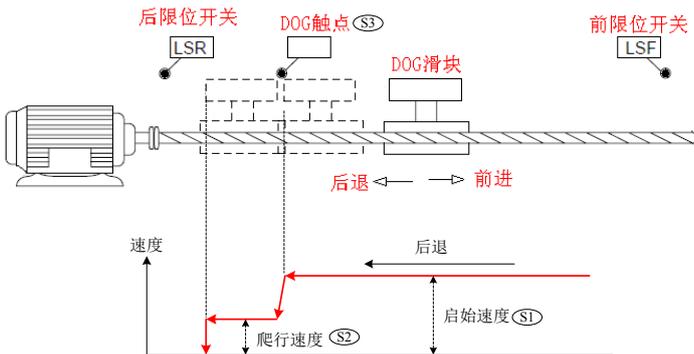
相对位置控制指令DRVI（FNC158）和绝对位置控制指令DRVA（FNC159）在执行时，控制器会计算自身已发出的正转脉冲或反转脉冲数，并将之保存在寄存器[D8141, D8140]（Y000）和[D8143, D8142]（Y001）。但该寄存器的数据在断电时会消失，故上电时和初始运行时，必须执行原点回归指令ZRN，以事先将机械动作的原点位置的数据写入。

指令举例：



本指令的动作是，当M10变为ON后，PLC从Y0高速输出口，开始以1000Hz发出脉冲，令步进电机向原点作后退，当遇到DOG信号变为ON时（此时DOG滑块刚好碰到DOG触点），输出频率降为80Hz，作低速爬行，直到DOG信号再变为OFF时，Y0停止输出脉冲，向当前值寄存器（Y000: [D8141, D8140], Y001: [D8143, D8142]）中写入0。另外，M8140（清零信号输出功能）ON时，同时输出清零信号。随后，当执行完成标志（M8029）置为ON的同时，脉冲输出中监控（Y000: [M8147], Y001: [M8148]）变为OFF。

参见下图：



本指令执行中，涉及的系统变量有：

- D8141（高位），D8140（低位）：Y000输出的当前值寄存器（使用32位）
- D8143（高位），D8142（低位）：Y001输出的当前值寄存器（使用32位）
- M8145: Y000脉冲输出停止（立即停止）
- M8146: Y001脉冲输出停止（立即停止）
- M8147: Y000脉冲输出中监控（BUSY/READY）
- M8148: Y001脉冲输出中监控（BUSY/READY）

FNC156
ZRN

由于伺服驱动器对位置信息具有掉电保持功能，该指令并不需要每次上电时都需进行；指令执行中，仅能单方向运动（后退方向），所以回原点动作必须在DOG信号前端开始。

注意事项：

- 定位指令（ZRN/PLSV/DRVI/DRVA）在程序中可多次使用，但不要对同一高速Y输出口同时进行操作；
- 当指令的驱动能流OFF之后，再次驱动ON时，必须在状态位（Y000: [M8147], Y001: [M8148]）OFF后，经过一个运算周期后，方能再次驱动。
- 定位指令再次驱动时，必须有1个周期以上的OFF时间，若以比上述条件更短的时间内执行再驱动时，将在最初指令执行（运算）时发生「运算错误」。

16bit	32bit	\square	FNC 157	PLSV	可变速度脉冲输出
✓	✓				
7步	13步		指令格式: PLSV (S) (D1) (D2)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(D1)		✓													
(D2)		✓	✓	✓											

该指令是按指定的端口、频率和运行方向输出脉冲频率，没有加减速过程，当驱动能流无效时，输出脉冲直接停止。只有晶体管输出PLC才能使用该指令。其中：

(S) 为指定的输出脉冲频率。16bit指令时，范围是1~32,767Hz；-1~-32,768Hz；32bit指令时，范围是1~100,000Hz；-1~-100,000Hz。其中负号表示反方向运行的指令信号；

(D1) 为脉冲输出端口。只能指定Y0/Y1/Y2；

(D2) 运行方向输出端口或位变量，输出为ON状态，表示为正向运行；否则为反向运行。

指令举例：



语句表示，当M1为ON时，由Y1端口输出10kHz频率的脉冲，Y4用于控制运行方向，若Y4=ON表示为正方向。

本指令执行中，涉及的系统变量有：

D8141（高字节），D8140（低字节）：Y000输出脉冲数。反转时减少。（使用32位）

D8143（高字节），D8142（低字节）：Y001输出脉冲数。反转时减少。（使用32位）

D8151（高字节），D8150（低字节）：Y002输出脉冲数。反转时减少。（使用32位）

D8153（高字节），D8152（低字节）：Y003输出脉冲数。反转时减少。（使用32位）

D8155（高字节），D8154（低字节）：Y004输出脉冲数。反转时减少。（使用32位）

M8145：Y000脉冲输出停止（立即停止）

M8146：Y001脉冲输出停止（立即停止）

M8152：Y002脉冲输出停止（立即停止）

M8153：Y003脉冲输出停止（立即停止）

M8154：Y004脉冲输出停止（立即停止）

M8147：Y000脉冲输出中监控（BUSY/READY）

M8148：Y001脉冲输出中监控（BUSY/READY）

M8149：Y002脉冲输出中监控（BUSY/READY）

M8150：Y003脉冲输出中监控（BUSY/READY）

M8151：Y004脉冲输出中监控（BUSY/READY）

FNC157

PLSV

16bit	32bit		FNC 158	DRVI	相对位置控制
✓	✓				
9步	17步		指令格式: DRVI (S1) (S2) (D1) (D2)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(S2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(D1)		✓													
(D2)		✓	✓	✓											

该指令是按指定的端口、频率和运行方向输出指定的脉冲数，令伺服执行机构在当前位置的基础上作给定偏移量的运动。只有晶体管输出PLC才能使用该指令。其中：

(S1) 为指定的输出脉冲数。16bit指令时，范围是-32768~32,767；32bit指令时，范围是-2,147,483,648~2,147,483,647。其中负号表示反方向；

(S2) 为指定的输出脉冲频率，16bit指令时，范围为10~32767Hz；32bit指令时，范围为10~100,000Hz；

(D1) 为脉冲输出端口。只能指定Y0/Y1/Y2；

(D2) 运行方向输出端口或位变量，输出为ON状态，表示为正向运行；否则为反向运行。

输出脉冲数，是相对于下面的当前值寄存器作为相对位置：

向 [Y000] 输出时，当前寄存器为[D8141 (高字节)，D8140 (低字节)] (使用32位)

向 [Y001] 输出时，当前寄存器为[D8143 (高字节)，D8142 (低字节)] (使用32位)

向 [Y002] 输出时，当前寄存器为[D8151 (高字节)，D8150 (低字节)] (使用32位)

向 [Y003] 输出时，当前寄存器为[D8153 (高字节)，D8152 (低字节)] (使用32位)

向 [Y004] 输出时，当前寄存器为[D8155 (高字节)，D8154 (低字节)] (使用32位)

反转时，当前值寄存器的数值减小。

在指令执行过程中，即使改变操作数的内容，也无法在当前运行中表现出来。只在下一次指令执行时才有效。

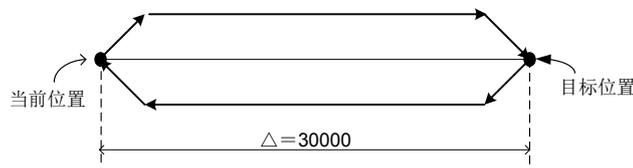
若在指令执行过程中，指令驱动的接点变为OFF时，将减速停止。此时执行完成标志M8029不会动作；

指令驱动接点变为OFF后，在脉冲输出中断标志M8147 (Y000)、M8148 (Y001)、M8149 (Y002)、M8150 (Y003)、M8151 (Y004) 处于ON时，将不接受指令的再次驱动。

指令举例：

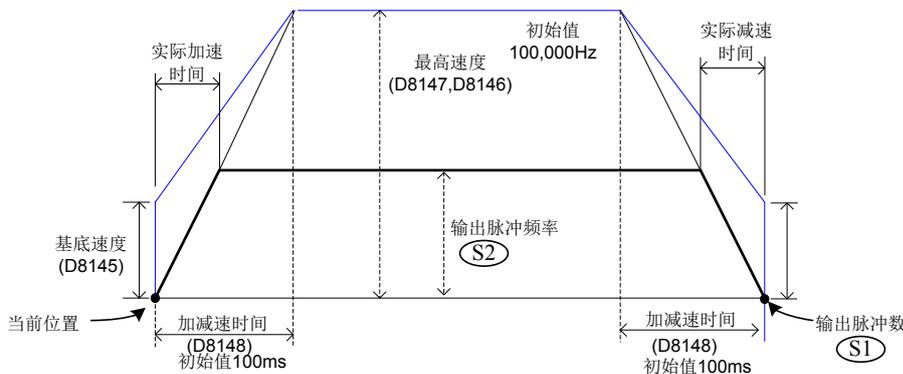


表示以4kHz的频率、由Y0端口输出30000个脉冲，令外部伺服执行机构运行，方向则由Y3决定。



脉冲输出过程中，频率会按预定值加减速：

FNC158
DRVI



实际能够输出的输出脉冲频率的最低频率数，根据以下公式所决定：

$$\text{输出脉冲频率的最低频率数} = \sqrt{\text{最高速度 [D8147, D8146] Hz} \div (2 \times (\text{加减速时间 [D8148]ms} \div 1000))}$$

即使指定了低于上面计算结果的数值，仍将输出计算值的频率。加速初期和减速最终部分的频率也不可低于上述计算结果。

指令执行中涉及的系统变量如下：

[D8145]：执行FNC158(DRVI)，FNC159(DRVA)指令时的基底速度。控制步进电机时，设定速度时需考虑步进电机的共振区域和自动起动频率。设定范围：最高速度（D8147，D8146）的1/10以下。超过该范围时，自动降为最高速度的1/10数值运行。

[D8147（高字节），D8146（低字节）]：执行FNC158(DRVI)，FNC159(DRVA)指令时的最高速度。S2指定的输出脉冲频率必须小于该最高速度。设定范围：10~100,000（Hz）

[D8148]：执行FNC158(DRVI)，FNC159(DRVA)指令时的加减速时间。加减速时间表示到达最高速度（D8147，D8146）所需的时间。因此，当输出脉冲频率S2低于最高速度（D8147，D8146）时，实际加减速时间会缩短。设定范围：50~5,000(ms)

- [M8145]：Y000脉冲输出停止（立即停止）
- [M8146]：Y001脉冲输出停止（立即停止）
- [M8152]：Y002脉冲输出停止（立即停止）
- [M8153]：Y003脉冲输出停止（立即停止）
- [M8154]：Y004脉冲输出停止（立即停止）
- [M8147]：Y000脉冲输出中监控（BUSY/READY）
- [M8148]：Y001脉冲输出中监控（BUSY/READY）
- [M8149]：Y002脉冲输出中监控（BUSY/READY）
- [M8150]：Y003脉冲输出中监控（BUSY/READY）
- [M8151]：Y004脉冲输出中监控（BUSY/READY）

注意事项：

定位指令（ZRN/PLSV/DRVI/DRVA）在程序中可多次使用，但不要对同一输出口同时进行输出操作；

当指令的驱动能流OFF之后，再次驱动ON时，必须在状态位（Y000：[M8147]，Y001：[M8148]，Y002：[M8149]，Y003：[M8150]，Y004：[M8151]）OFF后，经过一个运算周期后，方能再次驱动。

定位指令再次驱动时，必须有1个周期以上的OFF时间，若以比上述条件更短的时间内执行再驱动时，将在最初指令执行（运算）时发生「运算错误」。

注意：

在新版本的H1U/H2U系列PLC中，PLSR，DRVI，DRVA等指令的功能有增强：

- 1) 通过使用特殊位M8135~M8137(分别对应Y0~Y2)为ON，可以在运行中更改输出脉冲个数(可大可小)；同时加速时间和减速时间分别由如下寄存器定义：

端口号	使用特殊位	修改加速时间用寄存器
Y000	M8135	D8104
Y001	M8136	D8105
Y002	M8137	D8106

端口号	使用特殊位	修改减速时间用寄存器
Y000	M8135	D8165
Y001	M8136	D8166
Y002	M8137	D8167

2) 通过使用特殊位M8085~M8087(分别对应Y0~Y2)为ON, 可以实现如下功能:

在驱动特殊位为ON, 可以马上启动下条脉冲输出指令, 不需要上条能流无效的处理;

3) 通过使用特殊位M8090~M8092(分别对应Y0~Y2)为ON, 可以实现如下功能:

可以实现脉冲输出完成中断; 具体对应如下:

端口号	使用特殊位	对应的用户中断
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504

具体可参考附录8.6中的说明。

16bit	32bit	\square	FNC 159	DRVA	绝对位置控制
✓	✓				
9步	17步		指令格式: DRVA (S1) (S2) (D1) (D2)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(S2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(D1)		✓														
(D2)		✓	✓	✓												

该指令是按指定的端口、频率和运行方向输出脉冲，令伺服执行机构运动到指定目的点。只有晶体管输出PLC才能使用该指令。其中：

(S1) 为指定的目标位置(绝对位置)。16bit指令时，范围是-32768~32,767；32bit指令时，范围是-2,147,483,648~2,147,483,647。

若(D1) = [Y000]，对应[D8141 (高字节)，D8140 (低字节)] (使用32位) 为绝对位置；

若(D1) = [Y001]，对应[D8143 (高字节)，D8142 (低字节)] (使用32位) 为绝对位置；

若(D1) = [Y002]，对应[D8151 (高字节)，D8150 (低字节)] (使用32位) 为绝对位置；

其中负号表示反方向。反转时，当前值寄存器的数值减小。

(S2) 为指定的输出脉冲频率，范围为10~32,767Hz (16bit指令)；或10~100,000Hz (32bit指令)；

(D1) 为脉冲输出端口；H1U机型可以指定Y0/Y1/Y2；H2U机型中3624MT/2416MT型只能指定Y0或Y1，其他MT型可以指定Y0/Y1/Y2；

(D2) 运行方向输出端口或位变量，可根据(S1)和当前位置的差值决定，输出为ON状态，表示为正向运行；否则为反向运行。

在指令执行过程中，即使改变操作数的内容，也无法在当前运行中表现出来。只在下一次指令执行时才有效。

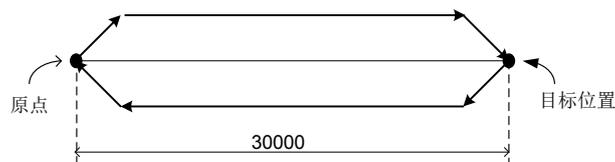
若在指令执行过程中，指令驱动的接点变为OFF时，将减速停止。此时执行完成标志M8029不会动作；

指令驱动接点变为OFF后，在脉冲输出中断标志M8147 (Y000)、M8148 (Y001) 处于ON时，将不接受指令的再次驱动。

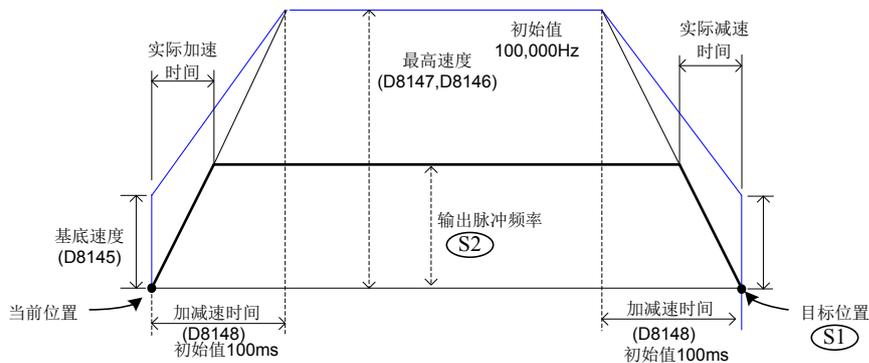
指令举例：



该指令是由指定原点向目标点运行的控制方式，



脉冲输出过程中，频率会按预定值有加减速：



实际能够输出的输出脉冲频率的最低频率数，根据以下公式所决定：

$$\text{输出脉冲频率的最低频率数} = \sqrt{\text{最高速度} [D8147, D8146] \text{Hz} \div (2 \times (\text{加减速时间} [D8148] \text{ms} \div 1000))}$$

即使指定了低于上面计算结果的数值，仍将输出计算值的频率。加速初期和减速最终部分的频率也不可低于上述计算结果。

FNC159
DRVA

指令执行中涉及的系统变量如下：

[D8145]：执行FNC158(DRVI)，FNC159(DRVA)指令时的基底速度。控制步进电机时，设定速度时需考虑步进电机的共振区域和自动起动频率。设定范围：最高速度（D8147，D8146）的1/10以下。超过该范围时，自动降为最高速度的1/10数值运行。

[D8147（高字节），D8146（低字节）]：执行FNC158(DRVI)，FNC159(DRVA)指令时的最高速度。S2指定的输出脉冲频率必须小于该最高速度。设定范围：10~100,000（Hz）

[D8148]：执行FNC158(DRVI)，FNC159(DRVA)指令时的加减速时间。加减速时间表示到达最高速度（D8147，D8146）所需的时间。因此，当输出脉冲频率S2低于最高速度（D8147，D8146）时，实际加减速时间会缩短。设定范围：50~5,000(ms)

[M8145]：Y000脉冲输出停止（立即停止）

[M8146]：Y001脉冲输出停止（立即停止）

[M8152]：Y002脉冲输出停止（立即停止）

[M8147]：Y000脉冲输出中监控（BUSY/READY）

[M8148]：Y001脉冲输出中监控（BUSY/READY）

[M8149]：Y002脉冲输出中监控（BUSY/READY）

注意事项：

定位指令（ZRN/PLSV/DRVI/DRVA）在程序中可多次使用，但不要对同一输出口进行输出操作；

当指令的驱动能流OFF之后，再次驱动ON时，必须在状态位（Y000：[M8147]，Y001：[M8148]，Y002：[M8149]）OFF后，经过一个运算周期后，方能再次驱动。

定位指令再次驱动时，必须有1个周期以上的OFF时间，若以比上述条件更短的时间内执行再驱动时，将在最初指令执行（运算）时发生「运算错误」。

注意：

在新版本的H1U/H2U系列PLC中，PLSR，DRVI，DRVA等指令的功能有增强；

4) 通过使用特殊位M8135~M8137(分别对应Y0~Y2)为ON，可以在运行中更改输出脉冲个数(可大可小)；同时加速时间和减速时间分别由如下寄存器定义：

端口号	使用特殊位	修改加速时间用寄存器
Y000	M8135	D8104
Y001	M8136	D8105
Y002	M8137	D8106

端口号	使用特殊位	修改减速时间用寄存器
Y000	M8135	D8165
Y001	M8136	D8166
Y002	M8137	D8167

5) 通过使用特殊位M8085~M8087(分别对应Y0~Y2)为ON，可以实现如下功能：

在驱动特殊位为ON，可以马上启动下条脉冲输出指令，不需要上条能流无效的处理；

6) 通过使用特殊位M8090~M8092(分别对应Y0~Y2)为ON，可以实现如下功能：

可以实现脉冲输出完成中断；具体对应如下：

端口号	使用特殊位	对应的用户中断
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504

具体可参考附录8.6中的说明。

FNC159
DRVA

4.3.12 时钟运算 (160~169)

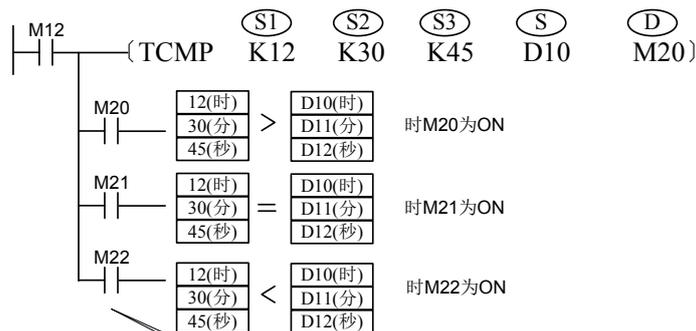
16bit	32bit	\overline{P}	FNC 160	TCMP	时钟数据比较
✓		✓			
11步		11步	指令格式: TCMP (S1) (S2) (S3) (S) (D)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(S2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(S3)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(S)											✓	✓	✓		
(D)		✓	✓	✓											

该指令是将指定的时、分、秒数值，与内部实时时钟进行比较，输出比较结果。其中：

- (S1) 为指定比较用时间的“时”，范围是0~23；
- (S2) 为指定比较用时间的“分”，范围是0~59；
- (S3) 为指定比较用时间的“秒”，范围是0~59；
- (S) 为实时时钟的时间寄存器的起始地址，通常为时钟读取TRD或MOV指令读取后的存放单元；
- (D) 为比较结果的存放变量起始地址，占用后续共3个变量单元；

指令举例：



当M12 ON时，M20~M22其中之一会ON。
 当M12由ON变OFF时，不执行TCMP指令，M20/M21/M22保持M12=OFF以前的状态不变。要清除M20~M22的比较结果可用RST或者ZRST对M20~M22进行清除。
 若需要得到≥、≤、≠的结果时，可将M20~M22 串并联即可取得。

FNC160
TCMP

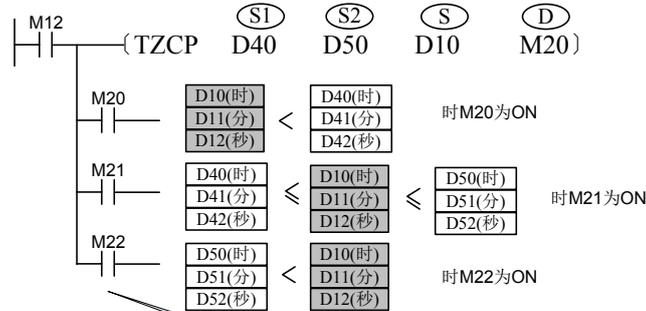
16bit	32bit		FNC 161	TZCP	时钟数据区间比较
✓		✓			
9步		9步	指令格式: TZCP (S1) (S2) (S) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)												✓	✓	✓		
(S2)												✓	✓	✓		
(S)												✓	✓	✓		
(D)		✓	✓	✓												

该指令是将内置实时时钟数据与指定的两组时/分/秒预设值进行区间比较，输出比较结果。其中：

- (S1) 为设定的时间比较下限，占用3个连续的变量单元，依次存储时、分、秒数据；
- (S2) 为设定的时间比较上限，占用3个连续的变量单元，依次存储时、分、秒数据；
- (S) 为实时时钟的时间寄存器的起始地址，通常为时钟读取TRD或MOV指令读取后的存放单元；
- (D) 为比较结果的存放变量起始地址，占用后续共3个变量单元；

指令举例：



当M12 ON时，M20~M22其中之一会ON。
 当M12由ON变OFF时，不执行TZCP指令，M20/M21/M22保持M12=OFF以前的状态不变。要清除M20~M22的比较结果可用RST或者ZRST对M20~M22进行清除。

16bit	32bit	\square	FNC 162	TADD	时钟数据加法运算
✓		✓			
7步		7步	指令格式: TADD (S1) (S2) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)												✓	✓	✓	✓	✓
(S2)												✓	✓	✓	✓	✓
(D)												✓	✓	✓	✓	✓

该指令是将2组时钟数据的时/分/秒对应相加，结果保存于指定的变量中。其中：

(S1) 为时间被加数，占用3个连续的变量单元，依次存储时、分、秒数据：

(S2) 为时间加数，占用3个连续的变量单元，依次存储时、分、秒数据：

(D) 为时间相加和，存储单元，占用3个连续的变量单元，依次存储时、分、秒数据：

若计算结果超过24小时，进位标志M8022置1，实际显示的时间会减去24: 00: 00的数值；

若计算结果为00: 00: 00，零标志M8020置1。

指令举例：



完成的操作如下：



如果加法运算结果超过24小时，则进位标志M8022置On。



16bit	32bit	\square	FNC 163	TSUB	时钟数据减法运算
✓		✓			
7步		7步	指令格式: TSUB (S1) (S2) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)												✓	✓	✓	✓	✓
(S2)												✓	✓	✓	✓	✓
(D)												✓	✓	✓	✓	✓

该指令是将2组时钟数据的时/分/秒对应相减，结果保存于指定的变量中。其中：

- (S1) 为时间被减数，占用3个连续的变量单元，依次存储时、分、秒数据；
- (S2) 为时间减数，占用3个连续的变量单元，依次存储时、分、秒数据；
- (D) 为时间相减差，存储单元，占用3个连续的变量单元，依次存储时、分、秒数据；

若计算结果为负，借位标志M8021置1，实际显示的时间会加上24: 00: 00的数值；

若计算结果00: 00: 00，零标志M8020置1；

指令举例：



完成的操作如下：



如果加法运算结果为负数时，则借位标志M8021置On。



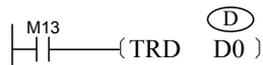
16bit	32bit	\square	FNC 166	TRD	时钟数据读取
✓		✓			
3步		3步	指令格式: TRD (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(D)												✓	✓	✓	✓	✓

该指令是读取PLC内置的实时时钟的年/月/日/时/分/秒/星期，将该7个数据保存于指定的寄存器中。

其中：(D) 为保存读取时间的起始存储单元，占用共7个连续的变量单元，地址由小到大依次存储：年、月、日、时、分、秒、星期等数据。

指令举例：



操作如下：

项目	系统变量		操作后 (D)
年(2000~2099)	D8018	→	D0
月(1~12)	D8017	→	D1
日(1~31)	D8016	→	D2
时(0~23)	D8015	→	D3
分(0~59)	D8014	→	D4
秒(0~59)	D8013	→	D5
星期[0(日)~6]	D8019	→	D6

注：一般情况下要使用可编程控制器的时钟，先用TDR指令将时钟读出来放到D寄存器里再使用，不要直接使用D8013~D8019的值。

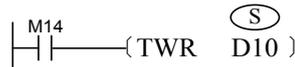
16bit	32bit	<input type="checkbox"/>	FNC 167	TWR	时钟数据写入
✓		✓			
3步		3步	指令格式: TWR (S)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)											✓	✓	✓	✓	✓	

该指令是将指定时钟数据 (S) (含年/月/日/时/分/秒/星期) 的7个数据写入PLC内置的实时时钟数据里。

其中: (S) 为保存读取时间的起始存储单元, 占用共7个连续的变量单元, 地址由小到大依次存储: 年、月、日、时、分、秒、星期等数据

指令举例一:



操作如下:

项目	数据源 (D)		系统变量
年(2000~2099)	D0	→	D8018
月(1~12)	D1	→	D8017
日(1~31)	D2	→	D8016
时(0~23)	D3	→	D8015
分(0~59)	D4	→	D8014
秒(0~59)	D5	→	D8013
星期[0(日)~6]	D6	→	D8019

注意在写入时钟的时候是7个数据全写入的, 在预先设值的时候不能缺少某个变量, 比如星期不写, 则默认是0, 为星期天; 如果月不预先赋值, 则此时的月变量是0, 则PLC认为你提供的月是错误的, 此次时钟的修改无效。

M8017每On一次, PLC内部时钟作±30秒校正动作, 这里的校正是指当PLC的内部时钟的秒针于1~29时, 会被自动归为“0”秒而分针不变、30~59时, 也会被自动归为“0”秒, 分针加1分钟。

M8015置ON可以停止时钟计时。

PLC内部时钟校时方法如下。

指令举例二:

将PLC的现在时间调整为2009年09月10日8时30分0秒, 星期四



提前一段时间将时间写到D0~D6当中, 当这个实际时间到来时将X7接通, 就将正确时间写到PLC里面了

M8017在ON的一瞬间, 可进行正负30秒的调整

注: 一般情况下要修改可编程控制器的时钟, 用TWR指令将时钟写入到D8013~D8019, 用MOV指令直接对D8013~D8019进行赋值时需要将M8015置位才能写入。

FNC167
TWR

16bit	32bit	\square	FNC 169	HOUR	计时器
✓	✓				
7步	13步		指令格式: HOUR (S) (D1) (D2)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(D1)													✓			
(D2)		✓	✓	✓												

该指令是记录驱动条件满足的累加时间，当达到设定的时间后，令指定输出有效。其中：

(S) 为设定时间，单位为“小时”，当累加时间达到该设定值后，令输出有效；

(D1) 累计时间起始单元；

(D2) 计时到达告警输出变量单元，当计时到达设定值后，该指定单元状态有效。

16bit时：(D1) 设置范围K0~K32,767，单位：小时。(D1)+1为未满1个小时的现在时间值，设置范围K0~K3599，单位：秒。此时的(D1)共占用2个单元，

32bit时：(D1)+1、(D1) 设置范围K0~K2147483647，单位：小时。(D1)+2为未满1个小时的现在时间值，设置范围K0~K3599，单位：秒。此时的(D1)共占用3个单元。

指令(D1)计时没有负数，若(D1)制定为非停电保持的寄存器区域，则在PLC由STOP到RUN或者在掉电的时候会将(D1)的值清零。若需要在PLC掉电的情况下仍能保持当前值数据，请将(D1)指定为停电保持区域的寄存器。

指令举例：



当M200=ON时，累计该状态的持续时间，将时记录在D300中，将不满1小时的秒记录在D301中，当D300累计时间达到2000小时后，Y10输出状态为ON。计时条件满足时，到达(S)指定数值后，累计计时仍继续进行，读数会继续增大；现在时间值D300到达最大数值32,767小时、D301达到3,599秒时会停止计时测量，要重新计时须将现在时间值D300、301清除为0。

4.3.13 外围设备 (170~177)

16bit	32bit		FNC 170	GRY	格雷码转换 (BIN→GRY)
✓	✓	✓			
5步	9步		指令格式: GRY (S) (D)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(D)								✓	✓	✓	✓	✓	✓	✓	✓

该指令是将BIN数值，转换为格雷码 (GRY)。其中：

(S) 为待转换的BIN的数据源或数据变量单元；16bit指令时范围0~32,767，32bit指令时范围0~2, 147, 483, 647，超出此范围时M8067、M8068会置On，指令不执行；

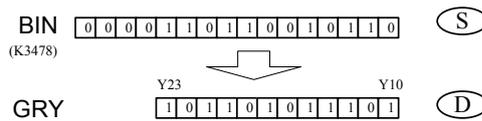
(D) 为转换为格雷码 (GRY) 后的存放单元。

BIN→GRY数学算法：从最右边一位起，依次将每一位与左边一位异或(XOR)，作为对应GRY该位的值，最左边一位不变(相当于左边是0)；

指令举例：



执行结果：



16bit	32bit		FNC 171	GBIN	格雷码逆转换 (GRY→BIN)
✓	✓	✓			
5步	9步		指令格式: GBIN (S) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(D)								✓	✓	✓	✓	✓	✓	✓	✓	

该指令是将GRY格雷码，转换为二进制数值。其中：

(S) 为待转换的GRY码数据源或数据变量单元；16bit指令时范围0~32,767，32bit指令时范围0~2,147,483,647，超出此范围时M8067、M8068会置On，指令不执行；

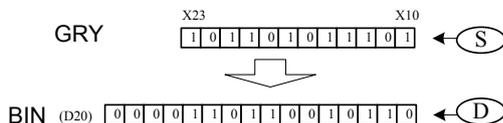
(D) 为转换为BIN后的存放单元。

GRY→BIN数学算法：从左边第二位起，将每位与左边一位解码后的值异或，作为该位解码后的值（最左边一位依然不变）。

指令举例：



执行结果举例：



16bit	32bit		FNC 176	RD3A	模拟量模块读取
✓		✓			
7步		7步	指令格式: RD3A (m1) (m2) (D)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(m1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(m2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(D)								✓	✓	✓	✓	✓	✓	✓	✓	

本指令提供与三菱FX0N~3A型模块的模拟量输入值的读取指令。其中：

- (m1) 为特殊模块号：(K0~K7)
- (m2) 为模拟量输入通道：(K1~K2)
- (D) 为读取回来后的值的存放地址

指令举例：



用FROM指令也可以读取FX0N~3A的模拟量输入值。

16bit	32bit		FNC 177	WR3A	模拟量模块写入
✓		✓			
7步		7步	指令格式: WR3A (m1) (m2) (D)		

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(m1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(m2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(D)								✓	✓	✓	✓	✓	✓	✓	✓

本指令提供与三菱FX0N~3A型模块的模拟量输出值的写入指令。其中：

(m1) 为特殊模块号；(K0~K7)

(m2) 为模拟量输入通道；(K1)

(D) 为主模块中参数寄存器地址，其参数作为写操作数据的来源。

指令举例：



用TO指令也可以对FX0N~3A的模拟量输出值进行写入。

4.3.14 接点比较 (224~246)

FNC NO.	指令助记符	《指令名称》
224	LD=	触点比较指令运算开始 (S1) = (S2) 时导通
225	LD>	触点比较指令运算开始 (S1) > (S2) 时导通
226	LD<	触点比较指令运算开始 (S1) < (S2) 时导通
228	LD< >	触点比较指令运算开始 (S1) ≠ (S2) 时导通
229	LD≤	触点比较指令运算开始 (S1) ≤ (S2) 时导通
230	LD≥	触点比较指令运算开始 (S1) ≥ (S2) 时导通
232	AND=	触点比较指令串联连接 (S1) = (S2) 时导通
233	AND>	触点比较指令串联连接 (S1) > (S2) 时导通
234	AND<	触点比较指令串联连接 (S1) < (S2) 时导通
236	AND< >	触点比较指令串联连接 (S1) ≠ (S2) 时导通
237	AND≤	触点比较指令串联连接 (S1) ≤ (S2) 时导通
238	AND≥	触点比较指令串联连接 (S1) ≥ (S2) 时导通
240	OR=	触点比较指令并联连接 (S1) = (S2) 时导通
241	OR>	触点比较指令并联连接 (S1) > (S2) 时导通
242	OR<	触点比较指令并联连接 (S1) < (S2) 时导通
244	OR< >	触点比较指令并联连接 (S1) ≠ (S2) 时导通
245	OR≥	触点比较指令并联连接 (S1) ≥ (S2) 时导通
246	OR≤	触点比较指令并联连接 (S1) ≤ (S2) 时导通

FNC177

WR3A

16bit	32bit		FNC 224~230	LD \odot	触点型比较指令
✓	✓				
5步	9步		指令格式: LD (S1) (S2)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(S2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

该指令将两个操作数进行比较，将比较结果以逻辑状态输出，参与比较的变量都按有符号数处理， \odot 比较符中有=、>、<、>=、<=、<>等。其中：

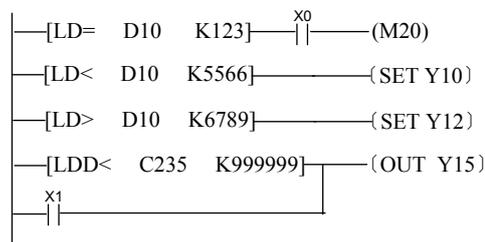
(S1) 为待比较的数据源或数据变量单元1；

(S2) 为待比较的数据源或数据变量单元2。

LD触点型比较方式有：

16bit指令	FNC NO	32bit指令	导通条件	非导通条件
LD=	224	LDD=	(S1) = (S2)	(S1) \neq (S2)
LD>	225	LDD>	(S1) > (S2)	(S1) <= (S2)
LD<	226	LDD<	(S1) < (S2)	(S1) >= (S2)
LD<>	228	LDD<>	(S1) <> (S2)	(S1) = (S2)
LD<=	229	LDD<=	(S1) <= (S2)	(S1) > (S2)
LD>=	230	LDD>=	(S1) >= (S2)	(S1) < (S2)

指令举例：



当D10 的内容等于K123且X0=On 的时候，M20=On 。

当D10 的内容小于K5566的时候，Y10=On 并保持住。

当D10 的内容大于K6789的时候，Y12=On 并保持住。

当C235 的内容小于K999999，或者X1=On 的时候，Y15=On 。

对于参与比较的数为32bit宽度的计数器，应使用32bit指令LDD \odot ，否则出错。

32位计数器(C200~C255)以本指令作比较时，一定要使用32位指令(LDD \odot)。

16bit	32bit	\square	FNC 232~238	AND \odot	接点型比较指令
✓	✓				
5步	9步		指令格式: AND (S1) (S2)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(S2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

该指令之前已有其他逻辑操作。该指令将两个操作数进行比较，将比较结果以逻辑状态形式参与程序能流的运算，指令中参与比较的变量都按有符号数处理， \odot 比较符中有=、>、<、>=、<=、<>等。其中：

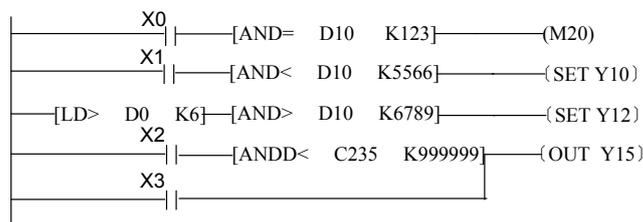
(S1) 为待比较的数据源或数据变量单元1；

(S2) 为待比较的数据源或数据变量单元2。

AND触点型比较方式有：

16bit指令	FNC NO	32bit指令	导通条件	非导通条件
AND=	232	ANDD=	(S1) = (S2)	(S1) \neq (S2)
AND>	233	ANDD>	(S1) > (S2)	(S1) <= (S2)
AND<	234	ANDD<	(S1) < (S2)	(S1) >= (S2)
AND<>	236	ANDD<>	(S1) <> (S2)	(S1) = (S2)
AND<=	237	ANDD<=	(S1) <= (S2)	(S1) > (S2)
AND>=	238	ANDD>=	(S1) >= (S2)	(S1) < (S2)

指令举例：



当X0=On 且D10 的值又等于K123 时，M20=On。

当X1=0n 且D10 的值又小于K5566 时，Y10=0n并保持住。

当D0的值大于K6且D10 的值又大于K6789 时，Y12=0n并保持住。

当X2=0n 且C235 的值又小于K999999，或者X3=0n时，Y15=0n并保持住。

对于参与比较的数为32bit宽度的计数器，应使用32bit指令ANDD \odot ，否则出错。

32位计数器(C200~C255)以本指令作比较时，一定要使用32位指令(ANDD \odot)。

16bit	32bit		FNC 240~246	OR	并联接点型比较指令
✓	✓				
5步	9步		指令格式: OR (S1) (S2)		

操作数	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
(S1)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(S2)					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

该指令将两个操作数进行比较，将比较结果以逻辑状态形式参与程序能流的或运算，指令中参与比较的变量都按有符号数处理，☆比较符中有=、>、<、>=、<=、<>等。其中：

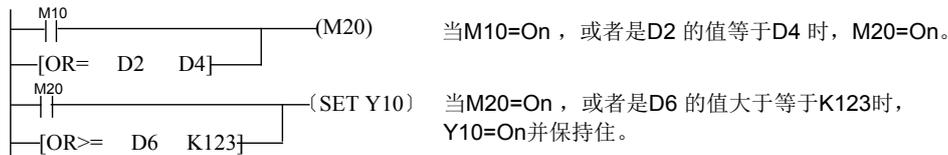
(S1) 为待比较的数据源或数据变量单元1；

(S2) 为待比较的数据源或数据变量单元2。

LD触点型比较方式有：

16bit指令	FNC NO	32bit指令	导通条件	非导通条件
OR=	240	ORD=	(S1) = (S2)	(S1) ≠ (S2)
OR>	241	ORD>	(S1) > (S2)	(S1) ≤ (S2)
OR<	242	ORD<	(S1) < (S2)	(S1) ≥ (S2)
OR<>	244	ORD<>	(S1) <> (S2)	(S1) = (S2)
OR<=	245	ORD<=	(S1) ≤ (S2)	(S1) > (S2)
OR>=	246	ORD>=	(S1) ≥ (S2)	(S1) < (S2)

指令举例：

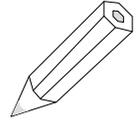


对于参与比较的数为32bit宽度的计数器，应使用32bit指令ORD☆，否则出错。

32位计数器(C200~C255)以本指令作比较时，一定要使用32位指令(ORD☆)。

Memo NO. _____

Date / /



A series of horizontal lines for writing, starting from the top of the page and extending to the bottom, providing a template for notes.



第五章 通信功能

5.1 概要	212
5.1.1 硬件及通信连线	212
5.1.2 H2U通信协议切换逻辑说明	213
5.2 HMI监控协议	217
5.3 并联协议（1:1）	217
5.4 N:N协议	218
5.5 N:N增强协议	221
5.6 计算机链接协议	223
5.7 MODBUS协议说明	230
5.7.1 MODBUS主站通讯应用	230
5.7.2 MODBUS从站通讯应用	234
5.7.3 多台PLC之间采用MODBUS协议通讯的编程方法	238
5.7.4 MODBUS配置与编程方法	240
5.7.5 汇川QLink协议与编程方法	243
5.7.6 PLC与MD380系列变频器的通讯编程方法	245
5.7.7 MD380配合PLC应用的功能码设置	245
5.7.8 PLC修改MD380功能码的编程	250
5.9 CAN通讯指令说明	252
5.9.1 CANlink3.0应用简介	252
5.9.2 主从架构	253
5.9.3 心跳机制	255
5.9.4 掉站信息	255
5.9.5 CANlink3.0的使用	256
5.9.6 软件配置CANlink3.0主从站的通信配置方法	257
5.9.7 CAN网络配置表的填写方法	259
5.9.8 CANlink常见故障与排除	270

第五章 通信功能

5.1 概要

H_{1U}/H_{2U}系列PLC主模块包含四个独立物理串行通信口，分别命名为COM0、COM1、COM2和COM3。COM0具有编程、监控功能；COM1、COM2和COM3功能即完全由用户自由定义。PLC标配MiniUSB通信口，具有上下载、监控、配置等功能。

主模块整机硬件标准配置，COM0硬件为标准RS422，接口端子为8孔鼠标头母座；COM1和COM2硬件为标准RS485，接口为接线端子；COM3预留。

除主模块标准配置的通信硬件外，还可以通过三款通信扩展卡供用户选择。H_{2U}-485-BD通信扩展卡接口方式为接线端子，可支持2线半双工RS485标准通信和4线制全双工RS422通信，需要用户配线。H_{2U}-422-BD通信扩展卡接口方式为8孔鼠标头，需要同时选配专用电缆，仅支持全双工RS422通信。H_{2U}-232-BD通信扩展卡接口方式为DB9，支持3线RS232标准通信，可选用标准RS232通信线连接用户设备。

5.1.1 硬件及通信连线

整机硬件标准配置，COM0硬件为标准RS422，接口端子为8孔鼠标头母座。USB通信口为MiniUSB接口。

接口定义：

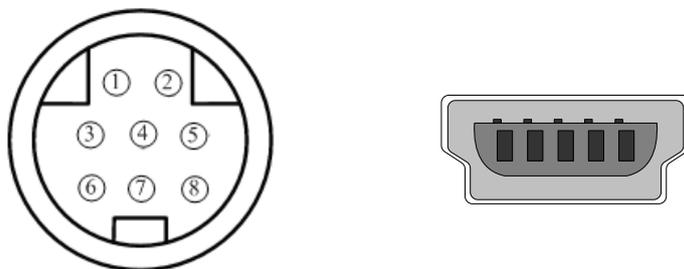


图 COM0 和MiniUSB程序下载口

管脚号	信号	描述
1	RXD-	接收负
2	RXD+	接收正
3	GND	地线，9、10没有电气连接
4	TXD-/RXD-	对外发送负
5	+5V	对外供电+5V，与内部用的逻辑+5V相同
6	CCS	通讯方向控制线，高电平表示发，低电平表示收，在串口作RS422时固定为高，4、7脚一直处于发送
7	TXD+/RXD+	对外发送正
8	NC	空脚

通过COM0，PLC与计算机或触摸屏的连接有两种方式。

方式1：PLC侧为RS422，计算机侧为USB。计算机通过专用的USB下载电缆连接到COM0的程序下载口。

方式2：PLC侧为RS422，计算机侧为RS232，计算机通过专用的串口下载电缆连接到COM0的程序下载口。

H_{2U}-XP 24136及以上版本，H_{1U}-XP 26127及以上版本支持USB下载。USB驱动安装程序见后台安装目录下“usb”文件夹。

COM1和COM2硬件为RS485，接口为接线端子，接口定义：

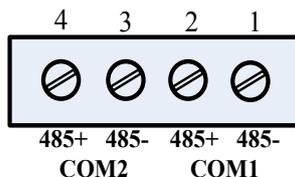


图 通信接线端子

COM1与其它设备通信连接方式。

通过接线端子，用户现场配线。

两串口若通过RS485通信，均只支持半双工通信模式。

COM3预留。

5.1.2 H₂U 通信协议切换逻辑说明

1) COM0协议

COM0协议固定为下载协议/HMI监控协议。

2) COM1协议切换逻辑

a) 在停机状态或第一次运行时，协议可以随时切换，系统按优先级检查协议，若存在优先级别较高的协议，将不再检查优先级低的协议。协议优先级如下：N:N，并联主站，并联从站，计算机链接。

b) N:N协议触发方式：从0步开始，存在以下指令，系统设置为N:N协议。

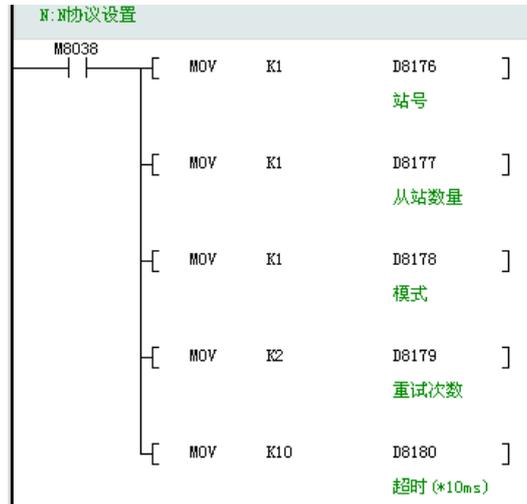


图 N:N协议设置

c) 并联协议，M8070置位为并联协议主站，M8071置位为并联协议从站，M8070与M8071不能同时置位，若同时置位并联协议无效。

d) 计算机链接协议，D8120的bit14 = 1，系统设置COM1协议为计算机链接协议。

e) 以上设置均不存在，协议由D8126决定，D8126在PLC第一个扫描周期内确定的值对协议有效，运行后D8126的更改不能改变协议，D8126与协议对应关系见协议设置表。

f) PLC运行后，协议不能改变。

备注：COM2和COM3的协议切换逻辑与COM1类同，只有D元件不同，设置见下面介绍。

3) 串口通信格式设置

a) 协议设置表

COM0协议	D8116设定	半双工/全双工模式	COM0通信格式
下载协议/HMI监控协议	01h	全双工	固定
并联协议主站	不支持	不支持	不支持
并联协议从站	不支持	不支持	不支持
N:N协议主站	不支持	不支持	不支持
N:N协议从站	不支持	不支持	不支持
计算机链接协议	不支持	不支持	不支持
MODBUS-RTU从站/Qlink协议	02h	半双工	由D8110决定
MODBUS-ASC从站	不支持	不支持	不支持
RS指令	不支持	不支持	不支持
MODBUS RTU指令	不支持	不支持	不支持
MODBUS-ASC指令	不支持	不支持	不支持

COM1通信协议由D8126设定，通信模式及通信格式由D8120设定；COM2通信协议由D8266设定，通信模式及通信格式由D8260设定；COM3通信协议由D8276设定，通信模式及通信格式由D8270设定。

COM1协议设置表如下：（COM2和COM3只有设定D元件不同，其他一样。）

COM1协议	D8126设定	半双工/全双工模式	COM1通信格式
HMI监控协议	01h	半双工	固定
并联协议主站	50h	半双工	固定

COM1协议	D8126设定	半双工/全双工模式	COM1通信格式
并联协议从站	05h	半双工	固定
N:N协议主站	40h	半双工	固定
N:N协议从站	04h	半双工	固定
N:N增强协议主站	41h	半双工	固定
N:N增强协议从站	14h	半双工	固定
计算机链接协议	06h	半双工	由D8120决定
MODBUS-RTU从站/Qlink协议	02h	半双工	由D8120决定
MODBUS-ASC从站	03h	半双工	由D8120决定
RS功能	10h	由D8120的Bit10设定*	由D8120决定
MODBUS RTU功能	20h	半双工	由D8120决定
MODBUS-ASC功能	30h	半双工	由D8120决定
MODBUS配置	60h	半双工	由D8120决定

*RS指令半双工/全双工模式由D8120的Bit10设定:

1:半双工RS485;

0:全双工RS232C/RS422;

(COM1本机标配为RS485, 若使用RS485, D8120的Bit10必须设置为ON; 若用户需要使用全双工的RS422或RS232C, 需要购买通讯扩展板, 并把D8120的Bit10置为OFF)。

协议与通信格式对照表

协议名称	波特率	数据位	校验位	停止位
N:N协议	默认为38400	固定为7	固定为偶校验E	固定为1位
N:N增强协议	固定为38400	固定为7	固定为偶校验E	固定为1位
5 并联协议	默认为19200	固定为7	固定为偶校验E	固定为1位
通信功能 HMI监控协议	固定为9600	固定为7	固定为偶校验E	固定为1位
计算机链接协议	串口0由D8110、串口1由D8120、串口2由D8260、串口3由D8270的Bit7~Bit4设定:	串口0由D8110、串口1由D8120、串口2由D8260、串口3由D8270的Bit0设定:	串口0由D8110、串口1由D8120、串口2由D8260、串口3由D8270的Bit2~Bit1设定:	串口0由D8110、串口1由D8120、串口2由D8260、串口3由D8270的Bit3设定:
MODBUS-RTU从站	0011b-300Bits/s	0b-7Bits	00b-无校验(N)	0-1Bits
MODBUS-ASC从站	0100b-600Bits/s	1b-8Bits	01b-奇校验(O)	1-2Bits
RS指令	0110b-2400Bits/s	注:	11b-偶校验(E)	
MODBUS-RTU指令	0111b-4800Bits/s	MODBUS-RTU从站协议及指令只支持8位数据位, 否则将造成通信出错		
MODBUS-ASC指令	1000b-9600Bits/s			
MODBUS配置	1001b-19200Bits/s 1010b-38400Bits/s 1011b-57600Bits/s 1100b-115200Bits/s			

4) 串口通信格式设置软元件一览表

COM0 串口设定			
M8110	保留	D8110注1	通讯格式, 界面配置设定, 默认为0
M8111	发送等待中 (RS指令)	D8111注1	站号设置, 界面配置设定, 默认为1
M8112	发送标志 (RS指令) 指令执行状态 (MODBUS)	D8112	传送剩余数据数量 (仅对RS指令)
M8113	接收完成标志 (RS) 通讯错误标志 (MODBUS)	D8113	接收到的数据数量 (仅对RS指令)
M8114	接收中 (仅对RS指令)	D8114	起始字符STX (仅对RS指令)
M8115	保留	D8115	终止字符ETX (仅对RS指令)
M8116	保留	D8116	通讯协议设定, 界面配置设定, 默认为0
M8117	保留	D8117	计算机链接协议接通要求数据起始地址号
M8118	保留	D8118	计算机链接协议接通要求发送数据数量
M8119	超时判断	D8119	通讯超时时间判断, 界面配置设定, 默认为10 (100ms)
COM1 串口设定			
M8120	保留	D8120注1	通讯格式, 界面配置设定, 默认为0

M8121	发送等待中 (RS指令)	D8121注1	站号设置, 界面配置设定, 默认为1
M8122	发送标志 (RS指令) 指令执行状态 (MODBUS)	D8122	传送剩余数据数量 (仅对RS指令)
M8123	接收完成标志 (RS) 通讯错误标志 (MODBUS)	D8123	接收到的数据数量 (仅对RS指令)
M8124	接收中 (仅对RS指令)	D8124	起始字符STX (仅对RS指令)
M8125	保留	D8125	终止字符ETX (仅对RS指令)
M8126	为ON时485BD扩展卡有效	D8126	通讯协议设定, 界面配置设定, 默认为0
M8127	保留	D8127	计算机链接协议接通要求数据起始地址号; SETEX协议: 从站 (PLC) 发送给主站 (显示屏) 缓冲区起始地址, 缓冲区大小为18。
M8128	保留	D8128	计算机链接协议接通要求发送数据数量; SETEX协议: 主站 (显示屏) 发送给从站 (PLC) 缓冲区起始地址, 缓冲区大小为19。
M8129	超时判断	D8129	通讯超时时间判断, 界面配置设定, 默认为10 (100ms)
COM2 串口设定			
M8260	保留	D8260注1	通讯格式, 界面配置设定, 默认为0
M8261	发送等待中 (RS2指令)	D8261注1	站号设置, 界面配置设定, 默认为1
M8262	发送标志 (RS2指令) 指令执行状态 (MODBUS2)	D8262	传送剩余数据数量 (仅对RS2指令)
M8263	接收完成标志 (RS2) 通讯错误标志 (MODBUS2)	D8263	接收到的数据数量 (仅对RS2指令)
M8264	接收中 (仅对RS2指令)	D8264	起始字符STX (仅对RS2指令)
M8265	保留	D8265	终止字符ETX (仅对RS2指令)
M8266	为ON时485BD扩展卡有效	D8266	通讯协议设定, 界面配置设定, 默认为0
M8267	保留	D8267	计算机链接协议接通要求数据起始地址号; SETEX协议: 从站 (PLC) 发送给主站 (显示屏) 缓冲区起始地址, 缓冲区大小为18。
M8268	保留	D8268	计算机链接协议接通要求发送数据数量; SETEX协议: 主站 (显示屏) 发送给从站 (PLC) 缓冲区起始地址, 缓冲区大小为19。
M8269	超时判断	D8269	通讯超时时间判断, 界面配置设定, 默认为10 (100ms)
COM3 串口设定			
M8270	保留	D8270注1	通讯格式, 界面配置设定, 默认为0
M8271	发送等待中 (RS2指令)	D8271注1	站号设置, 界面配置设定, 默认为1
M8272	发送标志 (RS2指令) 指令执行状态 (MODBUS2)	D8272	传送剩余数据数量 (仅对RS2指令)
M8273	接收完成标志 (RS2) 通讯错误标志 (MODBUS2)	D8273	接收到的数据数量 (仅对RS2指令)
M8274	接收中 (仅对RS2指令)	D8274	起始字符STX (仅对RS2指令)
M8275	保留	D8275	终止字符ETX (仅对RS2指令)
M8276	为ON时485BD扩展卡有效	D8276	通讯协议设定, 界面配置设定, 默认为0
M8277	保留	D8277	计算机链接协议接通要求数据起始地址号; SETEX协议: 从站 (PLC) 发送给主站 (显示屏) 缓冲区起始地址, 缓冲区大小为18。
M8278	保留	D8278	计算机链接协议接通要求发送数据数量; SETEX协议: 主站 (显示屏) 发送给从站 (PLC) 缓冲区起始地址, 缓冲区大小为19。
M8279	超时判断	D8279	通讯超时时间判断, 界面配置设定, 默认为10 (100ms)

注1: 掉电保存。

5) 通讯错误码一览表

类型	出错代码	出错内容	处理方法	
并行连接 通信出错 M8063(D8063) 继续运行	0000	无异常	检查双方的可编程控制器的电源是否为ON, 适配器和控制器之间, 以及适配器之间连接是否正确。	
	6301	奇偶出错; 超过出错; 成帧出错		
	6302	通信字符有误		
	6303	通信数据的和数不一致		
	6304	数据格式有误		
	6305	指令有误		
	6306	监视定时器溢出		
	6307~6311	无		
	6312	并行连接字符出错		
	6313	并行连接和数出错		
	6314	并行连接格式出错		
	6330	MODBUS从站地址设置错误		COM0通讯出错 请检查COM0的通讯电缆是否正确连接。
	6331	数据帧长度错误		
	6332	地址错误		
	6333	CRC检验错误		
	6334	不支持的命令码		
	6335	接收超时		
	6336	数据错误		
	6337	缓冲区溢出		
	6338	帧错误		
	6339	串口协议错误		
	6340	MODBUS从站地址设置错误	COM1通讯出错 请检查COM1的通讯电缆是否正确连接; 检查通讯双方通讯格式是否匹配。	
	6341	数据帧长度错误		
	6342	地址错误		
	6343	CRC检验错误		
	6344	不支持的命令码		
	6345	接收超时		
	6346	数据错误		
	6347	缓冲区溢出		
	6348	帧错误	COM2通讯出错 请检查COM2的通讯电缆是否正确连接; 检查通讯双方通讯格式是否匹配。	
	6349	串口协议错误		
	6350	MODBUS从站地址设置错误		
	6351	数据帧长度错误		
	6352	地址错误		
	6353	CRC检验错误		
	6354	不支持的命令码		
6355	接收超时			
6356	数据错误	COM3通讯出错。 请检查COM3的通讯电缆是否正确连接; 检查通讯双方通讯格式是否匹配。		
6357	缓冲区溢出			
6358	帧错误			
6359	串口协议错误			
6360	MODBUS从站地址设置错误			
6361	数据帧长度错误			
6362	地址错误			
6363	CRC检验错误			
6364	不支持的命令码			
6365	接收超时			
6366	数据错误			
6367	缓冲区溢出			
6368	帧错误			
6369	串口协议错误			

注: M8063, D8063在故障消失后仍然保持, 直到用户强行清除。

5.2 HMI监控协议

1) 硬件配置与软件设置:

通过COM0通信, 连接方式为RS422, 只能通过下载口连接。

通过COM1通信, 连接方式为RS485, 需要设置D8126=01h, 通过接线端子配线, 也可通过H2u-485-BD扩展卡配线连接。

通过COM1通信, 连接方式为RS422, 需要设置D8126=81h, 通过H2u-485-BD扩展卡配线或通过H2u-422-BD扩展卡连接。H2u-485-BD扩展卡需要配置为4线制。H2u-422-BD需要通过专用电缆连接。

本协议通信格式及波特率固定。

备注: COM2和COM3的硬件配置及软件设置与COM1类同, 只有D元件不同, COM2需要设置D8266; COM3需要设置D8276。

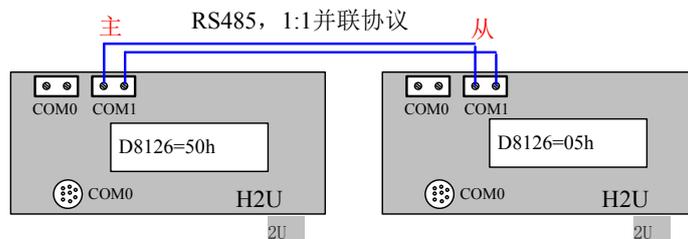
2) 协议说明:

HMI监控协议为PLC内部协议, 用于AUTOSHOP软件与PLC通信, AUTOSHOP通过该协议, 可以擦除、读取和下载用户程序; 可以对PLC实施遥测、遥调与遥控。具体为可监测PLC中任意元件的状态, 可强制更改任何元件, 还可以控制PLC的启动和停止。

为保护用户程序, 建议PLC工作中除调试程序外, 其它设备不要使用该协议连接PLC。

5.3 并联协议 (1:1)

当需要在两台H1u/H2u系列PLC主模块之间交换数据时, 采用1:1并联通讯协议进行通讯是最简单的方式之一, 硬件上只需要用双绞线将两个PLC的COM1端口RS485信号端对应并联就可以组成通讯网络。



1) 硬件配置与软件设置:

并联协议只能用于COM1, 设置COM1为并联协议有两种方法: 一、设置D8126=50h, PLC即为并联协议主站。设置D8126=05h, PLC即为并联协议从站。二、设置M8070=on, PLC即为并联协议主站。设置M8071=on, PLC即为并联协议从站。

本协议通信格式及波特率固定。

PLC系统软件内置有并联协议, 用户只需要在系统寄存器D8126中设置即可, 将其中一台PLC设置为并联协议主站, 并设定为普通通讯模式, 只需如下语句:



将另一台设置为并联协议从站, 并设定为普通通讯模式, 只需如下语句:



将两台PLC的COM1串口连接起来, 即可实现两台PLC间自动互相交换数据, 通讯交换的数据区地址是固定的, 收发对应各自固定的变量区域。依据交换的数据量大小, 分为两种通讯模式, 如下表:

	主站发送 (从站接收)	从站发送 (主站接收)
普通模式 M8162=0	M800~M899 D490~D499	M900~M999 D500~D509
高速模式 M8162=1	D490~D491	D500~D501

并联协议通讯与控制相关的变量如下：

M8070：设置1为并联连接主站；若该位为0，将D8126=50h也为并联主站；（M8070的设置优先于D8126）

M8071：设置1为并联连接从站；若该位为0，将D8126=05h也为并联从站；（M8071的设置优先于D8126）

M8162：高速并联连接模式

M8072：并联连接运行中

M8073：并行连接设定异常

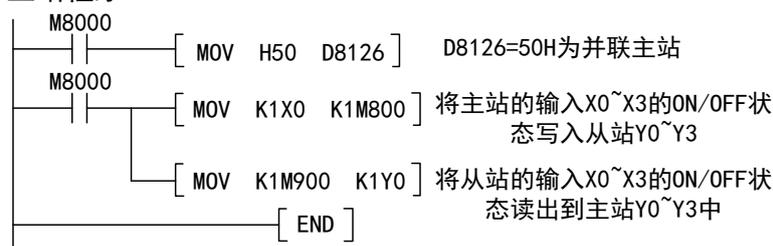
M8063：串行通信出错

D8070：判断出错的时间设定，默认为500

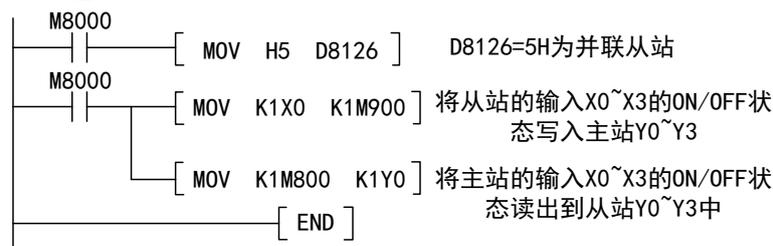
D8063：串行通信出错代码

2) 并联实例参考：

主站程序：



从站程序：

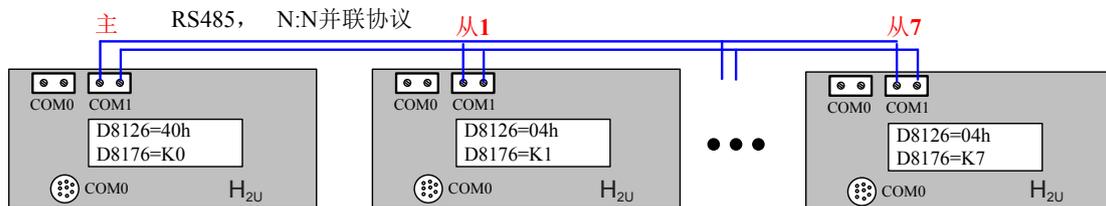


注：另一种设置：M8070置ON为主站；M8071置ON为从站。

注意：XP型COM2/COM3也可用并联协议，使用方法与COM1相同，设置M、D请参考相应的串口设定。

5.4 N:N协议

当一个设备中有多台（2~8台）PLC需要互相交换信息，协调地工作，可采用PLC内置的N:N网络协议，实现PLC之间的多方通讯。硬件上只需要用双绞线将所有PLC的COM1端口RS485信号端对应并联就可以组成通讯网络。

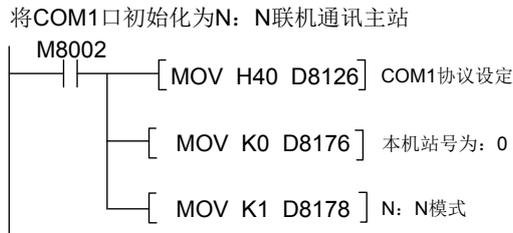


1) 硬件配置与软件设置：

该协议只能用于COM1，设置COM1为N:N协议有两种方法：一、设置D8126=40h，PLC即为N:N协议主站；设置D8126=04h，PLC即为N:N协议从站。二、在程序起始加一段程序。

本协议通信格式及波特率固定。

用户需要设置一台PLC为N:N协议主站，设置本站号D8176为0，速度模式D8178，例如：



其余PLC设置为N:N协议从站，设置本站站号D8176，速度模式D8178，例如：



PLC运行时，即可实现多台PLC间互相交换数据，用户程序在本PLC内部特定的数据区可读取到其他PLC的发送状态数据；本机用户程序将需要广播的数据复制到特定数据单元，便可让其他PLC读取。根据所需的通讯交互的信息量与通讯刷新速度两个指标要求，有三种模式可供选择，对应各自的变量区域定义如下：

模式	站点号	软元件号	
		位软元件 (M)	字软元件 (D)
		0个	4个
模式0 D8178=0 交换数据 0个M元件 4个D元件	第0号	无	D0到D3
	第1号	无	D10到D13
	第2号	无	D20到D23
	第3号	无	D30到D33
	第4号	无	D40到D43
	第5号	无	D50到D53
	第6号	无	D60到D63
	第7号	无	D70到D73
模式1 D8178=1 交换数据 32个M元件 4个D元件	第0号	M1000到M1031	D0到D3
	第1号	M1064到M1095	D10到D13
	第2号	M1128到M1159	D20到D23
	第3号	M1192到M1223	D30到D33
	第4号	M1256到M1287	D40到D43
	第5号	M1320到M1351	D50到D53
	第6号	M1384到M1415	D60到D63
	第7号	M1448到M1479	D70到D73
模式2 D8178=2 交换数据 64个M元件 8个D元件	第0号	M1000到M1063	D0到D7
	第1号	M1064到M1127	D10到D17
	第2号	M1128到M1191	D20到D27
	第3号	M1192到M1255	D30到D37
	第4号	M1256到M1319	D40到D47
	第5号	M1320到M1383	D50到D57
	第6号	M1384到M1447	D60到D67
	第7号	M1448到M1511	D70到D77

设置N:N联接协议的相关寄存器说明：

D8126: COM1通讯口通讯协议配置，设为40h表示N:N主站；设为04h表示N:N从站。

D8176: 站点号，范围0~7，0表示主站点；

D8177: 从站点的总数，范围1~7，仅主站需要设置；

D8178: 刷新范围（模式）设置，范围0~2，仅主站需要设置；

D8179: 重试次数设定，仅主站需要设置；

D8180: 通信超时设置, 单位为10ms, 仅主站需要设置;

M8183~M8190: 通信出错标志, M8183对应第0号站点(主站), M8184对应第1号站点, 依次类推, M8190对应第7号站点;

M8191: 正在执行数据传送。

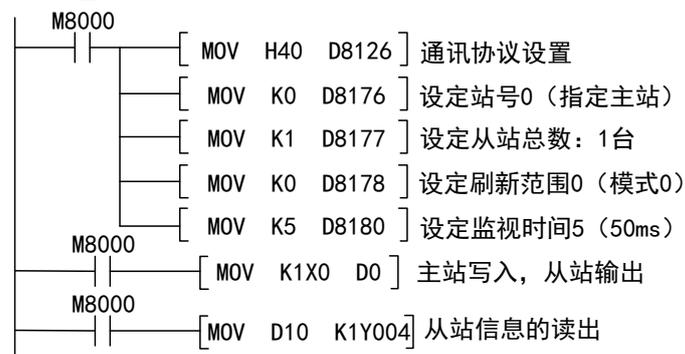
D8063	串行通信出错代码(见通信错误码一览表)
D8070	并联储机错误时间, 默认为500ms
D8173	本站站号设定状态
D8174	通讯子站设定状态
D8175	刷新范围设定状态
D8201	当前连接扫描时间
D8202	最大连接时间
D8203	主站通讯错误次数
D8204	从站1通讯错误次数
D8205	从站2通讯错误次数
D8206	从站3通讯错误次数
D8207	从站4通讯错误次数
D8208	从站5通讯错误次数
D8209	从站6通讯错误次数
D8210	从站7通讯错误次数
D8211	主站通讯错误代码
D8212	从站1通讯错误代码
D8213	从站2通讯错误代码
D8214	从站3通讯错误代码
D8215	从站4通讯错误代码
D8216	从站5通讯错误代码
D8217	从站6通讯错误代码
D8218	从站7通讯错误代码

5

通信功能

2) N: N通讯实例参考:

主站程序:



从站程序:



特别说明: XP型COM2/COM3也可用N:N协议, 使用方法与COM1相似, 只需要将通信协议寄存器进行相应的替换即可。例如: 使用COM2口作为N:N协议主站, 需设置D8266=40h, 其它寄存器无需修改。需要注意的是: 因为寄存器共用, 最多只能有一路COM口设定为N:N或者是N:N增强协议, 否则会导致未知错误。

5.5 N:N增强协议

N:N增强协议是相比于N:N协议可联网的PLC台数由8台增加至16台，并且可交换的数据量也得到扩充。

1) 通信原理:

通讯硬件设置：波特率固定为38400bps，通信格式7E1。

PLC运行时，即可实现多台PLC间互相交换数据，用户程序在本PLC内部特定的数据区可读取到其他PLC的发送状态数据；本机用户程序将需要广播的数据复制到特定数据单元，便可让其他PLC读取。

主机（地址为0）	数据区0：主机可写，其它机器读
从机1数据（地址为1）	数据区1：从机1可写，其它机器读
从机2数据（地址为2）	数据区2：从机2可写，其它机器读
从机3数据（地址为3）	数据区3：从机3可写，其它机器读
从机4数据（地址为4）	数据区4：从机4可写，其它机器读
从机5数据（地址为5）	数据区5：从机5可写，其它机器读
从机6数据（地址为6）	数据区6：从机6可写，其它机器读
从机7数据（地址为7）	数据区7：从机7可写，其它机器读
从机8数据（地址为8）	数据区8：从机8可写，其它机器读
从机9数据（地址为9）	数据区9：从机9可写，其它机器读
从机10数据（地址为10）	数据区10：从机10可写，其它机器读
从机11数据（地址为11）	数据区11：从机11可写，其它机器读
从机12数据（地址为12）	数据区12：从机12可写，其它机器读
从机13数据（地址为13）	数据区13：从机13可写，其它机器读
从机14数据（地址为14）	数据区14：从机14可写，其它机器读
从机15数据（地址为15）	数据区15：从机15可写，其它机器读

N:N增强协议相比于N:N协议刷新模式固定不可选，各台联网的PLC用于数据交互的M元件和D元件规划如下表所示。

站点号	软元件	
	位软元件（M）	字软元件（D）
	64个	32个
第0号	M500到M563	D200到D231
第1号	M570到M633	D240到D271
第2号	M640到M703	D280到D311
第3号	M710到M773	D320到D351
第4号	M780到M843	D360到D391
第5号	M850到M913	D400到D431
第6号	M920到M983	D440到D471
第7号	M990到M1053	D480到D511
第8号	M1060到M1123	D520到D551
第9号	M1130到M1193	D560到D591
第10号	M1200到M1263	D600到D631
第11号	M1270到M1333	D640到D671
第12号	M1340到M1403	D680到D711
第13号	M1410到M1473	D720到D751
第14号	M1480到M1543	D760到D791
第15号	M1550到M1613	D800到D831

2) 相关寄存器定义:

D8126：设为41h表示N:N增强协议主站；设为14h表示N:N增强协议从站。

D8176：站点号，范围0~15，0表示主站点；

D8177：从站点的总数，范围1~15，仅主站需要设置；

D8179：重试次数设定，仅主站需要设置，默认3；

D8180：通信超时设置，单位为10ms，仅主站需要设置，默认5；

M8183~M8190 / M8310~M8317: 通信出错标志, M8183对应第0号站点(主站), M8184对应第1号站点, 依次类推, M8317对应第15号站点;

其它相关寄存器见于下表

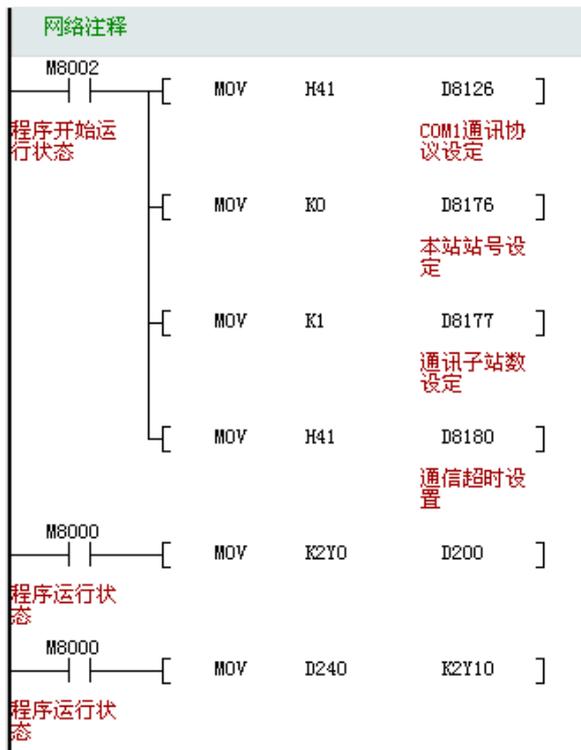
D8063	串行通信出错代码(见通信错误码一览表)
D8070	并联储机错误时间, 默认为500ms
D8173	本站站号设定状态
D8174	通讯子站设定状态
D8201	当前连接扫描时间
D8202	最大连接时间
D8203	主站通讯错误次数
D8204	从站1通讯错误次数
D8205	从站2通讯错误次数
D8206	从站3通讯错误次数
D8207	从站4通讯错误次数
D8208	从站5通讯错误次数
D8209	从站6通讯错误次数
D8210	从站7通讯错误次数
D8211	从站8通讯错误次数
D8212	从站9通讯错误次数
D8213	从站10通讯错误次数
D8214	从站11通讯错误次数
D8215	从站12通讯错误次数
D8216	从站13通讯错误次数
D8217	从站14通讯错误次数
D8218	从站15通讯错误次数

5
通信功能

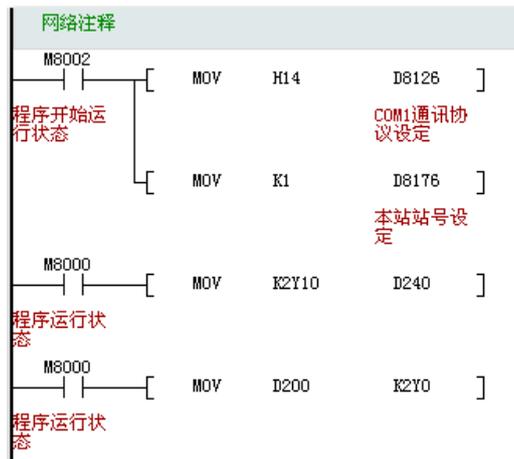
3) N:N增强协议范例:

通信要求: 通过N:N增强协议交换两台PLC的Y0-Y7和Y10-Y17的值。可以按照以下编写梯形图。

主机:



从机:



5.6 计算机链接协议

硬件配置与软件设置:

该协议只能用于COM1，设置COM1计算机链接协议有两种方法：一、设置D8126=06h，PLC即为计算机链接协议。二、D8120的bit14 = 1，波特率及通信格式由D8120决定。

协议说明:

1) 数据序列

格式	控制代码	站号	PC号	命令	消息等待时间	内容数据	[校验和]	[LF CR]
字节数	1	2	2	2	1	X	2	2

a) 控制代码

信号	代码 (16进制)	描述
STX	02H	文本起点
ETX	03H	文本终点
EOT	04H	传送结束
ENQ	05H	询问
ACK	06H	确认
LF	0AH	换行
CL	0CH	清除
CR	0DH	回车
NAK	15H	不确认

b) 站号: 即地址, 用来区分本PLC与其它PLC的通信, D8121设定, 取值00~0FH。

c) PC号: A系列用, FX系列固定为“FF”。

d) 命令:

			命令		描述	每一通讯中单元的最大数
			符号	ASCII代码		
软元件存储器	批读	位单元	BR	42H,52H	读一组位软元件 (X,Y,M,S,T,C), 结果以1点为单元	256
		字单元	WR	57H,52H	读一组位软元件 (X,Y,M,S), 结果以16点为单元	32字 512点
					读一组字软元件 (D,TN,CN), 结果以1元件为单元	64点
	批写	位单元	BW	42H,57H	写一组位软元件 (X,Y,M,S,T,C), 结果以1点为单元	160点
		字单元	WW	57H,57H	写一组位软元件 (X,Y,M,S), 结果以16点为单元	10字 160点
					写一组字软元件 (D,TN,CN), 结果以1元件为单元	64点
	测试	位单元	BT	42H,54H	有选择地以1点为单元设定/复位单独位软元件 (X,Y,M,S,T,C)	20点
		字单元	WT	57H,54H	有选择地以16点为单元设定/复位位软元件 (X,Y,M,S)	10字 160点
					有选择地以1元件为单元写字软元件 (D,TN,CN(除高速C200~255计数器))	10点

		命令		描述	每一通讯中单元的最大数
		符号	ASCII代码		
PC	远程运行	RR	52H,52H	向可编程控制器发送远程运行/停止请求	/
	远程停止	RS	52H,53H		
	读PC类型	PC	50H,43H	读PC类型名称(代码)	
接地		GW	47H,57H	对所有连接的可编程控制器设定/复位接地标记(对FX系列为M8126)	1点
接通要求				仅在1:1系统配置有可能从可编程控制器发送请求	最大64字
环路回送测试		TT	54H,54H	从计算机接收的字符直接被发送回到计算机	254字符

注：C字元件16位，32位要分开写，不可连写。

e) 消息等待时间

PLC收到消息后到回应前的最小延时，取值“0”~“F”，代表0~150ms。

f) 内容数据

例如序列：“M00160510101”：

“M0016”：表示从M0016开始；

“05”：表示操作5个变量；

“10101”：表示变量值。

g) 校验和

可选，是否添加由D8120的b13决定。

h) LF CR

可选，是否添加由所选协议来定，协议选定由D8120的b15决定。

2) 命令详解

a) 位元件的成批读(BR命令)

计算机命令(帧最小15字节)：

序号	名称	数据长度	例子
1	ENQ	1	\ENQ
2	站号	2	05
3	PC号	2	FF
4	命令BR	2	BR
5	消息等待时间	1	A
6	头元件	5	X0040
7	元件数N(1~255)	2	05
8	和校验(可选)	2	47
9	CR, LF(可选)	2	\CR\LF

通信数据例子：\ENQ05FFBRAX00400547\CR\LF

PLC响应：

序号	名称	数据长度	例子
1	STX	1	\STX
2	站号	2	05
3	PC号	2	FF
4	指定元件的数据	N	01101
5	ETX	1	\ETX
6	和校验(可选)	2	E7
7	CR, LF(可选)	2	\CR\LF

通信数据例子：\STX05FF01101\ETXE7\CR\LF

计算机确认:

序号	名称	数据长度	例子
1	ACK	1	\ACK
2	站号	2	05
3	PC号	2	FF

通信数据例子: \ACK05FF

b) 字软元件的成批读 (WR命令)

计算机命令 (帧最小15字节):

序号	名称	数据长度	例子
1	ENQ	1	\ENQ
2	站号	2	05
3	PC号	2	FF
4	命令WR	2	WR
5	消息等待时间	1	0
6	头元件	5	X0040
7	元件数 N(1~255/200/100)	2	02
8	和校验 (可选)	2	48
9	CR, LF (可选)	2	\CR\LF

通信数据例子: \ENQ05FFWR0X00400248\CR\LF

PLC响应:

序号	名称	数据长度	例子
1	STX	1	\STX
2	站号	2	05
3	PC号	2	FF
4	指定元件数据	N*4(16位元件)/ N*8(32位元件)	1234ABCD
5	ETX	1	\ETX
6	和校验 (可选)	2	08
7	CR, LF (可选)	2	\CR\LF

通信数据例子: \STX05FF1234ABCD \ETXC08\CR\LF

计算机确认:

序号	名称	数据长度	例子
1	ACK	1	\ACK
2	站号	2	05
3	PC号	2	FF

通信数据例子: \ACK05FF。

c) 位软元件的成批写 (BW命令)

计算机命令 (帧最小16字节):

序号	名称	数据长度	例子
1	ENQ	1	\ENQ
2	站号	2	05
3	PC号	2	FF
4	命令BW	2	BW
5	消息等待时间	1	0
6	头元件	5	M0903
7	元件数N(1~160)	2	05
8	指定元件的数据	N	01101
9	和校验 (可选)	2	2B

序号	名称	数据长度	例子
10	CR, LF (可选)	2	\CRLF

通信数据例子: \ENQ05FFBW0M090305011012B\CRLF

PLC响应:

序号	名称	数据长度	例子
1	ACK	1	\ACK
2	站号	2	05
3	PC号	2	FF

通信数据例子: \ACK05FF。

d) 字软元件的成批写 (WW命令)

计算机命令 (帧最小19字节):

序号	名称	数据长度	例子
1	ENQ	1	\ENQ
2	站号	2	05
3	PC号	2	FF
4	命令WW	2	WW
5	消息等待时间	1	0
6	头元件	5	M0640
7	元件数N(1~64)	2	02
8	指定元件的数据	N*4(16位元件)/ N*8(32位元件)	2347AB96
9	和校验 (可选)	2	0A
10	CR, LF (可选)	2	\CRLF

通信数据例子: \ENQ05FFWW0M0640022347AB960A\CRLF

PLC响应:

序号	名称	数据长度	例子
1	ACK	1	\ACK
2	站号	2	05
3	PC号	2	FF

通信数据例子: \ACK05FF。

e) 位软元件测试 (BT命令)

计算机命令 (帧最小16字节):

序号	名称	数据长度	例子
1	ENQ	1	\ENQ
2	站号	2	05
3	PC号	2	FF
4	命令BT	2	BT
5	消息等待时间	1	0
6	元件数N(1~20)	2	03
7	元件1	5	M0500
	元件1的数据 (0/1)	1	1
	元件2	5	S0100
	元件2的数据 (0/1)	1	0
	元件3	5	Y0001
	元件3的数据 (0/1)	1	1
	...		
8	和校验 (可选)	2	EC
9	CR, LF (可选)	2	\CRLF

通信数据例子: \ENQ05FFBT003M05001S01000Y00011EC\CRLF

PLC响应:

序号	名称	数据长度	例子
1	ACK	1	\ACK
2	站号	2	05
3	PC号	2	FF

通信数据例子: \ACK05FF。

f) 字软元件测试 (WT命令)

计算机命令 (帧最小19字节):

序号	名称	数据长度	例子
1	ENQ	1	\ENQ
2	站号	2	05
3	PC号	2	FF
4	命令WT	2	WT
5	消息等待时间	1	0
6	元件数N(1~10)	2	03
7	元件1	5	D0500
	元件1的数据 (0/1)	4	1234
	元件2	5	Y0100
	元件2的数据 (0/1)	4	BCA9
	元件3	5	CN100
	元件3的数据 (0/1)	4	0064
	...		
8	和校验 (可选)	2	07
9	CR, LF (可选)	2	\CR\LF

通信数据例子: \ENQ05FFWT003D05001234Y0100BCA9CN100006407\CR\LF。

PLC响应:

序号	名称	数据长度	例子
1	ACK	1	\ACK
2	站号	2	05
3	PC号	2	FF

通信数据例子: \ACK05FF。

g) 远程运行/停止 (RR/RS命令)

计算机命令(帧最小8字节):

序号	名称	数据长度	例子
1	ENQ	1	\ENQ
2	站号	2	05
3	PC号	2	FF
4	命令RR或RS	2	RR
5	消息等待时间	1	0
6	和校验 (可选)	2	C5
7	CR, LF (可选)	2	\CR\LF

通信数据例子: \ENQ05FFRR0C5\CR\LF。

PLC响应:

序号	名称	数据长度	例子
1	ACK	1	\ACK
2	站号	2	05
3	PC号	2	FF

通信数据例子：\ACK05FF。

有效执行命令条件：

远程运行：可编程控制器应处于停止状态；

远程停止：可编程控制器应为强制运行模式。

注意：断电后强制运行模式不会恢复。可编程控制器在强制运行模式时，若电源被关闭后再打开，特殊辅助继电器M8035、M8036、M8037都会复位到关，且可编程控制器保持停止。

h) 读取可编程控制器类型（PC命令）

计算机命令(帧最小8字节)：

序号	名称	数据长度	例子
1	ENQ	1	\ENQ
2	站号	2	0F
3	PC号	2	FF
4	命令PC	2	PC
5	消息等待时间	1	0
6	和校验（可选）	2	C5
7	CR, LF（可选）	2	\CR\LF

通信数据例子：\ENQ0FFPC0C5\CR\LF。

PLC响应：

序号	名称	数据长度	例子
1	STX	1	\STX
2	站号	2	0F
3	PC号	2	FF
4	内容PC号	2	02
5	ETX	1	\ETX
6	和校验（可选）	2	C5
7	CR, LF（可选）	2	\CR\LF

通信数据例子：\STX0FFRR\ETXC5\CR\LF。

计算机确认：

序号	名称	数据长度	例子
1	ACK	1	\ACK
2	站号	2	05
3	PC号	2	FF

通信数据例子：\ACK05FF。

i) 全局功能（GW命令）

计算机命令(帧最小9字节)：

序号	名称	数据长度	例子
1	ENQ	1	\ENQ
2	站号	2	FF
3	PC号	2	FF
4	命令GW	2	GW
5	消息等待时间	1	0
	控制标志（1：打开， 0：关闭）	1	1
6	和校验（可选）	2	17
7	CR, LF（可选）	2	\CR\LF

通信数据例子：\ENQFFFFGW01C5\CR\LF。

PLC不做响应：

注：站号FF表示所有站，打开全局变量标志后M8126置位，关闭全局变量标志后M8126复位。

疑问：无论是否打开或关闭全局变量标志，均无法通过地址“FF”控制或设置PLC，仍然只能通过PLC具体站号来控制或设置PLC。

j) 环路回送测试 (TT命令)

计算机命令(帧最小11字节):

序号	名称	数据长度	例子
1	ENQ	1	\ENQ
2	站号	2	05
3	PC号	2	FF
4	命令TT	2	TT
5	消息等待时间	1	0
6	字符数N	2	08
7	字符串	N	12345678
8	和校验 (可选)	2	FB
9	CR, LF (可选)	2	\CR\LF

通信数据例子: \ENQ05FFTT00812345678FB\CR\LF。

PLC响应:

序号	名称	数据长度	例子
1	STX	1	\STX
2	站号	2	05
3	PC号	2	FF
4	字符数	2	08
5	内容PC号	2	02
6	ETX	1	\ETX
7	和校验 (可选)	2	C5
8	CR, LF (可选)	2	\CR\LF

通信数据例子: \STX05FF0812345678\ETXC5\CR\LF。

k) 错误响应

PLC错误响应:

序号	名称	数据长度	例子
1	NAK	1	\NAK
2	站号	2	05
3	PC号	2	FF
4	错误码	2	02
5	CR, LF (可选)	2	\CR\LF

通信数据例子: \NAK 05FF02\CR\LF。

注:

I) 错误响应不存在和检验:

II) 错误码:

“02” :Check sum error, 和校验错误;

“03” :COMM Mode error, 通信格式错误;

“06” :char buf error, 字符区错误;

“07” :char error, 字符错误不在0~F间;

“10” :PC code error, PC号错误;

“18” :control error, 远程控制错误。

3) 接通要求 (PLC作主站发送数据到计算机)

若D8128非0, PLC将主动通过端口发送数据。发送数据的内容为D8127指定的软元件, 数据长度由D8128指定。D8127与D8128之和不超过8000, 即不能访问D8000以后的数据。

数据发送中, M8127置位, 发送完成后M8127复位。数据范围错误, M8128置位, 不发送数据。M8129由为ON, 数据寄存器按8位处理, OFF, 数据寄存器按16位处理。

例子:

用户程序: D8127 = 5, D8128 = 3, M8129 = OFF;

D5 = 501h, D6 = 602h, D7 = 703h。

PLC处理: M8127 = ON;

通过串口1发送如下数据: \STX05FE 050106020703\ETX;

M8127 = OFF。

PLC主动发送的数据:

序号	名称	数据长度	例子
1	STX	1	\STX
2	站号	2	05
3	PC号 (固定为FE)	2	FE
4	内容(D元件数据)	N*4 (或2)	050106020703
5	ETX	1	\ETX
6	和校验 (可选)	2	C5
7	CR, LF (可选)	2	\CR\LF

注意: XP型COM2/COM3也可用计算机链接协议, 使用方法与COM1相同, 设置M、D请参考相应的串口设定。

5

5.7 MODBUS协议说明

通信功能

注意: XP型COM2/COM3口使用MODBUS协议时, 使用RS2和MODBUS2指令, 其他使用方法与COM1相同, 设置M、D请参考相应的串口设定。

MODBUS通讯的底层为RS485信号, 采用双绞线进行联接就可以了, 因此传输距离较远, 可达1000米, 抗干扰性能比较好, 且成本低, 在工业控制设备的通讯中被广泛使用, 现在众多厂家的变频器、控制器都采用了该协议。

传送数据格式有HEX码数据和ASCII码两种, 分别称为MODBUS-RTU和MODBUS-ASC 协议, 前者为数据直接传送, 而后者需将数据转换为ASCII码后传送, 因此MODBUS-RTU协议的通讯效率较高, 处理简单, 使用得更多。

MODBUS为单主多从通讯方式, 采用的是主问从答方式, 每次通讯都是由主站首先发起, 从站被动应答。因此, 如变频器之类的被控设备, 一般内置的是从站协议, 而PLC之类的控制设备, 则需具有主站协议、从站协议。

5.7.1 MODBUS 主站通讯应用

在H1U/H2U系列PLC的COM1通讯口可以使用MODBUS-RTU和MODBUS-ASC指令, 只需给系统寄存器D8126设置相应的数值就可使用了。

MODBUS指令 (主站) 的通讯有RS扩展指令、MODBUS指令两种, 分别解释如下:

1) RS指令

硬件配置与软件设置:

RS指令只能用于COM1通信, 连接方式为全双工或半双工, 支持本机标准通信口, 也支持所有的通讯扩展卡。

可以设置D8126=10h来启动。

本协议通信格式及波特率由D8120设定。D8120的定义见下表:

位号	名称	内容	
		0 (OFF)	1 (ON)
b0	数据长	7位	8位
b2b1	奇偶性	00:无 01:奇校验 (ODD) 11:偶校验 (EVEN)	
b3	停止位	1位	2位

位号	名称	内容	
		0 (OFF)	1 (ON)
b7b6b5b4	波特率 (bps)	0011: 300 0100: 600 0101: 1200 0110: 2400 0111: 4800 1000: 9600 1001: 19200 1010: 38400 1011: 57600 1100: 115200	
b8	起始符	无	有, D8124为起始符
b9	终止符	无	有, D8125为起始符
b10	全半双工	全双工	半双工
b11	保留	不可使用	
b12	保留	不可使用	
b13	和校验	不附加	附加
b14	协议	不使用	使用
b15	控制顺序	方式1	方式4

注: b12~b14只用于计算机链接协议, 与RS指令无关。

协议说明:

RS指令格式为:

RS(TXDADDR, TXDLEN, RXDADDR, RXDLEN);

TXDADDR: 要发送数据地址, 必须是D元件;

TXDLEN: 发送数据长度, 可以是变量和常数;

RXDADDR: 接收数据地址: 必须是D元件;

RXDLEN: 接收数据长度, 可以是变量和常数。

发送请求命令: M8122, 若程序把M8122置为ON, 并且RS指令被驱动, 即从TXDADDR指定的D元件地址起, 发送TXDLEN个数据到COM1 (若指定有起始符或中止符, 会一起发出)。发送完成后系统自动复位M8122。

接收标志: M8123, 接收数据完成后, M8123自动置为on, 复位将进入下一次接收状态。

接收超时: 若接收数据不足, 且时间大于设定时间 (D8129×10ms), 接收超时, M8129置为on。注意: 若一个数据都没有收到, M8129是不会置位的。用户需要根据收到的数据量 (D8123) 来判断通讯是否正常。

2) MODBUS指令

硬件配置与软件设置:

RS指令只能用于COM1通信, 若D8126=20h, COM1协议为MODBUS-RTU主站 (指令); 若D8126=30h, COM1协议为MODBUS-ASC主站 (指令);

协议说明:

MODBUS指令对串口COM1有效, 用户可通过MODBUS指令编程, 把PLC作为主站与MODBUS从站设备通信。

MODBUS指令有两种, 一种符合MODBUS RTU协议, 一种符合MODBUS ASC协议, 通过D8126确定。用哪种指令由从站所支持的协议格式定, 若从站两种协议都支持而用户要求较快速的通信, 建议选用RTU协议。两种协议只是通信格式不一样, 对用户编程都一样, 下面仅就RTU协议做说明。

MODBUS指令可以同时存在多条并且全部被驱动, 系统内部会协调指令的顺序执行, MODBUS协议要求无论写还是读, 从站均需要有应答 (广播除外)。一条MODBUS指令可能需要执行较长时间, 一般需要多个扫描周期。在一个扫描周期内, 指令被驱动, 但不一定被执行。

若存在多条MODBUS指令, 其执行顺序是这样的: 从开机开始, 扫描第一条被驱动的MODBUS指令, 若扫描到, 把该MODBUS的参数记录下来, 在后台执行。执行完后, 返回用户程序, 从刚执行的MODBUS指令位置开始扫描下一条被驱动的MODBUS指令并执行, 周而复始。

指令格式: RS(ADDR&CMD, REGADDR, REGLN, DATABUF)

ADDR&CMD: 从机地址和MODBUS功能码, 高8位表示从机地址, 即目标设备地址。低8位表示MODBUS功能码, 由标准MODBUS协议定义, 目前支持功能码有0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x0f, 0x10。具体含义请参照标准MODBUS协议或目标设备MODBUS协议。

REGADDR: 所要读或写的从机线圈 (1位) 或寄存器 (16位) 地址, 取值参考从机MODBUS协议。可为元件或常数。

REGLN: 所要读写的从机线圈或寄存器个数, 可为元件或常数。

DATABUF: 只能为D元件。本机用于存放数据的起始寄存器, 即数据缓冲区。缓冲区长度与REGLN相关, 至少取1。若

MODBUS命令为读，指令成功执行完后，把从机数据读到缓冲区中，若MODBUS命令为写，把缓冲区发送给从机。用户在设计程序时需要计算缓冲区长度，预留足够的寄存器作缓冲区。

相关状态标志

M8122: MODBUS指令执行状态指示，OFF时表示指令执行完毕，ON时为执行中。若M8122为OFF，且指令在一个扫描周期内能流有效，M8122置为ON，系统将会把指令参数记录下来，转入后台执行该指令的通信要求。通信执行完后，当再次运行到此指令的位置时，无论该指令能流是否有效，均会把M8122复位为OFF，立即扫描下一条能流有效的指令，记录指令参数并转入后台执行该指令的通信要求。

M8123: 指令通信情况指示，ON表示通信异常，OFF表示通信正常；

M8063: 指令错误指示，错误码存于D8063；

D8063: 错误码（见通信错误码一览表）。

注意：

读写寄存器时，H_{1U}最大支持一次读写50个十六位寄存器。H_{2U}满足MODBUS协议标准要求，最大支持一次读125个寄存器，写120个寄存器。若读写超出最大值，将报参数错。

读写线圈时，H_{1U}最大支持一次读写800个线圈。H_{2U}满足MODBUS协议标准要求，最大支持一次读2000个线圈，写1968个线圈。若读写超出最大值，将报参数错。

例子1: 不断的读从机地址为100的寄存器，数据存于D10。

初始化:

D8126 = H0020 设定通信协议为MODBUS RTU指令；

D8120 = H0081 设定COM1通信格式为：9600，8N1；

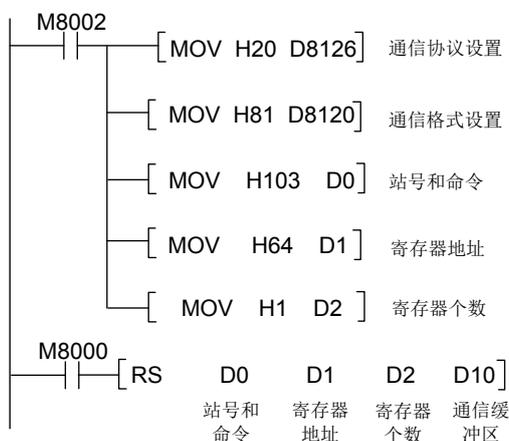
D0 = H0103 Addr&Cmd从机地址为01和MODBUS命令码为03，读寄存器；

D1 = H0064 RegAddr要操作的从机的寄存器地址；

D2 = H0001 RegLen要操作的寄存器的个数；

D10 Buf本PLC数据缓冲区，本例中读命令通信成功后数据存于D10。

梯形图如下：



执行结果：开机后，PLC不断读从机地址为100的寄存器，通过COM1发送以下一帧数据（16进制）：01 03 00 64 00 01 C5 D5

01: 代表从机地址，D0的高8位；

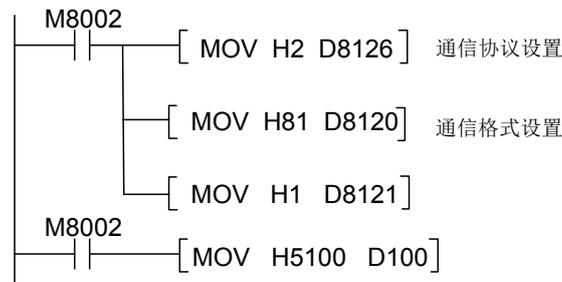
03: MODBUS命令码，D0的低8位，意义为读从机寄存器；

00 64: 所要读从机寄存器地址，D1的值；

00 01: 所要读的寄存器个数，D2的值；

C5 D5: CRC校验码。

若从机也是H_{1U}/H_{2U}系列PLC，设定为MODBUS RTU从站协议，梯形图如下：



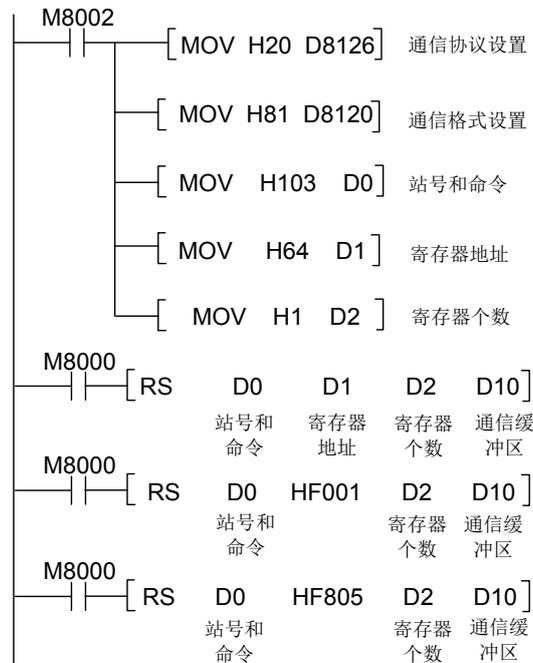
从机正确响应数据帧（16进制）：01 03 02 51 00 85 D4

从机把D100（寄存器地址为H0064）发给主机：

- 01: 代表从机地址；
- 03: MODBUS命令码；
- 02: 表示回复2个字节的有效数据；
- 51 00: 寄存器数据，即D100 的值；
- 85 D4: CRC校验码。

例子2: 用三条MODBUS指令，分别的读从机地址为H0064，F001和F805的寄存器，数据存于D10，D20和D30中。

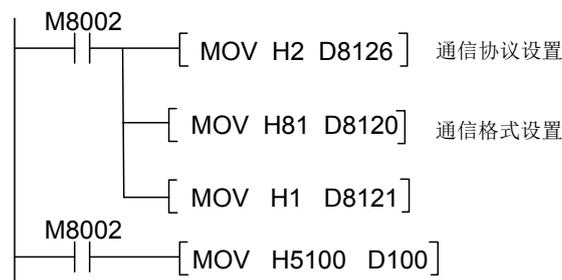
梯形图如下：



执行结果，PLC通过串口COM1依次循环发送以下三帧数据（16进制）：

- 01 03 00 64 00 01 C5 D5
- 01 03 F0 01 00 01 E6 CA
- 01 03 F8 05 00 01 A5 6B

回复：从机仍然是H1U/H2U系列PLC，设定为MODBUS RTU从站协议，梯形图如下：



从机响应:

对第一帧数, 从机响应数据帧为(16进制): 01 03 02 51 00 85 D4

意义是: 从机把D100(D100寄存器的地址为H0064)的值H5100发给主机;

对第二帧数, 从机响应数据帧为(16进制): 01 03 02 00 00 B8 44

意义是: 从机把T1(T1寄存器地址为F001, 参见5.7.2节中的《MODBUS从站通讯应用》)的值H0000发给主机;

对第三帧数, 从机响应数据帧为(16进制): 01 83 02 C0 F1

意义是: 读寄存器错误。

01: 从站地址;

83: 读寄存器错误;

02: 错误码, 地址错误, 原因是地址HF805的寄存器不存在;

C0 F1: CRC校验码。

5.7.2 MODBUS 从站通讯应用

有些工业应用中, PLC控制器作为工业自动化系统的一部分, 要接受自动化控制网络的监控, 典型的上位机有如DCS、运行组2态软件的工业PC等, 作为监控主机, 以MODBUS主站协议与PLC等设备进行通讯, 此时PLC的通讯口需要以MODBUS从站协议与上位机通讯。H1U/H2U系列PLC内置有MODBUS-RTU从站协议和MODBUS-ASC从站协议, 并且在COM0和COM1口都可以运行该协议。

3) 硬件配置与软件设置:

COM0设置: 若D8116=02h, COM0协议为MODBUS-RTU从站; 若D8116=03h, COM0协议为MODBUS-ASC从站;

COM1设置: 若D8126=02h, COM1协议为MODBUS-RTU从站; 若D8126=03h, COM1协议为MODBUS-ASC从站;

4) 协议说明:

5 MODBUS从站协议包括MODBUS RTU协议(以下简称RTU协议)和MODBUS ASC协议(以下简称ASC协议), 两者区别在与数据链路, 通信传送的数据RTU协议为真实数据, ASC协议传送的数据是转换为ASC码的数据。另外两者在帧结构上也有区别, RTU协议是以时间来区分数据帧的, 若通信中有3.5个字节的时间没有接收到数据, 则认为对方数据传送完毕; ASC协议即是以ASC码“:”为帧起始符, 以\r\n(0D0Ah)为帧结束符, 从通信效率来看, RTU协议高于ASC协议, 大概RTU协议大概为ASC协议到两倍。具体可参照标准MODBUS协议相关文档, 这些文档是开放到, 可在网上下载到或到MODBUS相关官方网上下载。

与控制相关的变量及标志表:

变量	说明	备注
D8120	通讯格式设定	例: 81h: 9600bps, 8N1 91h: 19200bps, 8N1
D8121	设定本PLC的从站地址	程序运行中随时更新有效
D8126	从站协议设定	02h: MODBUS RTU从站 03h: MODBUS ASC从站
M8063	MODBUS通讯错误指示	只读, 用户程序清除或关机到开机时清除
D8063	通信错误码(见通信错误码一览表)	只读, 用户程序清除或关机到开机时清除

5) MODBUS从站通讯协议说明(MODBUS功能码及数据编址)

H1U/H2U作为MODBUS从站时, 支持MODBUS协议功能码0x01, 0x03, 0x05, 0x06, 0x0f, 0x10; 通过这些功能码, 可读写的线圈有M, S, T, C, X(只读), Y等变量; 寄存器有D, T, C。

MODBUS通讯主机在访问(读取或改写)PLC从机的内部变量时, 必须遵循如下的通讯命令帧定义, 以及变量地址索引方法, 才能进行正常的通讯操作。

MODBUS帧格式(以MODBUS-RTU为例)

a) 功能码0x01(01)、0x02(02): 读线圈

请求帧格式: 从机地址+0x01+线圈起始地址+线圈数量+CRC检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1个字节	取值1~247, 由D8121设定
2	0x01(功能码)	1个字节	读线圈
3	线圈起始地址	2个字节	高位在前, 低位在后, 见线圈编址
4	线圈数量	2个字节	高位在前, 低位在后(N)
5	CRC校验	2个字节	低位在前, 高位在后

响应帧格式：从机地址+0x01+字节数+线圈状态+CRC检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1个字节	取值1~247, 由D8121设定
2	0x01 (功能码)	1个字节	读线圈
3	字节数	1个字节	值: $[(N+7) / 8]$
4	线圈状态	$[(N+7) / 8]$ 个字节	每8个线圈合为一个字节, 最后一个若不足8位, 未定义部分填0。前8个线圈在第一个字节, 地址最小的线圈在最低位。依次类推
5	CRC校验	2个字节	低位在前, 高位在后

错误响应：见错误响应帧

注：N, H_{2U}最大为2000, H_{1U}最大为800。

b) 功能码0x03 (03)、0x04 (04)：读寄存器

请求帧格式：从机地址+0x03+寄存器起始地址+寄存器数量+CRC检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1个字节	取值1~247, 由D8121设定
2	0x03 (功能码)	1个字节	读寄存器
3	寄存器起始地址	2个字节	高位在前, 低位在后, 见寄存器编址
4	寄存器数量	2个字节	高位在前, 低位在后 (N)
5	CRC校验	2个字节	低位在前, 高位在后

响应帧格式：从机地址+0x03+字节数+寄存器值+CRC检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1个字节	取值1~247, 由D8121设定
2	0x03 (功能码)	1个字节	读寄存器
3	字节数	1个字节	值: $N*2$
4	寄存器值	$N*2$ 个字节	每两字节表示一个寄存器值, 高位在前低位在后。寄存器地址小的排在前面
5	CRC校验	2个字节	低位在前, 高位在后

错误响应：见错误响应帧

注：N, H_{2U}最大为125, H_{1U}最大为50。

c) 功能码0x05 (05)：写单线圈

请求帧格式：从机地址+0x05+线圈地址+线圈状态+CRC检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1个字节	取值1~247, 由D8121设定
2	0x05 (功能码)	1个字节	写单线圈
3	线圈地址	2个字节	高位在前, 低位在后, 见线圈编址
4	线圈状态	2个字节	高位在前, 低位在后。非0即为有效
5	CRC校验	2个字节	低位在前, 高位在后

响应帧格式：从机地址+0x05+线圈地址+线圈状态+CRC检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1个字节	取值1~247, 由D8121设定
2	0x05 (功能码)	1个字节	写单线圈
3	线圈地址	2个字节	高位在前, 低位在后, 见线圈编址
4	线圈状态	2个字节	高位在前, 低位在后。非0即为有效
5	CRC校验	2个字节	低位在前, 高位在后

错误响应：见错误响应帧

d) 功能码0x06 (06)：写单个寄存器

请求帧格式：从机地址+0x06+寄存器地址+寄存器值+CRC检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1个字节	取值1~247, 由D8121设定

序号	数据(字节)意义	字节数量	说明
2	0x06 (功能码)	1个字节	写单寄存器
3	寄存器地址	2个字节	高位在前, 低位在后, 见寄存器值编址
4	寄存器值	2个字节	高位在前, 低位在后. 非0即为有效
5	CRC校验	2个字节	低位在前, 高位在后

响应帧格式: 从机地址+0x06+寄存器地址+寄存器值+CRC检验。

序号	数据(字节)意义	字节数量	说明
1	从机地址	1个字节	取值1~247, 由D8121设定
2	0x06 (功能码)	1个字节	写单寄存器
3	寄存器地址	2个字节	高位在前, 低位在后, 见寄存器编址
4	寄存器值	2个字节	高位在前, 低位在后. 非0即为有效
5	CRC校验	2个字节	低位在前, 高位在后

错误响应: 见错误响应帧。

e) 功能码0x0f (15): 写多个线圈

请求帧格式: 从机地址+0x0f+线圈起始地址+线圈数量+字节数+线圈状态+CRC检验。

序号	数据(字节)意义	字节数量	说明
1	从机地址	1个字节	取值1~247, 由D8121设定
2	0x0f (功能码)	1个字节	写多个单线圈
3	线圈起始地址	2个字节	高位在前, 低位在后, 见线圈编址
4	线圈数量	2个字节	高位在前, 低位在后. N, 最大为1968
5	字节数	1个字节	值: $[(N+7)/8]$
6	线圈状态	$[(N+7)/8]$ 个字节	每8个线圈合为一个字节, 最后一个若不足8位, 未定义部分填0. 前8个线圈在第一个字节, 最地址最小的线圈在最低位. 依次类推
7	CRC校验	2个字节	低位在前, 高位在后

响应帧格式: 从机地址+0x0f+线圈起始地址+线圈数量+CRC检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1个字节	取值1~247, 由D8121设定
2	0x0f (功能码)	1个字节	写多个单线圈
3	线圈起始地址	2个字节	高位在前, 低位在后, 见线圈编址
4	线圈数量	2个字节	高位在前, 低位在后。
5	CRC校验	2个字节	低位在前, 高位在后

错误响应: 见错误响应帧。

注: N, H₂₀最大为1968, H₁₀最大为800。

f) 功能码0x10 (16): 写多个寄存器

请求帧格式: 从机地址+0x10+寄存器起始地址+寄存器数量+字节数+寄存器值+CRC检验。

序号	数据(字节)意义	字节数量	说明
1	从机地址	1个字节	取值1~247, 由D8121设定
2	0x10 (功能码)	1个字节	写多个寄存器
3	寄存器起始地址	2个字节	高位在前, 低位在后, 见寄存器编址
4	寄存器数量	2个字节	高位在前, 低位在后. N, 最大为120
5	字节数	1个字节	值: $N*2$
6	寄存器值	$N*2 (N*4)$	
7	CRC校验	2个字节	低位在前, 高位在后

响应帧格式: 从机地址+0x10+寄存器起始地址+寄存器数量+CRC检验。

序号	数据(字节)意义	字节数量	说明
1	从机地址	1个字节	取值1~247, 由D8121设定
2	0x10 (功能码)	1个字节	写多个寄存器
3	寄存器起始地址	2个字节	高位在前, 低位在后, 见寄存器编址

序号	数据(字节)意义	字节数量	说明
4	寄存器数量	2个字节	高位在前, 低位在后。N, 最大为120
5	CRC校验	2个字节	低位在前, 高位在后

错误响应: 见错误响应帧。

注: N, H_{2U}最大为120, H_{1U}最大为50。

g) 功能码0x60: 指定将要读取的PLC元件

HMI请求帧格式: 从机地址+0x60+数据块标识+位元件个数+寄存器个数+位元件地址+寄存器地址+CRC校验。

序号	数据(字节)意义	字节数量	说明
1	从站地址	1	取值1~247
2	0x60 (功能码)	1	QLink自定义
3	数据块标识	2	唯一, 不允许为0xffff。
4	位元件个数N	1	
5	寄存器个数N	1	
6	分散的位元件地址	N*2	高位在前, 低位在后, 见线圈编址。
7	分散的寄存器地址	N*2	高位在前, 低位在后, 见寄存器编址。
8	CRC校验	2	低位在前, 高位在后

PLC响应帧格式: 从机地址+0x60+数据块标识+位元件个数+寄存器个数+位元件的值+寄存器的值+CRC校验。

序号	数据(字节)意义	字节数量	说明
1	从站地址	1	取值1~247
2	0x60 (功能码)	1	QLink自定义
3	数据块标识	2	唯一, 在后面数据中有用到
4	位元件个数N	1	
5	寄存器个数N	1	
6	位元件的值	(N+7)/8	按主站指定地址
7	寄存器的值	N*2或者N*4	按主站指定地址
8	CRC校验	2	低位在前, 高位在后

错误响应: 见错误响应帧。

注: N值H_{2U}最大为120, H_{1U}最大为50。

h) 功能码0x61: 数据块读取请求

HMI请求帧格式: 从机地址+0x61+数据块标识+位元件个数+寄存器个数+CRC检验。

序号	数据(字节)意义	字节数量	说明
1	从站地址	1	取值1~247
2	0x61 (功能码)	1	QLink自定义
3	数据块标识	2	唯一, 在后面数据中有用到
4	位元件个数N	1	
5	寄存器个数N	1	
6	CRC校验	2	低位在前, 高位在后

PLC响应帧格式: 从机地址+0x60+数据块标识+位元件个数+寄存器个数+位元件的值+寄存器的值+CRC校验。

序号	数据(字节)意义	字节数量	说明
1	从站地址	1	取值1~247
2	0x61 (功能码)	1	QLink自定义
3	数据块标识	2	唯一, 在后面数据中有用到
4	位元件个数N	1	
5	寄存器个数N	1	
6	位元件的值	(N+7)/8	按主站指定地址
7	寄存器的值	N*2或者N*4	按主站指定地址
8	CRC校验	2	低位在前, 高位在后

错误响应: 见错误响应帧。

注：N的值H2U最大为120，H1U最大为50。

i) 错误响应帧

错误响应：从机地址+（功能码+0x80）+错误码+CRC校验。

序号	数据(字节)意义	字节数量	说明
1	从机地址	1个字节	取值1~247，由D8121设定
2	功能码+0x80	1个字节	错误功能码
3	错误码	1个字节	1~4
4	CRC校验	2个字节	低位在前，高位在后

变量编址

a) 线圈编址

线圈：指位变量，只有两种状态0和1。在本PLC中包含M，S，T，C，X，Y等变量。

变量名称	起始地址	线圈数量	说明
M0~3071	0 (0)	3072	
M8000~M8255	0x1F40 (8000)	256	
S0~S999	0xE000 (57344)	1000	
T0~T255	0xF000 (61440)	256	
C0~C255	0xF400 (62464)	256	
X0~X255	0xF800 (63488)	256	
Y0~Y255	0xFC00 (64512)	256	

b) 寄存器编址

寄存器：指16位或32位变量，在本PLC中，16位变量包含D，T，C0~199；32位变量为C200~255。

变量名称	起始地址	寄存器数量	说明
D0~D8255	0 (0)	8256	
T0~T255	0xF000 (61440)	256	
C0~C199	0xF400 (62464)	200	
C200~C255	0xF700 (63232)	56	32位寄存器

说明：

通过MODBUS访问C200~C255段32位寄存器时，一个寄存器作两寄存器看待，一个32位寄存器占用两个16寄存器空间。比如用户要读或写C205~C208这4个寄存器，MODBUS地址为0xF70A (0xF700+10)，寄存器数量8 (4*2)。

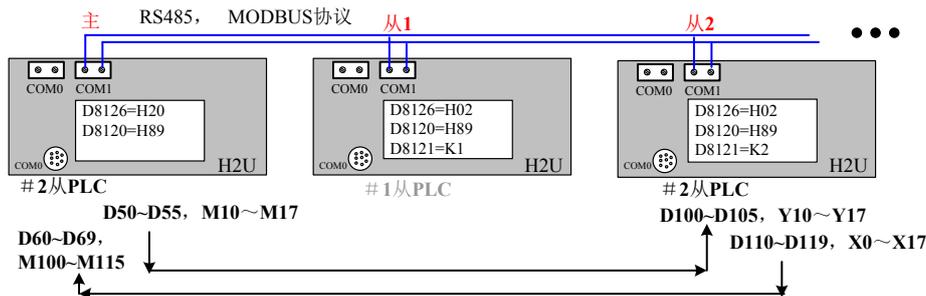
32位寄存器不支持写单个寄存器 (0x06) 功能码。

5.7.3 多台 PLC 之间采用 MODBUS 协议通讯的编程方法

对于有2台或更多的PLC通讯并机的系统，采用MODBUS协议的编程，具有简单灵活的特点，在有PLC+MDI等多种设备的组合的系统，更显方便。

MODBUS通讯的系统是一主多从方式，通讯所需的数据交换完全由主站发起，所有从站都是被动接收和响应，通讯相关的编程主要在主站的程序中进行，从站的通讯编程中，只需配置好通讯协议、通讯格式、本机站号即可，对通讯数据进行适当的处理即可。

例一：对于如下的多台PLC通讯联接，若要实现主PLC与#2从PLC之间的数据交换，如下图所示，交换的数据是：主 (D50~D55, M10~M17) → 从 (D100~D105, Y10~Y17)；从 (D110~D119, X0~X17) → 主 (D60~D69, M100~M115)。



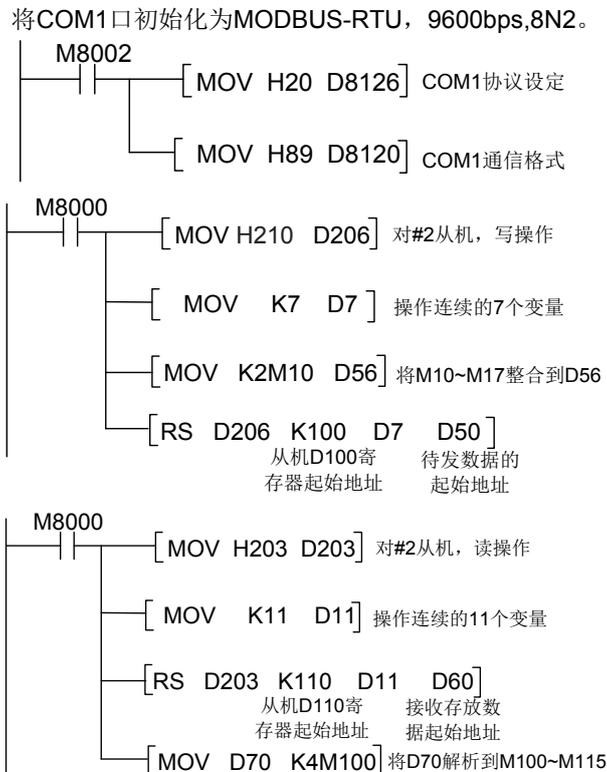
编程方法：

将主PLC的COM1通讯口配置为MODBUS主站协议，9600bps，8N2格式，数据的交换（读写）全部由主PLC完成。

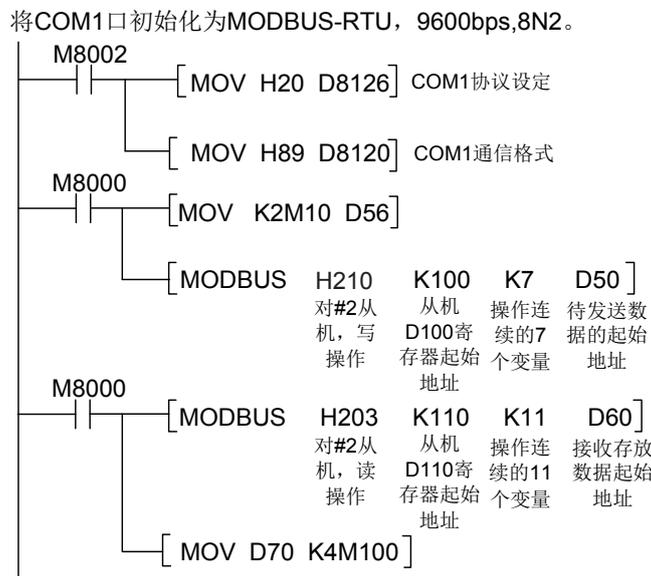
因其中还要部分位变量X、Y、M需要数据交换，将这些变量整合成D变量，在一片连续的D变量区域，成批交换，主从双方各自进行位变量的组合与解析，这样的交换效率高，编程简单。

这里将主站内的M10~M17变量组合成D56，这样主站要发送的数据为D100~D106共7个D变量；将从站内的X0~X17变量组合成D120，主站要读取的数据为D110~D120共11个D变量。

用RS扩展指令的主站编程举例为：



用MODBUS指令(V24120以上版本适用)的主站编程，完成同样的数据交换功能，语句精炼，可减少寄存器占用：



#2从站的编程相应要求是，将COM1端口配置为MODBUS-RTU从站，通讯格式与主站相同，即9600bps，8N2，将本机站号设置为2，及时刷新主站要读取的数据寄存器、主站由通讯写过来的数据的维护，编程举例如下：

将COM1口设置为MODBUS-RTU从站协议，9600bps,8N2，站号为2。



其他从站的编程方法，可参考上述从站的编程方法，注意设置的站号不要有重复的情况。

5.7.4 MODBUS 配置与编程方法

Modbus配置可实现对Modbus通讯数据的集中配置，将配置数据下载到Modbus主站。PLC单板软件根据配置信息实现与从站设备通讯，从而达到数据交换的目的；降低了Modbus的使用难度、减少了工作量、提升了用户体验的效果。

一、主站Modbus配置

1.1 配置界面

在“工程管理”窗口双击“Modbus配置”打开“Modbus配置”窗口。



配置操作：

新增：在配置数据末尾增加一条新的配置记录；

插入：在当前位置新增一条配置记录；

删除：删除选中的配置记录；

上移：调整选中行位置；

下移：调整选中行位置；

清除：清除窗口中所有配置信息；

确定：保存修改、关闭窗口；

取消：不保存修改、关闭窗口。

1.2 配置数据

1) 从站站号(H)：十六进制表示的从站站号，从站可以为PLC、MD、IS；

2) 通讯方式：通讯方式分为“触发”、“循环”；

触发：“触发条件”置“ON”通讯操作被触发；触发条件可以一下类型元件：M、C、T、S；

循环：“触发条件”为“空”，通讯操作被循环执行；

3) COM: 主站Modbus通讯串口；

4) 从站寄存器地址(H): 通讯操作从站端起始操作地址，用十六进制表示；

此文档只列出PLC编址，MDIS编址请参考相应设备的功能码说明；

PLC Modbus协议变量编址表：线圈编址

变量名称	起始地址	线圈数量	说明
M0~M3071	0(0)	3072	
M8000~M8256	0x1F40 (8000)	256	
S0~S999	0xE000 (57344)	1000	
T0~T255	0xF000 (61440)	256	
C0~C256	0xF400 (62464)	256	
X0~X255	0xF800 (63488)	256	
Y0~Y255	0xFC00 (64512)	256	

寄存器编址:

变量名称	起始地址	寄存器数量	说明
D0~D8255	0 (0)	8256	
T0~T255	0xF000 (61440)	256	
C0~C199	0xF400 (62464)	200	
C200~C255	0xF700 (63232)	56	32位寄存器

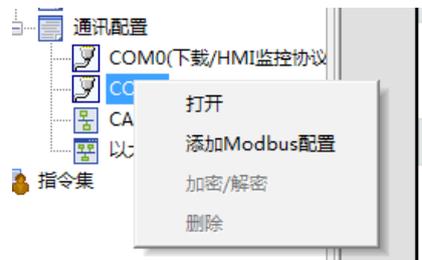
5) 数据长度: 读寄存器、写寄存器数据长度表示寄存器个数，即“从站寄存器地址”开始的n个寄存器；读线圈、写线圈数据长度表示线圈个数，即“从站寄存器地址”开始的n个线圈；

6) 主站缓冲区地址: 主、从站交换数据时主站端缓存区；缓冲区为PLC的D元件，元件个数有数据长度决定；

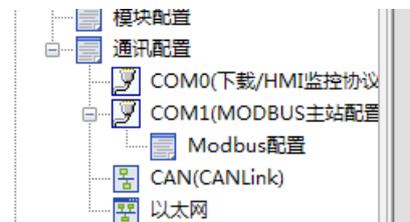
1.3 配置加密&解密

为方便用户对工程实施加密保护，提供了配置加密/解密功能。用户可根据需要进行设置，经过加密保护的配置参数，在重新下载到设备后，须按要求输入口令后才能查看。具体操作说明如下：

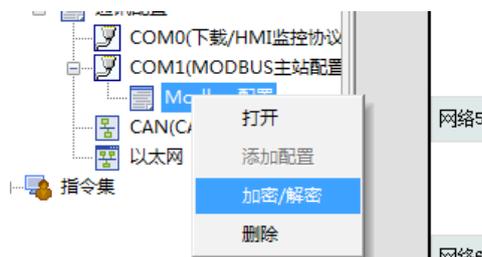
一、选择一个通讯口配置，在此以选择“COM1 (Modbus主站配置)”为例：



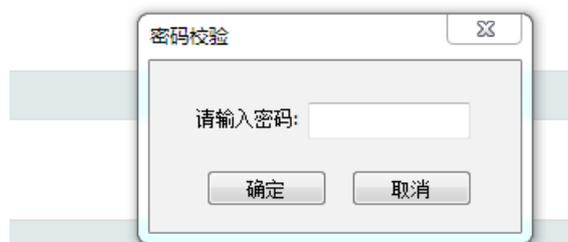
右键点击“添加Modbus配置”，添加后如下：



点击“Modbus配置”，右键进入“加密/解密”，进行具体的密码设置。



配置加密后，必须输入密码后才能查看。密码输入后原程序解密，通过上述路径，在“加密/解密”里选择解密，输入密码后实现解密。



【注意】加密后必须输入密码才可以查看配置，输入一次密码后配置将自动解密（即在该设备上以后不用再输入密码进行查看）。

1.4 主站通讯设置

1) 在“系统参数->COM配置->协议选择”中选中“Modbus配置”，设置站号等通讯配置信息;单击“确定”保存通讯配置；



2) 单击菜单项“PLC->下载”、工具栏“下载”按钮或快捷键F8，在打开的“下载”窗口勾选“系统参数”、“Modbus”配置单击“下载”即可完成主站通讯配置；



【注】通讯协议设置完成后只需下载“Modbus”即可；

二、PLC从站通讯配置

在“系统参数->COM配置->协议选择”中选中“Modbus-RTU/QLINK从站”，设置站号

等通讯配置信息，单击“确定”保存通讯配置；在“下载”窗口中勾选“系统参数”下载即可完成Modbus从站协议配置：



【注】从站无“Modbus配置”数据；

三、备注

- 1) Modbus更多信息，请参考“AutoShop帮助->PLC的通讯应用”；
- 2) MD\IS的Modbus从站通讯配置请参考相应设备的帮助文档；

5.7.5 汇川 QLink 协议与编程方法

1) 协议说明：

在HMI页面中，由于用户程序设计不能把变量集中，要多帧数据才能读取PLC数据，造成通信效率降低，数据刷新缓慢。为解决此问题，特开发Qlink协议，HMI通过此协议快速读取PLC数据。

本协议是基于MODBUS 协议，每个HMI页面最多允许两个配置帧，单条配置帧的数据长度设定最长240个字节，即120个寄存器或线圈(C200~C255占用两个寄存器)。

写和连续读的功能可参照MODBUS从站协议，通过正常功能完成通信。H1U/H2U PLC MODBUS从站协议是本协议的一个子集，本协议可正常响应MODBUS帧。

2) 编程方法：

第一步：用户需要首先将H2U配置为modbus通讯方式，并将协议类型选定为快速协议，便于与以往的modbus协议进行区分。当前通信物理串口，若是串口0，设置D8110（通信格式、波特率），D8111（从站地址），D8116（协议）。若是串口1，实际设置D8120（通信格式、波特率），D8121（从站地址），D8126（协议）。同时需要用户在HMI后台中指定使用H2U快速协议，方可正常通讯。

第二步：在HMI切换新页面时，HMI需要下载配置信息给PLC，最多能下载两条信息。PLC端接收配置信息后将该信息对应的寄存器和线圈数据返回给HMI，并将此配置信息一直存于PLC内存中，直至PLC复位或HMI下载最新配置信息覆盖了该信息。如果页面通读量超过两条配置信息能包含的量，剩下的寄存器或线圈使用标准的modbus协议处理。

第三步：若HMI仅停留于该页面，可通过数据块标识快速读取该数据块标识配对的配置信息中的PLC数据。

异常处理：当HMI与PLC通讯时,如果是数据块标识读取信息造成通讯超时(不包括通讯错或校验错)，HMI需要重新下载配置帧信息，因为此时PLC可能已经重启，配置信息可能已经不存在。

3) 编程示例：

如果设定通信的波特率为9600bps，通信的数据格式为8位数据位、无校验位、1位停止位、PLC站号设定为1号站。将HMI的COM1口和PLC的COM0口通过合适的电缆相连接，按下图进行设置即可通过汇川Qlink协议进行通信。

a) InoTouch Editor软件配置：

打开InoTouch Editor软件：文件>新建>设备>通信设置。HMI端配置如下图：

设备

设备名称: 汇川PLC QLink协议

设备型号: 汇川PLC QLink协议 连接方式: 本地连接

连接端口: COM1 设置 预设站号: 1

使用广播命令

PLC地址间隔(words): 5

PLC通讯最大读取字数: 120 PLC通讯最大写入字数: 120

确定 取消

通信设置

通信模式: RS-485_2w

COM: COM1

波特率: 9600

数据位: 8 bits

校验: None

停止位: 1 bit

超时: 0.4 sec

通信延时: 0 ms

响应延时: 0 ms

重发次数: 3

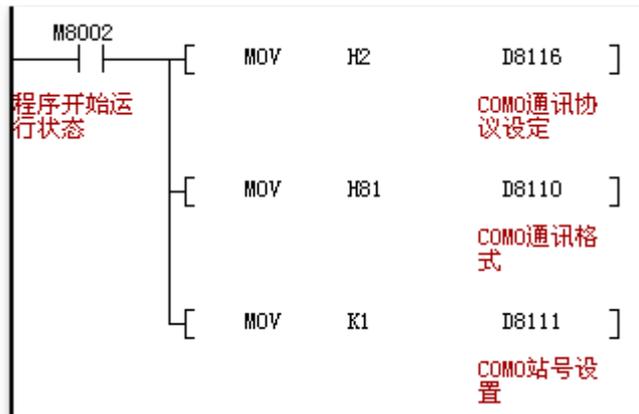
确定 取消

5

通信功能

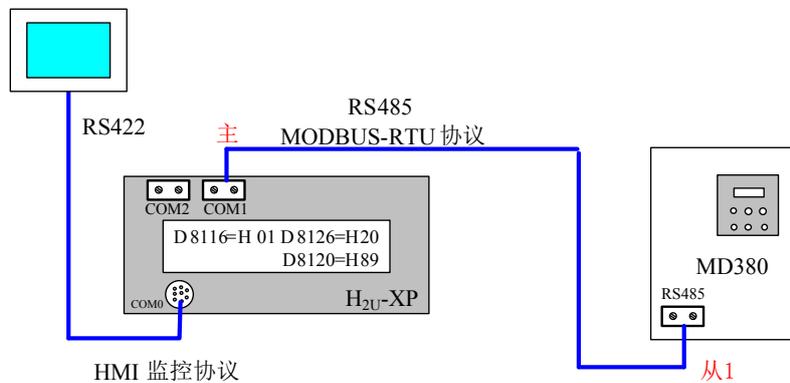
b) AutoShop软件配置:

打开AutoShop软件: 文件>新建工程。PLC端配置如下图:



5.7.6 PLC 与 MD380 系列变频器的通讯编程方法

MD系列变频器内置MODBUS-RTU从站协议, 当需要PLC—MDI之间两者之间需要以通讯方式数据交换时, 须采用MODBUS协议进行通讯, 对于如下典型系统:



5
通信功能

编程时, 将PLC的COM1口配置为MODBUS-RTU主站, 9600bps, 8N2, 以便与MD系列变频器的出厂默认值匹配, 减少设置变频器相关功能码的麻烦; 由PLC完成系统所需数据的交换工作, 即对从机(变频器默认为#1地址)读写命令对变频器进行启停控制、频率设定、运行参数读取等, 或将HMI设置数据转发到MDI;

将PLC的COM0口设为(默认的)HMI监控协议, 才能让PLC在运行或停止状态都能与HMI通讯, 方便编程下载;

(COM0口: 协议D8116= H01 ----下载与 HMI监控协议)

COM1口: 协议D8126= H20; 格式 D8120= H89;----modbus主站, 9600bps, 8N2)

变频器的通讯协议为MODBUS从站, 其默认地址为#1, 9600bps, 8N2, 只需初始化MD380后, 就会是该设置, 被动响应外部控制(在变频器面板上设FP-01=1, 按ENT键确认, 即可恢复默认值)

编程时要注意:

对MD系列变频器通讯访问时, 每个通讯帧只能对一个参数进行读或写操作, 不支持连续地址的多个变量进行批量读写, 即要求通讯指令中的寄存器个数n=K1。

变频器的功能码(地址为HF***)可以反复读取, 但不要反复改写, 防止损坏内部存储器件; 其他参数可以反复读或写。

5.7.7 MD380 配合 PLC 应用的功能码设置

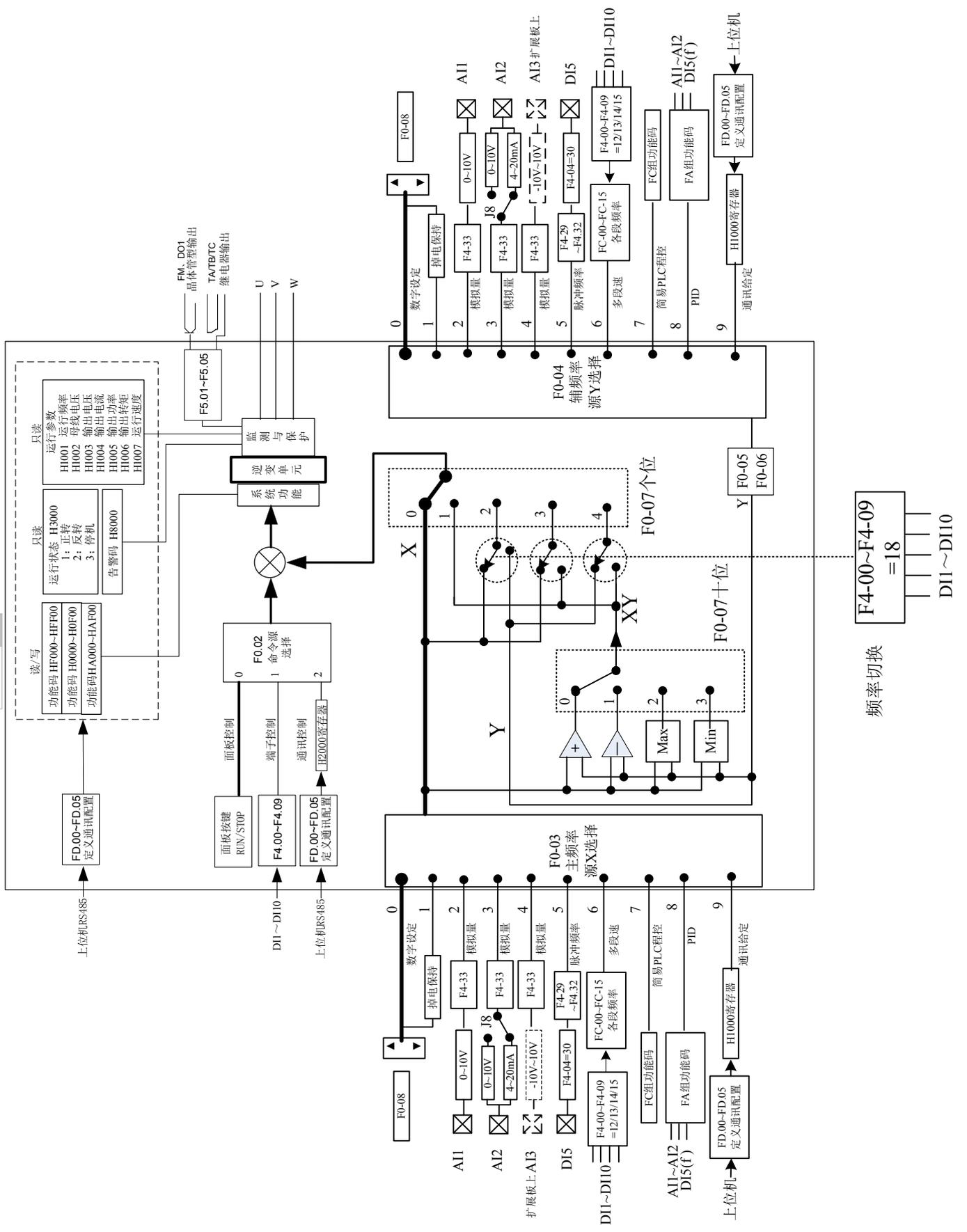
MD380变频器运行控制的主要功能码设定关系示意图如下, 根据所需的工作模式, 设定图中标明的功能码即可; 这些功能码可通过面板手动设置, 也可通过串口通讯进行设置。

其他MD系列的变频器的功能码设置可能不同, 但都有“启停命令源”和“频率命令源”等基本功能码的设置, 处理方法可以借鉴, 请查阅对应的使用手册。

端子启停, 多段速控制

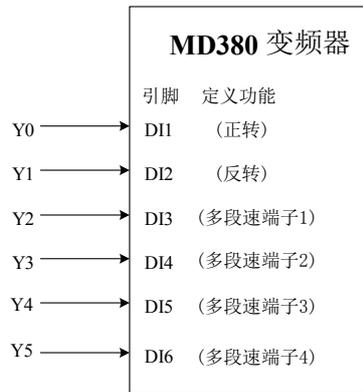
对于如下图的接线, 按图中要求定义逻辑输入DI引脚功能:

5 通信功能



F4-00~F4-09 =18
DII~DII10

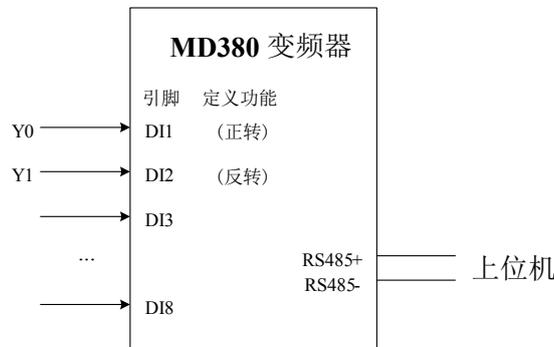
频率切换



设定项目	功能码设置	说明
将命令源选为“端子”	F0-02=1	命令源选为“端子”
“端子命令”方式为2线式	F4-11=0	为FRW/REV两个端子，默认值
按实际接入端子，定义其功能	DI1: F4-00=1	“正转运行”
	DI2: F4-01=2	“反转运行”
将频率源选为“多段速”	F0-03=6	频率源选为“多段速”
按实际多段速接入端子，定义其功能	DI3: F4-02=12	多段速端子1
	DI4: F4-03=13	多段速端子2
	DI5: F4-04=14	多段速端子3
	DI6: F4-05=15	多段速端子4
定义要使用到的多段速的频率，注意默认值为0.00Hz，要预赋值	FC-00 ... FC-015	DI6/DI5/DI4/DI3=0 0 0 0: 选择多段速1 (FC-00) ... DI6/DI5/DI4/DI3=1 1 1 1: 选择多段速16 (FC-15)
其他	F0-07=0, 初始化后默认值，只使用主频率源X。	

端子启停，通讯频率控制

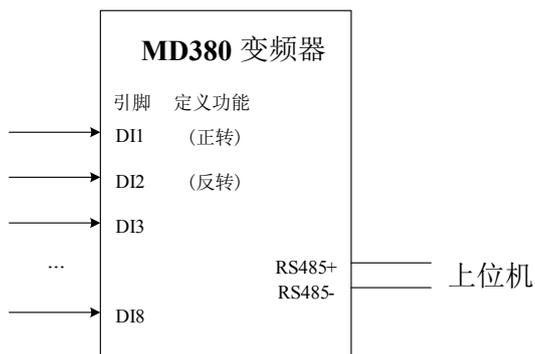
对于如下图的接线，按图中要求定义其中2个逻辑输入DI引脚功能，使用变频器通讯端口及其Modbus-RTU从站协议：



设定项目	功能码设置	说明
将命令源选为“端子”	F0-02=1	命令源选为“端子”
“端子命令”方式为2线式	F4-11=0	为FRW/REV两个端子，默认值
按实际接入端子，定义其功能	DI1: F4-00=1	“正转运行”
	DI2: F4-01=2	“反转运行”
将频率源选为“串口通讯”	F0-03=9	频率源选为“串口通讯”
将通讯配置为上位机相同的设置，建议采用默认值	FD-00=5 FD-01=0 FD-02=1 FD-05=1	“9600bps” 数据格式为“无校验” 本机通讯地址（默认为1） 选择“标准Modbus”协议
运行频率随时修改	“H1000”单元	以通讯式向H1000单元写入运行频率的%值，(-10000~10000)对应于最大频率的(-100.00%~100.00%)
其他	FD-03/FD-04: 通讯应答延迟、通讯超时设置酌情处理	

通讯启停，通讯频率控制

对于如下图的接线，不需使用逻辑输入DI引脚功能，使用变频器的通讯端口：



设定项目	功能码设置	说明
将命令源选为“串口通讯”	F0-02=2	命令源选为“串口通讯”
将频率源选为“通讯口”	F0-03=9	频率源选为“串口通讯”
将通讯配置为上位机相同的设置，建议采用默认值	FD-00=5 FD-01=0 FD-02=1 FD-05=1	“9600bps” 数据格式为“无校验”，（默认8N2固定） 本机通讯地址（默认为1） 选择“标准Modbus”协议
变频器的启停控制	“H2000”单元	分别写入参数： 1=正转运行 2=反转运行 3=正转点动 4=反转电动 5=自由停机 6=减速停机 7=故障复位
运行频率随时修改	“H1000”单元	以通讯式向H1000单元写入运行频率的%值， (-10000~10000)对应于最大频率的(-100.00%~100.00%)
其他	FD-03/FD-04: 通讯应答延迟、通讯超时设置酌情处理	

通讯读取变频器的运行参数，修改功能码。

上位机只要是采用Modbus协议，采用了与MD380变频器的相同的通讯格式配置，就可以实时通讯访问变频器，读取变频器的参数，甚至修改功能码。

读取运行参数部分：

参数地址	参数描述	参数地址	参数描述
1000H	*通信设定值（十进制） -10000~10000	1010H	PID设置
1001H	运行频率	1011H	PID反馈
1002H	母线电压	1012H	PLC步骤
1003H	输出电压	1013H	PULSE输入脉冲频率，单位0.01kHz
1004H	输出电流	1014H	反馈速度，单位0.1Hz
1005H	输出功率	1015H	剩余运行时间
1006H	输出转矩	1016H	AI1校正前电压
1007H	运行速度	1017H	AI2校正前电压
1008H	DI输入标志	1018H	AI3校正前电压
1009H	DO输出标志	1019H	线速度
100AH	AI1电压	101AH	当前上电时间
100BH	AI2电压	101BH	当前运行时间
100CH	AI3电压	101CH	PULSE输入脉冲频率，单位1Hz
100DH	计数值输入	101DH	通讯设定值
100EH	长度值输入	101EH	实际反馈速度
100FH	负载速度	101FH	主频率X显示
-	-	1020H	辅频率Y显示

注意：通信设定值是相对值的百分数（-100.00%~100.00%），可做通信读写操作。

控制命令输入到变频器：（只写）

命令字地址	命令功能
H2000	0001：正转运行
	0002：反转运行
	0003：正转点动
	0004：反转点动
	0005：自由停机
	0006：减速停机
	0007：故障复位

读取变频器运行状态：（只读）

参数地址	参数描述
H3000	1：正转
	2：反转
	3：停机
	其它：无意义

读取变频器故障告警码：（只读）

变频器故障地址	变频器故障信息	
8000H	0000：无故障	0015：参数读写异常
	0001：保留	0016：变频器硬件故障
	0002：加速过电流	0017：电机对地短路故障
	0003：减速过电流	0018：保留
	0004：恒速过电流	0019：保留
	0005：加速过电压	001A：运行时间到达
	0006：减速过电压	001B：用户自定义故障1
	0007：恒速过电压	001C：用户自定义故障2
	0008：缓冲电阻过载故障	001D：上电时间到达
	0009：欠压故障	001E：掉载
	000A：变频器过载	001F：运行时PID反馈丢失
	000B：电机过载	0028：快速限流超时故障
	000C：输入缺相	0029：运行时切换电机故障
	000D：输出缺相	002A：速度偏差过大
	000E：模块过热	002B：电机超速度
	000F：外部故障	002D：电机过温
	0010：通讯异常	005A：编码器线数设定错误
	0011：接触器异常	005B：未接编码器
	0012：电流检测故障	005C：初始位置错误
	0013：电机调谐故障	005E：速度反馈错误
0014：编码器/PG卡故障		

MD380功能码数据

功能码数据为变频器的重要设置参数，在MD320 仅有F组功能参数的基础上，MD380增加了A组功能参数

如下：

MD380 功能码数据	F组(可读写)	F0、F1、F2、F3、F4、F5、F6、F7、F8、F9、FA、FB、FC、FD、FE、FF
	A组(可读写)	A0、A1、A2、A3、A4、A5、A6、A7、A8、A9、AA、AB、AC、AD、AE、AF

功能码数据通讯地址定义如下：

1、当为通讯读取功能码数据时

对于F0~FF、A0~AF组功能码数据，其通讯地址高十六位直接为功能组编号，低十六位直接为功能码在功能组中序号，举例如下：

F0-16功能参数，其通讯地址为 F010H，其中F0H代表F0组功能参数，10H代表功能码在功能组中序号16的十六进制数据格式；

AC-08功能参数，其通讯地址为 AC08，其中ACH代表AC组功能参数，08H代表功能码在功能组中序号8的十六进制数据格式

2、当为通讯写入功能码数据时

对于F0~FF组功能码数据，其通讯地址高十六位，根据是否写入EEPROM，区分为00~0F或F0~FF，低十六位直接为功能码在功能组中序号，举例如下：

写功能参数F0-16

不需要写入EEPROM时，其通讯地址为 0010H

需要写入EEPROM时，其通讯地址为 F010H

对于A0~AF组功能码数据，其通讯地址高十六位，根据是否需要写入EEPROM，区分为40~4F或A0~AF，低十六位直接为功能码在功能组中序号，举例如下：

写功能参数AC-08

不需要写入EEPROM时，其通讯地址为4C08H

需要写入EEPROM时，其通讯地址为AC08H

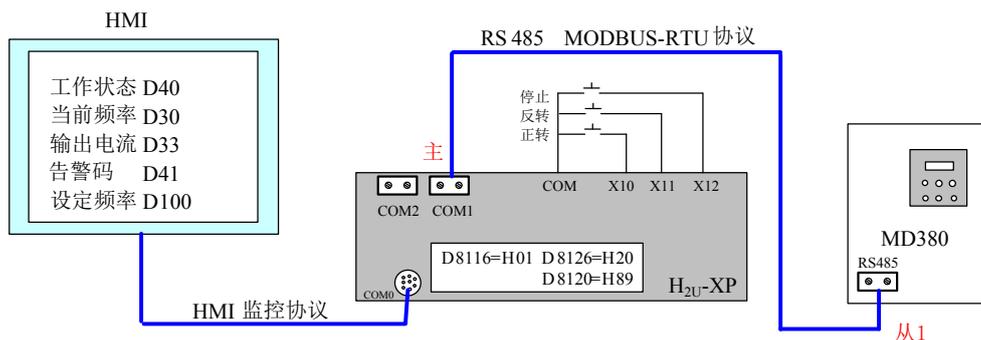
当读取或修改变频器功能码时，“寄存器地址”就是指“功能码”号，如要读取F0-01功能码，“寄存器地址”就是HF001，用Hex格式表示，其中高字节为功能码组号，低字节为功能码组内索引号，注意该索引号要有Hex格式。例如要读取FB-29号功能码，“寄存器地址”就是HFB1D，依此类推。

变频器的功能码有其修改属性限制，有些功能码可以随时修改，有些则只有在停机状态修改，有些修改值的范围受其他相关功能码的约束等，需要注意。

5.7.8 PLC 修改 MD380 功能码的编程

要求H2u—MDI采用通讯方式连接，由H2u进行变频器的启动、停止、正转、反转以及运行频率的控制，实时读取变频器的运行参数。例如如下的系统：

5
通信功能



4) 功能要求如下:

每次上电运行时，H2u自动将变频器的命令源（F0-02=2）设为串口，修改成功后不再发送修改命令；与H2u、MD380两个部件的上电顺序无关；

每次上电运行时，H2u自动将变频器的频率选择（F0-03=9）设为串口，修改成功后不再发送修改命令；与H2u、MD380两个部件的上电顺序无关；

按下X10，令变频器正转运行；按下X11，令变频器反转运行；按下X12，令变频器停止运行；

将D100的频率数值作为变频器的运行频率，循环下发给变频器；

循环读取变频器的当前运行状态、运行频率、输出电流、功率等参数

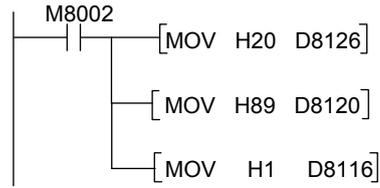
5) PLC编程要点解释:

将D8116=H01, 将COM0口设定为下载与 HMI监控协议；

将D8126=H20,D8120=H89就将COM1口设定为MODBUS-RTU协议，9600bps，8N2，此后用MODBUS指令或RS扩展指令对COM1口的操作进行编程。

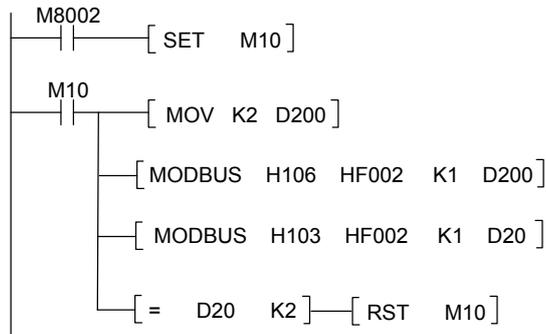
a) 设置MODBUS协议的语句举例（最好放在PLC程序的首行位置）

初始化COM0为下载协议；COM1串口为MODBUS-RTU主站协议
9600bps,8N2, RS指令即为MODBUS指令模式。

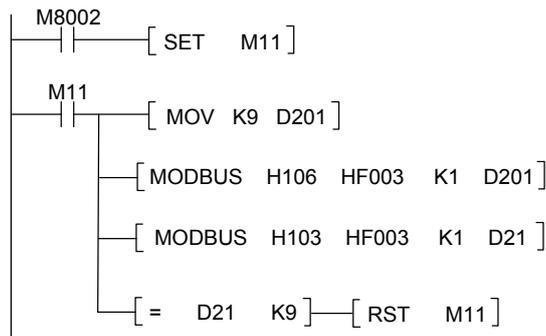


b) 对功能码的读取操作语句举例:

修改“命令源”功能码F0-02, 将之改为“通讯启停”, 即令F0-02=2:

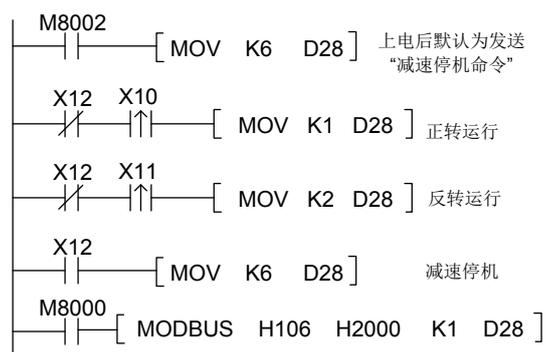


修改“频率源”功能码F0-03, 将之改为“通讯给定”, 即令F0-03=9:



c) 对按键的命令响应, 发送变频器正转运行、反转运行或停机指令, 停机键优先:

运行命令: X10正转, X11反转, X12停止

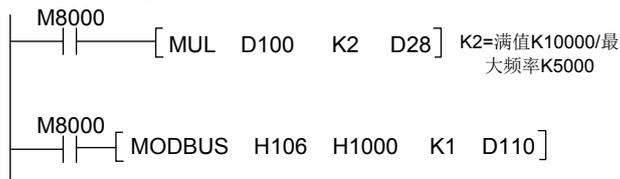


这里是将三种操作响应的命令字处理后, 由同一个MODBUS 或RS指令发送。

d) 改变变频器的运行频率:

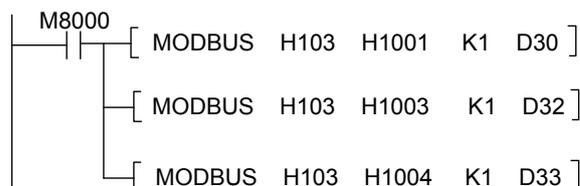
下发给变频器的频率指令, 并不是以0.01Hz为量纲的数据, 而是相对于“最大频率”的百分值, K10000为满刻度, 发送前需要折算一下, 例如变频器最大频率为50.00Hz, 希望以40.00Hz运行, 需要发送的数据为 $40.00 \times K10000 / 50.00 = K8000$ 。本例中将k10000/k5000直接以K2代替, 实际编程中若最大频率并不是50.00Hz, 最好如实地用指令进行计算, 指令采用循环发送:

D100为给定频率（×0.01Hz），注意发送前要折算为最大频率（F0-04=50.00Hz）的相对值，满值为K10000；
 $D110=D100 \times K10000 / K5000$

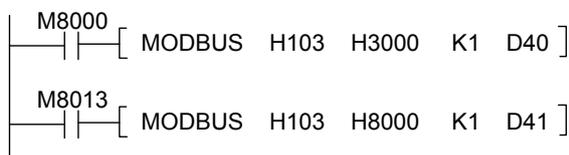


e) 对运行参数的循环读取操作举例:

读取变频器状态: D30为运行频率; D32为输出电压; D33为输出电流



读取变频器运行状态和告警码: D40= (1=正转运行; 2=反转运行; 3=停机) D41=告警码



5.9 CAN通讯指令说明

5

5.9.1 CANlink3.0 应用简介

通信功能

H1U/H2U 系列 PLC 具有 CAN 通信功能，硬件上需要配置 CAN 通信卡。在 PLC 主模块上，可支持远程模块访问指令 FROM/TO，同时支持 CAN-LINK 网络功能。

CAN-LINK 是汇川控制技术公司开发的基于 CAN 总线的网络协议，该协议是一个开放的协议，支持该协议的设备均可接入 CAN-LINK 网络。

H1U/H2U PLC 主模块在用户编程中，采用 FROM/TO 指令或者网络配置方式，可访问各种 H2U 系列远程扩展模块，IS 系列伺服驱动器，MD 系列变频器等智能设备。

若是 H1U 产品，要特别注意，必须人为预置 D8199 为“7”，即将接入的扩展卡类型配置字设置为“CAN 扩展卡”，否则 CAN 可能无法正常通信。

1) 网络拓扑



图 CAN-LINK 网络拓扑示意

5.9.2 主从架构

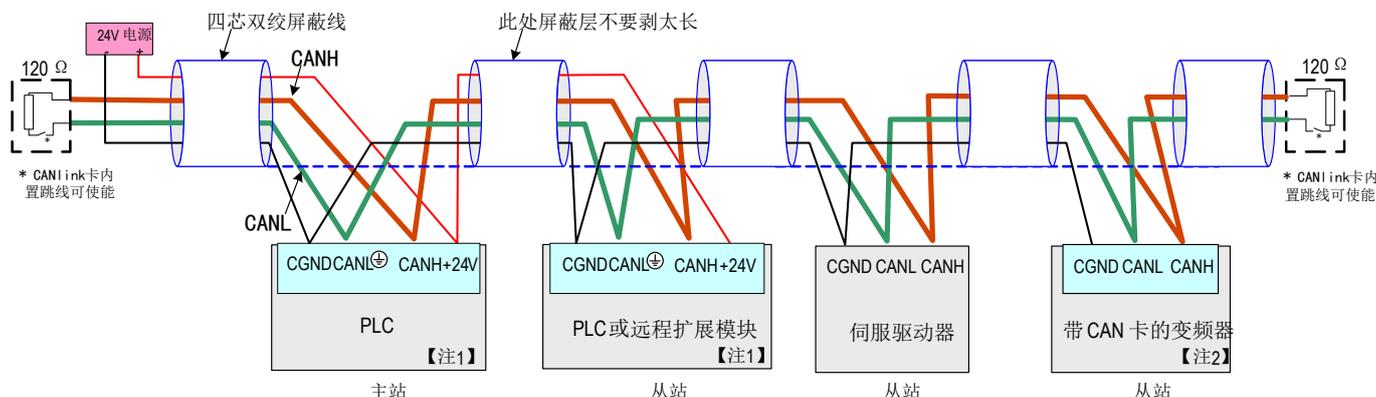
1 H1U/H2U CAN 扩展卡接口定义

信号排布图	管脚号	信号	描述
+24V	1	+ 24V	外接直流 24V 供电电源正
CANH	2	CANH	CAN 总线正
	3	保留	保留
CANL	4	CANL	CAN 总线负
CGND	5	CGND	外接直流 24V 供电电源负

组成 CANlink 网络时,所有设备的以上五根线均要一一对应连在一起。并且必要时在 + 24V 和 CGND 间需要外接 24V 直流电源。总线的两端均要加 120 欧姆的 CAN 总线匹配电阻 (亦称终端电阻), H1U/H2U 远程扩展卡和 CAN 接口卡均内置了匹配电阻, 可通过拨码开关接入或断开。

2 正确的信号连接

CANlink 接线图如下图所示, 设备组网时请按照此正确接线方式进行接线:



【注1】 H1U/H2U系列PLC需另外配置CAN扩展卡, H3U系列PLC自带CAN通信接口, 请根据需要选择;

【注2】 请根据变频器类型选择相应的CAN卡。

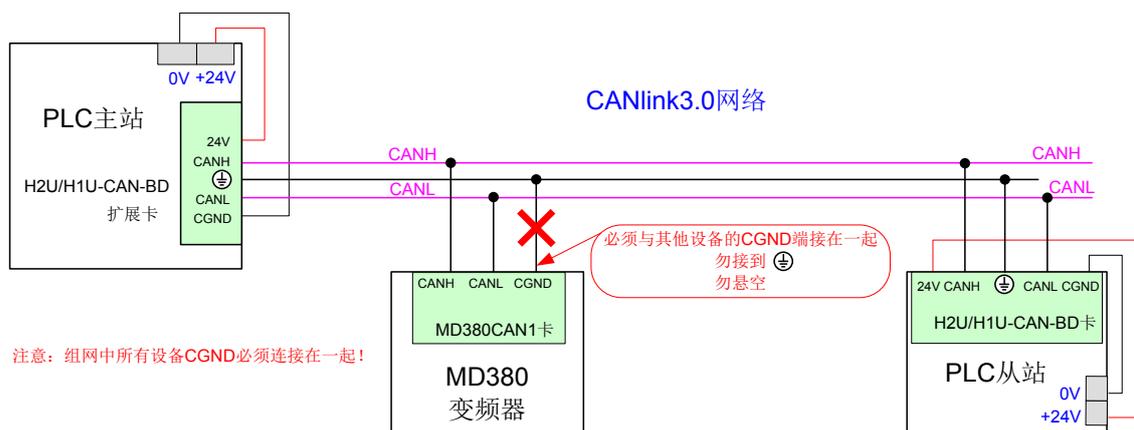
图 正确的 CANlink 网络接线

注: CAN通信电缆采用屏蔽电缆, 推荐型号: RVVP 2×2×0.5。

3 错误的信号连接

敬请注意: 如下不正确的连接方法, 容易导致通信接口损坏, 或工作异常!

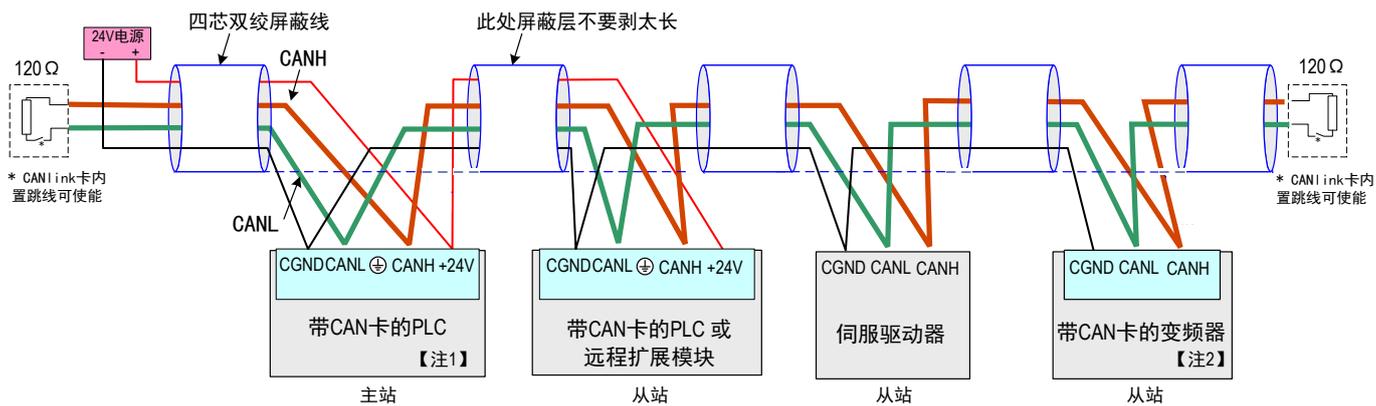
1) 在各设备CAN卡采用24V自行供电的情况下, CAN卡上的CGND接线错误:



注意: 组网中所有设备CGND必须连接在一起!

图 接线错误 --CAN 卡上 CGND 接线错误

2) 屏蔽层接地--接线错误:



【注1】请根据PLC类型选择相应的CANLink接口卡
 【注2】请根据变频器类型选择相应的CANLink接口卡

图 接线错误 -- 屏蔽层接地

3) 波特率与通信长度和线缆要求

估算通信线缆总长度，评估线缆质量，各波特率下通信的最大距离和站点数如下图所示，以下数据采集的条件为线缆必须为双绞线，且带金属屏蔽层，线截面积满足最小线径，所有公共端连接在一起（不能与电源地接到一起），所有站点可靠接地。

波特率	距离 (m)	最小线径 (mm ²)	最大接入点数
1000Kbps	30	0.3	18
500 Kbps	80	0.3	32
250 Kbps	150	0.3	63
125 Kbps	300	0.5	63
100 Kbps	500	0.5	63
50 Kbps	1000	0.7	63

CANlink 扩展卡上设置用拨码开关，开关定义如下：

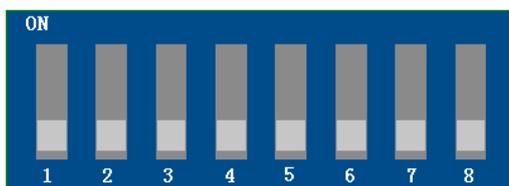


图 CANlink 拨码开关
 表：CANlink 拨码开关定义

拨码号	信号	描述
1	地址线 A1	此六位拨码开关由高到低组合成一个六位二进制数字，用来标识本站号。“ON”表示1，“OFF”表示0。A6在最高位，A1在最低位，依序进行组合。 例如 A1、A2 为 ON，其它位为 OFF，即二进制地址为：000011，十六进制为 H03，十进制为 K03。若 A5、A4 为 ON，其它为 OFF，即二进制地址为：011000，十六进制为 H18，十进制为 K24。 对于 PLC 主模块，CANlink 本机地址还可以通过 M8284、D8284 元件组合的软件方式进行设置
2	地址线 A2	
3	地址线 A3	
4	地址线 A4	
5	地址线 A5	
6	地址线 A6	
7	波特率	OFF：高速模式，波特率 500Kbps，ON：低速模式，波特率 100Kbps
8	匹配电阻	若拨码开关为 ON，表示接入 120 欧姆的终端匹配电阻，否则断开。

在 PLC 运行状态改变拨码开关，匹配电阻的设置可以及时生效，但波特率和地址并不能立即生效，需要让系统重新启动才可以使用新的设置参数。

注意：当 M8284、M8285 分别为 0 时，拨码开关设置的 CAN 地址和 CAN 波特率才分别有效。若需要以软件方式设置为其他的波特率，请查阅 M8285、D8285、D8286 的设置说明。

5.9.3 心跳机制

网络心跳监控主/从站运行状态,所有从站以该时间间隔发送心跳给主站,主站通过心跳机制监控网络中各站点的状态(不在线、在线),从站通过主站心跳监控主站状态。(建议设置时间大于 200ms)。

【注】网络心跳过短,会增加整个网络的负载状态;网络心跳设置范围为 10~20000ms,且不允许网络心跳占用网络带宽超过 10%。

5.9.4 掉站信息

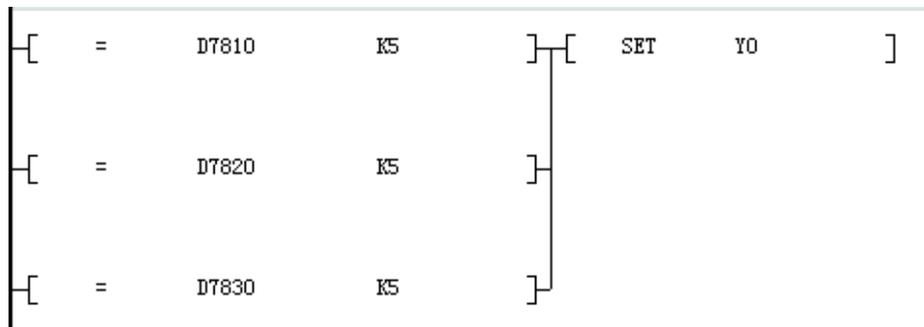
PLC 提供通过检测寄存器的值判断网络中掉站情况的功能。

D7800 是 PLC 从站检测主站是否正常,只有从站 PLC 的 D7800 才有意义。

D7801~D7863 是主站监控在网从站状态的寄存器,与主站号对应的寄存器无意义。

寄存器	定义	掉线检测寄存器的值定义		备注
D7800	主站状态			从站监控主站状态
D7801	1 号站状态	D78xx 值	定义	主站监控各从站状态,与主站站号对应的寄存器无意义。如主站号是 7,那么 D7807 无意义。
D7802	2 号站状态	0	无该站或该寄存器未定义	
D7803	3 号站状态	1	主站专用,用户有配置该站,但主站检测不到该站在网	
D7804	4 号站状态	2	该站正常	
.....	5	该站掉站	
D7861	61 号站状态			
D7862	62 号站状态			
D7863	63 号站状态			

主站可通过以下程序检测是否存在从站掉站(例中是 10 号、20 号、30 号从站中任一从站掉站时,Y0 导通)。



CANlink3.0 具有掉站后自动重连的特点,即从站掉站后,会自动检测网络,在通信电缆正常后自动接入网络,主站检测到掉站从站重新接入后会自动将相应寄存器恢复。变频器和伺服掉站后重新接入网络需手动清除面板上的错误字样,其它功能是否受影响请参考变频器和伺服用户手册。

注意: PLC 会根据实际状态修正相应寄存器(D7800-D7863)的值,不要尝试对其进行人为的修改。

5.9.5 CANlink3.0 的使用

CAN 通信扩展卡相关信息的特殊变量表 (VER 3.00)

M 元件	M 元件定义	D 元件	D 元件定义
M8280	协议标志 ※1 0: CANlink VER3.00 协议 1: CANlink VER1.00 协议	D8280	当前软件 CANlink 协议版本号 (默认 300)
M8281	0: 没有识别到扩展卡, 此时 CANlink 功能不能使用 1: 检测到扩展卡 (含 CANlink 卡)	D8281	扩展卡类型号, H2U 为自动识别, 在 D8280 可查询到接入的扩展卡类型号; H1U 不能自动识别, 需要通过 D8281 人为设定扩展卡类型。 扩展卡类型如下: 1~6 无效 7 CAN 扩展卡
M8282	保留	D8282	当前心跳时间 (只读, 单位 ms)

表: CAN 通信的地址和波特率相关设置的特殊变量表

M8284	0: 本机 CANlink 地址由拨码开关设定, 当前地址可在 D8284 查询		
	1: 本机 CANlink 地址由 D8284 设定, 设定范围 1~63 (0 保留)		
M8285	0: CANlink 波特率由拨码开关设定, 当前波特率可在 D8285 查询 ※1;		
	1: CANlink 波特率由软件设定, 由 D8286 决定波特率设置 ※1	CANlink 波特率由 D8286 分档设定模式 ※1	D8286 = 20, 20Kbps D8286 = 50, 50Kbps D8286 = 100, 100Kbps D8286 = 125, 125Kbps D8286 = 250, 250Kbps D8286 = 500, 500Kbps D8286 = 800, 800Kbps D8286 = 1000, 1000Kbps D8286 = 其它值, 500Kbps

表: CAN 通信的状态相关信息的特殊变量表

M8287~ M8289	保留	D8287~ D8289	保留
M8290	网络启停控制, ON 表示全体节点网络启动, OFF 表示全体节点网络停止	D8290	保留
M8291	同步传输启动元件, 上升沿时发送广播命令, 启动网络中所有同步传输配置, 完成后自动复位	D8291	保留
M8292~M8295	保留	D8292~M8295	保留
M8296	设备地址错误, CANlink 地址设置错误。	D8296	保留
M8297~ M8298	保留	D8297~ D8298	保留
M8299	CAN 接收超时 ※2, 指 FROM 与 TO 指令返回超时	D8299	CAN 接收超时设定 (ms) ※3
M8300~ M8306	保留	D8300~ D8306	保留
M8307	保留	D8307	命令帧错误寄存器, 主站显示格式为: 出错从站站号 *100+ 错误码
M8308	保留	D8308	配置帧错误寄存器, 主站显示格式为: 出错从站站号 *100+ 错误码

表: 故障信息 (增加 D 寄存器值说明)

故障查询寄存器	错误代码	出错内容
D8066 或 D8067	6380	发送超时
	6381	接收超时
	6382	CAN 发送忙
	6383	CAN 接收忙

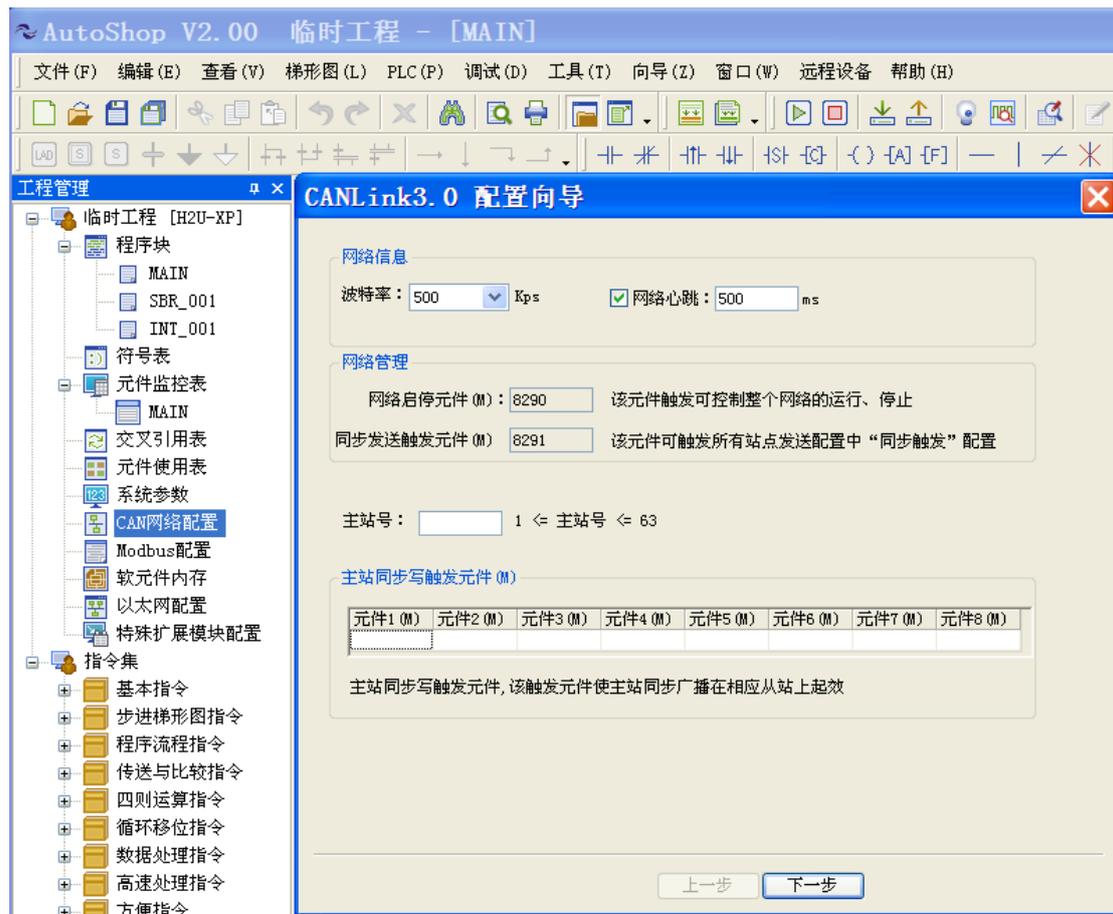
※1: 用户设定后需重新上电或 STOP->RUN 才能够识别。如果 PLC 所接远程模块中有一台及以上非 XP 平台模块则必须置 M8280 为 ON(切换为 CANlink1.0 协议), 并重新上电才能使模块正常工作。(远程扩展模块标签有标注是否为 XP 平台模块)

※2: 置位后需用户将其复位。

※3: FROM、TO 的接收等待时间, 如果设定小于 5 则默认认为是 5ms, 如果设定大于 1000 默认认为是 1000ms。表示从发送指令开始计算等待的时间, 对于 FROM、TO 指令, 超过此时间则 M8299 置位。建议波特率大时将此值减小, 波特率小时将此值增大, 否则会由于传输时间长导致没有接收到数据。波特率变小时, D8299 设置时间应变大, 例如 50K 设置为 10; 25K 设置为 25, 否则会出现异常。

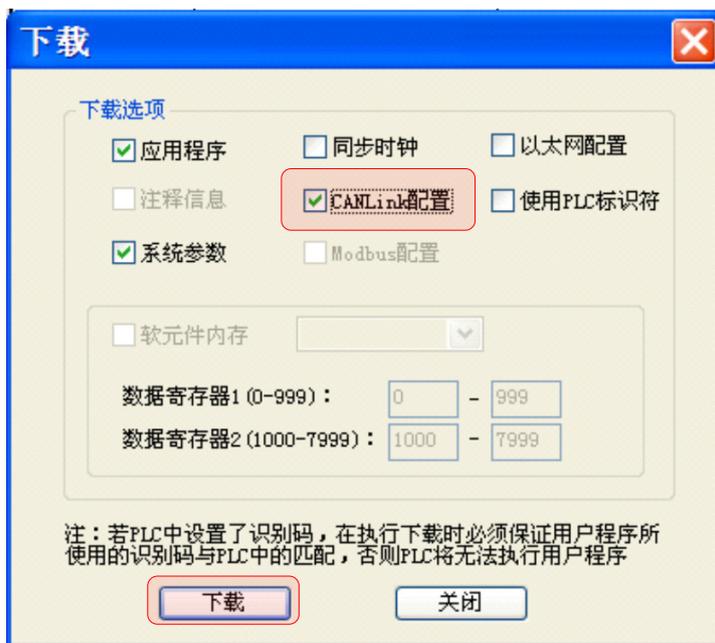
5.9.6 软件配置 CANlink3.0 主从站的通信配置方法

CANlink3.0 的通信应用编程时, 不是以往的 CAN 通信指令方式, 而是以“CAN 网络配置”方式, 将需要进行的通信交互内容事先配置好:



5
通信功能

在下载用户程序时, 同时将“CAN 网络配置”下载到 PLC 中:



先理解 CANlink3.0 网络配置的原理，有助于正确填写“CAN 网络配置”表：

- 1) CANlink3.0 网络中，必需只有 1 个通信主站，这个主站是 H2U-XP 或 H1U-XP 系列 PLC；
- 2) CANlink3.0 网络中，必需有 1 个或多个通信从站，可以是 H2U 系列 IO/AI/AO/AM/PT/TC 等远程扩展模块，带 CANlink3.0 接口卡的 MD 系列变频器、带 CANlink3.0 接口的 IS 系列伺服驱动器，可以是 H2U-XP/H1U-XP 系列 PLC，也可以是用户按照 CANlink3.0 协议开发的从站设备；
- 3) CANlink3.0 网络中的主站从站设备，均采用主动发送“通信写”数据的方式，而非询问应答方式。例如：
 - 主站要将数据发送给从站，实现方法是主站依据 CANlink 通信配置，在满足触发条件时，将指定寄存器的数据“写入”到指定从站的寄存器中；

- 主站需要向从站读取的数据，是从站依据 CANlink 通信配置，自动向主机发送数据，将数据通信“写入”到主站的接收单元中的方式实现；
 - 从站之间要交互的信息，是通过从站依据 CANlink 通信配置，自动向指定从站发送数据，将数据通信“写入”到指定从站的接收单元中的方式实现；
 - 站点要向多个站点发送的信息，是依据 CANlink 通信配置，自动向自己发送“写操作”数据（等效于广播），而其它站点将这些数据有选择性的接收，自动保持到预设的接收单元中的方式实现；
 - 为了提高网络通信中数据交互的效率，主站、从站都可以将“听到”的其他站号发出的广播数据保存下来，主站、从站中需要设置“接收配置”，将所需接收从站的站号地址事先设置好，对来自站号设置以外站点的广播数据，不予理睬。
- 4) 因 CANlink3.0 从站不需配置，而是通过 H2U-XP/H1U-XP 主站 PLC 向从站转达 CANlink 配置，故在主站的 CANlink3.0 通信配置项中，有对主站的配置、有分别对各从站的配置，这些对从站的配置项，是由 CANlink 主站通过配置帧进行转发的。
- 5) 在主站每次开始运行时，都会向 CANlink 从站发送一次配置帧，将各从站的“通信任务清单”布置下去，一旦运行起来，各从站按照该任务清单，主动对外发送各项数据；
- 6) CANlink3.0 配置项中内容包括：待发送数据的寄存器地址、目标接收从站地址、数据个数、接收寄存器地址、通信发送的时间间隔、触发条件等等，这些都一般的通信指令中必需的内容。
- 与一般的通信不同的是，这些“通信写”操作默认为不需要操作是否成功的应答的。
 - 对于需要多个从机同时动作响应的通信应用，例如由伺服驱动的多轴同步控制、位置控制的高速运动应用，需要主站配置中采用“同步写”的配置选项中填写。实际运行时，主站先分别对各从站写入数据后，再发送同时生效的广播命令帧，使得各从站同时操作。

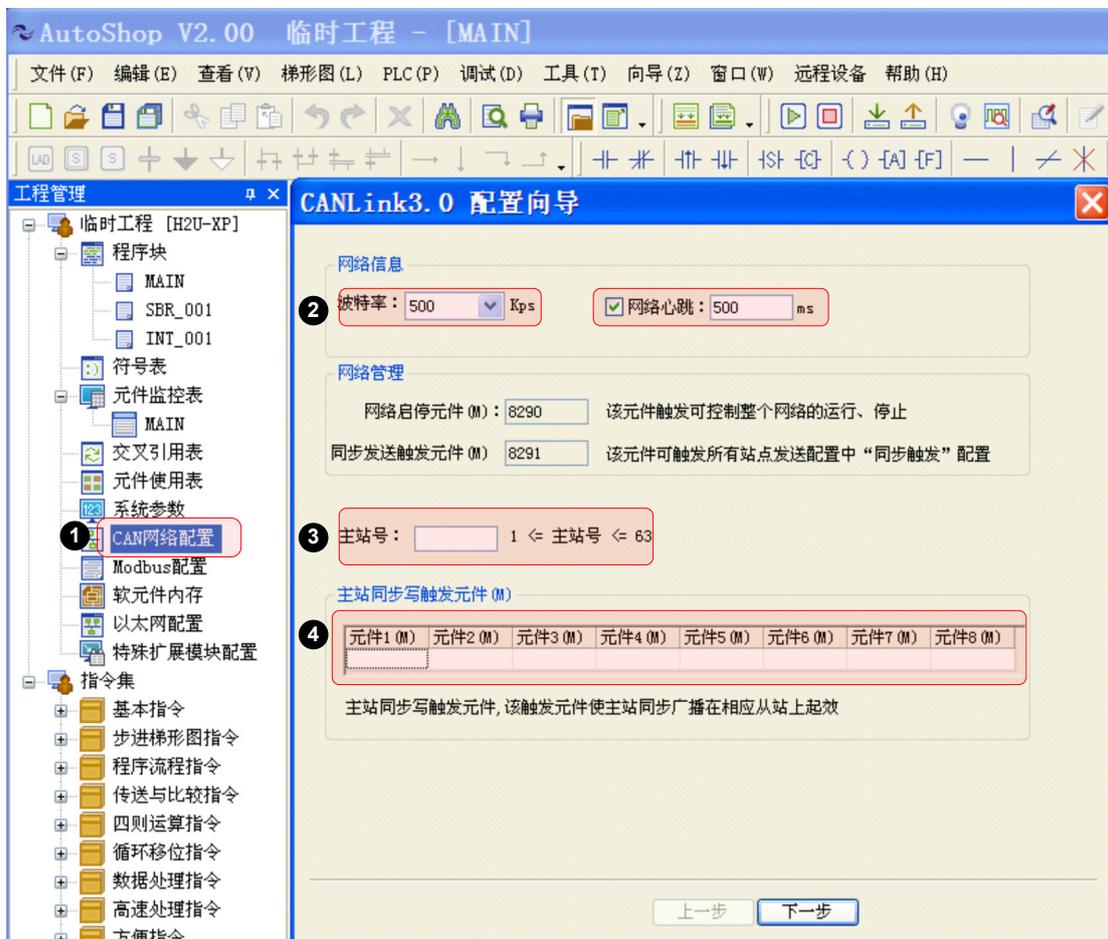
5.9.7 CAN 网络配置表的填写方法

建立工程后，在“工程管理”中双击“CAN 网络配置”即可打开 CANlink 配置表。

首次进入配置，则直接进入“CANlink3.0 配置”窗口。

1 配置向导

主站配置向导主要完成网络配置、主站配置、主站同步触发元件管理、从站添加功能，向导页面如图所示：



- 波特率(必选)

波特率有 20K, 50K, 100K, 125K, 250K, 500K, 800K, 1M 共八种, 以满足不同使用场合的需求, 可通过下拉选择需要的波特率, 配置下载到主站即可生效 (该参数仅对主站生效, 从站仍需手动修改), 用户可以根据总使用负载情况以及实际通信距离选择适当的波特率。

- 网络心跳 (可选)

所有站点以该时间间隔发送心跳给主站, 主站通过心跳机制监控网络中各站点的状态 (不在线、在线), 从站通过主站心跳监控主站状态。(建议设置时间大于 200ms) 如果将网络心跳前的勾去掉, 则网络心跳功能将取消, 将无法对网络进行监控。

【注】网络心跳过短, 会增加整个网络的负载状态; 网络心跳设置范围为 10~20000ms, 且不允许网络心跳占用网络带宽超过 10%。

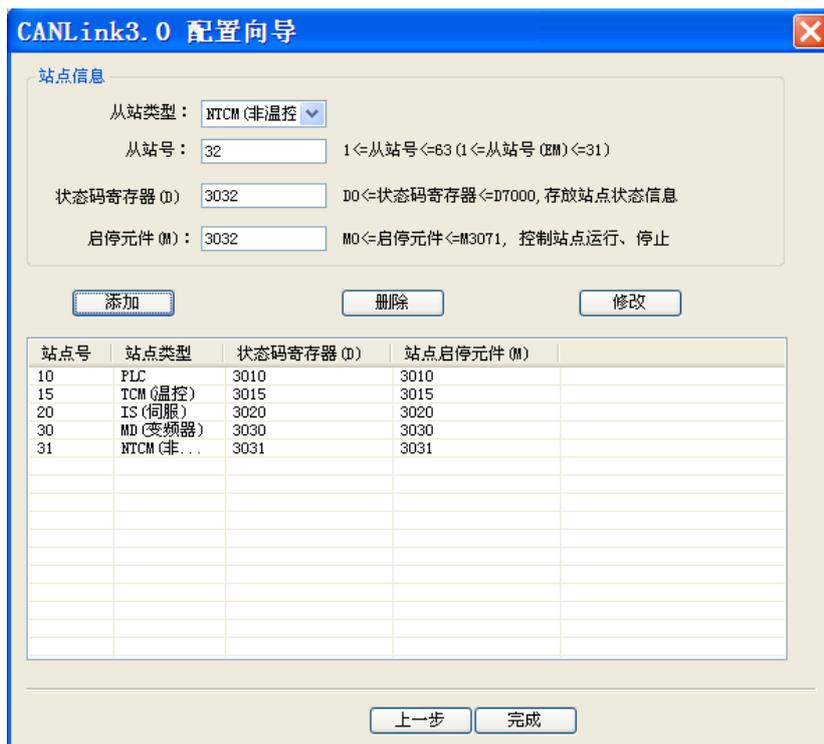
- 主站号 (必选)

主站号是整个网络中的主站站号, 及下载配置的 PLC 主站站点的站号 (这里需填写作为主站的 PLC 的站号), 仅配置作用, 不能在这里改变站点, 如果这里填写的站号与实际站号不一致, 则即使配置下载到 PLC, PLC 也不会执行, 而是把它当做无效配置处理。例如, 后台配置中主站号填写为 7, 那该配置下载到除了 7 号站外的其它 PLC 中都不能执行, 只有下载到 7 号站 PLC 才会有效, 7 号站再下发给网络中其他站号完成配置。整个 CANlink 网络配置下载到主站后分发到从站, 后台可通过主站监控、管理整个网络。

- 主站同步写配置触发元件 (可选)

主站同步配置的触发元件, 触发元件 (M) 置位则对应触发配置有效, 发送完成自动复位。

- 添加、修改或删除从站



- 删除

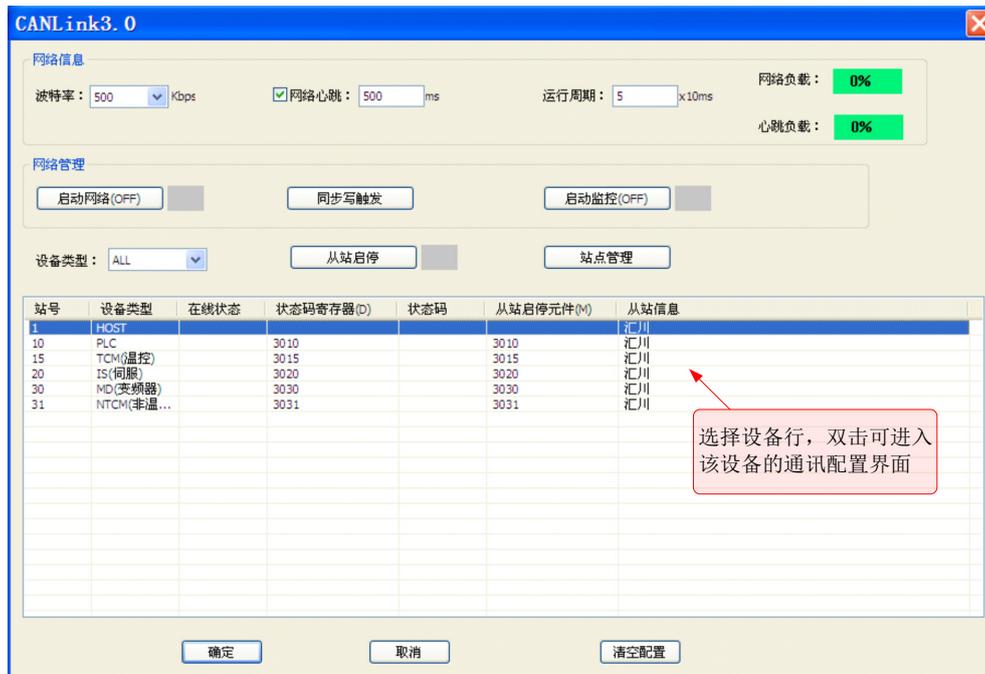
列表中选择站点后单击“删除”按钮, 在提示“是否删除”单击“确定”即可删除 (可同时选中多个删除);

- 修改

列表中选中单个从站, 在“站点信息”中修改相应信息, 单击“修改”按钮即可 (站点类型不能修改);

- 补充

在配置向导窗口中, 单击“完成”按钮可保存向导中修改并退出; 单击窗口右上角“X”则取消修改操作并退出。



2 站点配置

在“主界面”中双击任一站点，即可打开站点的通信配置窗口，通信配置包括三部分：

同步配置 (仅主站)、发送配置、接收配置。

● 主站同步配置

- 触发条件 M 置位时对应主站同步发送配置起效。选择不同“触发条件(M)”即可查看、添加、修改、删除该触发元件对应的主站同步配置信息。同步配置主要适用于需要同时启动某一操作的场合。
- 在图中，当主站的 M1=1 时，将会把上述三条配置依次发出，从站收到该配置后将数据存放到缓存区中，在最后一配置发送成功后，主站会自动发送一个生效命令，所有在网从站收到该命令后自动将缓冲区中的数据写到相应的元件或功能码中，在图中，10 号 PLC 将前面接收到的主站 D10 的值写入 D10，20 号伺服将前面接收到的主站 D20 的值写入 H200,30 号变频器将前面接收到的主站的 D30 的值写入 HF003，即上述所有的值在收到生效命令后同时写入。生效命令成功发出后，主站将自动复位触发元件 M1。触发元件使用规则见“事件触发”。

5
通信功能

【注】触发条件 (M): 每个触发条件最多关联 16 条配置，该“触发条件 (M)”可决定其关联的主站同步配置是否有效，整个网络允许最多 8 个不同的触发条件 (M); 点击触发元件 (M) 即可下拉切换不同的触发元件。

如需对伺服的 32 位寄存器进行同步配置，请在同一触发元件将数据和地址分为高 16 位和低 16 位进行操作，即在同一触发元件下写两条，一条对应伺服 32 位功能码的低地址位，一条对应高地址位，如只写一条或将两条分在两个不同触发元件下，伺服将报错而不能进行相关操作。





- 主站写伺服 32 位寄存器同步配置
- 上图中H1112 是伺服的32 位功能码，如要对该功能码进行设置，则需设置两条，分别对应其高低地址位。当M2 置位时，主站将D101、D102 的值写入伺服的H1112 这个32 位功能码中，当M2 下的五条全部发送后主站会发送生效命令使能各从站同时生效这些配置，然后M2 自动复位；当M3 置位时，主站将D201、D202 的值写入伺服的H1112，然后生效，复位M3。通过触发条件的选择栏可下拉选择切换不同的触发元件。
- 触发元件的设置向导中，见图：



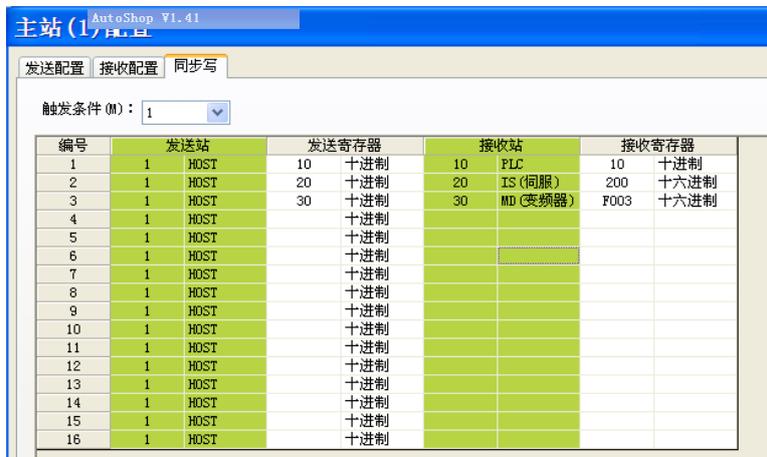
图 同步配置触发元件设置

如对 32 位功能码操作时只在单个触发元件下对一个地址进行操作，伺服将会返回错误导致同步不能继续执行，该错误记录在主站的 D8307 中，错误信息参见“主站错误代码及处理”。



图 错误的同步配置

- 同步配置 (仅主站)



3 发送配置

发送配置主要是用于配置本站向其他站点发送的数据，配置页面如图 所示：



图 发送配置

- 触发方式
 - 时间(ms)：适用所有设备，本站以固定间隔时间(触发条件)执行本条通信配置，设置范围1~30000ms。
 - 事件(M)：适用Host、PLC，本站触发条件(M 元件)置位时执行本条通信配置，允许使用相同M 元件触发，完成发送后自动复位。
 - 同步(M)：适用所有设备，主站元件M8291 置位时执行本条通信配置。
 - 事件(ms)：适用IS、MD、远程扩展模块(TCM/NTCM)，本站检测到发送寄存器的值发生改变且满足禁止时间(触发条件)执行本条通信配置。

※1: 禁止时间: 同一配置相邻两次发送的最小间隔时间。

※2: 发送配置允许单站最大条数: HOST 主站 256 条, 单个从站 16 条, 且从站总条数 256 条。

※3: 选择一条配置, 按“Insert”, 会在这条配置后增加一条空配置行。同样, 选择一条配置后按“Delete”会删除这条配置; 另外可通过快捷键或右键弹出菜单实现“复制”、“粘贴”、“删除”、“行插入”、“行删除”

● 寄存器

Host、PLC 中寄存器值为对应 D 元件; IS、MD 中寄存器值为对应功能码; TCM/NTCM 对应 BFM 区。

● 寄存器个数

表示发送、接收的连续 D 元件或功能码的个数。

● 点对多配置

发送、接收站相同时为点对多配置, 该配置不指定接收站; 任何将该发送站站号配置到“接收配置”的站点都可以接收该配置发送的数据, 接收寄存器为接收站点对应的 D 元件或功能码。

● 本站接收

红色分割线下的灰色部分, 是其它站点发送给本站的数据, 包括点对点、点对多两种数据。用户可以通过这部分直观的查看哪些站的哪些元件或功能码会对本站造成影响。

4 接收配置

接收配置主要用于该站点接收其他站点的点对多数据, 每个站点可以接收其它 8 个站点的点对多数据。

例: 如下配置

The screenshot shows the configuration interface for the CAN network. It is divided into four main sections: Master Station (1) configuration, Slave Station (10) configuration, Slave Station (20) configuration, and Slave Station (30) configuration. Each section has a '发送配置' (Send Configuration) and a '接收配置' (Receive Configuration) tab. The Master Station configuration table is as follows:

编号	触发方式	触发条件	发送站	发送寄存器	接收站	接收寄存器	寄存器个数
1	时间 (ms)	100	1 HOST	1000 十进制	1 HOST	8192 十进制	1
2			1 HOST				
3			1 HOST				

The Slave Station (10) configuration table is as follows:

编号	站点1	站点2	站点3	站点4	站点5
1	1	20	30		

The Slave Station (20) configuration table is as follows:

编号	站点1	站点2	站点3	站点4	站点5
1	1	10	30		

The Slave Station (30) configuration table is as follows:

编号	站点1	站点2	站点3
1	10	20	

图 接收配置

该例中, 主站 1 号每 100ms 将 D1000 的值以点对多数据帧的格式发往接收站的 D8192, 按照从站 20 号、30 号、40 号的接收配置的情况, 20 号和 30 号站将接收该数据帧并写入 D8192 中, 而 40 号站没有配置其接收 1 号站的点对多数据, 所以会将该数据帧直接忽略。

【注】: 点对多可以实现数据的同时生效, 相当于主站同步配置, 但已不局限于主站才能发送。每站最多可以接收 8 个不同站点的点对多数据, 但每个站发出的点对多数据不局限接收站的数目, 即网络中所有除了发出站自身外都可以接收, 只要接收配置里已经配置了接收该站点。

5 网络管理

网络管理页面如图, 当用户程序及 CAN 网络配置下载到 PLC 之后, 在这个页面, 可以进行网络的监控调试, 监控各从站的通信状况, 分析 CAN 网络的负载率等:



① 网络状态

非监控模式下通过网络负载带宽、从站配置计算获取，监控模式下从网络（通过主站）实时获取；网络负载 ≤ 50：绿色（良好）、50 < 网络负载 ≤ 75：黄色（警告）、75 < 网络负载 ≤ 90：红色（严重警告）、网络负载 > 90：ERR 红底（错误）

② 网络管理

通过 COM0 或 USB 连接主站 PLC 后，单击“启动网路 (OFF)”按钮成功执行后：按钮后图标变为“绿色”，按钮名改为“关闭网络 (ON)”；通过 COM0 或 USB 连接主站 PLC 后，单击“停止网路 (ON)”按钮成功执行后：按钮后图标变为“灰色”，按钮名改为“启动网络 (OFF)”；

【注】监控状态下将实时更新网络管理按钮的状态，该按钮与 M8290 相关联，可以在用户程序中操作来实现。

③ 同步发送触发

通过该按钮，可触发所有站点“发送配置”中“同步 (M)”类型配置；该操作主要通过置位“M8291”实现，用户程序中可通过该元件实现“同步 (M)”类型配置；

④ 监控管理

通过 COM0 或 USB 连接主站 PLC 后，单击“启动监控 (OFF)”即可打开监控功能：按钮后图标变为绿色，按钮名改为“停止监控 (ON)”；通过 COM0 或 USB 连接主站 PLC 后，单击“停止监控 (ON)”按钮即可关闭监控功能：按钮后图标变为灰色，按钮名改为“启动监控 (OFF)”

【注】网络连接失败，则自动转为状态：按钮后图标变为灰色，按钮名改为“启动监控 (OFF)”。

⑤ 设备类型

“设备类型”可过滤列表中显示的站点类型；

⑥ 从站管理

通过 COM0 或 USB 连接主站 PLC 后，在列表中选择站点：单击“启动从站 (OFF)”成功执行后：按钮后图标变为“绿色”，按钮名改为“关闭从站 (ON)”，列表中对应该站点“在线状态”为“运行”；通过 COM0 或 USB 连接主站 PLC 后，单击“关闭从站 (OFF)”成功执行后，按钮后图标变为“灰色”，按钮名改为“启动从站 (ON)”，列表中对应该站点“在线状态”为“停止”；

【注】监控状态下将实时更新从站管理按钮的状态，对于不在线的站点进行此操作无效；该按钮与“从站启停元件 (M)”相关联，也可以在用户程序中置位该元件进行操作。这部分 M 元件不允许复用。

⑦ 站点管理

单击“站点管理”按钮即可打开“CANlink”向导窗口，进入“主站配置”界面；

【注】网络管理、站点的增删和修改只能在向导中进行；监控状态下将实时更新从站管理按钮的状态，对于不在线的站点进行此操作无效；该按钮与“从站启停元件(M)”相关联，也可以在用户程序中置位该元件进行操作。这部分M元件不允许复用。

6 从站的访问举例(伺服、变频器)

当前支持 CANlink3.0 的产品包括汇川 PLC (H2u-XP、H1u-XP)，汇川变频器 MD380、MD500、MD310，汇川伺服 IS500A (214 非标)、IS620P、IS700，XP 系列远程扩展模块所需 AutoShop 后台为 V1.40 或以上版本

● 伺服驱动器的访问

• 伺服 CANlink 通信相关功能码

功能码	名称	设定范围	最小单位	出厂设定	生效时间	类别	相关模式
H00	02	非标版本号					
H0C	00	伺服轴地址	1~247	1	1	重新上电生效	运行中设定 PST
H0C	08	CAN 通信波特率设定	0:20K 1:50K 2:100K 3:125K 4:250K 5:500K 6:800K 7:1M	1	5	重新上电生效	运行中设定 PST
H0C	13	通信写入功能码值是否更新到 EEPROM	0: 不更新 EEPROM 1: 更新 EEPROM	1	1	立即生效	运行中设定 PST
H0C	15	CAN 通信协议选择	0: CANlink 协议 1: 预留 (CANOpen 协议)	1	0	重新上电生效	停机设定 PST

注：IS620P 的 H0C.08 没有 800K，选择 6 也会设定为 1Mbps。

请注意：

伺服功能码组号（前 2 位）不变，功能码号（后 2 位）需要从 10 进制转换成 16 进制。如：H08.22（22 为 10 进制）在 PLC 程序里表示为 H0816（16 为 16 进制）

请事先设置好伺服站地址（H0C.00）、波特率（H0C.08）、CAN 协议选择（H0C.15）这三个参数。并且伺服功能码有些是重新上电生效。

7 变频器的访问

由一台 H1U-XP、一台 MD380 变频器和一个 H2U-4DAR 组成 CANlink 网络，要求 PLC 控制变频器加速运行 20 秒后自由停机 20 秒，如此循环，控制电压由 4DAR 提供。H1U-XP 站号为 1,4DAR 站号为 2，MD380 站号为 3。鉴于站点数较少，且通信距离仅 10m 左右，波特率采用 500Kbps，心跳无特殊要求，采用默认值 500ms。新建工程“CANlink3.0 Example”，双击左侧“工程管理”的“CAN 网络配置”，如图配置：

主站 (1) 配置

发送配置 | 接收配置 | 同步配置

编号	触发方式	触发条件	发送站	发送寄存器	接收站	接收寄存器	寄存器个数
1	事件 (M)	100	1 HOST	500 十进制	3 MD (变频器)	2000 十六进制	1
2			1 HOST	十进制			
3			1 HOST	十进制			
4			1 HOST	十进制			
5			1 HOST	十进制			
6			1 HOST	十进制			
7			1 HOST	十进制			
8			1 HOST	十进制			
9			1 HOST	十进制			
10			1 HOST	十进制			
11			1 HOST	十进制			
12			1 HOST	十进制			
13			1 HOST	十进制			
14			1 HOST	十进制			
15			1 HOST	十进制			
16			1 HOST	十进制			
本站接收	---	---	---	---	---	---	---
1	事件 (M)	100	3 MD (变频器)	3000 十六进制	1 HOST	2000 十进制	1

设置变频器返回当前的状态，配置如下：

从站 (3) 配置

发送配置 | 接收配置

编号	触发方式	触发条件	发送站	发送寄存器	接收站	接收寄存器	寄存器个数
1	事件 (ms)	100	3 MD (变频器)	3000 十六进制	1 HOST	2000 十进制	1
2			3 MD (变频器)	十六进制			
3			3 MD (变频器)	十六进制			
4			3 MD (变频器)	十六进制			
5			3 MD (变频器)	十六进制			
6			3 MD (变频器)	十六进制			
7			3 MD (变频器)	十六进制			
8			3 MD (变频器)	十六进制			
9			3 MD (变频器)	十六进制			
10			3 MD (变频器)	十六进制			
11			3 MD (变频器)	十六进制			
12			3 MD (变频器)	十六进制			
13			3 MD (变频器)	十六进制			
14			3 MD (变频器)	十六进制			
15			3 MD (变频器)	十六进制			
16			3 MD (变频器)	十六进制			
本站接收	---	---	---	---	---	---	---
1	事件 (ms)	100	1 HOST	500 十进制	3 MD (变频器)	2000 十六进制	1

确定 取消

5
通信功能

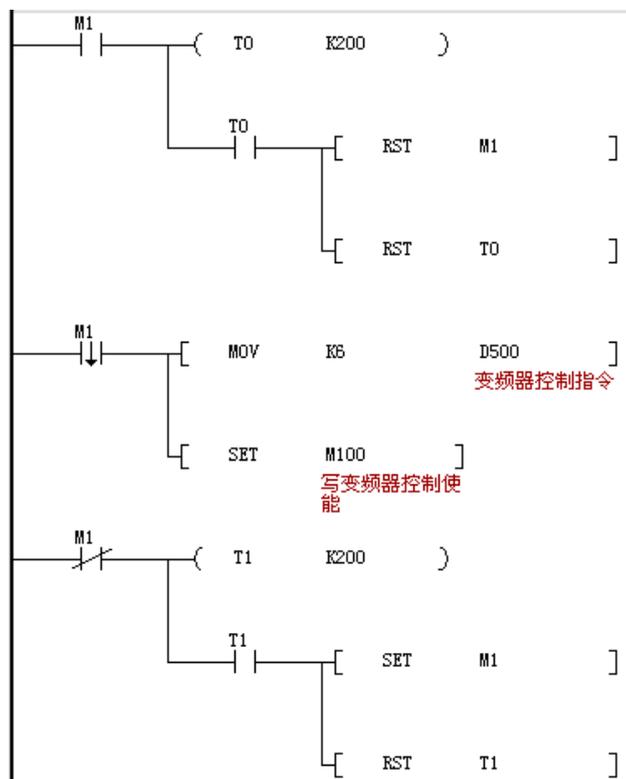
MD380 将当前状态 H3000 的值返回给主站，存放于主站的 D2000，返回的条件是 H3000 发生改变，且相邻两次发送的最小间隔为 100ms。

配置远程模块：M101 置位时，写 4DAR 的 BFM#20（模块复位寄存器）；每 50ms 发送输出值。

主站 (1) 配置											
发送配置			接收配置			同步配置					
编号	触发方式	触发条件	发送站		发送寄存器		接收站		接收寄存器		寄存器个数
1	时间 (ms)	50	1	HOST	502	十进制	2	NTCM (非温控)	1	十进制	1
2	事件 (M)	100	1	HOST	500	十进制	3	MD (变频器)	2000	十六进制	1
3	事件 (M)	101	1	HOST	501	十进制	2	NTCM (非温控)	20	十进制	1
4			1	HOST		十进制					
5			1	HOST		十进制					
6			1	HOST		十进制					
7			1	HOST		十进制					
8			1	HOST		十进制					
9			1	HOST		十进制					
10			1	HOST		十进制					
11			1	HOST		十进制					
12			1	HOST		十进制					
13			1	HOST		十进制					
14			1	HOST		十进制					
15			1	HOST		十进制					
16			1	HOST		十进制					
本站接收											
1	事件 (M)	100	3	MD (变频器)	3000	十六进制	1	HOST	2000	十进制	1

相关应用程序如下：





5.9.8 CANlink 常见故障与排除

1) 查看设备是否支持 CANlink3.0

设备	如何查看
PLC	请查看 D8280 的值： • 如 D8280=300，则表示支持 CANlink3.0，其他值则不支持。
变频器 / 伺服	请查看其软件版本： • MD380 要求 F7-11 为 4.71.06 或以上； • MD310 要求 F7-11 为 u37.18 或以上； • IS500 伺服要求 H00-02 的值为 214.xx； • IS620P 伺服要求 H01-00 的值为 6.0 或以上； • IS700 伺服要求 H01-00 的值为 301.05。
PLC 远程扩展模块	请查看软件版本 BFM#27 为 51210/52210 (16 进制 0xC80A/0xCB2F) 以上。

2) 检查匹配电阻

所有设备断电，用万用表测量网络任一端的 CANH 与 CANL 之间的阻值，应在 60Ω 左右，如果过小，则说明网络中不只是两端接入了匹配电阻，在其它位置还有错误接入，将错误接入的匹配电阻断开即可。如果只接入一个配备电阻，则会为 120Ω 左右，网络会通信质量很差。完全不接入配备电阻，网络无法通信。请接入网络首尾两个站点的匹配电阻。

3) 设备完全通信不上或者偶尔可以通信可能原因

- 波特率设置不正常。检查波特率，是否正常。设备波特率需要重新上下电或停止再运行后才可以生效。
- 电源连接不正常。PLC CAN 通信卡与 PLC 扩展模块需要外接 24V 电源，建议所有设备使用相同电源。如果是变频器或伺服为自身供电，请把 CANlink 电源的 CGND 端全部连接在一起，从而保持所有设备共 CAN 通信电源 CGND 端。
- 检查通信线、屏蔽线、电源间是否有短路现象。
- 如果以上都没有问题，那可能是现场干扰很大，在没有办法排除故障时，请尝试降低通信波特率。
- PLC 主站用户程序中不可以对 D7000~D7999 寄存器进行修改。如果修改将发生不可预知错误。D7800~D7863 仅用来监控从站状态。

4) PLC 主站的 FROM/TO 指令不可以与 CANlink3.0 配置功能同时应用。



第六章 扩展模块及扩展卡的使用方法

6.1 扩展模块概要.....	272
6.1.1 命名规则.....	272
6.1.2 基本型号.....	272
6.1.3 H _{2U} -CAN-BD (或H _{1U} -CAN-BD) 扩展卡.....	273
6.1.4 扩展模块地址编号.....	274
6.1.5 模块的安装尺寸及用户接口表.....	275
6.2 数字量输入/输出扩展模块.....	276
6.2.1 数字量输入扩展模块.....	276
6.2.2 数字量输出扩展模块.....	277
6.2.3 数字量输入输出扩展模块.....	278
6.3 模拟量输入模块.....	280
6.4 模拟量输出模块.....	285
6.5 模拟量混合(输入/输出)模块.....	289
6.5.1 H _{2U} -4AM 2通道模拟电流输入/2通道模拟电压电流输出混合扩展模块.....	289
6.5.2 H _{2U} -6AM 4通道模拟电流输入/2通道模拟电压电流输出混合扩展模块.....	295
6.5.3 H _{2U} -6CM 4通道模拟电压输入/2通道模拟电压电流输出混合扩展模块.....	301
6.6 温度控制模块.....	308
6.7 扩展卡使用说明.....	319
6.7.1 H _{2U} 系列6A/6B扩展卡使用说明.....	319
6.7.2 H _{2U} 系列以太网通讯扩展卡使用说明.....	321
6.7.3 H _{1U} /H _{2U} 系列通讯扩展卡使用说明.....	325
6.7.4 H _{1U} 系列4A扩展卡使用说明.....	327
6.8 差分转集电极输出模块使用说明.....	329

第六章 扩展模块及扩展卡的使用方法

6.1 扩展模块概要

6.1.1 命名规则

5) 数字量扩展模块:

H2U-0016ERNR

① ② ③ ④ ⑤ ⑥ ⑦ ⑧

- ①: 汇川控制器
- ②: 系列号
- ③: 输入点数
- ④: 输出点数
- ⑤: 产品类型。E-表示扩展模块
- ⑥: 输出类型。N-表示无输出；R-表示继电器输出；T-表示晶体管输出
- ⑦: 表示输入电源类型。N-表示无电源输入；D表示24VDC输入
- ⑧: 表示远程或本地扩展。R-表示远程扩展模块，缺省-表示本地扩展模块

6) 模拟量扩展模块:

H2U-4ADR

① ② ③ ④ ⑤

- ①: 汇川控制器
- ②: 系列号
- ③: 通道数
- ④: 特殊功能模块类型。AD-表示模拟量输入；DA-表示模拟输出；PT-表示热电阻输入；TC-表示热电偶输入；AM-表示混合模拟量模块；CM-表示混合模拟量模块。
- ⑤: 表示远程或本地扩展。R-表示远程扩展模块，缺省-表示本地扩展模块

6.1.2 基本型号

表: 基本型号

6 扩展模块 及扩展卡 使用方法	产品型号	产品名称	描述	适用机型	
				H1U-XP	H2U-XP
	H2U-0016ERN	H2U系列本地继电器输出扩展模块	16点继电器输出本地模块	—	√
	H2U-0016ETN	H2U系列本地晶体管输出扩展模块	16点晶体管输出本地模块	—	√
	H2U-1600ENN	H2U系列本地输入扩展模块	16点输入本地模块	—	√
	H2U-0800ENN	8点输入本地模块	H2U系列本地继电器输出扩展模块	—	√
	H2U-0008ERN	8点继电器输出本地模块	H2U系列本地继电器输出扩展模块	—	√
	H2U-0008ETN	8点晶体管输出本地模块	H2U系列本地晶体管输出扩展模块	—	√
	H2U-0404ERN	4点数字量输入4点继电器输出混合型本地模块	H2U系列本地继电器输出扩展模块	—	√
	H2U-0404ETN	4点数字量输入4点晶体管输出混合型本地模块	H2U系列本地晶体管输出扩展模块	—	√
	H2U-0808ERN	8点数字量输入8点继电器输出混合型本地模块	H2U系列本地继电器输出扩展模块	—	√
	H2U-0808ETN	8点数字量输入8点晶体管输出混合型本地模块	H2U系列本地晶体管输出扩展模块	—	√
	H2U-2AD	H2U系列本地模拟量输入模块	2通道电压电流输入本地模块	—	√
	H2U-2DA	H2U系列本地模拟量输出模块	2通道电压电流输出本地模块	—	√
	H2U-4AD	H2U系列本地模拟量输入模块	4通道电压电流输入本地模块	—	√
	H2U-4DA	H2U系列本地模拟量输出模块	4通道电压电流输出本地模块	—	√
	H2U-4HC	H2U系列计数模块	4通道计数模块	—	√
	H2U-4PT-XP	H2U系列本地热电阻输入模块	4通道热电阻输入本地模块	—	√
	H2U-4TC-XP	H2U系列本地热电偶输入模块	4通道热电偶输入本地模块	—	√
	H2U-4AM	H2U系列本地模拟量混合模块	2通道电压电流输入2通道电压电流输出本地模块	—	√
	H2U-6AM	H2U系列本地模拟量混合模块	4通道电流输入2通道电压电流输出本地模块	—	√
	H2U-6CM	H2U系列本地模拟量混合模块	4通道电压输入2通道电压电流输出本地模块	—	√
	H2U-2DC	H2U系列差分转集电极输出转换模块	2组差分转集电极输出转换模块	√	√

产品型号	产品名称	描述	适用机型	
			H1U-XP	H2U-XP
H2U-0016ERDR	远程继电器输出扩展模块	16点继电器输出远程模块	√	√
H2U-0016ETDR	远程晶体管输出扩展模块	16点晶体管输出远程模块	√	√
H2U-1600ENDR	远程输入扩展模块	16点输入远程模块	√	√
H2U-0808ERDR	8点数字量输入8点继电器输出混合型远程模块	远程继电器输出扩展模块	√	√
H2U-4ADR	远程模拟量输入模块	4通道电压电流输入远程模块	√	√
H2U-4DAR	远程模拟量输出模块	4通道电压电流输出远程模块	√	√
H2U-4PTR-XP	远程热电阻输入模块	4通道热电阻输入远程模块	√	√
H2U-4TCR-XP	远程热电偶输入模块	4通道热电偶输入远程模块	√	√

备注：H1U仅能接远程扩展模块，需要配H1U-CAN-BD卡；H2U分两种，通用机型仅能接本地模块，N系列带CAN机型可接本地和远程模块，接远程模块需要配H2U-CAN-BD卡。

6.1.3 H2U-CAN-BD (或 H1U-CAN-BD) 扩展卡

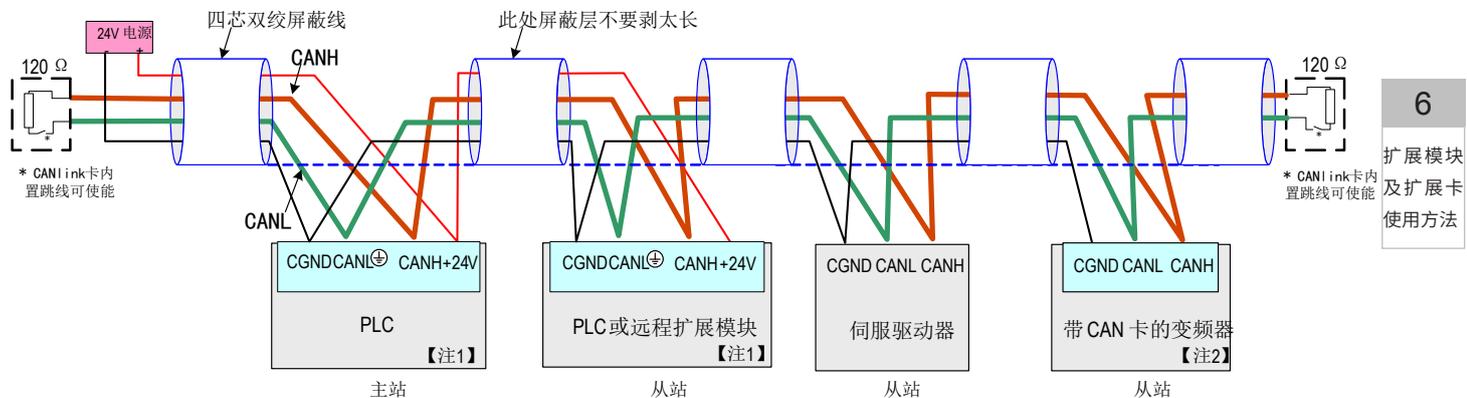
主模块CAN BD卡通信接口定义：

所有远程扩展模块通过CAN接口，连接到H2U-CAN-BD (或H1U-CAN-BD) 板，与PLC进行数据交互。CAN BD卡定义如下：

CAN端口定义：

信号排布图		管脚号	信号	描述
+24V	1	1	+ 24V	外接直流 24V 供电电源正
CANH	2	2	CANH	CAN 总线正
	3	3	保留	保留
CANL	4	4	CANL	CAN 总线负
CGND	5	5	CGND	外接直流 24V 供电电源负

组成CANlink 网络时，所有设备的以上五根线均要一一对应连在一起。并且+24Vcc和0V 间需要外接24V 直流电源。总线的两端均要加120 欧姆的CAN 总线匹配电阻，CAN BD卡本身有匹配电阻，使用时拨上即可，无需外接电阻。



【注1】 H1U/H2U系列PLC需另外配置CAN扩展卡，H3U系列PLC自带CAN通信接口，请根据需要选择；

【注2】 请根据变频器类型选择相应的CAN卡。

图 正确的 CANlink 网络接线

注：CAN通信电缆采用屏蔽电缆，推荐型号：RVVP 2×2×0.5。

6.1.4 扩展模块地址编号

1) 本地特殊扩展模块的地址编号:

除IO扩展以外的各种扩展模块（如4AD/4DA/4AM/4PT/CC-Link等模块），统称为特殊模块。PLC主模块每次上电时，会自动检查一次已接入的所有扩展模块，并分别对特殊模块和IO扩展端口进行“编号”，用户无法干预或更改其编号结果，除非改变模块的连接顺序。

主模块对特殊模块的地址编号方法是，由紧靠近主PLC模块开始进行，依次为#0、#1、…#7等编号，中间若插入的IO扩展模块不参与编号，如下图中，4AD扩展模块的模块地址编号为#0，4DA模块地址编号为#1，后续的4PT和4AM模块地址依次为#2和#3，以此类推，最多可接入8个特殊模块：



即使中间插有若干IO扩展模块，编号顺序也不受影响。了解了上述编址原则，用户在编程时就可以准确地访问指定模块。

上电前需要将模块的外部24V/COM电源端与PLC主模块的24V/COM对应连接妥当，以便主模块上电时，扩展模块也能同时上电。若扩展模块的外部电源上电滞后，可能导致PLC主模块不能正确辨识模块类型；PLC主模块只在电源上电时检查一次接入系统的所有扩展模块，对运行中插入的扩展模块，不会被PLC主模块检查到，无法正常对其进行访问；运行中插拔扩展模块，可能损坏器件，更严重的是可能导致控制输出状态的不可预知，导致用户设备故障。

2) 远程扩展模块的地址编号:

在远程扩展模块中，远程模块地址是：模块通信站号+100，最多允许有62个远程扩展模块（CANlink3.0版本最多允许30个远程扩展模块），每个远程模块都有一个八位的拨码开关，用户可以设定模块的站号，选择波特率，是否接终端匹配电阻（标配120Ω）。如下图所示，“ON”表示逻辑“1”。

表1: CANlink3.0版本拨码开关定义

终端电阻 8	定义	波特率		定义	地址设置					定义	
		7	6		5	4	3	2	1		
0	终端匹配电阻接入	0	0	500Kbps	0	0	0	0	0	0	保留
					0	0	0	0	1	1	1
		0	1	250Kbps	0	0	0	1	0	2	
					0	0	0	1	1	3	
1	终端匹配电阻断开	1	0	100Kbps	
					1	1	1	0	1	29	
		1	1	50Kbps	1	1	1	1	0	30	
					1	1	1	1	1	31	

表2: 非CANlink3.0版本拨码开关定义

终端电阻 8	定义	波特率 7	定义	地址设置					定义	
				6	5	4	3	2		1
0	终端匹配电阻接入	0	500Kbps	0	0	0	0	0	0	保留
				0	0	0	0	0	1	1
				0	0	0	0	1	0	2
				0	0	0	0	1	1	3
1	终端匹配电阻断开	1	100Kbps	
				1	1	1	1	0	1	29
				1	1	1	1	1	0	30
				1	1	1	1	1	1	63

终端电阻 → BR

波特率 → Baud

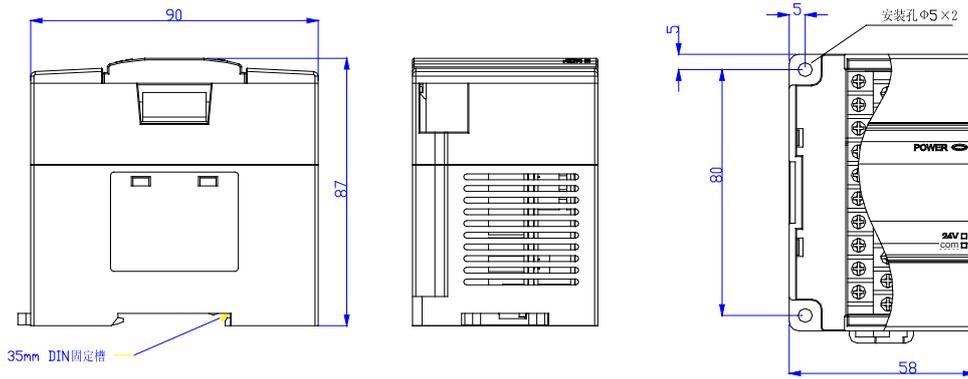
地址设置 → Station NO.

若改变拨码开关，除匹配电阻外，波特率和地址并不能立即生效，需要给系统重新上电才可以使新的设置参数。需注意CAN地址的唯一性，不能有重复的地址。

注：51210-0000和52210-0000（包括这两个版本）以上版本支持CANlink3.0。

6.1.5 模块的安装尺寸及用户接口表

1) 模块的安装尺寸:



2) 模块用户接口表:

项目	说明	
接线端子功能	V+: 通道的电压正输入\输出端子; I+: 通道的电流正输入\输出端子; VI-: 通道的输入\输出公共端; 24+/24-:外部24V输入端子; GND: 保护接地。	
指示灯	本地扩展模块	PWR: 当与主模块连接且主模块上电的情况下点亮。 COM: 当模块有访问是点亮, 无访问时熄灭 24V: 当外部24V接通时点亮。
	远程扩展模块	PWR: 模块24V供电正常 COM: 当使用FROM/TO指令访问时, 有访问点亮, 无访问熄灭。 COM与ERR: 当使用CANlink3.0配置访问时, 请参考CANlink3.0应用手册相关说明。
扩展端口	本地扩展模块	扩展输入端采用26针梯形连接头, 通过扁平电缆接入模块; 扩展输出端采用26针梯形连接头。
	远程扩展模块	扩展输入采用5针孔插头, 建议用4芯双绞屏蔽线或者网线做传输导线

6.2 数字量输入/输出扩展模块

6.2.1 数字量输入扩展模块

H2u-1600ENN H2u-1600ENDR H2u-0800ENN

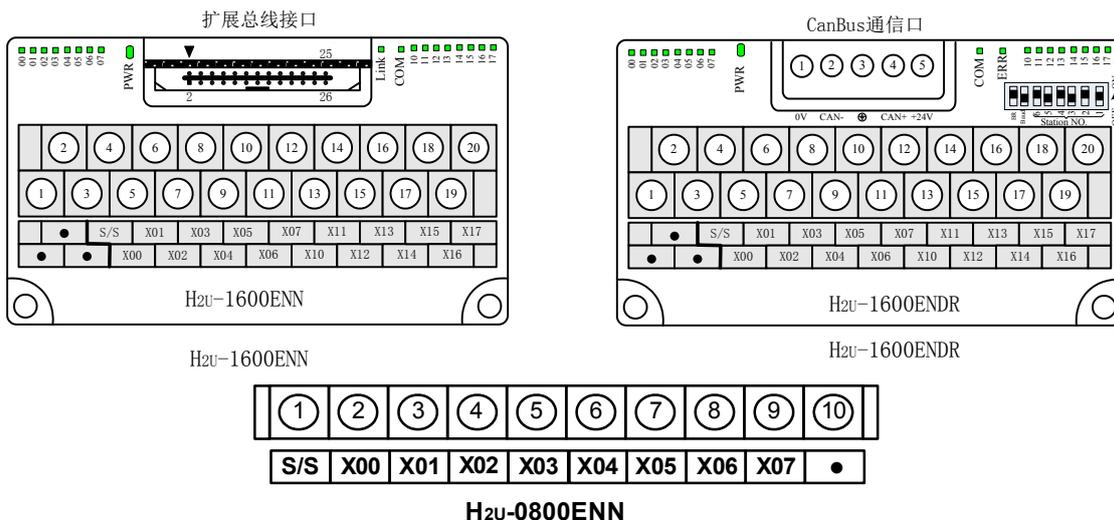
1) 电气规格:

项目		普通输入端
信号输入方式		漏型/源型方式。当S/S端子与24V短接时为漏型输入；当S/S端子与COM短接时为源型输入。
电气参数	检测电压	DC24V
	输入阻抗	4.3k
	输入为ON	3.5mA Min
	输入为OFF	1.5mA Max
滤波功能	数字滤波	无
	硬件滤波	滤波时间约10ms
公共接线端		只有一个公共端，为S/S

2) 模块用户接口表

项目	说明	
接线端子功能	X00~X17: 输入点 S/S: 输入点公共端, 漏型输入接+24V, 源型输入接0V	
指示灯	本地扩展模块	PWR: 模块数字电路供电正常 X00~ X17: 输入点指示灯 LINK: 主模块访问扩展模块时点亮 RUN: 当模块正常工作时点亮。
	远程扩展模块	PWR: 模块24V供电正常 X00~ X17: 输入点指示灯 COM: 当使用FROM/TO指令访问时, 有访问点亮, 无访问熄灭。 COM与ERR: 当使用CANlink3.0配置访问时, 请参考CANlink3.0应用手册的相关说明。
扩展端口	本地扩展模块	扩展输入端采用26针梯形连接头, 通过扁平电缆接入模块; 扩展输出端采用26针梯形连接头。
	远程扩展模块	扩展输入采用5针孔插头, 建议用4芯双绞屏蔽线或者网线做传输导线

3) 端口定义图



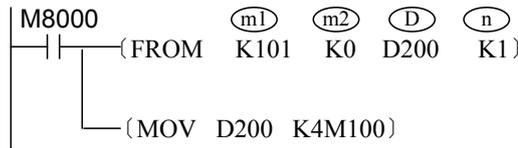
4) 本地扩展和远程扩展编程方法:

当本地扩展模块后接到主模块上后, 扩展模块上X端口的编号紧接主模块上X端口的编号, 依次向后编号, 例如当主模块为H2u-1616MR, 现在要接入H2u-1600ENN扩展模块, 因主模块最后的X端口编号为X17, 则扩展模块的X在编程时的访问编号为X20~X37, 即扩展模块上的X0点对应程序里的X20, 依次类推。

注意, 扩展模块的编号总是从8进制个位为0开始的, 例如, 当主模块为H2u-3624MR, 其最后的X端口编号为X43, 扩展模块的X在编程时的访问编号为X50~X67, 即主模块上空缺的X44~X47的端口号被丢弃。扩展模块上Y端口也采取了同样的处理方法。

远程数字量扩展模块不能直接访问X、Y点。必须通过FROM/TO指令访问, 编程方式如下:

6
扩展模块
及扩展卡
使用方法



本例子介绍远程数字量扩展模块H2U-1600ENDR的访问方法，因为16点的输入模块用一个BFM区代表了所有的输入点，即BFM区的K0（一个寄存器）的每个bit位对应每个输入点，故可以将模块的状态读取回来后用MOV D200 K4M100来分别取位，此时M100~M107为远程模块的X0~X7，M108~M115为远程模块的X10~X17，我们可以通过读取M100~M115就能读取X0~X17的状态，注意：为了保证能读取到数字量扩展模块输入点的最新状态，请使用M8000来驱动FROM 指令，以便在每个扫描周期都能更新。

6.2.2 数字量输出扩展模块

H2U-0016ERN H2U-0016ERDR H2U-0016ETN H2U-0016ETDR H2U-0008ERN H2U-0008ETN

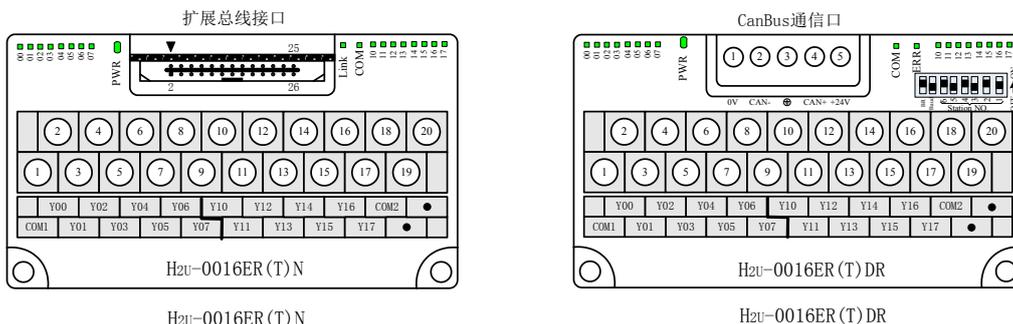
1) 电气规格:

项目		继电器输出端口	晶体管输出端口
回路电源电压		AC250V, DC30V以下	DC5~24V
隔离		继电器机械绝缘	光耦绝缘
动作指示		继电器输出触点闭合LED亮	光耦被驱动时LED点亮
开路时漏电流		/	小于0.1mA/DC30V
最小负载		2mA/DC5V	5mA (DC5~24V)
最大输出电流	电阻负载	2A/1点; 8A/4点组公共端; 8A/8点组公共端	0.5A/点; 0.8A/4点; 1.6A/8点
	感性负载	AC220V, 80VA	12W/DC24V
	电灯负载	AC220V, 100W	1.5W/DC24V
ON响应时间		20ms Max	0.5ms Max
OFF响应时间		20ms Max	
输出公共端		每一组共用一个公共端, 组与组之间隔离	
熔断器保护		无	

2) 模块用户接口表

项目	说明	
接线端子功能	Y00~Y17: 输出点 COM1~2: 输出点公共端	
指示灯	本地扩展模块	PWR: 模块数字电路供电正常 Y00~Y17: 输入点指示灯 LINK: 主模块访问扩展模块时点亮 RUN: 当模块正常工作时点亮。
	远程扩展模块	PWR: 模块24V供电正常 Y00~Y17: 输入点指示灯 COM: 当使用FROM/TO指令访问时, 有访问点亮, 无访问熄灭。 COM与ERR: 当使用CANlink3.0配置访问时, 请参考CANlink3.0应用手册的相关说明。
扩展端口	本地扩展模块	扩展输入端采用26针梯形连接头, 通过扁平电缆接入模块; 扩展输出端采用26针梯形连接头。
	远程扩展模块	扩展输入采用5针孔插头, 建议用4芯双绞屏蔽线或者网线做传输导线

3) 端口定义图



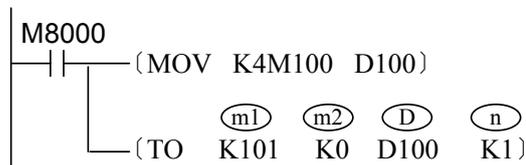
6
扩展模块及扩展卡使用方法

4) 本地扩展和远程扩展编程方法

当本地扩展模块后接到主模块上后，扩展模块上Y端口的编号按紧接主模块上Y端口的编号，依次向后编号，例如当主模块为H_{2U}-1616MR，现在要接入H_{2U}-0016ERN扩展模块，因主模块最后的Y端口编号为Y17，则扩展模块的Y在编程时的访问编号为Y20~Y37，即扩展模块上的Y0点对应程序里的Y20，依次类推。

注意：扩展模块的端口编号总是从8进制个位为0开始的。继电器输出扩展模块可以接到继电器或者晶体管主模块上面；同样，晶体管输出扩展模块也可以接到晶体管或者继电器主模块上面。

远程数字量扩展模块不能直接访问X、Y点。必须通过FROM/TO指令访问，编程方式如下：



本例子介绍远程数字量扩展模块H_{2U}-0016ERDR的访问方法，因为16点的输入模块用一个BFM区代表了所有的输出点，即BFM区的K0（一个寄存器）的每个bit位对应每个输出点，故可以将全部输出点的状态放到一个字里面后再传送到模块里面去，例子中M100~M107为远程模块的Y0~Y7，M108~M115为远程模块的Y10~Y17，我们通过改变M100~M115就可以改变远程模块中Y0~Y17的状态，在远程数字量输出模块中，可以在每次更新输出点后再驱动TO指令，也可以用M8000来驱动。

远程输出扩展模块新增加BFM#5的功能：为通讯超时判断时间，时间单位为1ms，当主模块与本远程模块的通讯失去通讯联系的时间超过设定值后，本远程模块的所有Y输出端口将全部呈关闭状态（默认等待通讯时间为500ms）；若BFM#5=0时，本功能无效，模块保持原输出状态。

6.2.3 数字量输入输出扩展模块

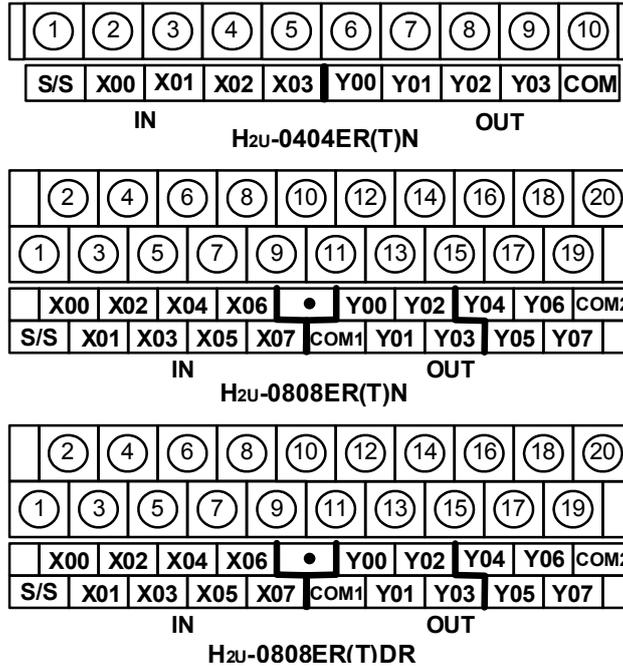
H_{2U}-0016ERN H_{2U}-0016ERDR H_{2U}-0016ETN H_{2U}-0016ETDR H_{2U}-0808ERN H_{2U}-0808ETN

1) 电气规格:

项目		普通输入端
信号输入方式		漏型/源型方式。当S/S端子与24V短接时为漏型输入；当S/S端子与COM短接时为源型输入。
电气参数	检测电压	DC24V
	输入阻抗	4.3k
	输入为ON	3.5mA Min
	输入为OFF	1.5mA Max
滤波功能	数字滤波	无
	硬件滤波	滤波时间约10ms
公共接线端		只有一个公共端，为S/S

项目		继电器输出端口	晶体管输出端口
回路电源电压		AC250V, DC30V以下	DC5~24V
隔离		继电器机械绝缘	光耦绝缘
动作指示		继电器输出触点闭合LED亮	光耦被驱动时LED点亮
开路时漏电流		/	小于0.1mA/DC30V
最小负载		2mA/DC5V	5mA (DC5~24V)
最大输出电流	电阻负载	2A/1点; 8A/4点组公共端; 8A/8点组公共端	0.5A/点; 0.8A/4点; 1.6A/8点
	感性负载	AC220V, 80VA	12W/DC24V
	电灯负载	AC220V, 100W	1.5W/DC24V
ON响应时间		20ms Max	0.5ms Max
OFF响应时间		20ms Max	
输出公共端		每一组共用一个公共端，组与组之间隔离	
熔断器保护		无	

2) 端口定义图



3) 本地扩展和远程扩展编程方法

当本地扩展模块后接到主模块上后，扩展模块上X、Y端口的编号紧接主模块上X、Y端口的编号，依次向后编号，例如当主模块为H2U-1616MR，现在要接入H2U-0808ENN扩展模块，因主模块最后的X端口编号为X17，则扩展模块的X在编程时的访问编号为X20~X27，主模块最后的Y端口编号为Y17，扩展模块的Y在编程时的访问编号为Y20~Y27，依次类推。

注意，扩展模块的编号总是从8进制个位为0开始的，例如，当主模块为H2U-3624MR，其最后的X端口编号为X43，扩展模块的X在编程时的访问编号为X50~X57，即主模块上空缺的X44~X47的端口号被丢弃。扩展模块上Y端口也采取了同样的处理方法。

远程数字量扩展模块不能直接访问X、Y点。必须通过FROM/TO指令访问。

4) 访问数字量远程扩展模块的BFM区：

模块型号	寄存器定义	备注
H2U-1600ENNR	BFM #0=扩展模块的输入端口状态（读取X输入）	按位对齐
H2U-0016ERNR	◆ BFM#0=扩展模块输出缓冲区（设置Y输出） ◆ BFM#5=CAN通讯超时自动关闭输出时间（ms单位，1~32767，出厂默认为500ms；设为0表示禁止该功能）	按位对齐
H2U-0808ERNR	◆ BFM#0=扩展模块输入缓冲区（读取X输入） ◆ BFM#1=扩展模块输出缓冲区（设置Y输出） ◆ BFM#5=CAN通讯超时自动关闭输出时间（ms单位，1~32767，出厂默认为500ms；设为0表示禁止该功能）	低8位对齐

6.3 模拟量输入模块

H2u-4AD (R) /H2u-2AD 4通道/2通道模拟量输入扩展模块

1) 简介

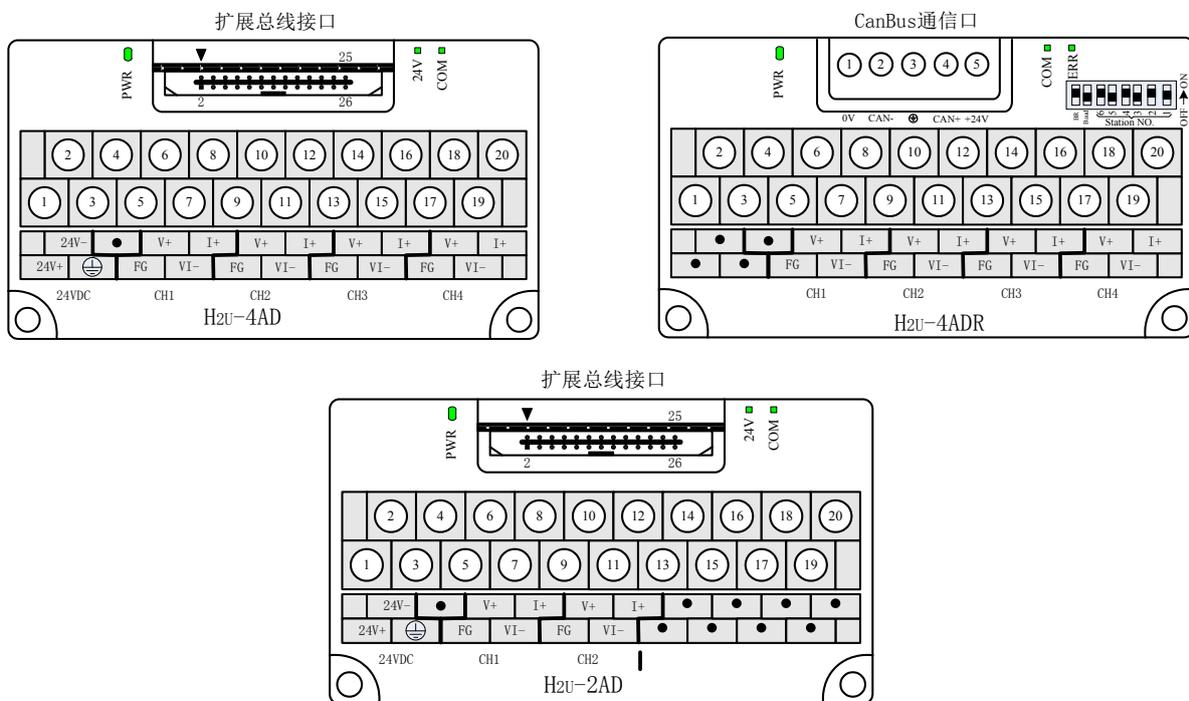
4AD (R) /2AD扩展模块可配合H2u系列主模块工作, 实现4个/2个模拟输入通道的信号检测, 将-10V~10V或-20mA~20mA的信号转换为12bit的数字量, 供PLC主模块读取。主模块通过FROM/TO指令访问本扩展模块内寄存器的BFM单元。

备注: 4AD (R) 与2AD的区别: 4AD (R) 比2AD在硬件上多了2个输入通道, BFM、使用方法、编程等一样。

2) 电气规格

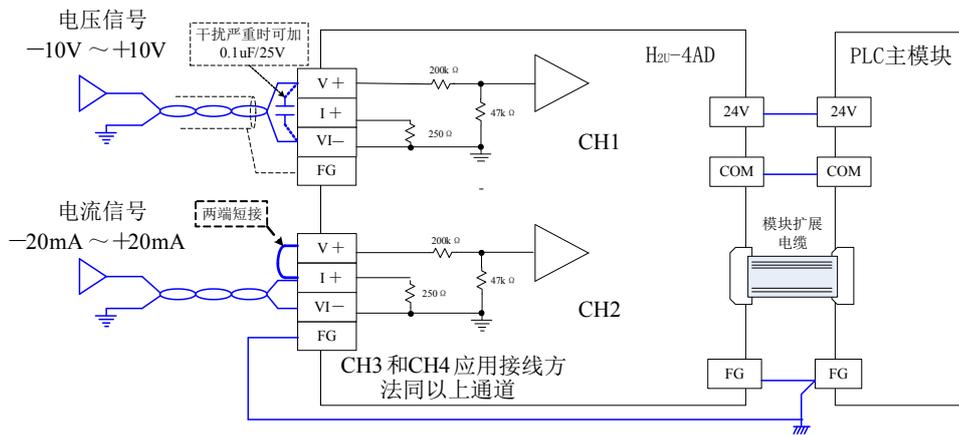
项目		指标	
电源	模拟电路	24V DC -15%/+20%, 最大允许纹波电压5% 电流消耗80mA(取自于主模块的24V/COM, 或其它的24VDC电源)	
	数字电路	5V DC 50mA(通过模块扩展电缆, 取自于主模块电源内部电源)	
	备注	远程扩展模块无需数字电路电源。	
输入通道数		4AD (R): 4通道; 2AD: 2通道	
模拟输入范围	电压输入	-10~10V DC, (输入阻抗为200K Ω)	用户需根据接入的信号类型, 设置相应的BFM区。
	电流输入	-20~20mA(输入阻抗为250 Ω)	
数字输出		-2048 ~ +2047	
分辨率	电压输入	5mV	
	电流输入	20 μ A	
转换速度		15ms/通道(常速), 6ms/通道(快速)	
精度		\pm 1%全范围	
占用I/O点数		不占用主模块I/O点数	
隔离		模拟电路和数字电路之间用光电耦合器进行隔离; 模拟电路和外部电源用DC/DC进行隔离; 模拟通道之间不隔离。	

3) 接线端子布局:



4) 输入信号与扩展电缆的接线:

模拟输入信号通过双绞线连接到扩展模块的输入端口, 布线时不要与交流电源线或干扰信号的线路靠近; 若模拟信号的干扰严重时, 可采用屏蔽线连接, 并在输入端口并联1只0.1 μ F/25V的高频电容; 信号源及其屏蔽线的外壳与H2u-4AD的信号接地端FG相连, 共同接地。



备注：若其中任意通道，输入信号超出规定范围，就会影响其他通道的精度。

5) 访问4AD (R) /2AD模块的BFM区：

PLC主模块是通过读取4AD (R) /2AD模块的寄存器缓存单元 (BFM区) 的方式读取数字化AD转换结果，通过改写特定BFM区的方式来设置模块状态。PLC主模块通过读写指令FROM/TO访问这些BFM单元。

扩展模块内设有EEPROM存储单元，用于保存一些BFM设定值,例如每个模拟输入通道的信号类型、偏移值、增益值等，这些单元的保存动作是由相应BFM单元的设置状态决定而自行完成的。

BFM区的每个寄存器宽度为16bit (即1Word)，按照4AD (R) /2AD模块的BFM区定义如下表： (2AD的通道3和通道4保留)

BFM	R/W属性	内容	
#0 (E)	WR	通道信号模式选择，每个HEX位代表1个输入通道，4AD (R)：最高位为ch4，最低位为ch1； (默认值=H0000)，2AD：取低8位中的高HEX位为ch2，低HEX位为ch1； (默认值=H00) 0=-10V~10V；对应数字输出：-2000~2000 1=4mA~20mA；对应数字输出：0~1000 2=-20mA~20mA；对应数字输出：-1000~1000 3=本通道关闭； 4=-10V~10V；对应数字输出：-10000~10000 5=4mA~20mA；对应数字输出：0~10000 6=-20mA~20mA；对应数字输出：-10000~10000	例图： □ □ □ □ ch 4 3 2 1
#1	WR	通道1	平均滤波常数，即用于平均计算的采样值个数，设定范围1~4096，默认值8。若要高速采集，可设定为1。当BFM#15改变时，自动恢复为默认值。
#2	WR	通道2	
#3	WR	通道3	
#4	WR	通道4	
#5	R	通道1	输入通道采集值平均滤波后的数据
#6	R	通道2	
#7	R	通道3	
#8	R	通道4	
#9	R	通道1	输入通道当前采集的数据，即未滤波处理的瞬时值。
#10	R	通道2	
#11	R	通道3	
#12	R	通道4	
#13~14	-	保留；	
#15	WR	ADC速率选择	0=正常速度，15ms/通道 (默认值)； 1=快速转换，6ms/通道； 1000~30000=高速采样，对应1ms~30ms/通道
#16~19	-	保留	
#20 (E)	WR	1=复位设定参数到默认值 (出厂值)。默认值=0	
#21 (E)	WR	1=允许调整偏移/增益 (默认值)； 2=禁止调整偏移/增益 (禁止修改输出转换的曲线特性，其中包括偏移、增益、通道信号模式)	

6
扩展模块及扩展卡使用方法

BFM	R/W属性	内容								
# 22	WR	低8bit位对应4个通道的操作4AD (R)	G4	O4	G3	O3	G2	O2	G1	O1
		低4bit位对应2个通道的操作2AD	—	—	—	—	G2	O2	G1	O1
		偏移/增益调整使能, 当非0时, 模块会将BFM23/24值写入其内部对应通道控制寄存器中, 初始值=H00								
# 23	WR	偏移值, 数字输出为0时的模拟输入值 (0、1、2模式) 初始值为0								
# 24	WR	增益值, 数字输出为+1000时的模拟输入值 (0、1、2模式) 初始值为5000								
# 25~26	—	保留								
# 27	R	4AD/2AD模块软件版本								
# 28	—	保留								
# 29	R	错误状态								
# 30	R	扩展模块识别码, H _{2U} -4AD (R) 的识别码为K2010; H _{2U} -2AD的识别码为K2011								
# 31	—	保留, 不可访问								

注释: 模块掉电后输出转换的曲线特性保存, 其中包括曲线的偏移增益特性。当前BFM区显示的偏移增益为将要生效的值, 并不是当前已经生效的偏移增益。

其中状态信息字BFM #29的意义说明如下:

BFM # 29位号	ON状态	OFF状态
b0:	存在错误。b0~b3中任一个非0, A/D转换停止	无错误
b1:	模块内EEPROM的偏移/增益设置有误	偏移/增益数据正确
b2:	保留	保留
b3:	模块硬件故障	硬件正常
b10:	数字输出超出-2048~2047的范围	数字输出值正常
b11:	采样滤波常数超出1~4096范围	采样滤波常数正常
b12:	禁止BFM#21的值设为K2	允许BFM#21=K2
BFM # 29的其它bit4~7, bit13~15等没有定义。		

其中表中的“(E)”字样的BFM单元为存入EEPROM的项目, 具有掉电保持特性。

寄存器改写说明:

BFM#0、#23、#24等带(E)单元的改写引发模块内部对EEPROM的写操作, 而写操作需要一定时间, 每个Word约需300ms时间, 因此在需要改写多个带(E)的BFM单元时, 注意PLC编程时, 用户程序中每写一个上述的BFM单元后要延时一段时间, 不要连续进行写操作, 确保写指令的正确完成。

注意: 本地扩展模块的外部24V电源失电, PLC主模块的系统标志M8067会置位, 出错代码D8067=K6708。编程时可定时检查该标志, 能及时发现该现象。

6) 部分BFM区解释

BFM#0 通道选择

通道的初始化, 默认4个通道都为-10V~10V, 由BFM #0的十六进制HXXXX控制, 最低位控制通道一, 依次顺序, 最高位控制通道四, 每个字符的控制方式如下:

X=0 预设范围-10V~10V (对应数字-2000~2000)

X=1 预设范围4mA~20 mA (对应数字0~1000)

X=2 预设范围-20mA~20 mA (对应数字-1000~1000)

X=3 本通道关闭

X=4 预设范围-10V~10V (对应数字-10000~10000)

X=5 预设范围4mA~20 mA (对应数字0~10000)

X=6 预设范围-20mA~20 mA (对应数字-10000~10000)

例如: BFM#0为H1230, 表示通道一为-10V~10V; 通道二关闭; 通道三为-20mA~20 mA; 通道四为4mA~20 mA。

没有用到的通道, 可以关闭, 也可以不关闭, 关闭的通道不占用转换时间 (BFM#15), 例子中通道二关闭了, 则整个通道转换一次时间为没有关闭的三个通道的转换时间 (3×BFM#15)。

BFM #1~#4 平均采样数

每个通道(BFM#9~#12)对应的采样值累加采样数 (BFM #1~#4) 的个数后再除以采样数 (BFM #1~#4) , 存放到 (BFM #5~#8)

BFM #5~#8 存放平均采样值

BFM #9~#12 存放即时采样值

BFM #15 ADC速率时间

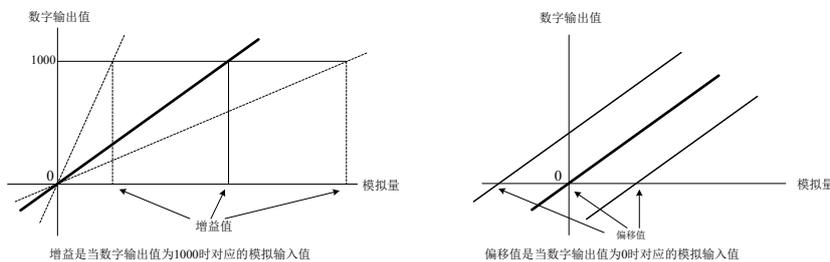
每个通道转换一次需要的时间, 需注意更新一次数据需要的时间是BFM#15的时间乘以没有关闭的通道数。

例如: BFM #0为H3310, BFM #1为K7, BFM #2为K6, BFM #15为K10; 则BFM #9和BFM #10刷新一次数据时间为 $BFM\#0 \times BFM\#15 = 2 \times 10 = 20MS$, BFM #5刷新一次数据时间为 $BFM\#0 \times BFM\#15 \times BFM\#1 = 2 \times 10 \times 7 = 140MS$, BFM #6刷新一次数据时间为 $BFM\#0 \times BFM\#15 \times BFM\#2 = 2 \times 10 \times 6 = 120MS$ 。程序中FROM/TO指令比较耗时, 故此模块参数在程序中采集BFM#5的数据可以用LDP M8012 FROM K0 K5 D10 K1指令读取, 和LD M8000 FROM K0 K5 D10 K1的效果是一样的, 但是后面用M8000驱动指令每个扫描周期都读取一次, 大大加长了程序的扫描周期。

BFM #20 回归出厂值

将#20设定为1可以恢复到默认值。

BFM #21~#24 偏移和增益的定义与设定方法:



偏移和增益可以独立设定或一起设定, 正常的增益值设定范围是1V~15V或4mA~32mA。正常的偏移值设定范围为-5V~5V, 或-20mA~20mA。

增益/偏移设定前, 需先将BFM#21设为1, 再修改BFM#23/24; 然后开各个通道的允许偏移增益BFM#22, 修改完毕, 应将BFM#21再设为2, 避免再次被改变。

例如: 在BFM#0为模式0中, 需要修改通道一、通道二的偏移和增益分别为0.5V和6V, 则需要按如下步骤操作:

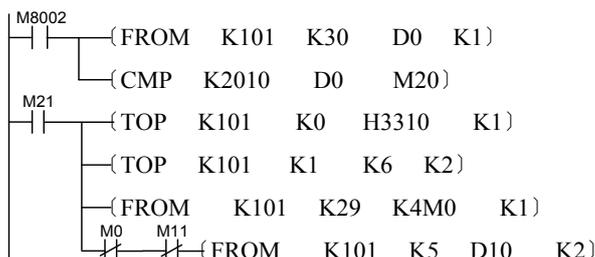
先将BFM#21改为1; 过300ms以后将K500和K6000分别送到BFM#23和BFM#24去; 再过300MS后开允许增益BFM#22, 本例子中BFM#22应为二进制的00001111, 即是将BFM#22修改为H000F; 修改完毕。最后将BFM#21改为2, 以防再次被修改。

编程举例1:

一只H2u-4AD扩展模块接于PLC主模块后方, 按编号原则为#0号模块, 其中CH1端口需要采集-10V~10V的电压信号, CH2需要采集4~20mA的电流信号, CH3/CH4未使用。要求改滤波次数为6, 将两个通道采集得到的数据分别存于D10、D11。编写的用户程序如下:



例子中如果改成H2u-4ADR远程模块, CAN站号为1。则此例子程序如下:



6
扩展模块
及扩展卡
使用方法

编程举例2:

一只H_{2u}-4AD扩展模块接于PLC主模块后方，按编号原则为#0号模块，要求通过一个X20按钮引发对其CH1通道的偏移量为0V，增益量为2.5V。编程如下:



使用注意事项:

禁止带电插拔，只有在主模块和应用系统停电的情况下，才能进行扩展模块的接入或拆除工作，以保证人身安全，防止因带电插拔损坏器件。

6.4 模拟量输出模块

H2u-4DA (R) / H2u-2DA 4通道/2通道模拟量输出扩展模块

1) 简介

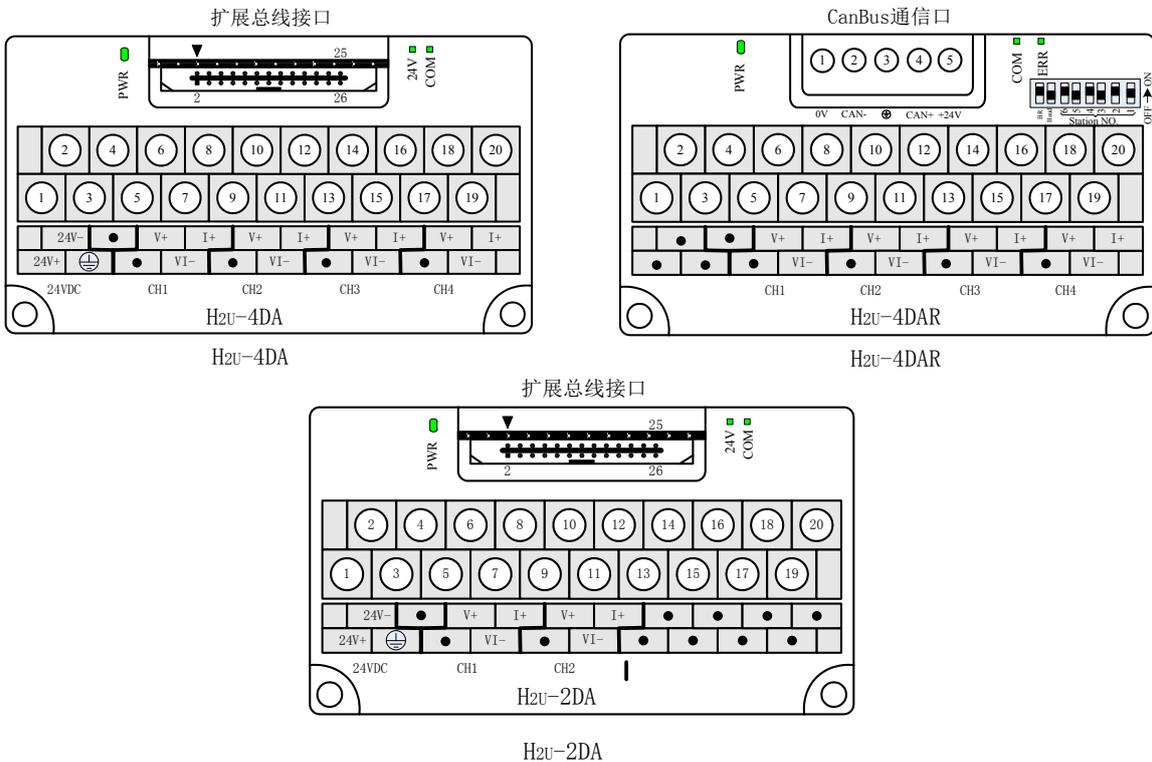
4DA (R) /2DA扩展模块可配合H2u系列主模块工作, 实现4个/2个模拟信号通道的输出, 每个通道都具有电压信号及电流信号输出端口, 信号幅值分别可为-10~10V或0~20mA。主模块通过FROM/TO指令访问本扩展模块内存寄存器的BFM单元, 实现模拟输出信号的控制。

备注: 4DA (R) 与2DA的区别: 4DA (R) 比2DA在硬件上多了2个输出通道, BFM、使用方法、编程等一样。

2) 电气规格:

项目		指标	
电源	模拟电路	24V DC -15%/+20%, 最大允许纹波电压5% ; 电流消耗200mA((取自于主模块的24V/COM, 或其它的24VDC电源))	
	数字电路	5V DC 50mA(取自于主电源内部电源或有源扩展单元)	
	备注	远程扩展模块无需数字电路电源。	
输出通道数		4DA (R) : 4通道; 2DA: 2通道	
模拟输出范围	电压输出	-10~10V DC(外部负载阻抗为2KΩ)	用户需根据接入的信号类型, 设置相应的BFM区。
	电流输出	4DA (R) : 4~20mA(外部负载阻抗为100Ω~500Ω) 2DA: 0~20mA(外部负载阻抗为500Ω或更小)	
数字输入		默认设置: -2000 ~ 2000	允许范围: -10000 ~ 10000
分辨率	电压输出	5mV	对应于10V/2000
	电流输出	20 μ A	对应于20mA /1000
转换速度		4通道: 4ms/通道 2通道: 2.1ms/通道	
精度		±1% 全范围	
占用I/O点数		不占用主模块I/O点数	
隔离		模拟电路和数字电路之间用光电耦合器进行隔离; 模拟电路电源和外部电源用DC/DC进行隔离; 模拟通道之间不隔离。	

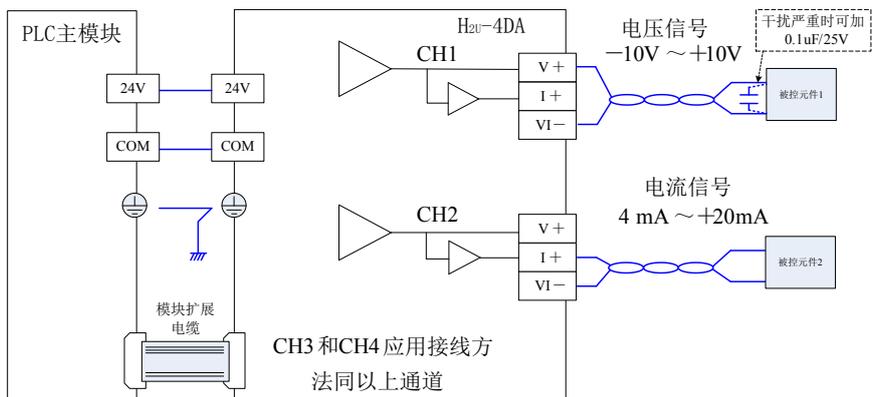
3) 接线端子布局如下图:



6
扩展模块
及扩展卡
使用方法

4) 输出信号的接线:

模拟输入信号通过双绞线连接到扩展模块的输入端口，布线时不要与交流电源线或干扰严重的线路靠近；若现场的干扰严重时，可采用屏蔽线进行信号连接，并在被控设备电压输入端口并联1只0.1 μF/25V的高频电容；信号源及其屏蔽线的外壳与PLC的FG端、被控设备的信号接地端相连，共同接地。



5) 访问4DA(R) / 2DA模块的BFM区:

PLC主模块是通过向4DA(R) / 2DA模块的寄存器缓存单元 (BFM区) 的写入数字值，由4DA(R) / 2DA转换为模拟量输出，通过改写特定BFM区的方式来设置模块状态。PLC主模块通过读写指令FROM/TO访问这些BFM单元。

扩展模块内设有EEPROM存储单元，用于保存一些BFM设定值,例如每个模拟输入通道的信号类型、偏移值、增益值等，这些单元的保存动作是由相应BFM单元的设置状态决定而自行完成的。

BFM区的每个寄存器宽度为16bit (即1Word)，按照4DA(R) / 2DA模块的BFM区定义如下表：(2DA的通道3和通道4保留)

BFM	R/W属性	内容			
#0 (E)	WR	输出模式选择，每个HEX位代表1个输入通道，4DA(R)：最高位为ch4，最低位为ch1：(默认值=H0000)；2DA：取低8位中的高HEX位为ch2，低HEX位为ch1：(默认值=H00) 0=-10V~10V；对应数字输出：-2000~2000 1=4mA~20mA；对应数字输出：0~1000 2=0mA~20mA；对应数字输出：0~1000 3=本通道关闭； 4=-10V~10V；对应数字输出：-10000~10000 5=4mA~20mA；对应数字输出：0~10000 6=0mA~20mA；对应数字输出：0~10000			
#1	WR	通道1	通道输出值，初始值为0		
#2	WR	通道2			
#3	WR	通道3			
#4	WR	通道4			
#5 (E)	WR	PLC停机时数值保留模式，每个HEX位代表1个输出通道(4DA(R)：Hxxxx,最高位为CH4,最低位为CH1；2DA：取低8位中的高HEX位为ch2，低HEX位为ch1，)，当：x=0，保持停机前的输出；x=1，将输出复位到偏移设定值			
#6~7	—	保留			
#8	WR	G2	O2	G1	O1
		CH2/CH1的偏移/增益设置命令，按HEX(4个2进制的bit)位设置，初始值=H0000 0=禁止改变；1=允许改变EEPROM对应数据			
#9	WR	G4	O4	G3	O3
		CH3/CH4的偏移/增益设置命令，按HEX(4个2进制的bit)位设置，初始值=H0000 0=禁止改变；1=允许改变对应BFM的数据			
#10	WR	偏移数据CH1	单位：mV或μA 初始偏移值：0 初始增益值：+5000,对应模式0		
#11	WR	增益数据CH1			
#12	WR	偏移数据CH2			
#13	WR	增益数据CH2			
#14	WR	偏移数据CH3			
#15	WR	增益数据CH3			
#16	WR	偏移数据CH4			
#17	WR	增益数据CH4			
#18~19	—	保留			

6
扩展模块
及扩展卡
使用方法

BFM	R/W属性	内容
# 20 (E)	WR	初始值=0, 当写入1时, 所有BFM单元将初始化为默认值。
# 21 (E)	WR	1=允许调整偏移/增益 (默认值); 2=禁止调整偏移/增益 (禁止修改输出转换的曲线特性, 其中包括偏移、增益、通道信号模式)
# 22~26	—	保留
# 27	R	扩展模块软件版本
# 28	—	保留
# 29	R	错误状态字
# 30	R	模块识别码, 4DA (R) 模块的识别码为K3020; 2DA模块的识别码为K3021。
# 31	—	保留

注释: 模块掉电后输出转换的曲线特性保存, 其中包括曲线的偏移增益特性。当前BFM区显示的偏移增益为将要生效的值, 并不是当前已经生效的偏移增益。

其中状态信息字BFM #29的意义说明如下:

BFM # 29位号	ON状态	OFF状态
b0:	存在错误。b1~b3中任一非0, D/A转换停止	无错误
b1:	模块内EEPROM的偏移/增益设置有误	偏移/增益值数据正确
b2:	保留	保留
b3:	模块硬件故障	硬件正常
b10:	写入数字值超出指定的范围 (-2350~2350)	数字输出值正常
b12:	禁止BFM#21的值没有设为K1	BFM#21=K1
BFM # 29的其它bit4~7, bit11, bit13~15等没有定义。		

寄存器改写说明:

表格中 (E) 表示存于EEPROM中, 注意BFM # 8、# 9、# 20等带 (E) 的单元不能频繁写操作, 以免损坏EEPROM单元。

BFM # 0单元的改写引发模块内部对EEPROM中BFM # 10~ # 17多个单元的写操作, 而写操作需要一定时间, 共约需3s时间。注意PLC编程时, 在改写BFM # 21与BFM # 10~ # 17指令之间大约需要3秒延时; 确保写指令的正确完成, 且不要连续进行写操作, 以免损坏EEPROM单元。

b10: 当BFM0的通道使用0、1、2时, 对应 (-2350~2350); 当BFM0通道使用4、5、6时, 对应 (-11750~11750)。

注意: 本地扩展模块的外部24V电源失电, PLC主模块的系统标志M8067会置位, 出错代码D8067=K6708。编程时可定时检查该标志, 能及时发现该现象。

6) 部分BFM区解释

BFM#0 通道选择

通道的初始化, 默认4个通道都为-10V~10V, 由BFM #0的十六进制HXXXX控制, 最低位控制通道一, 依次顺序, 最高位控制通道四, 每个字符的控制方式如下:

- X=0 预设范围-10V~10V (对应数字-2000~2000)
- X=1 预设范围4mA~20 mA (对应数字0~1000)
- X=2 预设范围0mA~20 mA (对应数字0~1000)
- X=3 本通道关闭
- X=4 预设范围-10V~10V (对应数字-10000~10000)
- X=5 预设范围4mA~20 mA (对应数字0~10000)
- X=6 预设范围0mA~20 mA (对应数字0~10000)

例如: BFM#0为H1220, 表示通道一为-10V~10V; 通道二和三为0mA~20 mA; 通道四为4mA~20 mA。

BFM #1~#4 通道输出值

用TO指令往BFM #1~#4写数据, 可以控制模拟量输出。初始值均为0。

BFM #5 数据保持模式

当主模块在由RUN至STOP状态时RUN最后的模式被保持(X=0)或者为偏移值(X=1)输出。

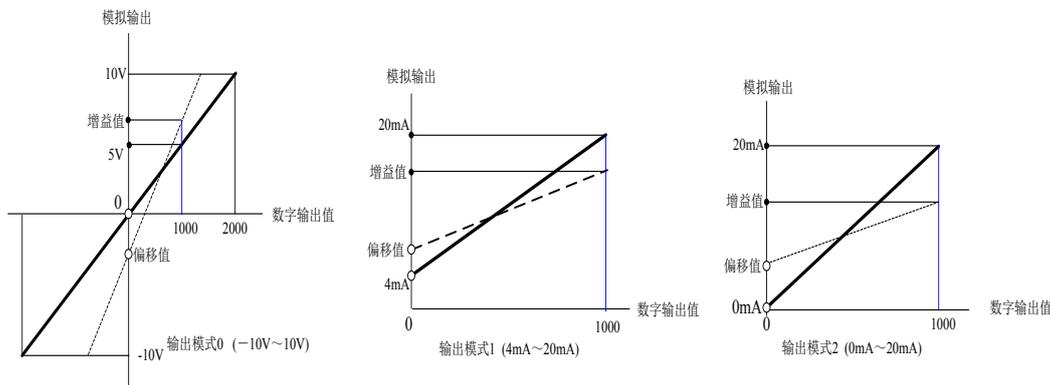
例如: BFM#0为H0000, BFM#5为H0101, 4个通道的偏移值均为0.1V, 由RUN至STOP变化时BFM#1~#4里面的值均为

6
扩展模块
及扩展卡
使用方法

1500 (7.5V)，当STOP后，通道1和通道3输出电压变成0.1V，通道2和通道4输出电压保持在7.5V。

BFM #8~#17、BFM #21偏移和增益的定义与设定方法：

H2U-4DA(R)/ 2DA共有三种工作模式，特性曲线如下图：



其中，增益值为数字1000时对应的模拟量输出；偏移值则是数字为0时对应的模拟量输出。偏移和增益可以独立设定或一起设定，设定参数的单位是mV（模式0）和μA（模式1和2）。

BFM #8~#9为偏移、增益设置命令，十六进制的每HEX（二进制的4个bit位组成）位来控制禁止或者允许，注意在AD输入模块中是二进制的每个bit位来控制，DA模块和AD模块在偏移、增益设置命令有所区别。BFM #10~#17为偏移、增益设置值，BFM #21为曲线特性设置命令。

增益/偏移设定前，需先将BFM #21设为1，再根据需要修改BFM #10~#17的相关单元数值；再向BFM #8、#9单元写入操作允许字，修改完毕，应将BFM #21再设为2，避免再次被改变。

例如：在BFM#0为模式H0000中，需要修改通道一的偏移和增益分别为0.2V和5.5V；通道三的偏移和增益分别为0.5V和6V，则需要按如下步骤操作：

先将BFM#21改为1；过300ms以后将K200、K5500、K500、K6000分别送到BFM#10、#11、#14、#15中去；紧接着BFM#8~#9开允许偏移、增益，即是将BFM#8、BFM#9都修改为H0011；修改完毕。最后将BFM#21改为2，以防再次被修改。

BFM #20 回归出厂值

将BFM #20设定为1可以恢复到默认值。

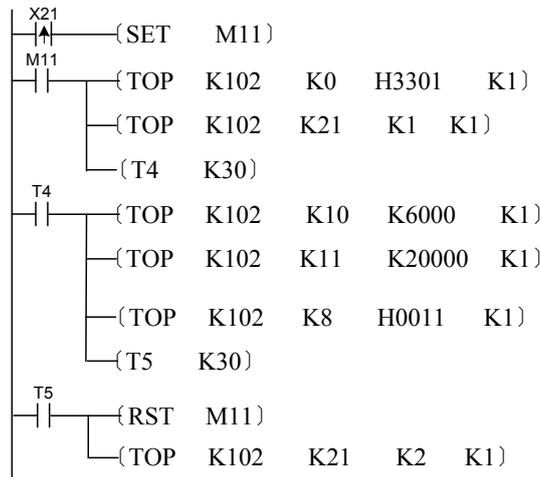
编程举例1：

一台H2U-4AD扩展模块接于PLC主模块后方，紧接着接了一台H2U-4DA扩展模块，按编号原则H2U-4DA为#1号模块，其中需要CH1端口输出6mA~20mA的电流信号，CH2端口输出-10V~10V的电压信号，CH3/CH4未使用。要求X21端口的闭合引发4DA模块的初始化操作。编写的用户程序如下：



例子中如果改成H2U-4DAR远程模块，CAN站号为2。则此例子程序如下：

6
扩展模块
及扩展卡
使用方法



编程举例2:

接上例本地模块的举例，若要求将D100、D101的值分别从CH1、CH2端口输出，每1秒中检查一次4DA模块的工作状态。编程如下:



其中若4DA模块的外部24V电源失电，系统标志M8067会置位，出错代码D8067=K6708。

使用注意事项:

禁止带电插拔，只有在主模块和应用系统停电的情况下，才能进行扩展模块的接入或拆除工作，以保证人身安全，防止因带电插拔损坏器件。

6.5 模拟量混合（输入/输出）模块

6.5.1 H2U-4AM 2 通道模拟电流输入 /2 通道模拟电压电流输出混合扩展模块

1) 简介

H2U本地扩展模块可配合H2U系列主模块工作，实现2个模拟输入通道的信号检测和2个模拟信号通道的输出。每个输入通道将0V~10V或0mA~20mA的信号转换为12bit的数字量，供PLC主模块读取；每个输出通道都具有电压信号及电流信号输出端口，信号幅值分别为-10~10V或0~20mA。主模块通过FROM/TO指令访问本扩展模块内寄存器的BFM单元，实现模拟输出信号的控制。

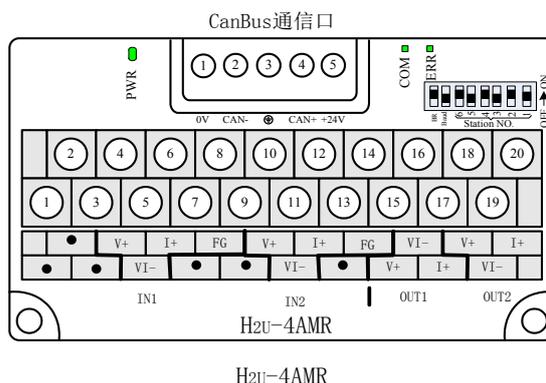
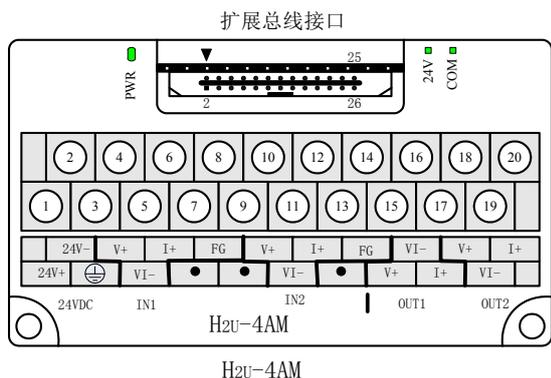
2) 电气规格:

电源	模拟电路	24V DC -15%/+20%，最大允许纹波电压5% 输入电流200mA(来自于主单元的外部电源)	
	数字电路	5V DC 50mA(来自于主电源内部电源)（远程扩展模块无需数字电路电源）	
项目		AD部分指标	
占用I/O点数		不占用主模块I/O点数	
转换速度		15ms/通道（常速），6ms/通道（快速），1ms/通道（最快）	
模拟输入范围	电压输入	0~10V DC,（输入阻抗为200KΩ）	通过设定BFM可进行输入范围选择
	电流输入	0~20mA(输入阻抗为250Ω)	
数字输出		默认设置为：0~+2000	
分辨率	电压输入	5mV	
	电流输入	20μA	
精度		±1%	
隔离		模拟电路和数字电路之间用光电耦合器进行隔离。模拟电路和外部电源用DC/DC进行隔离。模拟通道之间不隔离。	

6
扩展模块
及扩展卡
使用方法

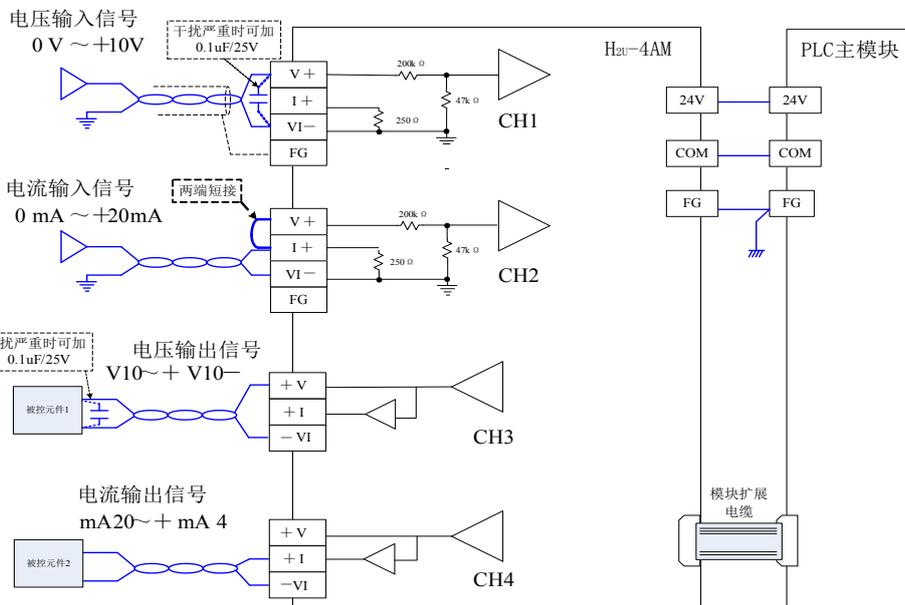
项目		DA部分指标
占用I/O点数		不占用主模块I/O点数
转换速度		2通道2.1ms
模拟输出范围	电压输出	-10~10V DC(外部负载阻抗为2KΩ~1MΩ)
	电流输出	0~20mA(外部负载阻抗为500Ω或更小)
数字输入		默认设置: -2000~2000, 允许范围: -10000~10000
分辨率	电压输出	5mV(10V/2000)
	电流输出	20 μA(20mA/1000)
总体精度		±1% (对于10V的全范围) ±1% (对于20mA的全范围)
隔离		模拟电路和数字电路之间用光电耦合器进行隔离。模拟电路电源和外部电源用DC/DC进行隔离。模拟通道之间不隔离。

3) 接线端子布局如下图:



外部配线:

模拟信号通过双绞线连接到扩展模块的输入/输出端口, 布线时不要与交流电源线或干扰信号的线路靠近; 若模拟信号的干扰严重时, 可采用屏蔽线连接, 并在输入端口并联1只0.1 μF/25V的高频电容; 信号源/被控元件及其屏蔽线的外壳与H2U-4AM的信号接地端FG相连, 共同接地。



4) 访问4AM 模块的BFM区:

PLC主模块通过读写指令FROM/TO访问这些BFM单元。

扩展模块内设有EEPROM存储单元, 用于保存一些BFM设定值, 例如每个模拟输入/输出通道的信号类型、偏移值、增益值等, 这些单元的保存动作是由相应BFM单元的设置状态决定而自行完成的。

BFM区的每个寄存器宽度为16bit (即1Word), 按照4AM 模块的BFM区定义如下表:

6
扩展模块
及扩展卡
使用方法

BFM	R/W属性	输入通道内容								
#0 (E)	WR	通道信号模式选择，每个HEX位代表1个输入通道，4AM模块取低8位中的高HEX位为ch2，低HEX位为ch1：（默认值=H00） 0=0V~10V；对应数字输出：0~2000 1=4mA~20mA；对应数字输出：0~1000 2=0mA~20mA；对应数字输出：0~1000 3=本通道关闭； 4=0V~10V；对应数字输出：0~10000 5=4mA~20mA；对应数字输出：0~10000 6=0mA~20mA；对应数字输出：0~10000								
#1	WR	通道1	平均滤波常数，即用于平均计算的采样值个数，设定范围1~4096，默认值8。若要高速采集，可设定为1。当BFM#15改变时，自动恢复为默认值							
#2	WR	通道2								
#3~4	—	保留								
#5	R	通道1	输入通道采集值平均滤波后的数据							
#6	R	通道2								
#7~8	—	保留								
#9	R	通道1	输入通道当前采集的数据，即未滤波处理的瞬时值。							
#10	R	通道2								
#11~14	—	保留								
#15	WR	ADC速率选择	0=正常速度，15ms/通道（默认值）； 1=快速转换，6ms/通道； 1000~30000=高速采样，对应1ms~30ms/通道							
#16~19	—	保留								
#20 (E)	WR	1=复位设定，将所有参数恢复到默认值（出厂值）。默认值=0								
#21 (E)	WR	1=允许调整偏移/增益（默认值）； 2=禁止调整偏移/增益（禁止修改输出转换的曲线特性，其中包括偏移、增益、通道信号模式）								
#22	WR	低4位对应2个通道的操作	-	-	-	-	G2	O2	G1	O1
偏移/增益调整使能，当非0时，模块会将BFM#23/ BFM#24值写入其内部对应通道控制寄存器中										
#23	WR	偏移值，数字输出为0时的模拟输入值（0、1、2模式）								
#24	WR	增益值，数字输出为+1000时的模拟输入值（0、1、2模式）								
#25~26	—	保留								
#27	R	4AM模块软件版本								
#28	—	保留								
#29	R	输入通道错误状态								
#30	R	扩展模块识别码，H2U-4AM的识别码为K4051								
#31	—	保留，不可访问								

BFM	R/W属性	输出通道内容								
#32 (E)	WR	输出模式选择，每个HEX位代表1个输出通道，4AM模块取低8位中的高HEX位为ch4，低HEX位为ch3：（默认值=H00） 0=-10V~10V；对应数字输出：-2000~2000 1=4mA~20mA；对应数字输出：0~1000 2=0mA~20mA；对应数字输出：0~1000 3=本通道关闭； 4=-10V~10V；对应数字输出：-10000~10000 5=4mA~20mA；对应数字输出：0~10000 6=0mA~20mA；对应数字输出：0~10000								
#33	WR	通道3	通道输出值，初始值为0							
#34	WR	通道4								
#35~36	—	保留								
#37 (E)	WR	PLC停机时数值保留模式，每个HEX位代表1个输出通道，4AM模块取低8位中的高HEX位为ch4，低HEX位为ch3，当：x=0，保持停机前的输出；x=1，将输出复位到偏移设定值								
#38~39	—	保留								
#40	WR	G2	O2	G1	O1					
CH4/CH3的偏移/增益设置命令，按HEX位设置，初始值=H0000 0=禁止改变；1=允许改变EEPROM对应数据										
#41	—	保留								

6
扩展模块
及扩展卡
使用方法

BFM	R/W属性	输出通道内容	
# 42	WR	偏移数据CH3	单位: mV或 μ A 初始偏移值: 0 初始增益值: +5000,对应模式0 (0、1、2模式)
# 43	WR	增益数据CH3	
# 44	WR	偏移数据CH4	
# 45	WR	增益数据CH4	
# 46~51	—	保留	
# 52 (E)	—	保留	
# 53 (E)	WR	1=允许调整偏移/增益(默认值); 2=禁止调整偏移/增益(禁止修改输出转换的曲线特性,其中包括偏移、增益、通道信号模式)	
# 54~60	—	保留	
# 61	R	输出通道错误状态	

注释: 模块掉电后输出转换的曲线特性保存, 其中包括曲线的偏移增益特性。当前BFM区显示的偏移增益为将要生效的值, 并不是当前已经生效的偏移增益。

其中状态信息字BFM #29的意义说明如下:

BFM #29位号	ON状态	OFF状态
b0:	存在错误。b0~b3中任一非0, A/D转换停止	无错误
b1:	模块内EEPROM的偏移/增益设置有误	偏移/增益数据正确
b2:	保留	保留
b3:	模块硬件故障	硬件正常
b10:	数字输出超出-2048~2047的范围	数字输出值正常
b11:	采样滤波常数超出1~4096范围	采样滤波常数正常
b12:	禁止BFM#21的值设为K2	允许BFM#21=K2
BFM #29的其它bit4~7, bit13~15等没有定义。		

状态信息字BFM #61的意义说明如下:

BFM #61位号	ON状态	OFF状态
b0:	存在错误。b1~b3中任一非0, D/A转换停止	无错误
b1:	模块内EEPROM的偏移/增益设置有误	偏移/增益数据正确
b2:	保留	保留
b3:	模块硬件故障	硬件正常
b10:	写入数字值超出指定的范围(-2350~2350)	数字输出值正常
b12:	禁止BFM#21的值没有设为K1	BFM#21=K1
BFM #61的其它bit4~7, bit11, bit13~15等没有定义。		

其中表中的“(E)”字样的BFM单元为存入EEPROM的项目, 具有掉电保持特性。

寄存器改写说明:

BFM#0、#32等带(E)单元的改写引发模块内部对EEPROM的写操作, 而写操作需要一定时间, 每个Word约需300ms时间, 因此在需要改写多个BFM单元时, 注意PLC编程时, 用户程序中每写一个上述的BFM单元后要延时一段时间, 不要连续进行写操作, 确保写指令的正确完成。

注意: 本地扩展模块的外部24V电源失电, PLC主模块的系统标志M8067会置位, 出错代码D8067=K6708。编程时可定时检查该标志, 能及时发现该现象。

5) 输入通道部分BFM区解释

BFM0# 输入通道选择

输入通道的初始化, 默认2个通道都为0V~10V, 由BFM 0#的十六进制HXX控制, 最低位控制通道一, 第二位控制通道二, 每个字符的控制方式如下:

X=0 预设范围0V~10V (对应数字0~2000)

X=1 预设范围4mA~20 mA (对应数字0~1000)

X=2 预设范围0mA~20 mA (对应数字0~1000)

X=3 本通道关闭

X=4 预设范围0V~10V (对应数字0~10000)

X=5 预设范围4mA~20 mA (对应数字0~10000)

X=6 预设范围0mA ~20 mA （对应数字0~10000）

例如：BFM#0为H30，表示通道一为0V~10V；通道二关闭。

没有用到的通道，可以关闭，也可以不关闭，关闭的通道不占用转换时间（BFM#15），例子中通道二关闭了，则整个通道转换一次时间为通道一乘以BFM#15。

BFM #1~#2 输入平均采样数

每个通道(BFM#9~#10)对应的采样值累加采样数（BFM #1~#2）的个数后再除以采样数（BFM #1~#2），存放到（BFM #5~#6）

BFM #5~#6 输入存放平均采样值

BFM #9~#10 输入存放即时采样值

BFM #15 输入ADC速率时间

每个输入通道转换一次需要的时间，需注意更新一次数据需要的时间是BFM#15的时间乘以没有关闭的通道数。

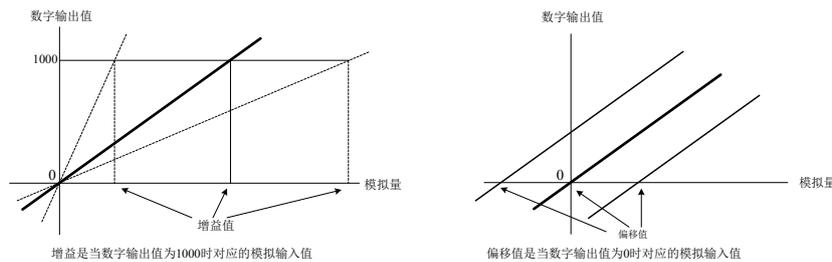
例如：BFM #0为H10，BFM #1为K7，BFM #2为K6，BFM #15为K10；则BFM #9和BFM #10刷新一次数据时间为 $BFM\#0 \times BFM\#15 = 2 \times 10 = 20MS$ ，BFM #5刷新一次数据时间为 $BFM\#0 \times BFM\#15 \times BFM\#1 = 2 \times 10 \times 7 = 140MS$ ，BFM #6刷新一次数据时间为 $BFM\#0 \times BFM\#15 \times BFM\#2 = 2 \times 10 \times 6 = 120MS$ 。程序中FROM/TO指令比较耗时，故此模块参数在程序中采集BFM#5的数据可以用LDP M8012 FROM K0 K5 D10 K1指令读取，和LD M8000 FROM K0 K5 D10 K1的效果是一样的，但是后面用M8000驱动指令每个扫描周期都读取一次，大大加长了程序的扫描周期。

BFM #20 输入通道回归出厂值

将#20设定为1可以恢复到默认值。

BFM #21~#24 输入通道偏移和增益的定义与设定方法：

H_{2U}-4AM的输入通道共有两种工作模式，特性曲线如下图：



偏移和增益可以独立设定或一起设定，正常的增益值设定范围是1V~15V或4mA~32mA。正常的偏移值设定范围为0V~5V，或0mA~20mA。

增益/偏移设定前，需先将BFM#21设为1，再修改BFM#23/ BFM#24；然后开各个通道的允许偏移增益BFM#22，修改完毕，应将BFM#21再设为2，避免再次被改变。

例如：在BFM#0为模式0中，需要修改通道一、通道二的偏移和增益分别为0.5V和6V，则需要按如下步骤操作：

先将BFM#21改为1；过300ms以后将K500和K6000分别送到BFM#23和BFM#24去；再过300MS后开允许增益BFM#22，本例子中BFM#22应为二进制的1111，即是将BFM#22修改为HF；修改完毕。最后将BFM#21改为2，以防再次被修改。

6) 输出通道部分BFM区解释

BFM#32 输出通道选择

输出通道的初始化，默认2个通道都为-10V~10V，由BFM 32#的十六进制HXX控制，最低位控制通道三，第二位控制通道四，每个字符的控制方式如下：

X=0 预设范围-10V~10V （对应数字-2000~2000）

X=1 预设范围4mA~20 mA （对应数字0~1000）

X=2 预设范围0mA~20 mA （对应数字0~1000）

X=3 本通道关闭

X=4 预设范围-10V~10V （对应数字-10000~10000）

X=5 预设范围4mA~20 mA （对应数字0~10000）

X=6 预设范围0mA~20 mA （对应数字0~10000）

例如：BFM#32为H10，表示通道三为-10V~10V；通道四为4mA~20mA。

6
扩展模块
及扩展卡
使用方法

BFM #33~#34 输出通道输出值

用TO指令往BFM #33~#34写数据，可以控制模拟量输出。初始值均为0。

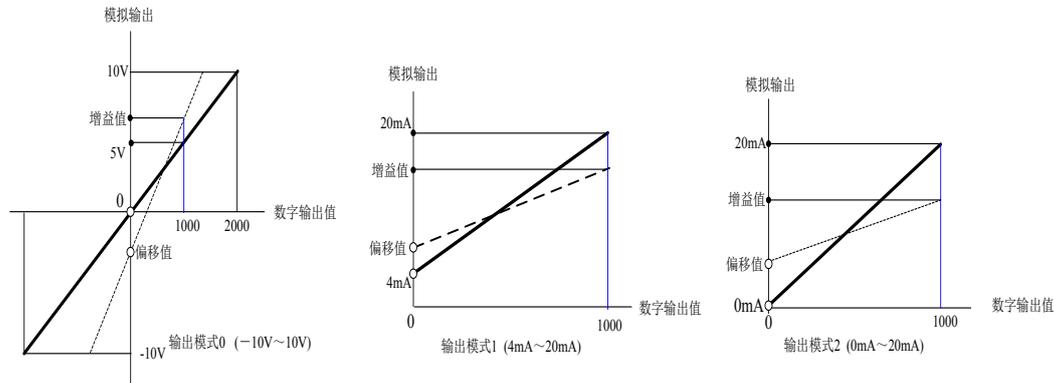
BFM #37 输出通道数据保持模式

当主模块在由RUN至STOP状态时RUN最后的模式被保持(X=0)或者为偏移值(X=1)输出。

例如：BFM#32为H00，BFM#37为H01，2个通道的偏移值均为0.1V，由RUN至STOP变化时BFM#33~#34里面的值均为1500（7.5V），当STOP后，通道3输出电压变成0.1V，通道4输出电压保持在7.5V。

BFM #40、BFM #42~#45、BFM #53 输出通道偏移和增益的定义与设定方法：

H_{2U}-4AM的输出通道共有三种工作模式，特性曲线如下图：



其中，增益值为数字1000时对应的模拟量输出；偏移值则是数字为0时对应的模拟量输出。偏移和增益可以独立设定或一起设定，设定参数的单位是mV（模式0）和 μ A（模式1和2）。

BFM #40为偏移、增益设置命令，十六进制的每HEX（二进制的4个bit位组成）位来控制禁止或者允许，注意在AD输入模块中是二进制的每个bit位来控制，DA模块和AD模块在偏移、增益设置命令有所区别。BFM #10~#13为偏移、增益设置值，BFM #53为曲线特性设置命令。

增益/偏移设定前，需先将BFM #53设为1，再根据需要修改BFM #42~#45的相关单元数值；再向BFM #40单元写入操作允许字，修改完毕，应将BFM #53再设为2，避免再次被改变。

例如：在BFM#40为模式H00中，需要修改通道三的偏移和增益分别为0.2V和5.5V，则需要按如下步骤操作：

先将BFM#53改为1；过300MS以后将K200、K5500分别送到BFM#42、#43中去；紧接着BFM#40开允许偏移、增益，即将BFM#40修改为H0011；修改完毕。最后将BFM#53改为2，以防再次被修改。

BFM #52 输出通道回归出厂值

将BFM #52设定为1可以恢复到默认值。

编程举例1：

一只H_{2U}-4AMR远程扩展模块，CAN站号为12，其中CH1端口需要采集0V~10V的电压信号，要求改滤波次数为6，将采集得到的数据存于D10。CH3端口由D12输出6mA~20mA的电流信号。编写的用户程序如下

6



使用注意事项:

禁止带电插拔，只有在主模块和应用系统停电的情况下，才能进行扩展模块的接入或拆除工作，以保证人身安全，防止因带电插拔损坏器件。

6.5.2 H2U-6AM 4 通道模拟电流输入 /2 通道模拟电压电流输出混合扩展模块

1) 简介

H2U本地扩展模块可配合H2U系列主模块工作，实现4个模拟输入通道的信号检测和2个模拟信号通道的输出。每个输入通道将0mA~20mA的信号转换为12bit的数字量，供PLC主模块读取；每个输出通道都具有电压信号及电流信号输出端口，信号幅值分别为-10V~10V或0~20mA。主模块通过FROM/TO指令访问本扩展模块内寄存器的BFM单元，实现模拟输出信号的控制。

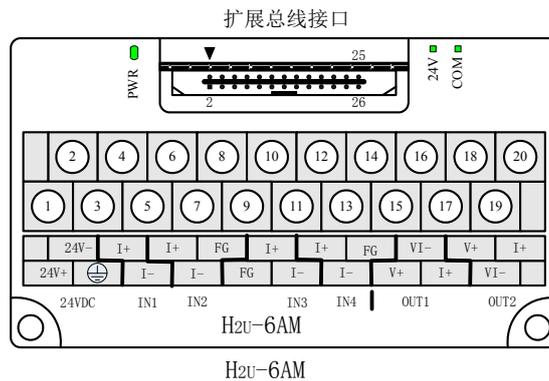
2) 电气规格:

电源	模拟电路	24V DC -15%/+20%，最大允许纹波电压5% 输入电流200mA(来自于主单元的外部电源)
	数字电路	5V DC 50mA(来自于主电源内部电源)（远程扩展无需数字电路电源）
项目		AD部分指标
占用I/O点数		不占用主模块I/O点数
转换速度		15ms/通道（常速），6ms/通道（快速），1ms/通道（最快）
模拟输入范围	电流输入	0~20mA(输入阻抗为250Ω)
	数字输出	默认设置为：0 ~ +1000
分辨率	电流输入	20 μA
精度		±1%
隔离		模拟电路和数字电路之间用光电耦合器进行隔离。模拟电路和外部电源用DC/DC进行隔离。模拟通道之间不隔离。

6
扩展模块
及扩展卡
使用方法

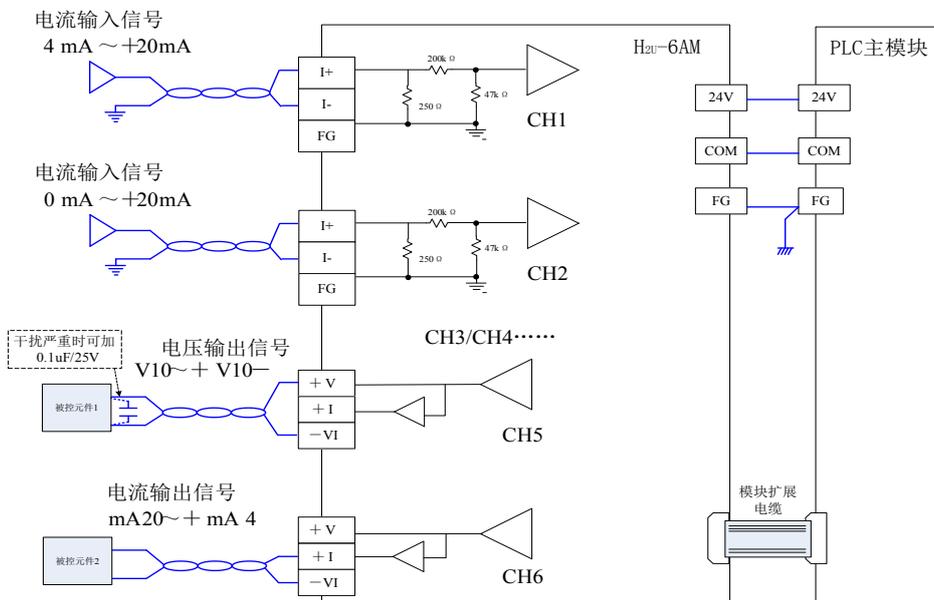
项目		DA部分指标
占用I/O点数		不占用主模块I/O点数
转换速度		2通道2.1ms
模拟输出范围	电压输出	-10~10V DC(外部负载阻抗为2KΩ~1MΩ)
	电流输出	0~20mA(外部负载阻抗为500Ω或更小)
数字输入		默认设置: -2000~2000, 允许范围: -10000~10000
分辨率	电压输出	5mV(10V/2000)
	电流输出	20 μA(20mA/1000)
总体精度		±1% (对于10V的全范围) ±1% (对于20mA的全范围)
隔离		模拟电路和数字电路之间用光电耦合器进行隔离。模拟电路电源和外部电源用DC/DC进行隔离。模拟通道之间不隔离。

3) 接线端子布局如下图:



4) 外部配线:

模拟信号通过双绞线连接到扩展模块的输入\输出端口，布线时不要与交流电源线或干扰信号的线路靠近；若模拟信号的干扰严重时，可采用屏蔽线连接，并在输入端口并联1只0.1 μF/25V的高频电容；信号源/被控元件及其屏蔽线的外壳与H2U-6AM的信号接地端FG相连，共同接地。



5) 访问6AM 模块的BFM区:

PLC主模块通过读写指令FROM/TO访问这些BFM单元。

扩展模块内设有EEPROM存储单元，用于保存一些BFM设定值,例如每个模拟输入/输出通道的信号类型、偏移值、增益值等，这些单元的保存动作是由相应BFM单元的设置状态决定而自行完成的。

BFM区的每个寄存器宽度为16bit (即1Word)，按照6AM 模块的BFM区定义如下表:

6
扩展模块
及扩展卡
使用方法

BFM	R/W属性	输入通道内容									
#0 (E)	WR	通道信号模式选择，每个HEX位代表1个输入通道，最高位为ch4，最低位为ch1：（默认值=H1111） 1=4mA~20mA；对应数字输出：0~1000 2=0mA~20mA；对应数字输出：0~1000 3=本通道关闭； 5=4mA~20mA；对应数字输出：0~10000 6=0mA~20mA；对应数字输出：0~10000	<pre> ----- 例图: □ □ □ □ ch4 3 2 1 ----- </pre>								
#1	WR	通道1	平均滤波常数，即用于平均计算的采样值个数，设定范围1~4096，默认值8。若要高速采集，可设定为1。当BFM#15改变时，自动恢复为默认值								
#2	WR	通道2									
#3	WR	通道3									
#4	WR	通道4									
#5	R	通道1	输入通道采集值平均滤波后的数据								
#6	R	通道2									
#7	R	通道3									
#8	R	通道4									
#9	R	通道1	输入通道当前采集的数据，即未滤波处理的瞬时值。								
#10	R	通道2									
#11	R	通道3									
#12	R	通道4									
#13~14	—	保留；									
#15	WR	ADC速率选择	0=正常速度，15ms/通道（默认值）； 1=快速转换，6ms/通道； 1000~30000=高速采样，对应1ms~30ms/通道								
#16~19	—	保留									
#20 (E)	WR	1=复位设定，将所有参数恢复到默认值（出厂值）。默认值=0									
#21 (E)	WR	1=允许调整偏移/增益（默认值）； 2=禁止调整偏移/增益（禁止修改输出转换的曲线特性，其中包括偏移、增益、通道信号模式）									
#22	WR	低8位对应4个通道的操作	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td>G4</td> <td>O4</td> <td>G3</td> <td>O3</td> <td>G2</td> <td>O2</td> <td>G1</td> <td>O1</td> </tr> </table>	G4	O4	G3	O3	G2	O2	G1	O1
G4	O4	G3	O3	G2	O2	G1	O1				
		偏移/增益调整使能，当非0时，模块会将BFM23/24值写入其内部对应通道控制寄存器中									
#23	WR	偏移值，数字输出为0时的模拟输入值（0、1、2模式）初始值4000									
#24	WR	增益值，数字输出为+1000时的模拟输入值（0、1、2模式）初始值20000									
#25~26	—	保留									
#27	R	6AM模块软件版本									
#28	—	保留									
#29	R	输入通道错误状态									
#30	R	扩展模块识别码，H2u-6AM 的识别码为K4050									
#31	—	保留，不可访问									

6

扩展模块
及扩展卡
使用方法

BFM	R/W属性	输出通道内容	
#32 (E)	WR	输出模式选择，每个HEX位代表1个输出通道，6AM 模块取低8位中的高HEX位为ch6，低HEX位为ch5：（默认值=H00） 0=-10V~10V；对应数字输出：-2000~2000 1=4mA~20mA；对应数字输出：0~1000 2=0mA~20mA；对应数字输出：0~1000 4=-10V~10V；对应数字输出：-10000~10000 5=4mA~20mA；对应数字输出：0~10000 6=0mA~20mA；对应数字输出：0~10000	
#33	WR	通道5	通道输出值，初始值为0
#34	WR	通道6	
#35~36	—	保留	
#37 (E)	WR	PLC停机时数值保留模式，每个HEX位代表1个输出通道，6AM 模块取低8位中的高HEX位为ch6，低HEX位为ch5，当：x=0，保持停机前的输出；x=1，将输出复位到偏移设定值	
#38~39	—	保留	

BFM	R/W属性	输出通道内容				
# 40	WR	低4bit位对应2个通道的操作	G2	O2	G1	O1
		CH6/CH5的偏移/增益设置命令，按HEX位设置，初始值=H0000 0=禁止改变；1=允许改变EEPROM对应数据				
# 41	—	保留				
# 42	WR	偏移数据CH5	单位：mV或 μ A 初始偏移值：0 初始增益值：+5000,对应模式0（0、1、2模式）			
# 43	WR	增益数据CH5				
# 44	WR	偏移数据CH6				
# 45	WR	增益数据CH6				
# 46~51	—	保留				
# 52（E）	WR	保留				
# 53（E）	WR	1=允许调整偏移/增益（默认值）； 2=禁止调整偏移/增益（禁止修改输出转换的曲线特性，其中包括偏移、增益、通道信号模式）				
# 54~60	—	保留				
# 61	R	输出通道错误状态				

注释：模块掉电后输出转换的曲线特性保存，其中包括曲线的偏移增益特性。当前BFM区显示的偏移增益为将要生效的值，并不是当前已经生效的偏移增益。

其中状态信息字BFM #29的意义说明如下：

BFM #29位号	ON状态	OFF状态
b0:	存在错误。b0~b3中任一非0，A/D转换停止	无错误
b1:	模块内EEPROM的偏移/增益设置有误	偏移/增益数据正确
b2:	保留	保留
b3:	模块硬件故障	硬件正常
b10:	数字输出超出-2048~2047的范围	数字输出值正常
b11:	采样滤波常数超出1~4096范围	采样滤波常数正常
b12:	禁止BFM#21的值为K2	允许BFM#21=K2
BFM # 29的其它bit4~7，bit13~15等没有定义。		

状态信息字BFM #61的意义说明如下：

BFM #61位号	ON状态	OFF状态
b0:	存在错误。b1~b3中任一非0，D/A转换停止	无错误
b1:	模块内EEPROM的偏移/增益设置有误	偏移/增益数据正确
b2:	保留	保留
b3:	模块硬件故障	硬件正常
b10:	写入数字值超出指定的范围（-2350~2350）	数字输出值正常
b12:	禁止BFM#21的值没有设为K1	BFM#21=K1
BFM # 61的其它bit4~7，bit11，bit13~15等没有定义。		

其中表中的“（E）”字样的BFM单元为存入EEPROM的项目，具有掉电保持特性。

寄存器改写说明：

BFM#0、#32等带（E）单元的改写引发模块内部对EEPROM的写操作，而写操作需要一定时间，每个Word约需300ms时间，因此在需要改写多个BFM单元时，注意PLC编程时，用户程序中每写一个上述的BFM单元后要延时一段时间，不要连续进行写操作，确保写指令的正确完成。

注意：本地扩展模块的外部24V电源失电，PLC主模块的系统标志M8067会置位，出错代码D8067=K6708。编程时可定时检查该标志，能及时发现该现象。

6) 输入通道部分BFM区解释

BFM#0 输入通道选择

输入通道的初始化，默认4个通道都为4mA~20mA，由BFM #0的十六进制HXXXX控制，最低位控制通道一，依次顺序，最高位控制通道四，每个字符的控制方式如下：

X=1 预设范围4mA~20 mA（对应数字0~1000）

X=2 预设范围0mA~20 mA（对应数字0~1000）

X=3 本通道关闭

X=5 预设范围4mA ~20 mA （对应数字0~10000）

X=6 预设范围0mA ~20 mA （对应数字0~10000）

例如：BFM#0为H2235，表示通道一为4mA~20 mA；通道二关闭；通道三和通道四为0mA~20 mA。

没有用到的通道，可以关闭，也可以不关闭，关闭的通道不占用转换时间（BFM#15），例子中通道二关闭了，则整个通道转换一次时间为没有关闭的三个通道的转换时间（3×BFM#15）。

BFM #1~#4 输入通道平均采样数

每个通道(BFM#9~#12)对应的采样值累加采样数（BFM #1~#4）的个数后再除以采样数（BFM #1~#4），存放到（BFM #5~#8）

BFM #5~#8 输入通道存放平均采样值

BFM #9~#12 输入通道存放即时采样值

BFM #15 输入通道ADC速率时间

每个通道转换一次需要的时间，需注意更新一次数据需要的时间是BFM#15的时间乘以没有关闭的通道数。

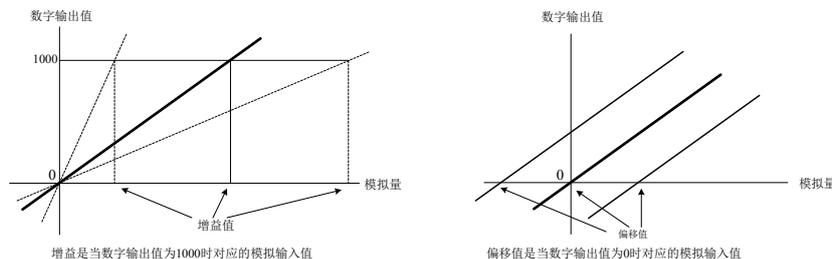
例如：BFM #0为H3311，BFM #1为K7，BFM #2为K6，BFM #15为K10；则BFM #9和BFM #10刷新一次数据时间为BFM#0×BFM#15=2×10=20MS，BFM #5刷新一次数据时间为BFM#0×BFM#15×BFM#1=2×10×7=140MS，BFM #6刷新一次数据时间为BFM#0×BFM#15×BFM#2=2×10×6=120MS。程序中FROM/TO指令比较耗时，故此模块参数在程序中采集BFM#5的数据可以用LDP M8012 FROM K0 K5 D10 K1指令读取，和LD M8000 FROM K0 K5 D10 K1的效果是一样的，但是后面用M8000驱动指令每个扫描周期都读取一次，大大加长了程序的扫描周期。

BFM #20 输入通道回归出厂值

将#20设定为1可以恢复到默认值。

BFM #21~#24 输入通道偏移和增益的定义与设定方法：

H2U-6AM 的输入通道共有两种工作模式，特性曲线如下图：



输入通道偏移和增益可以独立设定或一起设定，正常的增益值设定范围是4mA~32mA。正常的偏移值设定范围为-20mA~20mA。

增益/偏移设定前，需先将BFM#21设为1，再修改BFM#23/24；然后开各个通道的允许偏移增益BFM#22，修改完毕，应将BFM#21再设为2，避免再次被改变。

例如：在BFM#0为模式1111中，需要修改通道一、通道二的偏移和增益分别为5mA和18mA，则需要按如下步骤操作：

先将BFM#21改为1；过300ms以后将K5000和K18000分别送到BFM#23和BFM#24去；再过300MS后开允许增益BFM#22，本例子中BFM#22应为二进制的00001111，即是将BFM#22修改为H000F；修改完毕。最后将BFM#21改为2，以防再次被修改。

7) 输出通道部分BFM区解释

BFM#32 输出通道选择

输出通道的初始化，默认2个通道都为-10V~10V，由BFM 32#的十六进制HXX控制，最低位控制通道五，第二位控制通道六，每个字符的控制方式如下：

X=0 预设范围-10V~10V （对应数字-2000~2000）

X=1 预设范围4mA~20 mA （对应数字0~1000）

X=2 预设范围0mA~20 mA （对应数字0~1000）

X=4 预设范围-10V~10V （对应数字-10000~10000）

X=5 预设范围4mA~20 mA （对应数字0~10000）

X=6 预设范围0mA~20 mA （对应数字0~10000）

例如：BFM#32为H10，表示通道五为-10V~10V；通道六为4mA~20mA。

BFM #33~#34 输出通道输出值

用TO指令往BFM #33~#34写数据，可以控制模拟量输出。初始值均为0。

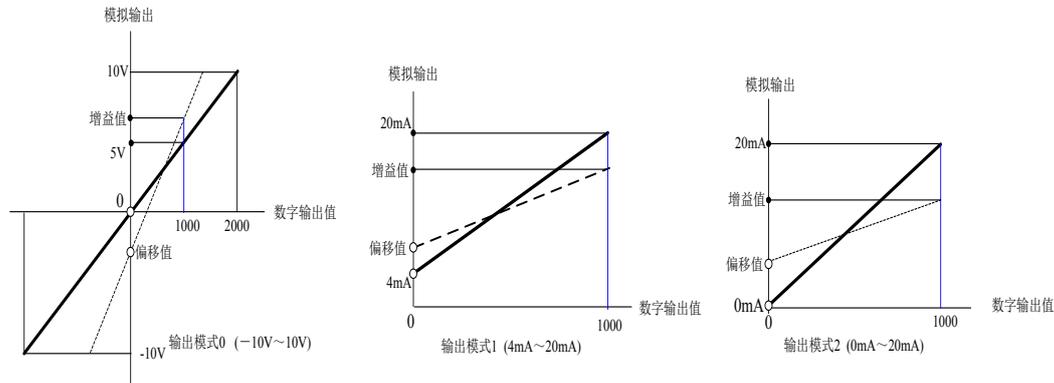
BFM #37 输出通道数据保持模式

当主模块在由RUN至STOP状态时RUN最后的模式被保持(X=0)或者为偏移值(X=1)输出，

例如：BFM#32为H00，BFM#37为H01，2个通道的偏移值均为0.1V，由RUN至STOP变化时BFM#33~#34里面的值均为1500（7.5V），当STOP后，通道5输出电压变成0.1V，通道6输出电压保持在7.5V。

BFM #40、BFM #42~#45、BFM #53 输出通道偏移和增益的定义与设定方法：

H_{2U}-6AM 的输出通道共有三种工作模式，特性曲线如下图：



其中，增益值为数字1000时对应的模拟量输出；偏移值则是数字为0时对应的模拟量输出。偏移和增益可以独立设定或一起设定，设定参数的单位是mV（模式0）和 μ A（模式1和2）。

BFM #40为偏移、增益设置命令，十六进制的每HEX（二进制的4个bit位组成）位来控制禁止或者允许，注意在AD输入模块中是二进制的每个bit位来控制，DA模块和AD模块在偏移、增益设置命令有所区别。BFM #10~#13为偏移、增益设置值，BFM #53为曲线特性设置命令

增益/偏移设定前，需先将BFM #53设为1，再根据需要修改BFM #42~#45的相关单元数值；再向BFM #40单元写入操作允许字，修改完毕，应将BFM #53再设为2，避免再次被改变。

例如：在BFM#40为模式H00中，需要修改通道五的偏移和增益分别为0.2V和5.5V，则需要按如下步骤操作：

先将BFM#53改为1；过300MS以后将K200、K5500分别送到BFM#42、#43中去；紧接着BFM#40开允许偏移、增益，即将BFM#40修改为H0011；修改完毕。最后将BFM#53改为2，以防再次被修改。

BFM #52 输出通道回归出厂值

将BFM #52设定为1可以恢复到默认值。

编程举例1：

一只H_{2U}-6AM扩展模块接于PLC主模块后方，按编号原则为#0号模块，其中CH1端口需要采集4mA~20mA的电压信号，要求改滤波次数为6，将采集得到的数据存于D10。CH5端口由D12输出6mA~20mA的电流信号。编写的用户程序如下

6



使用注意事项:

禁止带电插拔，只有在主模块和应用系统停电的情况下，才能进行扩展模块的接入或拆除工作，以保证人身安全，防止因带电插拔损坏器件。

6.5.3 H2U-6CM 4 通道模拟电压输入 /2 通道模拟电压电流输出混合扩展模块

1) 简介

H2U本地扩展模块可配合H2U系列主模块工作，实现4个模拟输入通道的信号检测和2个模拟信号通道的输出。每个输入通道将0V~10V的信号转换为12bit的数字量，供PLC主模块读取；每个输出通道都具有电压信号及电流信号输出端口，信号幅值分别可为-10~10V或0~20mA。主模块通过FROM/TO指令访问本扩展模块内寄存器的BFM单元，实现模拟输出信号的控制。

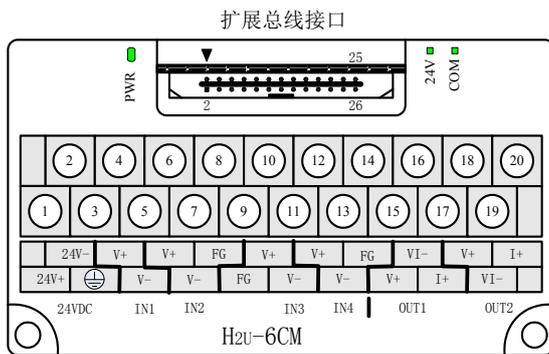
2) 电气规格:

电源	模拟电路	24V DC -15%/+20%，最大允许纹波电压5% 输入电流140mA(来自于主单元的外部电源)
	数字电路	5V DC 50mA(来自于主电源内部电源) (远程扩展无需数字电路电源)
项目		AD部分指标
占用I/O点数		无
转换速度		15ms/通道 (常速)， 6ms/通道 (快速)， 1ms/通道 (最快)
模拟输入范围	电压输入	0~10V DC (输入阻抗为200KΩ)
数字输出		默认设置为: 0 ~ +2000
分辨率	电压输入	5mV
精度		±1%
隔离		模拟电路和数字电路之间用光电耦合器进行隔离。模拟电路和外部电源用DC/DC进行隔离。模拟通道之间不隔离。
项目		DA部分指标
占用I/O点数		不占用主模块I/O点数
转换速度		2通道2.1ms

6
扩展模块及扩展卡使用方法

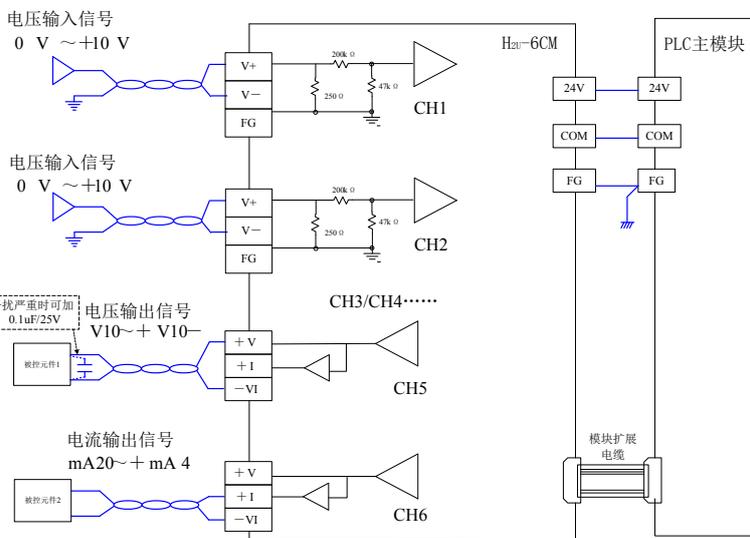
模拟输出范围	电压输出	-10~10V DC(外部负载阻抗为2KΩ~1MΩ)
	电流输出	0~20mA(外部负载阻抗为500Ω或更小)
数字输入		默认设置: -2000~2000, 允许范围: -10000~10000
分辨率	电压输出	5mV(10V/2000)
	电流输出	20 μA(20mA/1000)
总体精度		±1% (对于10V的全范围) ±1% (对于20mA的全范围)
隔离		模拟电路和数字电路之间用光电耦合器进行隔离。模拟电路电源和外部电源用DC/DC进行隔离。模拟通道之间不隔离。

3) 接线端子布局如下图:



4) 外部配线:

模拟信号通过双绞线连接到扩展模块的输入/输出端口, 布线时不要与交流电源线或干扰信号的线路靠近; 若模拟信号的干扰严重时, 可采用屏蔽线连接, 并在输入端口并联1只0.1μF/25V的高频电容; 信号源/被控元件及其屏蔽线的外壳与H2u-6CM的信号接地端FG相连, 共同接地。



5) 访问6CM 模块的BFM区:

PLC主模块通过读写指令FROM/TO访问这些BFM单元。

扩展模块内设有EEPROM存储单元, 用于保存一些BFM设定值, 例如每个模拟输入/输出通道的信号类型、偏移值、增益值等, 这些单元的保存动作是由相应BFM单元的设置状态决定而自行完成的。

BFM区的每个寄存器宽度为16bit (即1Word), 按照6CM 模块的BFM区定义如下表:

BFM	R/W属性	输入通道内容
#0 (E)	WR	通道信号模式选择, 每个HEX位代表1个输入通道, 最高位为ch4, 最低位为ch1: (默认值=H0000) 0=0V~10V; 对应数字输出: 0~2000 3=本通道关闭; 4=0V~10V; 对应数字输出: 0~10000

BFM	R/W属性	输入通道内容										
# 1	WR	通道1	平均滤波常数，即用于平均计算的采样值个数，设定范围1~4096，默认值8。若要高速采集，可设定为1。当BFM#15改变时，自动恢复为默认值									
# 2	WR	通道2										
# 3	WR	通道3										
# 4	WR	通道4										
# 5	R	通道1	输入通道采集值平均滤波后的数据									
# 6	R	通道2										
# 7	R	通道3										
# 8	R	通道4										
# 9	R	通道1	输入通道当前采集的数据，即未滤波处理的瞬时值。									
# 10	R	通道2										
# 11	R	通道3										
# 12	R	通道4										
# 13~14	—		保留；									
# 15	WR	ADC速率选择	0=正常速度，15ms/通道（默认值）； 1=快速转换，6ms/通道； 1000~30000=高速采样，对应1ms~30ms/通道									
# 16~19	—		保留									
# 20 (E)	WR		1=复位设定，将所有参数恢复到默认值（出厂值）。默认值=0									
# 21 (E)	WR		1=允许调整偏移/增益（默认值）； 2=禁止调整偏移/增益（禁止修改输出转换的曲线特性，其中包括偏移、增益、通道信号模式）									
# 22	WR	低8位对应4个通道的操作	G4	O4	G3	O3	G2	O2	G1	O1	偏移/增益调整使能，当非0时，模块会将BFM# 23/ BFM# 24值写入其内部对应通道控制寄存器中	
# 23	WR	偏移值，数字输出为0时的模拟输入值（0、1、2模式）初始值4000										
# 24	WR	增益值，数字输出为+1000时的模拟输入值（0、1、2模式）初始值20000										
# 25~26	—		保留									
# 27	R		6CM模块软件版本									
# 28	—		保留									
# 29	R		输入通道错误状态									
# 30	R		扩展模块识别码，H2U-6CM 的识别码为K4052									
# 31	—		保留，不可访问									

BFM	R/W属性	输出通道内容										
# 32 (E)	WR	输出模式选择，每个HEX位代表1个输出通道，6CM 模块取低8位中的高HEX位为ch6，低HEX位为ch5：（默认值=H00） 0=-10V~10V；对应数字输出：-2000~2000 1=4mA~20mA；对应数字输出：0~1000 2=0mA~20mA；对应数字输出：0~1000 4=-10V~10V；对应数字输出：-10000~10000 5=4mA~20mA；对应数字输出：0~10000 6=0mA~20mA；对应数字输出：0~10000										
# 33	WR	通道5	通道输出值，初始值为0									
# 34	WR	通道6										
# 35~36	—		保留									
# 37 (E)	WR	PLC停机时数值保留模式，每个HEX位代表1个输出通道，6CM 模块取低8位中的高HEX位为ch6，低HEX位为ch5，当：x=0，保持停机前的输出；x=1，将输出复位到偏移设定值										
# 38~39	—		保留									
# 40	WR	低4bit位对应2个通道的操作	G2	O2	G1	O1	CH6/CH5的偏移/增益设置命令，按HEX位设置，初始值=H0 0=禁止改变；1=允许改变EEPROM对应数据					
# 41	—		保留									



6
扩展模块
及扩展卡
使用方法

BFM	R/W属性	输出通道内容	
# 42	WR	偏移数据CH5	单位: mV或 μ A 初始偏移值: 0 初始增益值: +5000,对应模式0 (0、1、2模式)
# 43	WR	增益数据CH5	
# 44	WR	偏移数据CH6	
# 45	WR	增益数据CH6	
# 46~51	—	保留	
# 52 (E)	WR	初始值=0, 当写入1时, 所有输出通道BFM单元 (BFM#32~BFM#64) 将初始化为默认值。	
# 53 (E)	WR	1=允许调整偏移/增益 (默认值); 2=禁止调整偏移/增益 (禁止修改输出转换的曲线特性, 其中包括偏移、增益、通道信号模式)	
# 54~60	—	保留	
# 61	R	输出通道错误状态	

注释: 模块掉电后输出转换的曲线特性保存, 其中包括曲线的偏移增益特性。当前BFM区显示的偏移增益为将要生效的值, 并不是当前已经生效的偏移增益。

其中状态信息字BFM #29的意义说明如下:

BFM # 29位号	ON状态	OFF状态
b0:	存在错误。b0~b3中任一非0, A/D转换停止	无错误
b1:	模块内EEPROM的偏移/增益设置有误	偏移/增益数据正确
b2:	保留	保留
b3:	模块硬件故障	硬件正常
b10:	数字输出超出-2048~2047的范围	数字输出值正常
b11:	采样滤波常数超出1~4096范围	采样滤波常数正常
b12:	禁止BFM#21的值设为K2	允许BFM#21=K2
BFM # 29的其它bit4~7, bit13~15等没有定义。		

状态信息字BFM #61的意义说明如下:

BFM # 61位号	ON状态	OFF状态
b0:	存在错误。b1~b3中任一非0, D/A转换停止	无错误
b1:	模块内EEPROM的偏移/增益设置有误	偏移/增益数据正确
b2:	保留	保留
b3:	模块硬件故障	硬件正常
b10:	写入数字值超出指定的范围 (-2350~2350)	数字输出值正常
b12:	禁止BFM#21的值没有设为K1	BFM#21=K1
BFM # 61的其它bit4~7, bit11, bit13~15等没有定义。		

其中表中的“(E)”字样的BFM单元为存入EEPROM的项目, 具有掉电保持特性。

寄存器改写说明:

BFM#0、#32等带(E)单元的改写引发模块内部对EEPROM的写操作, 而写操作需要一定时间, 每个Word约需300ms时间, 因此在需要改写多个BFM单元时, 注意PLC编程时, 用户程序中每写一个上述的BFM单元后要延时一段时间, 不要连续进行写操作, 确保写指令的正确完成。

注意: 本地扩展模块的外部24V电源失电, PLC主模块的系统标志M8067会置位, 出错代码D8067=K6708。编程时可定时检查该标志, 能及时发现该现象。

6) 输入通道部分BFM区解释

BFM#0 输入通道选择

输入通道的初始化, 默认4个通道都为0V~10V, 由BFM #0的十六进制HXXXX控制, 最低位控制通道一, 依次顺序, 最高位控制通道四, 每个字符的控制方式如下:

X=0 预设范围0V~10V; (对应数字0~2000)

X=3 本通道关闭

X=4 预设范围0V~10V; (对应数字0~10000)

例如: BFM#0为H0034, 表示通道一为0V~10V; 通道二关闭; 通道三和通道四为0V~10V。

没有用到的通道, 可以关闭, 也可以不关闭, 关闭的通道不占用转换时间 (BFM#15), 例子中通道二关闭了, 则整个通道转换一次时间为没有关闭的三个通道的转换时间 (3×BFM#15)。

BFM #1~#4 输入通道平均采样数

每个通道(BFM#9~#12)对应的采样值累加采样数（BFM #1~#4）的个数后再除以采样数（BFM #1~#4），存放到（BFM #5~#8）

BFM #5~#8 输入通道存放平均采样值

BFM #9~#12 输入通道存放即时采样值

BFM #15 输入通道ADC速率时间

每个通道转换一次需要的时间，需注意更新一次数据需要的时间是BFM#15的时间乘以没有关闭的通道数。

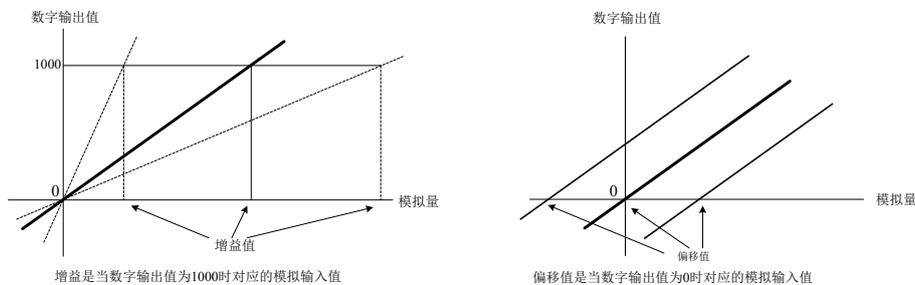
例如：BFM #0为H3311，BFM #1为K7，BFM #2为K6，BFM #15为K10；则BFM #9和BFM #10刷新一次数据时间为 $BFM\#0 \times BFM\#15 = 2 \times 10 = 20MS$ ，BFM #5刷新一次数据时间为 $BFM\#0 \times BFM\#15 \times BFM\#1 = 2 \times 10 \times 7 = 140MS$ ，BFM #6刷新一次数据时间为 $BFM\#0 \times BFM\#15 \times BFM\#2 = 2 \times 10 \times 6 = 120MS$ 。程序中FROM/TO指令比较耗时，故此模块参数在程序中采集BFM#5的数据可以用LDP M8012 FROM K0 K5 D10 K1指令读取，和LD M8000 FROM K0 K5 D10 K1的效果是一样的，但是后面用M8000驱动指令每个扫描周期都读取一次，大大加长了程序的扫描周期。

BFM #20 输入通道回归出厂值

将#20设定为1可以恢复到默认值。

BFM #21~#24 输入通道偏移和增益的定义与设定方法：

H2U-6CM 的输入通道共有两种工作模式，特性曲线如下图：



输入通道偏移和增益可以独立设定或一起设定，正常的增益值设定范围是1V~15V；正常的偏移值设定范围为0V~5V。

增益/偏移设定前，需先将BFM#21设为1，再修改BFM#23/ BFM#24；然后开各个通道的允许偏移增益BFM#22，修改完毕，应将BFM#21再设为2，避免再次被改变。

例如：在BFM#0为模式0中，需要修改通道一、通道二的偏移和增益分别为0.5V和6，则需要按如下步骤操作：

先将BFM#21改为1；过300ms以后将K500和K6000分别送到BFM#23和BFM#24去；再过300MS后开允许增益BFM#22，本例子中BFM#22应为二进制的00001111，即是将BFM#22修改为H000F；修改完毕。最后将BFM#21改为2，以防再次被修改。

7) 输出通道部分BFM区解释

BFM#32 输出通道选择

输出通道的初始化，默认2个通道都为-10V~10V，由BFM 32#的十六进制HXX控制，最低位控制通道五，第二位控制通道六，每个字符的控制方式如下：

- X=0 预设范围-10V~10V （对应数字-2000~2000）
- X=1 预设范围4mA~20 mA （对应数字0~1000）
- X=2 预设范围0mA~20 mA （对应数字0~1000）
- X=4 预设范围-10V~10V （对应数字-10000~10000）
- X=5 预设范围4mA~20 mA （对应数字0~10000）
- X=6 预设范围0mA~20 mA （对应数字0~10000）

例如：BFM#32为H10，表示通道五为-10V~10V；通道六为4mA~20mA。

BFM #33~#34 输出通道输出值

用TO指令往BFM #33~#34写数据，可以控制模拟量输出。初始值均为0。

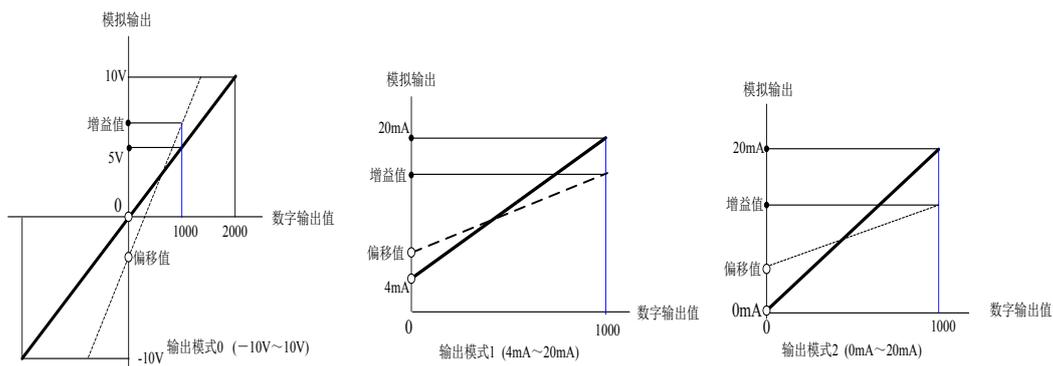
BFM #37 输出通道数据保持模式

当主模块在由RUN至STOP状态时RUN最后的模式被保持(X=0)或者为偏移值(X=1)输出。

例如：BFM#32为H00，BFM#37为H01，2个通道的偏移值均为0.1V，由RUN至STOP变化时BFM#33~#34里面的值均为1500（7.5V），当STOP后，通道5输出电压变成0.1V，通道6输出电压保持在7.5V。

BFM #40、BFM #42~#45、BFM #53 输出通道偏移和增益的定义与设定方法：

H_{2U}-6CM 的输出通道共有三种工作模式，特性曲线如下图：



其中，增益值为数字1000时对应的模拟量输出；偏移值则是数字为0时对应的模拟量输出。偏移和增益可以独立设定或一起设定，设定参数的单位是mV（模式0）和 μ A（模式1和2）

BFM #40为偏移、增益设置命令，十六进制的每HEX（二进制的4个bit位组成）位来控制禁止或者允许，注意在AD输入模块中是二进制的每个bit位来控制，DA模块和AD模块在偏移、增益设置命令有所区别。BFM #10~#13为偏移、增益设置值，BFM #53为曲线特性设置命令。

增益/偏移设定前，需先将BFM #53设为1，再根据需要修改BFM #42~#45的相关单元数值；再向BFM #40单元写入操作允许字，修改完毕，应将BFM #53再设为2，避免再次被改变。

例如：在BFM#40为模式H00中，需要修改通道五的偏移和增益分别为0.2V和5.5V，则需要按如下步骤操作：

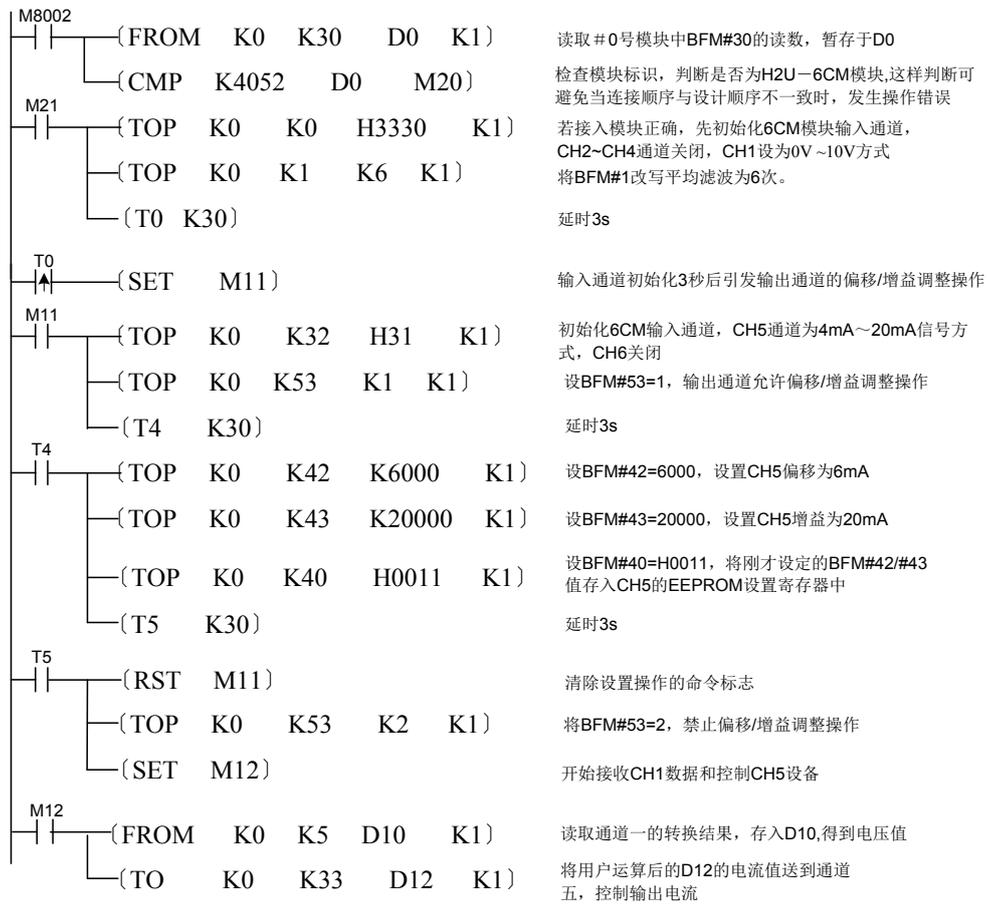
先将BFM#53改为1；过300ms以后将K200、K5500分别送到BFM#42、#43、中去；紧跟着BFM#40开允许偏移、增益，即是将BFM#40修改为H0011；修改完毕。最后将BFM#53改为2，以防再次被修改。

BFM #52 输出通道回归出厂值

将BFM #52设定为1可以恢复到默认值。

编程举例1：

一只H_{2U}-6CM扩展模块接于PLC主模块后方，按编号原则为#0号模块，其中CH1端口需要采集0V~10V的电压信号，要求滤波次数为6，将采集得到的数据存于D10。CH5端口由D12输出6mA~20mA的电流信号。编写的用户程序如下：



使用注意事项:

禁止带电插拔，只有在主模块和应用系统停电的情况下，才能进行扩展模块的接入或拆除工作，以保证人身安全，防止因带电插拔损坏器件；

6.6 温度控制模块

1 H_{2U}-4PT(R) 4 通道热电阻输入温度控制模块

H_{2U}-4PT(R)扩展模块可配合H2U系列主模块工作，实现4路热电阻温度控制，并将温度信号转换为12bit精度的数字量，供PLC主模块读取。主模块可以通过CANlink配置访问模块。

H_{2U}-4PT(R)扩展模块主要在原有4PT(R)模块基础上将温度控制功能加入到模块中。用户可将参数通过通讯方式设置到4PT(R)温控模块中，运行时通过采集到的温度与设置参数直接在扩展模块中进行温度控制，最后将输出保存在制定地址中。由于扩展模块没有输出点，PLC需通过通讯方式在主机上进行输出控制。

H_{2U}-4PT(R)扩展模块具备温控器的基本功能。主要具有ON/OFF控制方式、PID控制方式、单侧控制方式、加热冷却控制方式、数字输出控制、继电器输出控制、PID自整定功能和温度曲线功能。同时对基本设置参数进行实时限制，以及对运行状态、告警进行实时显示。

2 H_{2U}-4TC(R) 4 通道热电偶输入温度控制模块

H_{2U}-4TC(R)扩展模块可配合H2U系列主模块工作，实现4路热电偶温度控制，并将温度信号转换为12bit精度的数字量，供PLC主模块读取。主模块可以通过CANlink配置访问模块。

H_{2U}-4TC(R)扩展模块主要在原有4TC(R)模块基础上将温度控制功能加入到模块中。用户可将参数通过通讯方式设置到4TC(R)温控模块中，运行时通过采集到的温度与设置参数直接在扩展模块中进行温度控制，最后将输出保存在制定地址中。由于扩展模块没有输出点，PLC需通过通讯方式在主机上进行输出控制。

H_{2U}-4TC(R)扩展模块具备温控器的基本功能。主要具有ON/OFF控制方式、PID控制方式、单侧控制方式、加热冷却控制方式、数字输出控制、继电器输出控制、PID自整定功能和温度曲线功能。同时对基本设置参数进行实时限制，以及对运行状态、告警进行实时显示。

3 电气规格

电源	模拟电路	24V DC -15%/+20%，最大允许纹波电压5%；电流消耗80mA(取自于主模块的24V/COM，或其它的24VDC电源)
	数字电路	5V DC 50mA(通过模块扩展电缆，取自主模块电源内部电源)
	备注	远程扩展模块无需数字电路电源。

4PT(R) 指标:

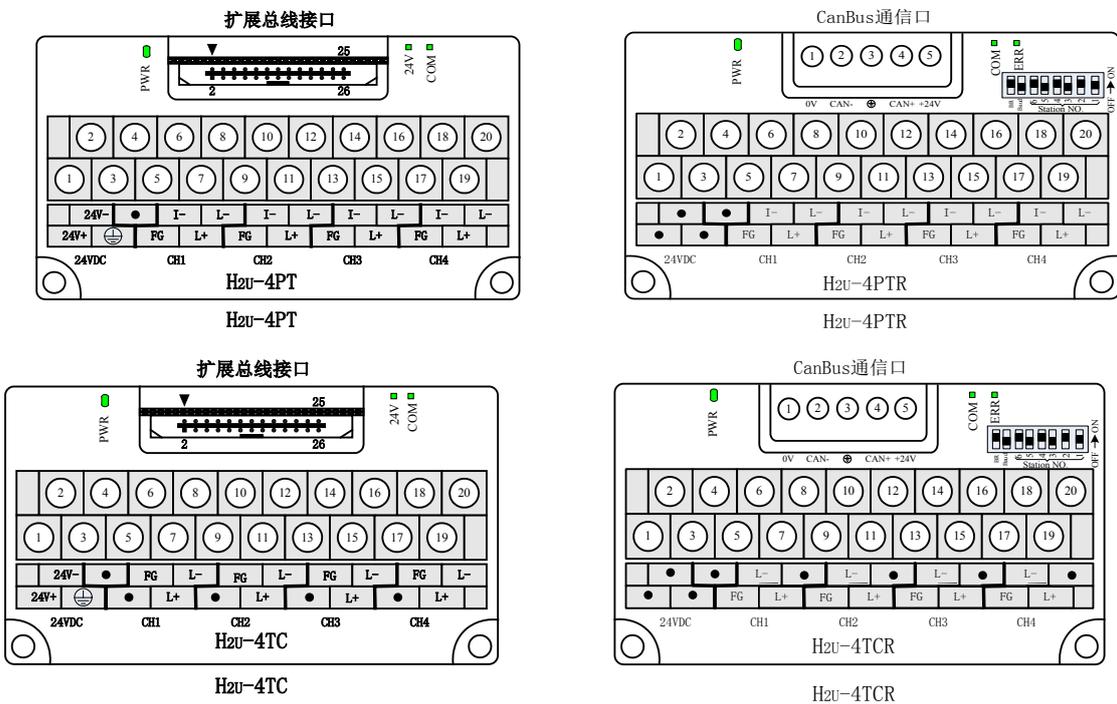
项 目	4PT(R) 指 标	
温度检测传感器	Pt100	Cu50
温度检测范围	摄氏: -100℃~600℃ 华氏: -148°F~1112°F	摄氏: -50℃~150℃ 华氏: -58°F~302°F
输入通道数	4通道	—
转换速度	15ms/通道	—
数字输出	摄氏: -1000~6000 华氏: -1480~11120	摄氏: -500~1500 华氏: -580~3020
分辨率	0.1℃	—
总精度	±1%全范围	—
占用I/O点数	8点	—
隔离设计	模拟电路和数字电路之间用光电耦合器隔离； 模拟电路和外部电源之间用DC/DC进行隔离； 模拟输入信号通道之间不隔离。	

4TC(R) 指标:

	摄氏	华氏
温度检测传感器	B/E/N/J/K/R/S/T型热电偶	
温度检测范围	B型: 200~1800℃ E型: -100~800℃ N型: -100~1200℃ J型: -100~850℃ K型: -200~1300℃ R型: -10~1700℃ S型: -10~1700℃ T型: -200~400℃	B型: 392~3272°F E型: -148~1472°F N型: -148~2192°F J型: -148~1562°F K型: -328~2372°F R型: 14~3092°F S型: 14~3092°F T型: -328~752°F

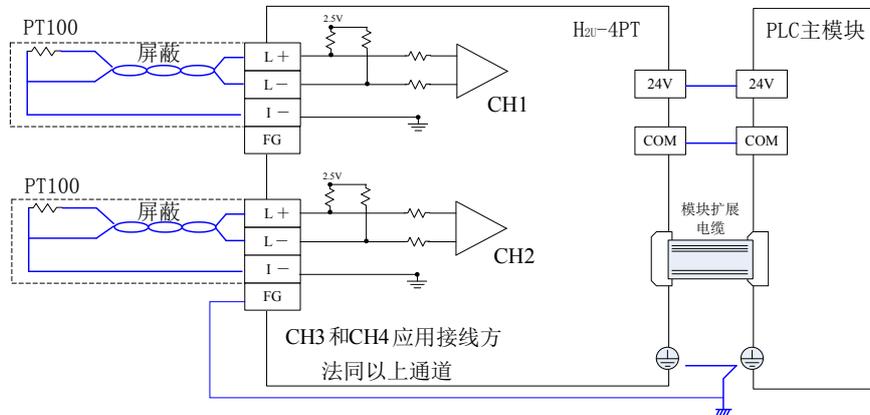
	摄氏	华氏
输入通道数	4通道	
转换速度	800ms/4通道	
数字输出	B型: 2000~18000 E型: -1000~8000 N型: -1000~12000 J型: -1000~8500 K型: -2000~13000 R型: -100~17000 S型: -100~17000 T型: -2000~4000	B型: 3920~32720 E型: -1480~14720 N型: -1480~21920 J型: -1480~15620 K型: -3280~23720 R型: 140~30920 S型: 140~30920 T型: -3280~7520
分辨率	0.1°C	0.1F
总精度	±0.5%全范围+1°C	
占用I/O点数	8点	
隔离设计	模拟电路和数字电路之间用光电耦合器隔离; 模拟电路和外部电源之间用DC/DC进行隔离; 模拟通道之间不隔离。	

接线端子布局:

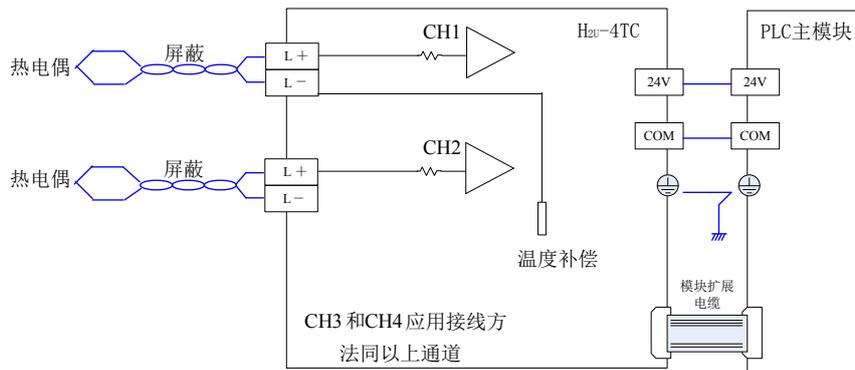


8) 输入信号与扩展电缆的接线:

模拟输入信号通过双绞线连接到扩展模块的输入端口，布线时不要与交流电源线或干扰信号的线路靠近；信号源及其屏蔽线的外壳与H2U-4PT (R) / H2U-4TC (R)的信号接地端FG相连，共同接地。



6
扩展模块
及扩展卡
使用方法



9) 4PT (R) 采集部分

访问4PT (R) /的BFM区：PLC主模块是通过读取4PT (R) 模块的寄存器缓存单元（BFM区）的方式读取数字化AD转换结果，通过改写特定BFM区的方式来设置模块状态。PLC主模块通过读写指令FROM/TO访问这些BFM单元。

BFM区的每个寄存器宽度为16bit（即1Word），按照4PT (R) 模块的BFM区定义如下表：

BFM	R/W属性	内容	
#0	WR	4PT (R) : 热电阻型号选择, 每个HEX位代表1个输入通道, 最高位为ch4, 最低位为ch1: (默认值=H0000) 0: PT100 5: Cu50	例图: [ch4 □ □ □ □] [3 2 1]
#1	WR	通道1	平均滤波常数, 即用于平均计算的采样值个数, 设定范围1~256, 默认值8。若要高速采集, 可设定为1。
#2	WR	通道2	
#3	WR	通道3	
#4	WR	通道4	
#5	R	通道1	CH1到CH4在0.1°C单位下的平均温度
#6	R	通道2	
#7	R	通道3	
#8	R	通道4	
#9	R	通道1	CH1到CH4在0.1°C单位下的当前温度
#10	R	通道2	
#11	R	通道3	
#12	R	通道4	
#13	R	通道1	CH1到CH4在0.1°F单位下的平均温度
#14	R	通道2	
#15	R	通道3	
#16	R	通道4	
#17	R	通道1	CH1到CH4在0.1°F单位下的当前温度
#18	R	通道2	
#19	R	通道3	
#20	R	通道4	
#21~26	—	保留	
#27	R	4PT 模块软件版本	
#28	R/W	数字范围错误锁存 (可检测热电阻断线)	
#29	R	错误状态	
#30	R	扩展模块识别码, H2U-4PT(R)的识别码为K2040	
#31	—	保留, 不可访问	

其中状态信息字BFM #28的意义说明如下：

b15到b8	b7	b6	b5	b4	b3	b2	b1	b0
未用	高	低	高	低	高	低	高	低
	CH4		CH3		CH2		CH1	

低：当温度测量值低于最低可测量温度时，锁存ON。

高：当温度测量值高于最高可测量温度时，或者热电偶断开时，锁存ON。

若出现错误之后，温度回到正常范围，则错误仍然被锁存在BFM#28中。

用TO指令向BFM28写入K0或者关闭电源，可清除错误。

注意：本地扩展模块的外部24V电源失电，PLC主模块的系统标志M8067会置位，出错代码D8067=K6708。编程时可定时检查该标志，能及时发现该现象。

其中状态信息字BFM #29的意义说明如下：

BFM #29位号	ON状态	OFF状态
b0:	存在错误。b0~b3中任一非0，A/D转换停止	无错误
b1:	保留	保留
b2:	保留	保留
b3:	模块硬件故障	硬件正常
b10:	数字输出超出指定范围	数字输出值正常
b11:	采样滤波常数超出1~256范围	采样滤波常数正常
b12:	保留	保留
BFM #29的其它bit4~7，bit13~15等没有定义。		

10) 4TC (R) 采集部分

访问4TC (R) 模块的BFM区：PLC主模块是通过读取4TC (R) 模块的寄存器缓存单元 (BFM区) 的方式读取数字化AD转换结果，通过改写特定BFM区的方式来设置模块状态。PLC主模块通过读写指令FROM/TO访问这些BFM单元。

BFM区的每个寄存器宽度为16bit (即1Word)，按照4TC (R) 模块的BFM区定义如下表

表六：4TC (R) 模块的BFM区定义

BFM	R/W属性	内容	
#0	WR	4TC (R)：热电偶型号选择，每个HEX位代表1个输入通道，最高位为ch4，最低位为ch1：（默认值=H0000） 0: K型 3: B型 6: S型 1: J型 4: N型 7: T型 2: E型 5: R型	
#1	WR	通道1	平均滤波常数，即用于平均计算的采样值个数，设定范围1~256，默认值8。若要高速采集，可设定为1。
#2	WR	通道2	
#3	WR	通道3	
#4	WR	通道4	
#5	R	通道1	CH1到CH4在0.1°C单位下的平均温度
#6	R	通道2	
#7	R	通道3	
#8	R	通道4	
#9	R	通道1	CH1到CH4在0.1°C单位下的当前温度
#10	R	通道2	
#11	R	通道3	
#12	R	通道4	
#13	R	通道1	CH1到CH4在0.1°F单位下的平均温度
#14	R	通道2	
#15	R	通道3	
#16	R	通道4	
#17	R	通道1	CH1到CH4在0.1°F单位下的当前温度
#18	R	通道2	
#19	R	通道3	
#20	R	通道4	
#21~26	—	保留	
#27	R	4TC模块软件版本	
#28	R/W	数字范围错误锁存（可检测热电偶断线）	
#29	R	错误状态	
#30	R	扩展模块识别码，H2U-4TC(R)的识别码为K2030	
#31	—	保留，不可访问	

其中状态信息字BFM #28的意义说明如下：

b15到b8	b7	b6	b5	b4	b3	b2	b1	b0
未用	高	低	高	低	高	低	高	低
	CH4		CH3		CH2		CH1	

低：当温度测量值低于最低可测量温度时，锁存ON。

高：当温度测量值高于最高可测量温度时，或者热电偶断开时，锁存ON。

若出现错误之后，温度回到正常范围，则错误仍然被锁存在BFM#28中。

用TO指令向BFM28写入K0或者关闭电源，可清除错误。

注意：本地扩展模块的外部24V电源失电，PLC主模块的系统标志M8067会置位，出错代码D8067=K6708。编程时可定时检查该标志，能及时发现该现象。

其中状态信息字BFM #29的意义说明如下：

BFM # 29位号	ON状态	OFF状态
b0:	存在错误。b0~b3中任一非0，A/D转换停止	无错误
b1:	保留	保留
b2:	保留	保留
b3:	模块硬件故障	硬件正常
b10:	数字输出超出指定范围	数字输出值正常
b11:	采样滤波常数超出1~256范围	采样滤波常数正常
b12:	保留	保留
BFM # 29的其它bit4~7, bit13~15等没有定义。		

11) 4PT (R) 温控部分（只有型号后面增加XP的含温控功能。例如：H2U-4PT-XP）

在4PT (R) 温控模块中原有参数保留原有含义，并新增加部分参数。在BFM # 40~ # 69中进行温控启动控制和保存当前周期输出值的设置，如表温控启动及输出数据参数定义。BFM # 100~ # 199为第1通道温度控制设置参数，BFM # 200~ # 299为第2通道温度控制设置参数，BFM # 300~ # 399为第3通道温度控制设置参数，BFM # 400~ # 499为第4通道温度控制设置参数。其中各个通道参数位置均相同只是首地址不同，所以以第1通道为例介绍BFM # 100以后参数。

表：温控启动及输出数据参数定义

BFM	R/W属性	内容	
# 51	WR	通道1	温度控制启动标志 0: 不进行温度控制 1: 启动温度控制
# 52	WR	通道2	
# 53	WR	通道3	
# 54	WR	通道4	
# 55	WR	通道1	PID控制暂停标志 0: PID控制正常运行 1: PID暂停
# 56	WR	通道2	
# 57	WR	通道3	
# 58	WR	通道4	
# 71	R	通道1	加热控制数字输出值
# 72	R	通道2	
# 73	R	通道3	
# 74	R	通道4	
# 75	R	通道1	加热控制继电器输出，仅输出1和0 如果为1表示继电器导通 如果为0表示继电器断开
# 76	R	通道2	
# 77	R	通道3	
# 78	R	通道4	
# 81	R	通道1	冷却控制数字量输出
# 82	R	通道2	
# 83	R	通道3	
# 84	R	通道4	
# 85	R	通道1	冷却控制继电器输出，仅输出1和0 如果为1表示继电器导通 如果为0表示继电器断开
# 86	R	通道2	
# 87	R	通道3	
# 88	R	通道4	

BFM	R/W属性	内容	
#91	WR	通道1	设定温度 单位: 0.1℃
#92	WR	通道2	
#93	WR	通道3	
#94	WR	通道4	

备注: 数字输出如果不设置范围, 默认为: -32768~32767。

表: 第1通道基本参数定义

BFM	R/W属性	内容
#100	WR	温度控制方式设置 Bit0: 0:继电器输出 1:数字输出 Bit1: 0:PID控制 1:ON/OFF控制 Bit2: 0:单侧控制 1:加热/冷却控制 Bit3: 0:正动作(冷却) 1:逆动作(加热) Bit4: 0:无动作 1:PID自整定 Bit5: 0:正常控制 1:温度曲线控制
#101	WR	参数限制设置(每Bit置1表示限制, 置0表示不限制) Bit0: 输入值(PV)限制 Bit1: 偏差带(PV-SV)限制 Bit2: 输出(MV)限制 Bit3: 设定值(SV)限制 Bit4: 输入变化值限制 Bit5: 输出变化值限制 Bit6: 冷端输出限制
#110	WR	KP (PID比例系数) 1~32767[%]
#111	WR	TI (PID积分系数) 0~32767(*100ms) 0: 没有积分
#112	WR	TD (PID微分系数) 0~32767(*100ms) 0: 没有微分
#113	WR	TS (PID采样周期) 1~32767(ms)
#114	WR	滤波系数 0~99[%] 0: 没有采样
#116	WR	自整定输出值
#117	R	当前目标温度
#118	WR	PID暂停输出值
#121	WR	继电器输出控制周期(加热)单位: 秒
#122	WR	继电器输出控制周期(冷却)单位: 秒
#124	WR	加热滞后(ON/OFF控制有效)
#125	WR	冷却滞后(ON/OFF、PID控制均有效)
#126	WR	冷却系数(数字输出冷却比例系数)
#130	WR	斜坡时间(温度曲线时间单位) 0: 单位分 1: 单位秒
#131	WR	斜坡值(单位0.1℃)
#140	WR	输入(PV)上限
#141	WR	输入(PV)下限
#142	WR	偏差带(PV-SV)
#143	WR	输出(MV)上限
#144	WR	输出(MV)下限
#145	WR	冷却输出上限
#146	WR	冷却输出下限
#148	WR	设定(SV)上限
#149	WR	设定(SV)下限
#150	WR	输入变化上限
#152	WR	输入变化下限

BFM	R/W属性	内容
#153	WR	输出变化上限
#154	WR	输出变化下限
#156	R	运行状态 0: 停止运行 1: ON/OFF控制 2: PID自整定 3: PID运算
#157	R	告警（停止运行）具体告警码见表BFM157告警含义
#158	R	告警（继续运行）具体告警码见表BFM158告警含义
#160-199	R	PID运行中间值

其中未涉及到变量作为保留，为以后扩展使用。BFM#200~#499分别为通道2、3和4所用，地址偏移与通道1相同。

注意：在启动温控模块之后，禁止改写BFM#100和BFM#101的变量设置，在温控模块停止后，可以改写变量。

建议温控模块运行时，不要修改PID相关参数。

表：BFM#157告警含义

告警码	BFM#157告警（停止运行）含义
1	输入限制参数错误
2	偏差带参数错误
3	输出限制参数错误
4	设定值限制参数错误
5	输入变化限制参数错误
6	输出变化限制参数错误
7	冷端限制参数错误
8	温度曲线参数错误
9	加热周期参数错误
10	冷却周期参数错误
30	采样时间TS<0
31	输入滤波常数不正常
32	输入比例系数不正常
33	冷却系数不正常
34	积分时间不正常
35	微分时间不正常
50	PID自整定设定值与偏差值过小 ※1
51	PID自整定最初方向与动作不一致 ※2
52	PID自整定动作不明显 ※3

※1：自整定开始时的测定值和目标值的差为150以下则结束，确认测定值、目标值后，请再次进行自整定。

※2：从自整定开始时，测定值预测的动作方向和自整定输出时实际动作方向不一致。请使目标值、自整定输出值、测定值的关系正确后，再次进行自整定。

※3：自整定中测定值因上下变化不能正确动作。请使采样时间远远大于输出的变化周期，增大输入滤波常数。设定变更后请再次进行自整定。

表：BFM#158告警含义

告警码	BFM#158告警（继续运行）含义
Bit0	输入值（PV）超限
Bit1	设定值（SV）超限
Bit2	输出值（MV）超限
Bit3	偏差带超限
Bit4	输入变化超限
Bit5	输出变化超限
Bit6	冷端输出超限

12) 4TC (R) 温控部分（只有型号后面增加XP的含有温控功能。例如：H₂U-4TC-XP）

在4TC温控模块中原有参数保留原有含义，并新增加部分参数。

表七：温控启动及输出数据参数定义

BFM	R/W属性	内容	
#32	WR	通道1	设定温度 单位：0.1°C
#33	WR	通道2	
#34	WR	通道3	
#35	WR	通道4	
#36	WR	通道1	KP (PID比例系数) 1~32767[%]
#37	WR		TI (PID积分系数) 0~32767(*100ms) 0: 没有积分
#38	WR		TD (PID微分系数) 0~32767(*100ms) 0: 没有微分
#39	WR	通道2	KP (PID比例系数) 1~32767[%]
#40	WR		TI (PID积分系数) 0~32767(*100ms) 0: 没有积分
#41	WR		TD (PID微分系数) 0~32767(*100ms) 0: 没有微分
#42	WR	通道3	KP (PID比例系数) 1~32767[%]
#43	WR		TI (PID积分系数) 0~32767(*100ms) 0: 没有积分
#44	WR		TD (PID微分系数) 0~32767(*100ms) 0: 没有微分
#45	WR	通道4	KP (PID比例系数) 1~32767[%]
#46	WR		TI (PID积分系数) 0~32767(*100ms) 0: 没有积分
#47	WR		TD (PID微分系数) 0~32767(*100ms) 0: 没有微分
#51	WR	通道1	温度控制启动标志 0: 不进行温度控制 1: 启动温度控制
#52	WR	通道2	
#53	WR	通道3	
#54	WR	通道4	
#55	WR	通道1	温度控制方式设置
#56	WR	通道2	Bit0: 0:继电器输出 1:数字输出
#57	WR	通道3	Bit1: 0:PID控制 1:ON/OFF控制
#58	WR	通道4	Bit2: 0:单侧控制 1:加热冷却控制
			Bit3: 0:正动作 (冷却) 1:逆动作 (加热)
			Bit4: 0:无动作 1:PID自整定 (请参见下文 “自整定使用方法”)
#59	WR	通道1	PID控制强度, 1~10 PID控制强度越大, 越快到达设定温度, 但超调量也会越大
#60	WR	通道2	
#61	WR	通道3	
#62	WR	通道4	
#63	WR	通道1	继电器输出死区时间, 1~32767(ms) 继电器输出连续动作最小时间间隔, 当时间小于该值时继电器输出将不动作。
#64	WR	通道2	
#65	WR	通道3	
#66	WR	通道4	
#67	WR	通道1	加热PID采样周期 1~32767(ms)
#68	WR	通道2	
#69	WR	通道3	
#70	WR	通道4	加热数字量输出 使用继电器控制方式时, 表示占空比 (0.00~100.00%)
#71	WR	通道1	
#72	WR	通道2	
#73	WR	通道3	
#74	WR	通道4	加热数字/继电器输出 数字输出, 0~10000 继电器输出, 仅输出1和0 如果为1表示继电器导, 0表示继电器断开
#75	R	通道1	
#76	R	通道2	
#77	R	通道3	
#78	R	通道4	冷却数字量输出 使用继电器控制方式时, 表示占空比 (0.00~100.00%)
#81	WR	通道1	
#82	WR	通道2	
#83	WR	通道3	
#84	WR	通道4	冷却数字/继电器输出 数字输出, 0~10000 继电器输出, 仅输出1和0 如果为1表示继电器导, 0表示继电器断开
#85	WR	通道1	
#86	WR	通道2	
#87	WR	通道3	
#88	WR	通道4	

BFM	R/W属性	内容	
#95	R	通道1	停止输出温差，单位：0.1℃ 当采样温度-设定温度大于该值时，停止加热输出 当设定温度-采样温度大于该值时，停止冷却输出
#96	R	通道2	
#97	R	通道3	
#98	R	通道4	

备注：对于温度控制方式使用单侧加热PID控制的场合，完全兼容52213-0000及以前软件关于4TC的BFM设置方式。

注意：在启动温控模块之后，禁止改写温度控制方式设置，在温控模块停止后，可以改写变量。

建议温控模块运行时，不要修改PID相关参数。

13) 部分BFM区解释

BFM#100 继电器输出：继电器输出通断占空比，输出单位是百分比，默认范围是0~100。以第一通道加热控制为例：BFM#75=1时，表示加热导通；BFM#75=0时，表示加热断开；

数字输出：当加热装置为模拟量控制时，可选择数字输出方式，用户可根据输出的数字量进行数模转换，来控制模拟量，默认为0~32767。如果设定输出上下限，假设设为0和10000。那么输出最小是0，最大是10000。

PID控制：选择PID控制模式，温控模块可根据PID参数自动调节温度。

ON/OFF控制：ON/OFF控制为最基本的通断控制，当前温度小于设定温度时，启动加热，当前温度大于设定温度时，停止加热，

单侧控制/加热冷却控制：选择单侧控制只有加热端有输出；选择加热冷却控制加热端和冷却端都有输出。

正动作/逆动作：当温度控制为制冷设备时应选择正动作，此时主动作为冷却；当温度控制为加热设备时应选择逆动作，此时主动作为加热。

PID自整定：当BFM#100的Bit4=1时，会先进入PID自整定状态，当自整定结束后，Bit4自动变为0，然后自动进入PID调节状态。

注意：当BFM#100的Bit1=1时选择ON/OFF控制，此时PID自整定功能无效。

温度曲线控制：当选择温度曲线控制时，设定温度会从当前温度开始，以设定的温度曲线模式，阶段性升高到目标温度。

BFM#101 参数限制设置：当BFM#101某一位置1时，表示进行相应的参数限制，如果超限，将产生告警来提示用户。

BFM#110~BFM#114 PID参数：温控模块根据PID参数来控制温度，采样周期和滤波系数需要用户设定，如果选择自整定模式，自整定结束后，系统会自动将整定的参数赋予BFM#110~BFM#112；如果没有选择自整定模式，需要用户自行设定。

注意：自整定的采样周期应远大于PID控制的采样周期，此处需要用户自行更改。

BFM#116 自整定输出值：如果自整定时需要设定输出值，此时将输出值输入到此变量中。自整定输出值与输出值一样，如果BFM #101有输出限制，自整定输出值不会超过BFM#143和#BFM#144的值。如果没有限制数字量输出默认是0~32767，继电器模式自整定输出值范围是0~100单位是百分比（假设为1000超出100会报输出值超限错误），如果超出界限，程序会自动更正为限制值，同时出现输出告警。

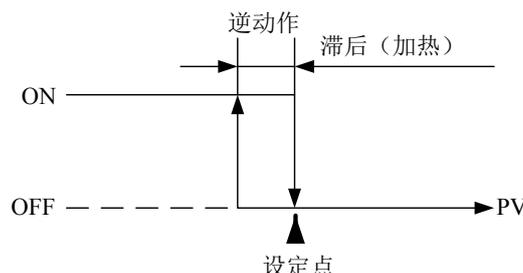
BFM#117 当前目标温度：设数据为只读，当选择温度曲线模式时才有效，温度曲线控制时，目标温度会根据设定参数分阶段上升，而BFM#117显示此刻的目标温度值。

BFM#118 PID暂停输出值：当BFM#55~BFM#58中设置1时，PID控制停止，此时的输出值由BFM#118来控制。当PID控制重新启动后，会继续进行PID控制，同时如果传感器断线，输出值也由BFM#118控制，此参数只在PID控制模式下有效。

BFM#121~BFM#122 继电器输出控制周期：当选择继电器输出控制模式时，需要设置继电器输出控制周期参数；虽然较短的周期，但可以提供更好的控制性能。

例如：设定2s，输出为50时，系统会加热导通1s，加热停止1s钟；如果输出为80时，系统会加热导通1.6s，加热停止0.4s。

BFM#124~BFM#125 滞后：在ON/OFF控制中，滞后用于在ON和OFF间切换时稳定运行。可分别在“滞后（加热）”和“滞后（冷却）”参数中设置控制输出（加热）和控制输出（冷却）。加热滞后的功能是：由设定温度减去加热滞后得出的值是ON/OFF临界点，无论正动作还是逆动作，它只对加热功能起作用。



BFM#126 冷却系数：当PID控制和加热冷却控制同时选择时有效，当PID控制时冷却端只有比例控制，冷却系数即表示冷却

端的P值。

BFM#130~BFM#131 温度曲线参数：当选择温度曲线控制时，需设置BFM#130和BFM#131这两个参数，BFM#130表示斜坡时间单位，BFM#131表示单位温度的变化值。

例如：BFM#130=0，BFM#131=10，则表示目标温度从当前温度开始以1分钟升高1℃的速度，直至升高到设定的温度。

BFM#140~BFM#154 限制参数：当BFM#101某一位置1时，需要设置相关的限制参数，如果控制过程中某个参数超出设定的界限，则会告警。如果设定输出上下限，例如分别设为30和70，表示最大的输出为70%，最小的输出是30%。

输入与输出变化限制：表示PID控制每次采样的变化限制，ON/OFF与自整定没有此功能。

注意：每个要设定的限制上限均不能小于下限，否则会停机告警。

BFM#156 运行状态：显示当前运行状态。

BFM#157 停机告警：当控制过程中遇到致命错误时，则温控功能停止，同时BFM#157显示相应告警码。

BFM#158 运行告警：当某一限制参数超限，则由BFM#158显示相应告警码，但此超限不会停机，系统会继续运行，此处告警只是用来提示用户。

BFM#160~BFM#199 PID运行中间值：中间值不可写入数据。

14) 自整定使用方法

使用自整定功能为了能得到最佳PID控制，用阶跃反应法自动设定重要常数：动作方向、比例增益、积分时间、微分时间。

自整定方法：

- a) 传送自整定输出值至BFM#116，自整定输出值请根据输出设备在输出最大值的50%~100%范围内使用。
- b) 设置自整定相关参数（采样时间、输入滤波等）以及目标值等。
- c) 自整定结束后，自动将自整定的参数保存在相应的变量中（如果PID功能导通则继续PID运算功能）。

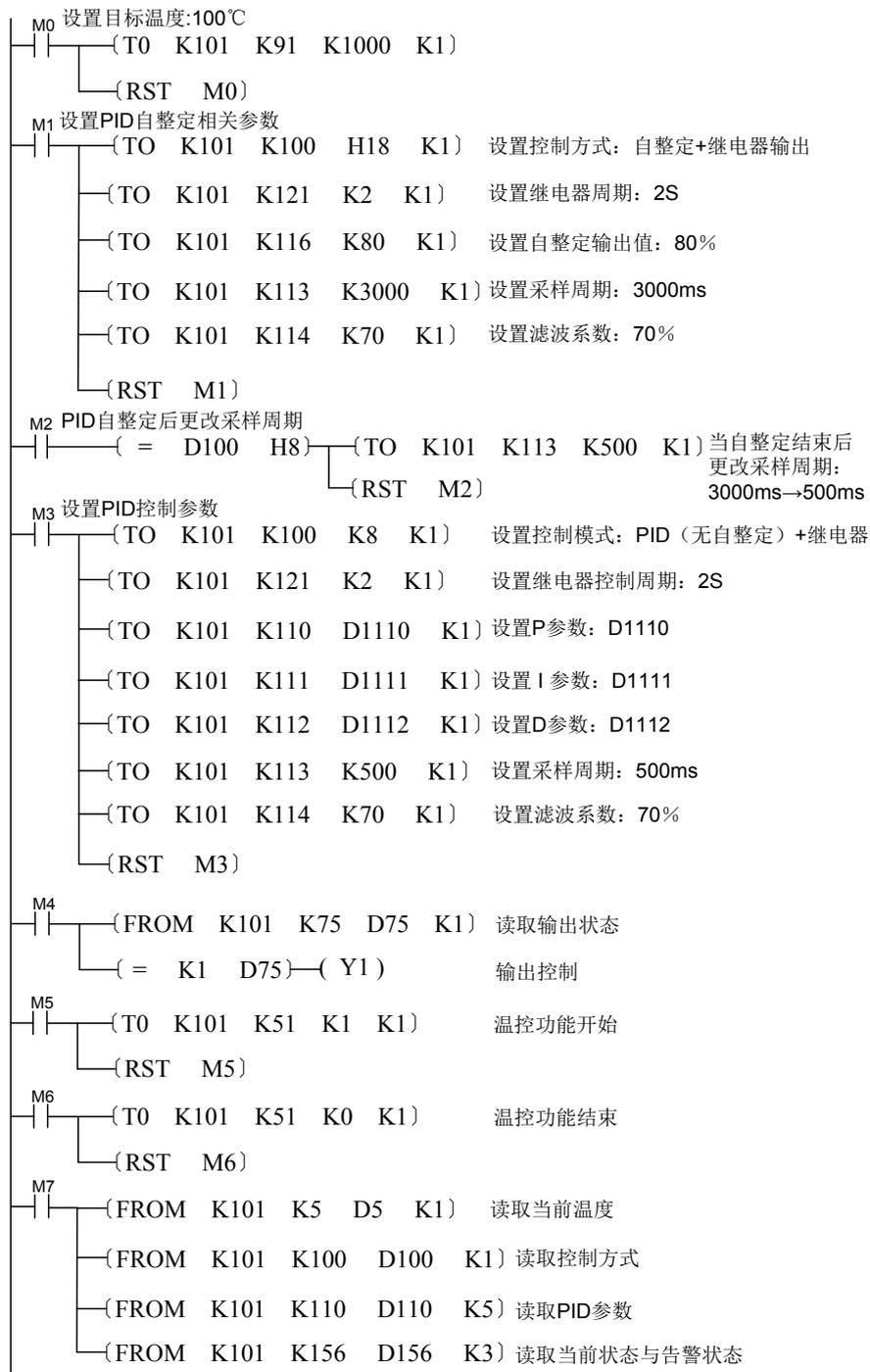
注意：

- a) 进行自整定时，若不满足上述自整定方法（1）、（2）两点，则出现不能正确自整定的情况，请特别注意。
- b) 自整定开始时的测定值和目标值的差如果在150以下则不能进行自整定。因此，若在150以下情况时，先设定自整定的目标值，待自整定完成后，再次设定目标值。
- c) 采样时间自整定时采样时间必须在1秒（1000ms）以上。另外本采样时间推荐使用大大长于输出变化周期的时间值。
- d) 自整定功能请在系统处于稳定状态时开始。如在不稳定的状态开始，则不能正确进行自整定。

使用注意事项：

禁止带电插拔，只有在主模块和应用系统停电的情况下，才能进行扩展模块的接入或拆除工作，以保证人身安全，防止因带电插拔损坏器件。

举例：一只H2U-4TCR-XP远程扩展模块，CAN站号为1，以CH1，继电器输出，设置温度：100℃。编写的用户程序如下：



6.7 扩展卡使用说明

6.7.1 H2U 系列 6A/6B 扩展卡使用说明

1) 简介

H2U-6A-BD扩展卡可实现2路电压输入（0V~10V）、2路电流输入（0~20mA）、2路电压电流模拟量输出，可满足有模拟量检测与控制的应用需求。

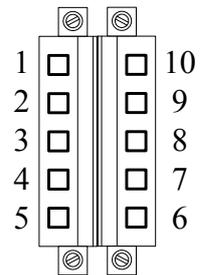
H2U-6B-BD扩展卡可实现4路电流(0~20mA)输入、2路电压电流模拟量输出，可满足有模拟量检测与控制的应用需求。

型号	独立通道数	输入端口		输出端口	
		0~10V型	0~20mA型	0~10V型	0~20mA型
H2U-6A-BD	4入2出	V1/V2	I3/I4	V1/V2	I1/I2
H2U-6B-BD	4入2出	—	I1/I2/I3/I4	V1/V2	I1/I2

2) 接线端口功能定义

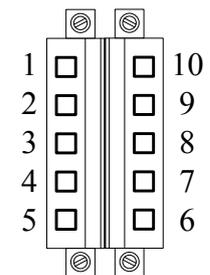
6A扩展卡引脚功能：

引脚	信号	功能描述	引脚	信号	功能描述
1	V1+	Ch1电压信号输入端	10	VO2+	Ch2电压输出端
2	V2+	Ch2电压信号输入端	9	IO2+	Ch2电流输出端
3	I3+	Ch1电流信号采样电阻端	8	VO1+	Ch1电压输出端
4	I4+	Ch2电流信号采样电阻端	7	IO1+	Ch1电流输出端
5	GND	输入公共接地端	6	GND	输出公共接地端



6B扩展卡引脚功能：

引脚	信号	功能描述	引脚	信号	功能描述
1	I1+	Ch1电流信号输入端	10	VO2+	Ch2电压输出端
2	I2+	Ch2电流信号输入端	9	IO2+	Ch2电流输出端
3	I3+	Ch3电流信号输入端	8	VO1+	Ch1电压输出端
4	I4+	Ch4电流信号输入端	7	IO1+	Ch1电流输出端
5	GND	输入公共接地端	6	GND	输出公共接地端



3) 编程参考

PLC主模块与扩展卡之间采用了通讯帧的方式进行数据交互，通讯帧由PLC系统软件自动组织和收发处理，使用时，用户程序只需在特定的系统缓冲区写入需要访问的扩展卡类型、输出值，在特定的寄存器中读取输入端口的检测数据即可。

通过设置D8220可实现两种功能：1、当D8220写入C000H时，扩展卡暂停，各通道只能读取当前值，不能修改参数；若重新启用扩展卡，请把D8220清0。2、当D8220写入C001H时，可以修改滤波时间，修改之后，请保持200ms，再把D8220清0。见下表：

设置扩展卡暂停命令：

PLC发送到扩展卡	访问扩展卡时的写入值
D8220	C000H暂停命令。此时无法修改相关数据。
D8221~ D8228	保留
D8229	PLC系统自动计算，用户程序不能更改
PLC从扩展卡接收	扩展卡回应的数据
D8230	C000H暂停命令已经生效。
D8231	模拟量输入通道1的当前值（量程：0-10000）
D8232	模拟量输入通道2的当前值（量程：0-10000）
D8233	模拟量输入通道3的当前值（量程：0-10000）
D8234	模拟量输入通道4的当前值（量程：0-10000）
D8235	模拟量输出通道1的当前值（量程：0-10000）
D8236	模拟量输出通道2的当前值（量程：0-10000）
D8237~ D8238	保留
D8239	通信错误计数，若通信正常，自动置0；非0，表示通信连续出错次数

设置滤波时间:

PLC发送到扩展卡	访问扩展卡的写入值
D8220	C001H写滤波时间命令。
D8221	滤波时间。范围1~32767单位: ms 默认滤波时间是1ms, 若超出设定范围, 则按照默认时间处理。
D8222 ~D8228	保留
D8229	PLC系统自动计算, 用户程序不能更改
PLC从扩展卡接收	扩展卡回应的数据
D8230	C001H写滤波时间命令已生效。
D8231	当前生效的滤波时间。范围1~32767单位: ms
D8232~D8238	保留
D8239	通信错误计数, 若通信正常, 自动置0; 非0, 表示通信连续出错次数

使用扩展卡时的相关寄存器及其定义:

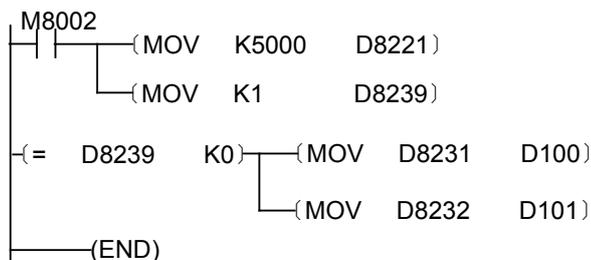
PLC发送到扩展卡	访问H _{2u} -6A-BD时的写入值	访问H _{2u} -6B-BD时的写入值
D8220	/	/
D8221	模拟量输出通道1的输出设定值 (量程: 0~10000)	模拟量输出通道1的输出设定值 (量程: 0~10000)
D8222	模拟量输出通道2的输出设定值 (量程: 0~10000)	模拟量输出通道2的输出设定值 (量程: 0~10000)
D8223~ D8228	保留	保留
D8229	PLC系统自动计算, 用户程序不能更改	
PLC从扩展卡接收	H _{2u} -6A-BD回应的数据	H _{2u} -6B-BD回应的数据
D8230	6A42h	6B42h
D8231	模拟量输入通道1的当前值 (量程: 0~10000) 对应0~10V	模拟量输入通道1的当前值 (量程: 0~10000) 对应0~20mA
D8232	模拟量输入通道2的当前值 (量程: 0~10000) 对应0~10V	模拟量输入通道2的当前值 (量程: 0~10000) 对应0~20mA
D8233	模拟量输入通道3的当前值 (量程: 0~10000) 对应0~20mA	模拟量输入通道3的当前值 (量程: 0~10000) 对应0~20mA
D8234	模拟量输入通道4的当前值 (量程: 0~10000) 对应0~20mA	模拟量输入通道4的当前值 (量程: 0~10000) 对应0~20mA
D8235	模拟量输出通道1的当前值 (量程: 0~10000) 对应0~10V, 或对应0~20mA	模拟量输出通道1的当前值 (量程: 0~10000) 对应0~10V, 或对应0~20mA
D8236	模拟量输出通道2的当前值 (量程: 0~10000) 对应0~10V, 或对应0~20mA	模拟量输出通道2的当前值 (量程: 0~10000) 对应0~10V, 或对应0~20mA
D8237~ D8238	保留	保留
D8239	通信错误计数, 若通信正常, 自动置0; 非0, 表示通信连续出错次数	通信错误计数, 若通信正常, 自动置0; 非0, 表示通信连续出错次数

由上面表格的说明可知, 6A/6B扩展卡的使用编程方法相同, 只需要向系统的专用D变量进行读(输入)或写(输出)操作即可, 以下以6A扩展卡的编程举例来说明使用方法。

应用编程举例1:

使用H_{2u}-6A-BD卡, 要求AO输出通道输出5V; 将AI1输入通道的采样值放入D100, AI2输入通道的采样值放入D101。

程序如下:

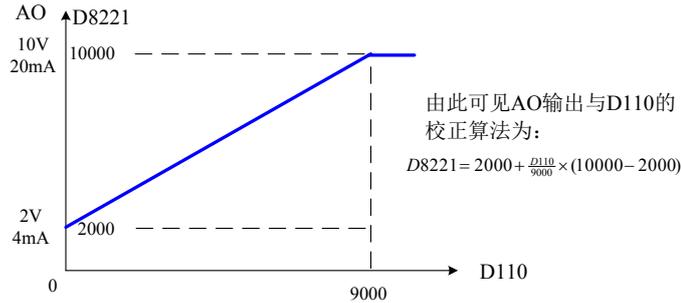


说明: 当D8239的值非0时, 表示PLC尚未正确读取到扩展卡的值, 编程时需注意; AO通电的输出值D8221可根据具体应用用户在用户程序中进行刷新。

应用编程举例2:

接上例, 使用H_{2u}-6A -BD卡, 将AI1输入通道的采样值放入D100, AI2输入通道的采样值放入D102; 但要求其AO电流输出与D110寄存器当前值相关, 当D110=0时, 输出4mA; 当D110=9000时, 输出20mA。

分析：AO的电流输出信号与电压输出信号，分别是（0~20mA）或（0~10V）对应D8221专用寄存器的（0~10000），若要4~20mA对应用户的寄存器值，需要用户程序作校正后写入D8221，才能得到希望的信号电流。（注意：校正后AO的电压信号特性也发生了变化。）



程序如下：

```

M8002 ┌──┴──┐ (MOV K1 D8239)
      └──┬──┘ (= D8239 K0) ┌──┴──┐ (MOV D8231 D100)
          └──┬──┘ (MOV D8232 D102)
M8000 ┌──┴──┐ (> D110 K9000) ┌──┴──┐ (MOV K9000 D110)
      └──┬──┘ (MOV D110 D0) //计算中，乘运算结果会超过1word，需采用32bit运算
          └──┬──┘ (MOV K0 D1) //将D110放（D1、D0）32bit临时单元
              └──┬──┘ (DMUL D0 K8000 D2) //将（D1、D0）乘8000，结果存（D3、D2）
                  └──┬──┘ (DDIV D2 K9000 D4) //将（D3、D2）除以9000，结果存（D5、D4）临时单元
                      └──┬──┘ (ADD D4 K2000 D8221) //将（D4）加2000，写入AO单元
                          (END)
    
```

应用编程举例3：
 设置滤波时间100ms，程序如下：

```

X5 ┌──┴──┐ (SET M0) M0置ON
    └──┬──┘ (MOV HC001 D8220) HC001写入D8220，可修改滤波时间
        └──┬──┘ (MOV K100 D8221) 设滤波时间100ms
M0 ┌──┴──┐ (T0 K2) 延时200ms
    └──┬──┘ (RST M0) 200ms之后M0复位
        └──┬──┘ (MOV H0 D8220) D8220清0
    
```

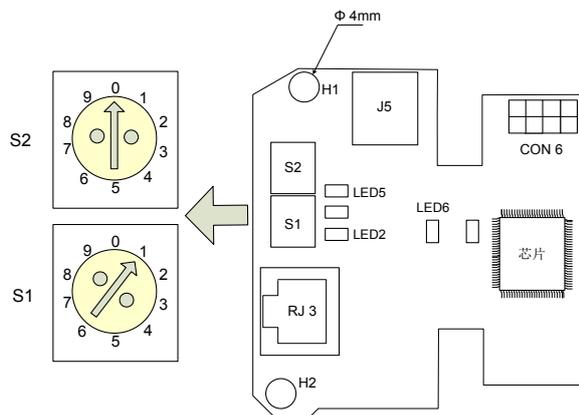
6.7.2 H2U 系列以太网通讯扩展卡使用说明

H_{2U}-ENET-BD以太网卡安装说明

为使H_{2U}-XP系列PLC实现以太网通讯功能，在H_{2U}-XP系列PLC主模块上配置H_{2U}-ENET-BD以太网通讯卡。安装之后，可支持PLC接入TCP/IP以太网，在应用层通过Modbus TCP进行数据交换。（主站加从站最多8台设备，通讯距离最远100米）。



H_{2U}-ENET-BD以太网通讯卡外观图



H_{2U}-ENET-BD以太网通讯卡端子分布

端子分布及功能说明

丝印名称	端子名称	功能
RJ45	RJ45接口	以太网通讯接口
S1	拨码盘	IP地址最后一段的个位（0-9）
S2	拨码盘	IP地址最后一段的十位（十位+个位，组成的范围为0-99）
H1/H2	固定安装螺钉孔	H ₂ U-ENET-BD的固定螺钉为M3*6自攻丝螺钉
J5	Mini USB	厂家保留
LED5	通信指示灯（绿色）	闪烁时表示以太网卡与PLC通讯正常，不亮表示无通信
LED2	通讯故障灯（红色）	常亮表示以太网卡与PLC无法通讯；闪烁表示以太网卡与其他设备通讯故障。故障可查看《错误诊断表》3-1
LED6	网卡指示灯	-
CON6	程序烧写口	厂家保留

以太网卡参数设置

1 以太网从站配置

以太网卡做从站（服务器）时，支持读保持寄存器（0x03）、写单个寄存器（0x06）、写多个寄存器（0x10），其他功能暂不支持。上位机可访问带以太网卡的PLC，并对PLC中元件数据进行读写。此时需要配置以太网从站。用户程序设置D8270为C1h（十进制193）、D8271为1、D8276为2。

- 1) 在AutoShop 后台软件的“工程管理”窗口，双击“以太网配置”打开“以太网配置”窗口。



2) IP地址栏

(1) IP地址：是设备在网络中身份的标识，必须确保每个设备IP地址的唯一性，否则设备将无法连接网络。IP地址由AutoShop后台软件设置，只设置前3段，最后一段由以太网卡面板拨码开关控制，取值范围1-99。

(2) 子网掩码：子网掩码作用是在同一个网络地址下为多个物理网络编址。掩码用于划分子网地址和主机ID的设备地址。获取子网地址的方法是：保留IP地址中与包含1的掩码的位置相对应的位，然后用0替换其他位。如无特殊要求，子网掩码均为255.255.255.0。

(3) 网关：可将网络消息路由到不在当前网络中的设备。如果没有网关，则网关地址设置为0.0.0.0。

3) 端口栏

TCP 502端口是为MODBUS通信保留的，以太网卡强制侦听这个端口。

4) 监控数据栏

共享寄存器地址：设置用户要访问的PLC字元件的首地址，同时也作为主站的数据存放区域。

共享寄存器数量：共享寄存器的数量。

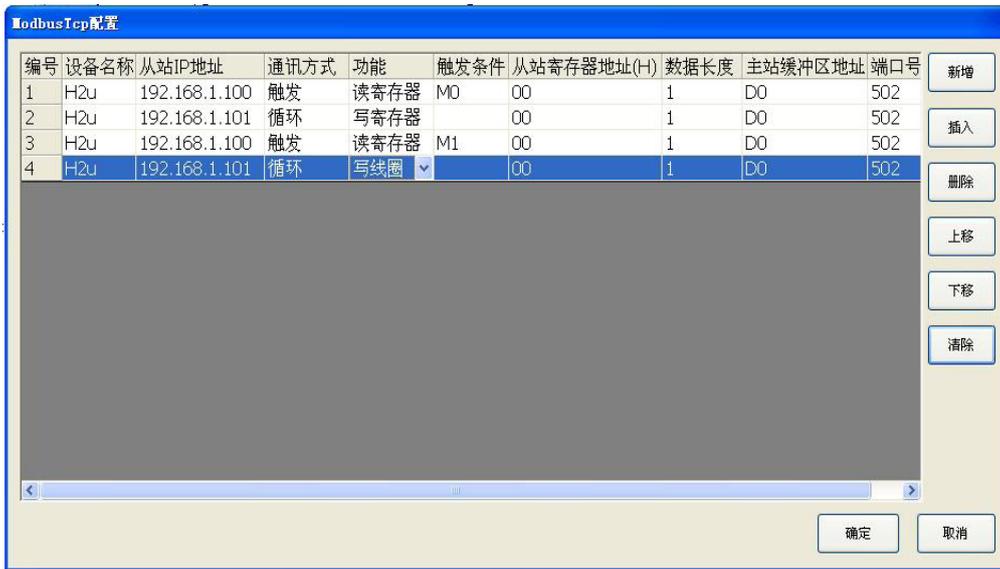
比如上图设置，通过ModbusTCP可以读取或者改变D200-D300的任何一个寄存器的值。

为确保PLC主模块中的数据安全，本产品只能监控连续的一段D寄存器，长度不超过120个字。其它元件如果要监控，可以通过PLC用户程序同步。例如要监控M10，可以在用户程序中写：“MOV M10 D200”指令，此时，监控D200等效于监控M10。

2 以太网主站配置

以太网卡可以做主机（客户端），支持读线圈（0x01）、读保持寄存器（0x03）、写单个线圈（0x05）、写多个线圈（0x0F）、写多个寄存器（0x10），去访问支持ModbusTCP协议的从机设备，其他功能码暂不支持。

1) 在“以太网配置”窗口，单击“以太网主站配置”按钮，打开“MdobusTcp配置”窗口；



2) 配置界面：

新增：在配置数据末尾增加一条新的配置记录

插入：在当前位置新增一条配置记录

删除：删除选中的配置记录

上移：调整选中行位置

下移：调整选中行位置

清除：清除窗口中所有配置信息

确定：保存修改、关闭窗口

取消：不保存修改、关闭窗口

3) 配置数据

通讯方式： 1) 循环：每个通讯周期都会执行一次； 2) 触发：当触发条件ON时，每个通讯周期执行1次，否则不执行。

功能：包含读寄存器、写寄存器、读线圈、写线圈。

触发条件：当通讯方式为触发时，才需要设置触发条件。触发元件可以为M、S、T、C元件。

从站寄存器地址：即Modbus协议中的从站寄存器地址，必须为16进制，并确保数据可读可写。

主站缓冲区地址：即Modbus命令读回数据时的存放地址，必须在从站设置的共享寄存器地址范围之内，否则会报“不是有效的主站缓冲区地址”。

端口号：TCP/IP链接时的端口号，由具体设备设置确定。当汇川PLC作从站时，端口号固定为502。

3 实例

现有PLC A，用其做ModbusTcp主站，IP地址设为192.168.50.54，需要周期读取IP地址为192.168.50.12的PLC B的寄存器D500，并将其保存到PLC A的寄存器D205中。

1) 设置PLC A的以太网从站和主站参数

▽ 设置IP地址

将以太网卡IP地址拨码开关的十位和个位分别拨到5和4。同时在后台软件设置其他IP参数。

从站参数

IP地址

IP地址: 192 . 168 . 50 . *

子网掩码: 255 . 255 . 255 . 0

网关地址: 192 . 168 . 50 . 0

注: 后台软件只设置IP地址的前3段, 最后一段由产品面板拨码开关控制, 取值范围0-99

▽ 设置共享寄存器

由于读取的数据要存到PLC A的D205中, D205需落到共享寄存器地址范围内, 如下设置, 则共享寄存器地址的范围为D200-D300。

监控数据

共享寄存器地址: D 200

共享寄存器数量: 100 (1-120)

▽ 主站配置

打开以太网主站配置界面, 设置从站IP地址(PLC B的IP地址), 循环读取, D500对应的16进制为1F4, 读取1个数据长度, 读取的数据放入PLC A的主站缓冲区D205中。

ModbusTcp配置									
编号	设备名称	从站IP地址	通讯方式	功能	触发条件	从站寄存器地址(H)	数据长度	主站缓冲区地址	端口号
1	H2u	192.168.50.12	循环	读寄存器		1F4	1	D205	502

2) 设置PLC B的从站参数

PLC B不用设置主站, 它只作为ModbusTcp服务器, 供PLC A读取数据。

▽ 设置IP地址

将以太网卡IP地址拨码开关的十位和个位分别拨到1和2。同时在后台软件设置其他从站参数。

从站参数

IP地址

IP地址: 192 . 168 . 50 . *

子网掩码: 255 . 255 . 255 . 0

网关地址: 192 . 168 . 50 . 0

注: 后台软件只设置IP地址的前3段, 最后一段由产品面板拨码开关控制, 取值范围0-99

▽ 设置共享寄存器

PLC B提供给PLC A访问的共享寄存器为D500, 因此需要让D500落到共享寄存器区域内。如下, 共享寄存器区域为D500-D600。

监控数据

共享寄存器地址: D 500

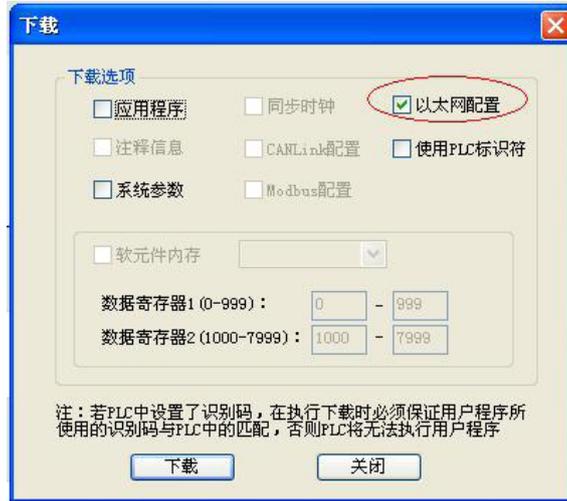
共享寄存器数量: 100 (1-120)

端口

侦听端口: 502

3) 分别下载以太网配置到PLC A、PLC B

下载时勾选以太网配置:



4) 下载完成后，需要把PLC A和PLC B进行一次下电上电操作，配置才能生效。

4 错误码信息

当以太网通讯错误时，可通过PLC寄存器D8063的值查看错误信息，如下表所示：

类型	错误代码	错误内容	处理方法
D8063	81xxh	不支持的功能码	修改IP地址为xx主站配置Modbus功能码。
	82xxh	寄存器数量超出	检查IP地址为xx主站配置中的数据长度是否超限。
	83xxh	寄存器地址超出范围	检查IP地址为xx主站配置中的从站寄存器地址是否超限。
	84xxh	与远程从站连接错误	检查IP地址为xx的设备是否可以连接。
	85xxh	串口通讯错误	检查以太网卡与PLC的连接是否正常；检查D8270是否为C1h（十进制193）、D8271是否为1、D8276是否为2
备注：错误代码中xx表示通讯错误设备的IP地址（最后一段）00h~FFh。			

举例，假如PLC A（IP地址 192.168.50.54），作为主站，连接PLC B远端从站（IP地址：192.168.50.12），当连接失败时D8063的数据为840Ch，表示PLC A与远程PLC B连接错误。

6.7.3 H1U/ H2U 系列通讯扩展卡使用说明

扩展卡类型号，H2U为自动识别，D8199显示此卡的类型号；H1U不能自动识别，需要通过D8199设定扩展卡类型。扩展卡类型如下：

1	RS232扩展卡	5	6A/6B扩展卡
2	RS422/485通讯扩展卡	6	高速扩展卡
3	AD扩展卡	7	CAN扩展卡
4	DA扩展卡	8	测试

比如我们需要在H1U中使用RS485扩展卡，则需要在程序中执行LD M8002 MOV K2 D8199。RS232扩展卡和RS422/485通讯扩展卡占用主模块COM1，其使用和设置方法与COM1口一致。

1) H2U-232-BD/H1U-232-BD扩展卡

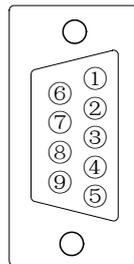
H2U-232-BD扩展卡为H2U系列PLC主模块用通讯扩展卡（H1U-232-BD扩展卡为H1U系列PLC主模块用通讯扩展卡），该卡的DB9信号插座提供了标准的RS232电平，在物理上是PLC主模块的COM1通讯端口，因此该板的通讯口与PLC原配RS485电平的COM1口不能同时使用，否则会发生通讯冲突。使用该扩展板，能实现PLC与PC机、HMI、MODEM、PLC、智能仪表等设备的通讯，通讯格式可由编程决定。

产品规格：

端口接头型式	DB9型，公座
信号标准	RS-232C
传输距离	与波特率有关，最远为15米
通讯方式	支持全双工、半双工方式
协议	MODBUS主、从协议；用于自定义协议
工作电源	5VDC，PLC内部电源提供
信号隔离	与PLC内部逻辑电路不隔离

引脚定义:

引脚号	信号名称	功能描述
1	CD	数据载波检查端（输入）
2	RXD	接收数据端口（输入）
3	TXD	发送数据端口（输出）
4	DTR	发送数据请求（输出）
5	GND	信号地
6	DSR	发送使能（输入）
7, 8	—	两引脚内部直接连接
9	—	内部无连接



与电脑、触摸屏等的连接方式和设置请参考《H2U通讯扩展卡用户手册》

2) H2U-485-BD（或H2U-485IF-BD）扩展卡

H2U-485-BD（或H2U-485IF-BD，IF：带隔离的产品，推荐使用）扩展卡为H2U系列PLC主模块用扩展通讯卡，该卡提供了RS485电平，将TX和RX信号线分别连接，又可作为RS422信号端口使用。在物理上是PLC主模块的COM1通讯端口，因此该板的通讯口与PLC原配RS485电平的COM1口不能同时使用，否则会发生通讯冲突。使用该扩展板，能实现PLC与PC机、HMI、PLC、智能仪表等设备的通讯，通讯格式可由编程决定。此卡需要M8126为ON时才有效。

端口功能定义:

引脚号	信号名称	功能描述
1	RA	485信号的RXD信号+
2	RB	485信号的RXD信号-
3	TA	485信号的TXD信号+
4	TB	485信号的TXD信号-
5	GND	信号地

与电脑、触摸屏等的连接方式和设置请参考《H2U通讯扩展卡用户手册》

3) H2U-422-BD扩展卡

H2U-422-BD扩展卡是H2U系列PLC主模块用通讯扩展卡，该卡的Mini DIN8信号插座提供了RS422电平，在物理上是PLC主模块的COM1通讯端口，因此该板的通讯口与PLC原配RS485电平的COM1口不能同时使用，否则会发生通讯冲突。使用该扩展板，能实现PLC与PC机、HMI、PLC、智能仪表等设备的通讯，通讯格式可由编程决定。H2U系列PLC使用该类型扩展卡后，可使得PC和HMI与之同时通讯。

端口功能定义:

引脚号	信号名称	功能描述
1	RB	485信号的RXD信号-
2	RA	485信号的RXD信号+
3	GND	信号地
4	TB	485信号的TXD信号-
5	VDD	5V+
6	NC	NC
7	TA	485信号的TXD信号+
8	NC	NC
金属壳	GND	信号地

该插座引脚的功能定义与COM0通讯口插座的定义相同

与电脑、触摸屏等的连接方式和设置请参考《H2U通讯扩展卡用户手册》。

4) H1U-CAN-BD/H2U-CAN-BD扩展卡

注：H1U-CAN-BD/H2U-CAN-BD扩展卡的详细介绍请参考《5.8 CAN通讯指令说明》。

6.7.4 H1U 系列 4A 扩展卡使用说明

1) H1U-4A-BD电气规格

项 目		参数说明
AI	输入通道数	2ch
	信号电平（差模）	电压通道：0V~10V，对应数字输出0~10000 电流通道：0mA~20mA，对应数字输出0~10000
	允许共模电压	5Vpp
	输入阻抗	不小于100kΩ
	ADC采样分辨率	12bit
	量化误差	0.3%
	采样速率	1ms/每通道（采样速率可设定）
AO	输出通道数	2ch
	信号电平	输出电压：0~10VDC
	DAC分辨率	12bit
	电压信号允许最小负载	2kΩ
	刷新速率	1ms（DA响应时间与用户程序扫描有关）

2) H1U-4A-BD扩展卡引脚功能

引脚	信号	功能描述	引脚	信号	功能描述
1	24V+	扩展卡供电电压正输入端	7	I2+	通道2电流正输入端，应用时要与V2+短接
2	0V	扩展卡供电电压负输入端	8	V/I2-	通道2电压/电流负输入端
3	V1+	通道1电压正输入端	9	VO1+	通道1电压输出正端
4	I1+	通道1电流正输入端，应用时要与V1+短接	10	VO2+	通道2电压输出正端
5	V/I1-	通道1电压/电流负输入端	11	VO-	电压输出通道负公共端
6	V2+	通道2电压正输入端	—	—	—

注意：扩展卡必须接24V电源，否则可能产生不可预知的测量值。

3) 编程参考

地址单元说明

地址单元	W/R属性(3)	功能描述
D8199	WR	扩展卡使用配置：H1U-4A-BD卡要配置为5
D8220	WR	命令寄存器(1)
D8221	WR	模拟量输出通道1的设定值（量程：0~10000）(2)
D8222	WR	模拟量输出通道2的设定值（量程：0~10000）
D8230	R	识别扩展卡类型：4A22H（表示是H1U-4A-BD卡）
D8231	R	输入通道1的当前值（量程：0~10000）(2)
D8232	R	输入通道2的当前值（量程：0~10000）
D8233	R	模拟量电压输出通道1的当前值（量程：0~10000）
D8234	R	模拟量电压输出通道2的当前值（量程：0~10000）
D8239	WR	通讯错误计数，若通讯正常，自动置0；非0，表示通讯连续出错次数

注：

(1) 命令说明：C000H暂停命令；C001H滤波常数设置命令；FFFFH版本号查看命令。

(2) 当D8220=C001H时，D8221为滤波常数设定值，D8231为滤波常数的当前值。另两个AI通道的滤波常数是一致的，不能分开设置。

(3) WR表示寄存器可读可写；R表示只读，对其进行写操作无效。

应用编程举例

由于H1U不能自动识别扩展卡类型，所以需要在用户程序的开始对特定寄存器D8199置值以启动扩展卡，设置语句如下：

```

    M8002
    ┌───┴───┐
    │         │
    └───┬───┘ [ MOV K5 D8199 ]

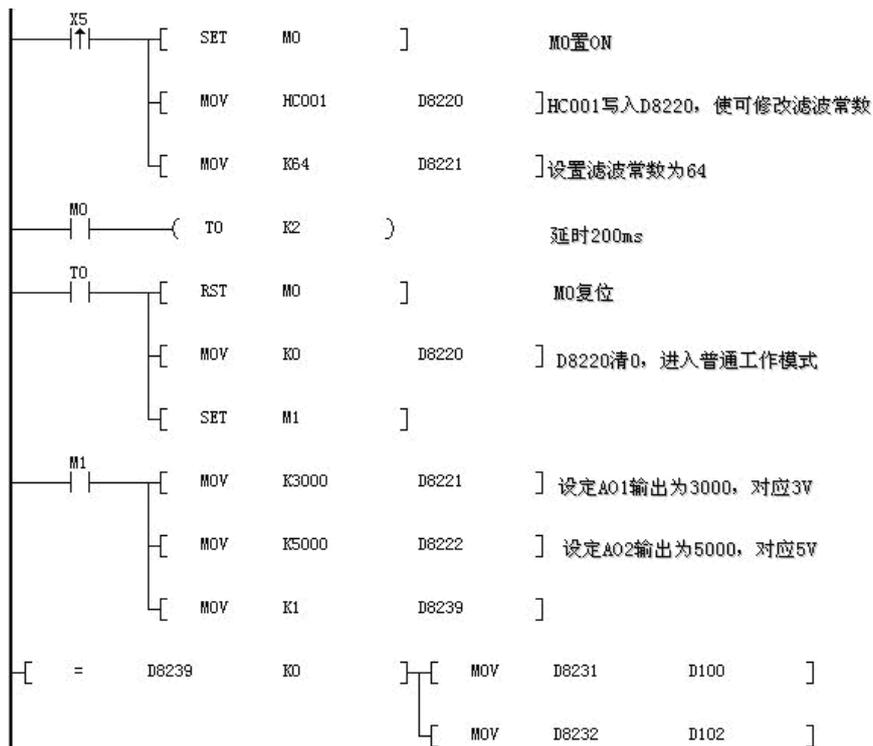
```

普通使用及滤波常数设置：

启动运行后，可以向D8221和D8222中置数决定AO1和AO2的输出大小，限定范围为0~10000（对应0~10V），同时可以从D8231和D8232中读出AI1和AI2的检测值，量程为0~10000，当选择电压输入时对应0~10V，电流输入时对应0~20mA。

注意：同一通道只能选择一种输入，即当选择电压输入时，电流输入端需要悬空，否则将影响检测结果。

例如，使用H1U-4A-BD卡，要求将滤波常数设置为64，AO1输出3V，AO2输出5V，同时，将AI1输入通道的采样值放入D100，AI2输入通道的采样值放入D102。程序如下：



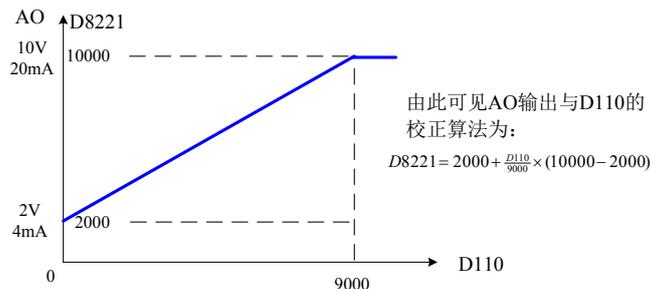
说明：滤波常数的有效设定范围为1~256，若写入值超出范围，则保持上次写入的有效值，本次超限写入值不会生效。滤波常数的当前值存放寄存器为D8231（D8220=HC001时）。

当D8239的值非0时，表示PLC尚未正确读到扩展卡的值，编程时需注意；AO通道的输出值D8221和D8222可根据具体应用在用户程序中进行刷新。

偏移增益设置

接上例，AI两通道的采样值存放地址不变，要求AO1电压输出与D110寄存器的当前值相关，当D110=0时，输出2V；当D110=9000时，输出10V。

分析：AO的电压输出信号，（0~10V）对应D8221专用寄存器的（0~10000），若要2~10V对应用户的寄存器值，需要用户程序校正后写入D8221，才能得到希望的信号电压。



程序如下：



6.8 差分转集电极输出模块使用说明

1) H_{2U}-2DC模块电气规格

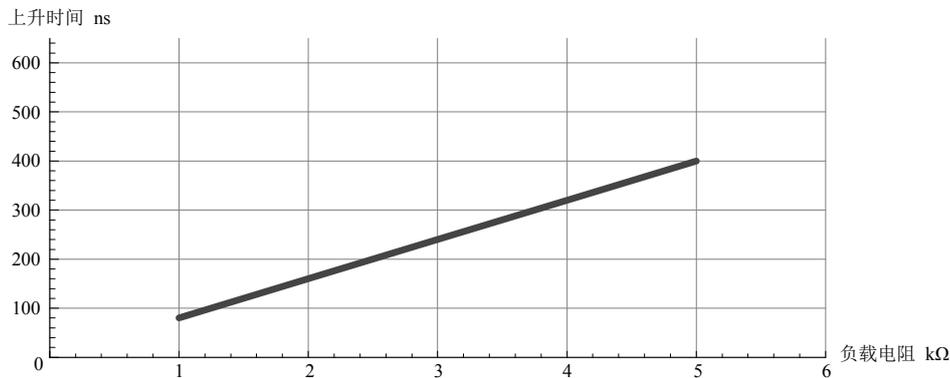
极限参数

参数名称	极限值
工作电源电压	32V
差分输入电压 (V+)-(V-)	7V
集电极输出峰值电流	400mA
集电极输出工作电压	33V

特性参数

参数名称	额定参数	工作范围	
		最小值	最大值
工作电源电压	DC24V	DC10V	DC30
差分输入VIL	0V	-6V	1.5V
差分输入VIH	5V	3V	6V
输入低电平脉冲时间	/	600ns	∞
输入高电平脉冲时间	/	600ns	∞
集电极输出导通压降	0.3V(100mA Load)	/	/
集电极输出工作电压	24V	9V	30V
集电极输出峰值电流	/	/	200mA
集电极输出连续电流	/	/	100mA
集电极输出下降沿	20ns	/	/
集电极输出上升沿	/	/	300ns(3K ohm RL)
最高工作频率	/	/	600kHz

集电极输出的上升沿时间与负载电阻有关系，上升沿时间与负载电阻关系如下图：



若负载电阻过大，将会导致集电极输出上升沿时间变长，如：600kHz工作时，一个脉冲的宽度为1670ns，如果使用负载电阻为5kΩ，将会导致有400ns的上升沿，约占了整个周期的25%，从而波形畸变将会比较严重。

2) 接线端子定义

0V	BO1	AO2	ZO2	A1+	B1+	Z1+	A2+	B2+	Z2+
VCC	AO1	ZO1	BO2	A1-	B1-	Z1-	A2-	B2-	Z2-

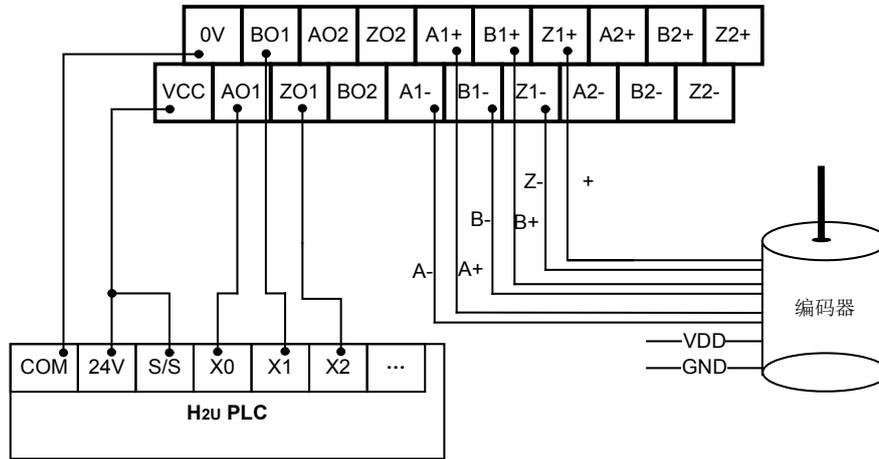
其中0V与VCC为模块的供电电源，A1+,A1-构成一组差分输入，其对应的集电极输出为AO1，其余通过名字类推。

该产品输入输出采用电气隔离，输入的6个通道之间也是隔离的，而6路集电极输出端的公共端是采取该产品的供电的0V。

3) 差分转集电极输出应用举例

ABZ型差分编码器差分输出信号，通过本产品H_{2U}-2DC进行信号转换后，接入汇川通用型PLC。连接图如下：

6
扩展模块
及扩展卡
使用方法



H_{2u}-2DC供电使用汇川PLC的外置24V供电，以A相为例，ABZ型差分编码器输出A+，A-接H_{2u}-2DC的A1+，A1-；AO1接PLC的X0，其余B，Z相对应接入。另外PLC输入侧的S/S引脚接24V。即可实现PLC对ABZ型差分编码器进行计数。



第七章 特色功能

7.1 禁止上载	332
7.1.1 禁止上载功能介绍.....	332
7.1.2 PLC加密设置	332
7.2 工程授权保护使用（24147及以上支持）	334
7.2.1 功能简介	334
7.2.2 工程授权保护.....	334
7.3 子程序加密/解密	336
7.4 PLC标识符.....	336
7.4.1 PLC标识符的设置界面.....	336
7.5 固件升级步骤指导.....	338
7.5.1 准备	338
7.5.2 升级操作.....	338

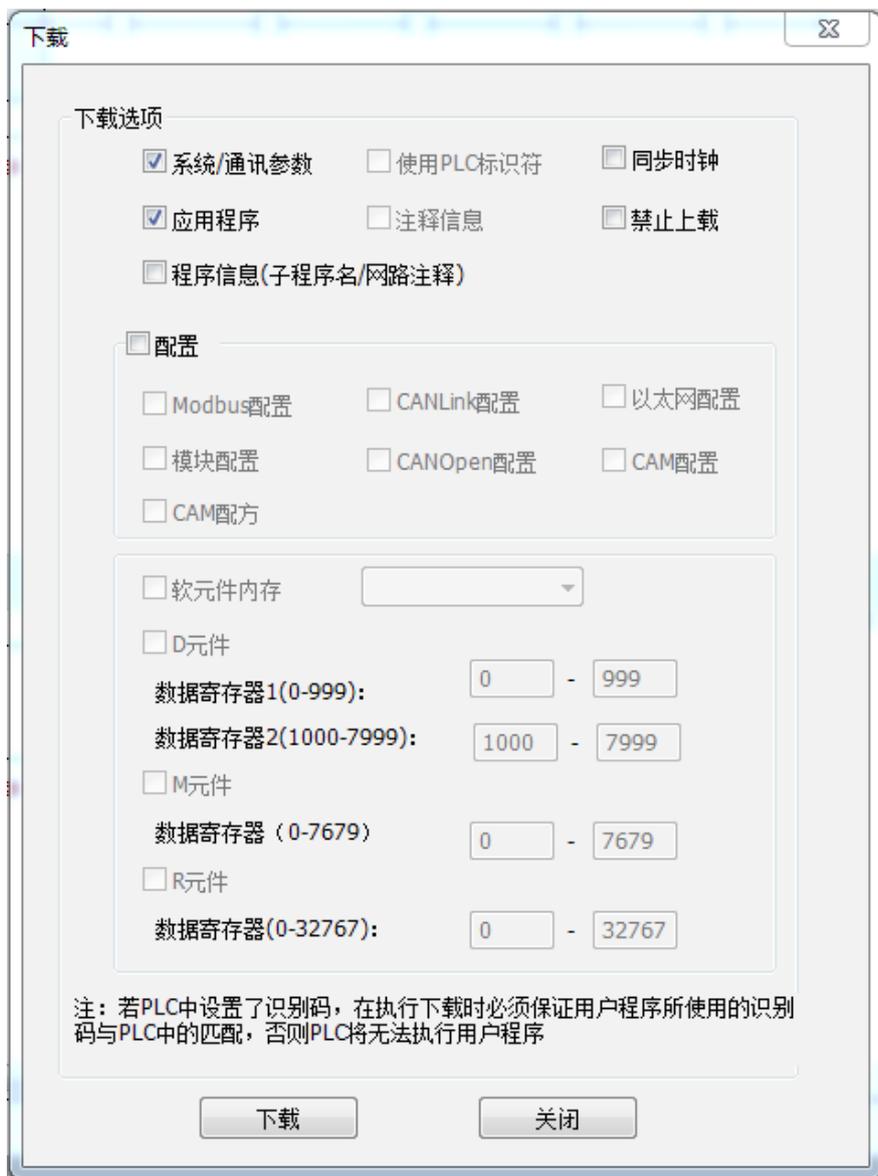
第七章 特色功能

7.1 禁止上载

7.1.1 禁止上载功能介绍

1、勾选“禁止上载”，则不能上载已下载的用户程序等数据；反之，未勾选“禁止上载”，若未设置上载密码则可直接上载用户程序等数据，设置上载密码，上载密码校验正确后，可以上载用户程序等数据；

2、如果要清除禁止上载功能，需重新下载一遍用户程序，下载时不勾选“禁止上载”，则清除该功能；



【注】：若PLC中设置了识别码，在执行下载时必须保证用户程序所使用的识别码与PLC中的匹配，否则PLC将无法执行用户程序。

7.1.2 PLC 加密设置

PLC加密设置功能可对PLC上载、下载、时钟设置操作进行权限确认；如果设置了密码，则需成功进行密码校验后才可完成操作；

上载与下载加密的作用是保护用户编写的PLC程序，防止非授权的访问。当设置上载密码后，每次执行上载的操作时都需要输入上载密码。类似的，当设置下载密码后，每次执行下载的操作时都需要输入下载密码。

密码长度为8位，可以是数字、字母以及其它所有可输入字符（比如+*/等字符也可以）。

【注】：当设置密码时，PLC必须处于停机状态。并且设置完成后，密码并不会立即生效，需要给PLC断电，并重新启动才可以。

<p>1) 加密设置</p> <p>点击菜单栏【PLC】-【加密设置】选项，便会弹出PLC加密设置对话框：</p>	
<p>2) 解开密码</p> <p>点击上载密码中的“解开密码”按钮，如果之前用户没有对PLC设置上载密码，系统便会弹出提示框，提示“PLC没有设置上载密码，无需解锁”；反之，便会弹出如右对话框：</p>	 <p>输入正确的密码后，点击“确定”，此时，在信息输出窗口会提示“上载密码解锁通讯命令执行成功”。当用户上传程序时便不需要进行上载密码校验了。</p>
<p>3) 设置密码</p> <p>点击上载密码中的“设置密码”按钮，如果之前用户没有对PLC设置上载密码，弹出的对话框如右图所示：</p>	 <p>如上图所示，需要输入新密码和确认新密码便可，然后点击“确定”，此时，在信息输出窗口会提示“上载密码修改通讯命令执行成功”。</p> <p>如果用户之前已经设置了PLC上载密码，则会弹出如下对话框：</p>  <p>用户需要输入正确的旧密码，然后输入新密码，再确认新密码后，点击“确定”，此时，在信息输出窗口会提示“上载密码修改通讯命令执行成功”。</p>
<p>4) 清除密码</p> <p>点击上载密码中的“清除密码”按钮，弹出“密码校验”对话框：</p>	 <p>如果用户之前没有设置上载密码，当点击“确定”按钮后，系统会弹出提示对话框；反之，当输入正确的上载密码后，点击“确定”，在信息输出窗口会提示“上载密码清除通讯命令执行成功”。</p>

7.2 工程授权保护使用 (24147及以上支持)

7.2.1 功能简介

已授权保护的工程只能用于特定授权PLC上, 实现用户工程与特定PLC的绑定, 即用户A对工程PrjA设置授权保护后, 则该工程只能运行在用户A授权的PLC上; 限制授权保护工程的使用范围;

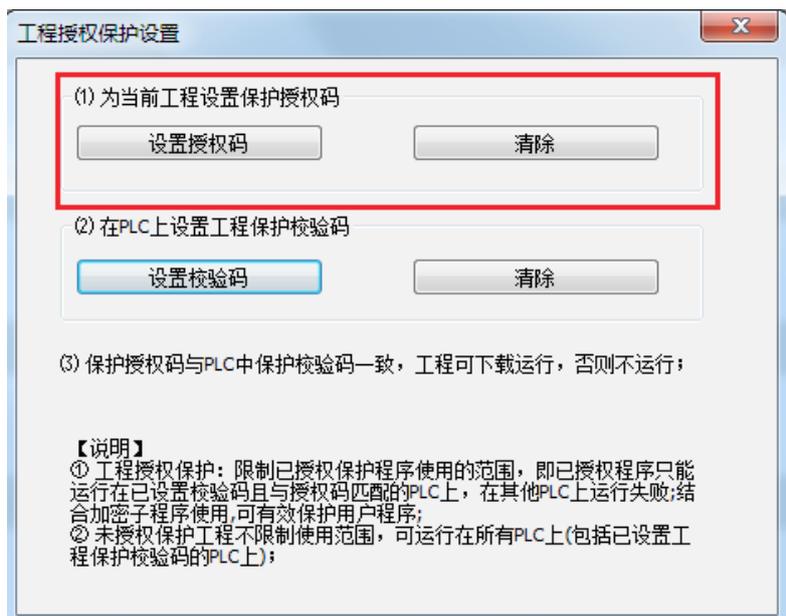
子程序加密可对工程中部分代码加密保护, 可对用户的核心算法、工艺进行有效保护;

通过工程授权保护与子程序加密结合使用, 既可防止工程中核心算法、工艺被窃取, 又可防止工程被非法使用, 实现对用户工程的灵活、有效保护;

7.2.2 工程授权保护

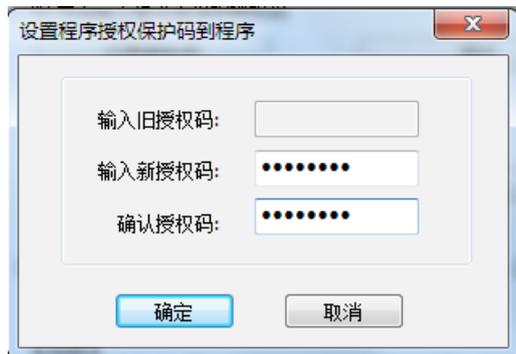
5) 工程中设置授权保护

单击菜单项 PLC|工程授权保护设置 打开“工程授权保护设置”窗口, 可为工程设置授权保护码、更改授权保护码、清除授权保护码如下图所示:



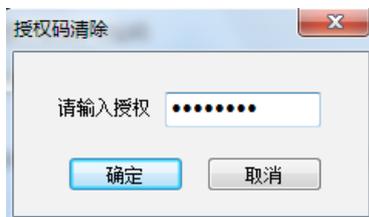
▽ 设置授权保护

单击“设置授权码”按钮, 打开授权码设置窗口, 如果首次设置授权保护, 则直接输入密码设置即可; 如果当前工程已设置授权保护, 则同时输入旧密码(授权保护码)、新密码(授权码), 旧密码校验成功收设置新授权码到工程中; 如下图所示:



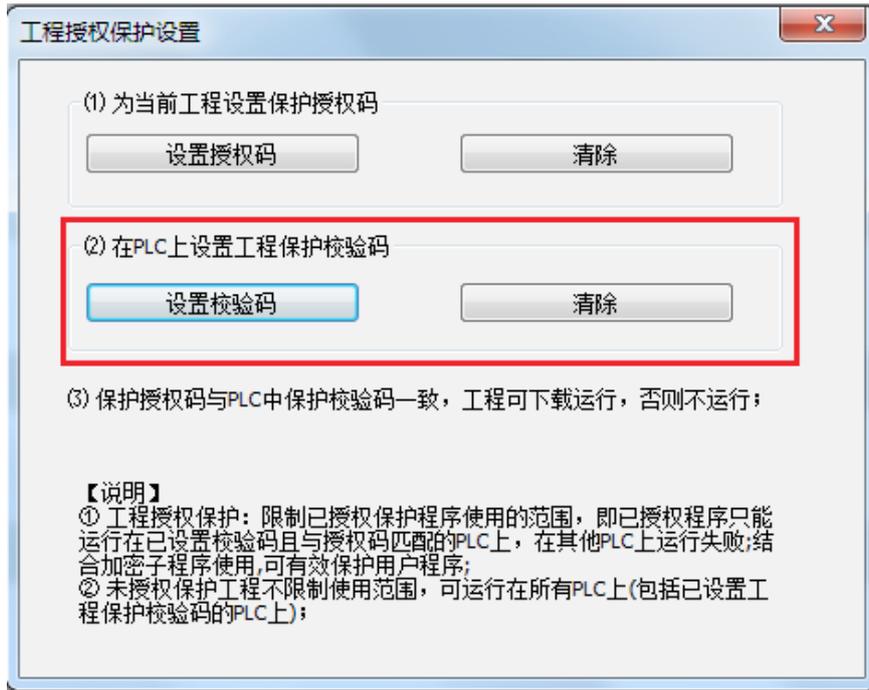
▽ 删除授权

单击“清除”按钮, 弹出“授权码清除”窗口, 输入授权码单击“确定”即可清除, 如下图所示:



6) PLC中设置校验码

单击菜单项 PLC|工程授权保护设置 打开“工程授权保护设置”窗口，如下图所示：



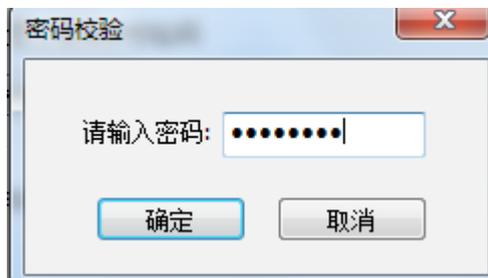
▽ 设置校验码到PLC

连接PLC后，单击“设置校验码”按钮，打开校验码设置窗口，输入密码后单击“确定”完成设置；如果当前PLC已设置校验码，需同时输入则旧密码、新秘密，旧密码校验成功后才能将新密码设置到PLC中；如下图所示：



▽ 清除校验码

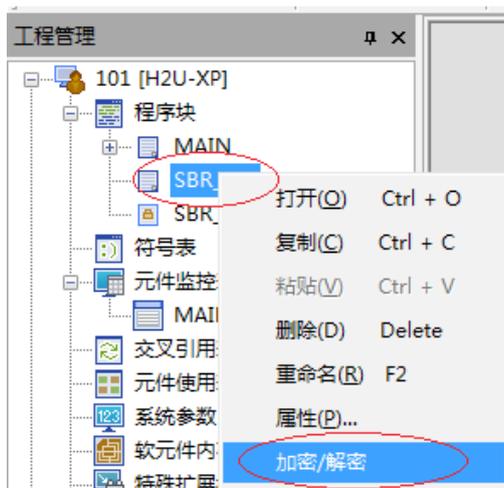
单击“清除”按钮后，弹出如下图所示窗口，输入密码校验成功后清除校验码；如下图所示：



7.3 子程序加密/解密

子程序加密的作用是将程序中的部分内容进行隐藏，这样可以将关键内容进行保护，公开非关键的部分，供其他人使用。

选中子程序单击右键，弹出如下菜单项：



加密：选中为普通子程序，然后执行“加密/解密”菜单，则会弹出密码输入窗口。输入想要设置的密码后，再点击“确定”，AutoShop就会将选中的子程序加密。加密后的子程序会显示一个锁型图标，加密后的子程序会自动关闭，不能被查看。

解密：双击加密程序会弹出密码窗口，输入正确后方能查看。

备注：PLC中设置的工程授权保护校验码不能通过“清除PLC程序存储空间”清除，只能通过“工程授权保护设置”窗口中“清除”按钮清除；如果忘记密码则无法清除在PLC上设置的程序授权保护校验码、清除校验码，需停机操作。

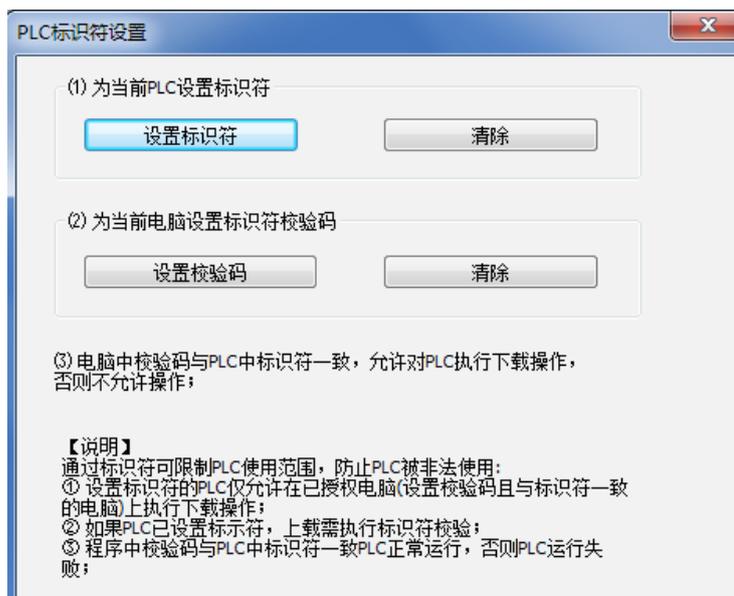
7.4 PLC标识符

PLC标识符是用于标志PLC的一个串号，相当于给PLC添加了一个唯一的标志。它的作用在于帮助PLC使用者保护自己的客户：添加了PLC标识符的PLC必须和添加了PLC标识符的程序匹配，PLC才可以正常工作。这里起到两个作用：

添加了标识符的PLC只能运行添加了标识符的程序；

添加了标识符的程序只能运行在添加了标识符的PLC上。

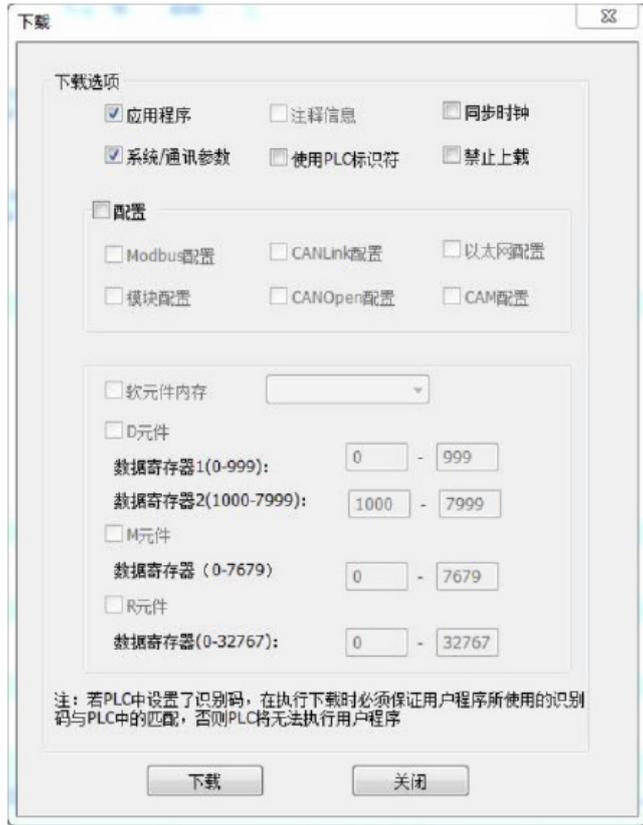
7.4.1 PLC 标识符的设置界面



上图中“(1) 为当前PLC设置标识符”指保存于PLC中的标识符，“(2) 为当前电脑设置标识符校验码”指保存于当前计算机中的标识符（使用这台计算机的AutoShop编写的程序都会使用这个标识符）。仅当这两者匹配时，程序才可以正常运行。所以通常这两者一起修改，这样可以保证PLC可以正常运行。

软件也允许单独修改“(1) 为当前PLC设置标识符”或“(2) 为当前电脑设置标识符校验码”。当单独修改“(1) 为当前PLC设置标识符”时，仅修改了保存于PLC中的标识符，当出现标识符不匹配的情况时可以进行这样的操作，需要PLC处于连通状态。当单独修改“(2) 为当前电脑设置标识符校验码”时，仅修改了保存于当前计算机中的标识符。

同其它密码相同，标识符的长度为8位，可以是数字、字母以及其它所有可输入字符（比如+*/等字符也可以）。当本机设置了“程序中的标识符”后，进行程序下载时，下载选项中的“使用PLC标识符”变为可选项，如图：



下载前，AutoShop首先会检查PLC是否已经设置了标识符。如果已经设置，则判断与当前程序中的PLC标识符是否匹配，如果匹配则继续下载，如果不匹配，则不能下载；如果没有设置过标识符，那么AutoShop会将当前标识符设置到PLC中。

上载操作类似，如果PLC中的标识符与当前程序中的标识符不匹配，则不能上载。

【注】：PLC中的标识符一经设置，就不能被清除，只允许进行修改。所以如果设置了标识符，请牢记密码！如果遗忘了，只能将返回汇川公司进行处理。

7.4.2 PLC标示符修改

设置PLC标示符，则下载用户程序时PLC端标示符与后台保存标示符进行自动匹配，匹配成功则继续下载操作；否则下载操作终止；

☞ 勾选“PLC中的标示符”

单击“设置”按钮弹出“PLC标示符设置”，可实现对PLC设备、后台保存的“PLC标示符”进行修改；



单击“清除”可实现对PLC设备、后台保存的“PLC标示符”



☞ 勾选“程序中的标示符”

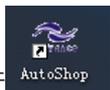
操作与“PLC中的标示符”一样，相应操作仅对后台的“PLC标示符”起效。

7.5 固件升级步骤指导

7.5.1 准备

- ▽ Autoshop版本：1.20版本及以上；
- ▽ PLC版本：H2U-XP:24136及以上；H1U-XP:26127及以上。
- ▽ 升级线缆：mini USB电缆（公司物料编码 15041200）
- ▽ 不需准备220VAC供电，可直接由电脑USB口给PLC供电。
- ▽ 固件为.upgrade文件，将升级所需.upgrade文件放置于Autoshop安装目录：C(D):\Inovance Control\AutoShop\Upgrade。

7.5.2 升级操作



- 1、打开汇川PLC编程软件
- 2、用mini USB电缆连接需要烧录软件的单板，另一头连接电脑USB口，PLC由USB口供电，不需上220VAC电源。

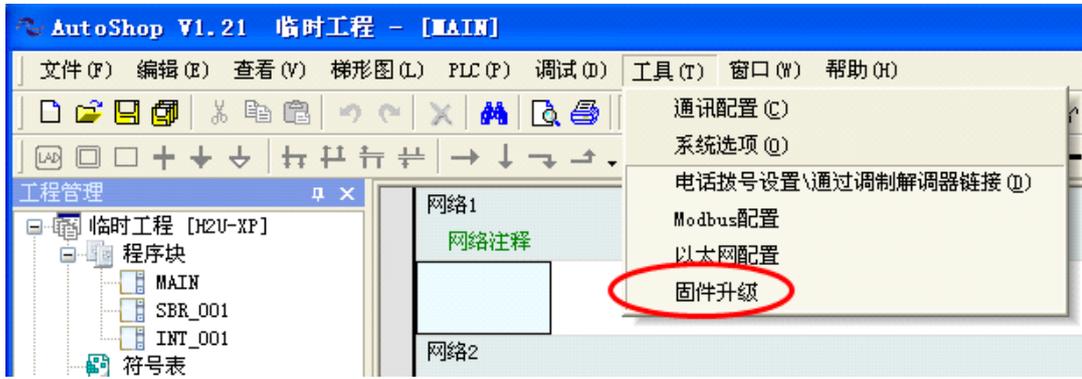
连接成功，显示器右下角有该图标显示(确认USB驱动安装是成功的)；未安装驱动时会出现如下硬件安装提示，点击自动安装软件即可。



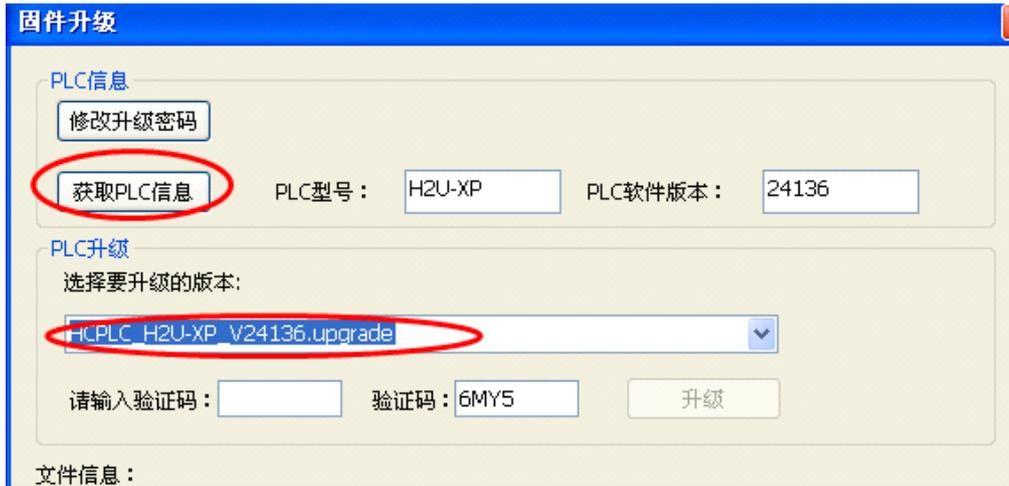
- 3、Autoshop 通讯设置中PC和PLC相连接方式选择USB，连接正常与否，可点击测试。



- 4、在PLC编程软件AutoShop界面里面参考如下提示找到“工具==》固件升级”功能选项



5、确认USB驱动是安装成功的，如下图示，点击获取PLC版本信息；则可以显示当前PLC的软件版本；点击“选择要升级的版本”下拉箭头，选择要升级的固件版本。



6、输入验证码后，按弹出界面提示操作，需要将拨码开关置“stop”状态；点击“升级”按钮；整个烧录过程需要等待几秒



7、软件烧录成功后如下提示

The screenshot shows the 'PLC升级' (PLC Upgrade) section of the software. A modal dialog box titled 'AutoShop' is displayed in the center, with the message '升级成功!' (Upgrade Successful!) and a '确定' (OK) button. The background interface includes the following elements:

- PLC信息** (PLC Information): A '修改升级密码' (Modify Upgrade Password) button and an '获取PLC信息' (Get PLC Information) button. The 'PLC型号' (PLC Model) is 'H2U-XP' and the 'PLC软件版本' (PLC Software Version) is '24136'.
- PLC升级** (PLC Upgrade): A section for selecting the version to upgrade to, with a dropdown menu showing 'HCPLC_H2U-XP_V24136(1).upgrade'. Below this are input fields for '请输入验证码' (Please enter the verification code) and '验证码' (Verification code), both containing 'O9XS', and an '升级' (Upgrade) button.
- 文件信息** (File Information): A section for recording changes, including fields for '公司信息' (Company Information) with value 'HC', '软件信息' (Software Information) with value 'CMR3232AV2110 (24136-00000)', and '版本号' (Version Number).

8、再次获取一次PLC信息，此时为最新下载的版本号；软件烧录完成。

This screenshot shows the same 'PLC升级' interface as the previous one, but with the 'PLC软件版本' (PLC Software Version) field highlighted by a red circle. The value in this field is '24136', indicating that the software has been updated to the latest downloaded version.

7

9、下载操作

如果工程已授权保护，则下载程序到已设置校验码且与工程中授权码一致的PLC中可正常运行，否则运行失败；
如果工程未授权保护，可运行在任意PLC上，即使PLC上已设置校验码，也可正常运行。



第八章 附录

8.1 系统特殊软元件	342
8.2 系统错误记录功能.....	351
8.2.1 错误代码及内容	351
8.2.2 PLC运行时间/异常记录.....	354
8.2.3 使用Autoshop进行故障诊断和处理.....	355
8.3 错误代码寄存器	356
8.4 H2U系列MTQ型与MT型的软件差别	356
8.5 H2U系列MTP型软件补充说明	358
8.5.1 DRVI、DRVA特殊功能的描述	358
8.5.2 ZRN指令的描述.....	360
8.6 H2U系列MTQ/MTP机型不支持的指令一览	361
8.7 H2U系列高速处理指令的增强功能说明	362
8.8 程序流程控制指令的相互关系.....	366
8.9 部分特殊继电器和寄存器功能说明	368

第八章 附录

8.1系统特殊软元件

M8000~M8511, D8000~D8511被定义为特殊元件种类, 及其功能如下表所述。

M元件	M元件的描述	D元件	D元件的描述
系统运行状态			
M8000	用户程序运行时置为ON状态	D8000	用户程序运行的监视定时器
M8001	M8000状态取反	D8001	单板程序版本, 如24100H2U=24, 100版本V1.00
M8002	用户程序开始运行的第一个周期为ON	D8002	程序容量, 4K, 8K, 16K等
M8003	M8002状态取反	D8003	固定为0X10, 为可编程控制器内存储
M8004	当M8060~M8067[除M8062]中任意一个处于ON, 则M8004有效	D8004	错误的M8060~M8067的BCD值, 正常为0。
M8005	电池电压过低时动作 (XP型不具备)	D8005	电池电压的BCD当前值 (XP型不具备)
M8006	电池电压低有出现就动作[锁存]	D8006	电池电压过低的检测值, 初始值为2.6V
M8007	当交流失电5ms后M8007&M8008动作, 但在D8008之内程序继续运行	D8007	保存M8007动作的次数, 当失电时该单元清0处理
M8008	当D8008时间内都失电, 当M8008由ON→OFF时, 用户程序不运行。M8000为OFF	D8008	交流失电检测时间, 默认为10ms
M8009	扩展单元24V掉电时动作	D8009	扩展单元24V失电的模块号
系统时钟			
M8010	保留	D8010	当前扫描时间, 从用户程序0步开始(0.1ms)
M8011	10ms时钟周期的振荡时钟	D8011	扫描时间的最小值(0.1ms)
M8012	100ms时钟周期的振荡时钟	D8012	扫描时间的最大值(0.1ms)
M8013	1S时钟周期的振荡时钟	D8013	时钟秒(0~59)
M8014	1分钟时钟周期的振荡时钟	D8014	实时时钟分(0~59)
M8015	时钟停止和预置	D8015	实时时钟小时(0~23)
M8016	时钟读取显示停止	D8016	实时时钟日(1~31)
M8017	±30秒修正	D8017	实时时钟月(1~12)
M8018	安装检测	D8018	实时时钟公历年(2000~2099)
M8019	实时时钟(RTC)出错	D8019	实时时钟星期
指令标志			
M8020	运算零标志	D8020	X000~X007的输入滤波常数1~60[默认10ms]
M8021	运算借位标志	D8021	保留
M8022	运算进位标志	D8022	保留
M8023	保留	D8023	保留
M8024	BMOV指令的方向	D8024	保留
M8025	HSC指令模式	D8025	保留
M8026	RAMP指令模式	D8026	保留
M8027	PR模式	D8027	保留
M8028	保留	D8028	和Z0同一变量地址
M8029	部分指令(多周期执行指令)执行完成	D8029	和V0同一变量地址
系统模式			
M8030	为ON时, 屏蔽电池低告警	D8030	保留
M8031	为ON时, 清除所有非保存存储器	D8031	保留
M8032	为ON时, 清除所有保存存储器	D8032	保留
M8033	为ON时, 停机状态所有的软元件不变	D8033	保留
M8034	为ON时, PLC所有的输出都为OFF状态	D8034	DRVI,DRVA的偏置运行速度, 由M8135决定是否有效[Y0] [默认100]
M8035	强制运行命令1	D8035	DRVI,DRVA的偏置运行速度, 由M8136决定是否有效[Y1] [默认100]
M8036	强制运行命令2	D8036	DRVI,DRVA的偏置运行速度, 由M8137决定是否有效[Y2] [默认100]
M8037	强制停止命令	D8037	保留
M8038	通讯参数设定标志	D8038	保留
M8039	恒定扫描控制	D8039	恒定扫描时间, 默认0, 以ms为单位

M元件	M元件的描述			D元件	D元件的描述
步进阶梯					
M8040	转移禁止			D8040	将S0~S899的最小动作地址号保存在D8040中，其它依次，最大的地址号保存在D8047中。
M8041	转移开始			D8041	
M8042	对应启动输入的脉冲输出			D8042	
M8043	原点回归状态的结束标志			D8043	
M8044	检测到机械原点动作			D8044	
M8045	所有输出复位禁止			D8045	
M8046	M8047动作后，当S0~S999中任何一个为ON，M8046为ON			D8046	
M8047	STL监视有效[D8040~D8047有效]			D8047	
M8048	M8049=ON，S900~S999任何一个有效，M8048有效			D8048	
M8049	信号报警有效，[D8049有效]			D8049	
中断禁止					
M8050	驱动I00□中断禁止			D8050	DRVI,DRVA的最大运行速度[默认100000]由M8135决定是否有效[Y0]
M8051	驱动I10□中断禁止			D8051	
M8052	驱动I20□中断禁止			D8052	DRVI,DRVA的最大运行速度[默认100000]由M8136决定是否有效[Y1]
M8053	驱动I30□中断禁止			D8053	
M8054	驱动I40□中断禁止			D8054	DRVI,DRVA的最大运行速度[默认100000]由M8137决定是否有效[Y2]
M8055	驱动I50□中断禁止			D8055	
M8056	驱动I6□□中断禁止			D8056	保留
M8057	驱动I7□□中断禁止			D8057	
M8058	驱动I8□□中断禁止			D8058	保留
M8059	驱动计数器中断禁止			D8059	
系统错误检测					
元件	名称	错误灯	运行		
M8060	I/O构成错误[]	OFF	RUN	D8060	I/O构成错误的未安装I/O起始地址号
M8061	PLC硬件错误	闪烁	STOP	D8061	PLC硬件错误的错误代码序号
M8062	PLC通讯错误	OFF	RUN	D8062	PLC通讯错误代码
M8063	联机错误/一般通讯错误	OFF	RUN	D8063	并行联机错误代码
M8064	参数错误	闪烁	STOP	D8064	参数错误代码
M8065	语法错误	闪烁	STOP	D8065	语法错误的代码
M8066	回路错误	闪烁	STOP	D8066	回路错误的代码
M8067	运算错误	OFF	RUN	D8067	运算错误的代码
M8068	运算错误锁存	OFF	RUN	D8068	锁存运算错误程序的步号
M8069	保留			D8069	M8065~M8067的错误发生的步号
联机功能					
M8070	联机主站驱动			D8070	并联联机错误时间，默认为500ms
M8071	联机从站驱动			D8071	保留
M8072	并联连接运行中为ON			D8072	保留
M8073	并联连接M8070/M8071设定不良			D8073	保留
采样跟踪					
M8074	保留			D8074	采样剩余次数
M8075	取样跟踪准备开始指令			D8075	采样次数的设定(1~512)
M8076	取样跟踪准备完成，执行开始指令			D8076	采样周期
M8077	取样跟踪执行中监控			D8077	触发指定
M8078	取样跟踪执行完成监控			D8078	触发条件元件地址号设定
M8079	取样跟踪超过D8075的数据			D8079	采样数据指针
M8080	保留			D8080	位元件地址号No.0
M8081	保留			D8081	位元件地址号No.1
M8082	保留			D8082	保留
M8083	保留			D8083	保留
M8084	高速计数器多段中断使能（默认OFF）（XP型不具备）			D8084	高速计数器多段中断的计数器序号（XP型不具备）
M8085	Y0端口的输出初始化标志			D8085	多段中断的数据默认为0（XP型不具备）
M8086	Y1端口的输出初始化标志			D8086	对应的D元件序号（XP型不具备）
M8087	Y2端口的输出初始化标志			D8087	保留
M8088	保留			D8088	保留
M8089	保留			D8089	保留
M8090	Y0输出完成中断使能			D8090	保留
M8091	Y1输出完成中断使能			D8091	保留
M8092	Y2输出完成中断使能			D8092	保留
M8093	保留			D8093	保留
M8094	保留			D8094	保留
M8095	Y0使用PWM功能，输出脉冲单位由ms改为us的标志			D8095	保留

M元件	M元件的描述	D元件	D元件的描述
M8096	Y1使用PWM功能, 输出脉冲单位由ms改为us的标志	D8096	字元件地址号No.0
M8097	Y2使用PWM功能, 输出脉冲单位由ms改为us的标志	D8097	字元件地址号No.1
M8098	保留	D8098	字元件地址号No.2
高速环形计数器			
M8099	高速环形计数器计数启动	D8099	[0~32767]上升动作环形计数器(0.1ms)
其它功能使用			
M8100	SPD(X000)具有-脉冲个数/分钟	D8100	保留
M8101	SPD(X001)具有-脉冲个数/分钟	D8101	单板程序版本, 如24100H2U=24, 100版本V1.00
M8102	SPD(X002)具有-脉冲个数/分钟	D8102	系统提供给用户程序的程序容量
M8103	SPD(X003)具有-脉冲个数/分钟	D8103	保留
M8104	SPD(X004)具有-脉冲个数/分钟	D8104	DRVI, DRVA执行时加速时间[默认100]由M8135决定是否有效[Y0]
M8105	SPD(X005)具有-脉冲个数/分钟	D8105	DRVI, DRVA执行时加速时间[默认100]由M8136决定是否有效[Y1]
M8106	保留	D8106	DRVI, DRVA执行时加速时间[默认100]由M8137决定是否有效[Y2]
M8107	保留	D8107	保留
M8108	保留	D8108	保留
M8109	输出刷新错误	D8109	输出刷新错误的输出地址编号
COM0 通讯链接			
M8110	保留	D8110	通讯格式, 界面配置设定, 默认为0
M8111	发送等待中 (RS指令)	D8111	站号设置, 界面配置设定, 默认为1
M8112	发送标志 (RS指令) 指令执行状态 (MODBUS)	D8112	传送剩余数据数量 (仅对RS指令)
M8113	接收完成标志 (RS) 通讯错误标志 (MODBUS)	D8113	接收到的数据数量 (仅对RS指令)
M8114	接收中 (仅对RS指令)	D8114	起始字符STX (仅对RS指令)
M8115	保留	D8115	终止字符ETX (仅对RS指令)
M8116	保留	D8116	通讯协议设定, 界面配置设定, 默认为0
M8117	保留	D8117	计算机链接协议接通要求数据起始地址号
M8118	保留	D8118	计算机链接协议接通要求发送数据数量
M8119	超时判断	D8119	通讯超时时间判断, 界面配置设定, 默认为10 (100ms)
COM1 通讯链接			
M8120	保留	D8120	通讯格式, 界面配置设定, 默认为0
M8121	发送等待中 (RS指令)	D8121	站号设置, 界面配置设定, 默认为1
M8122	发送标志 (RS指令) 指令执行状态 (MODBUS)	D8122	传送剩余数据数量 (仅对RS指令)
M8123	接收完成标志 (RS) 通讯错误标志 (MODBUS)	D8123	接收到的数据数量 (仅对RS指令)
M8124	接收中 (仅对RS指令)	D8124	起始字符STX (仅对RS指令)
M8125	保留	D8125	终止字符ETX (仅对RS指令)
M8126	为ON时485BD扩展卡有效	D8126	通讯协议设定, 界面配置设定, 默认为0 计算机链接协议接通要求数据起始地址号;
M8127	保留	D8127	SETEX协议: 从站 (PLC) 发送给主站 (显示屏) 缓冲区起始地址, 缓冲区大小为18。
M8128	保留	D8128	计算机链接协议接通要求发送数据数量; SETEX协议: 主站 (显示屏) 发送给从站 (PLC) 缓冲区起始地址, 缓冲区大小为19。
M8129	超时判断	D8129	通讯超时时间判断, 界面配置设定, 默认为10 (100ms)
高速&定位			
M8130	HSZ指令平台的控制模式	D8130	HSZ高速比较平台使用(记录号)
M8131	和M8130联合使用	D8131	HSZ&PLSY速度模型使用(记录号)
M8132	HSZ&PLSY速度模式	D8132	HSZ&PLSY速度模型频率使用
M8133	和M8132联合使用	D8133	
M8134	保留	D8134	HSZ&PLSY速度模型比较脉冲数使用
M8135	Y0在通用上加减速时间和脉冲更改有效[ON-PLSR, DRVI, DRVA, PLSY]	D8135	
M8136	Y1在通用上加减速时间和脉冲更改有效[ON-PLSR, DRVI, DRVA, PLSY]	D8136	
M8137	Y2在通用上加减速时间和脉冲更改有效[ON-PLSR, DRVI, DRVA, PLSY]	D8137	
M8138	保留	D8138	保留
M8139	保留	D8139	保留

M元件	M元件的描述	D元件	D元件的描述
M8140	ZRN的CLR信号输出功能有效	D8140	
M8141	X0~X5中断停止Y0脉冲输出使能	D8141	PLSY&PLSR输出Y000对应的脉冲个数累积值
M8142	X0~X5中断停止Y1脉冲输出使能	D8142	
M8143	X0~X5中断停止Y2脉冲输出使能	D8143	PLSY&PLSR输出Y001对应的脉冲个数累积值
M8144	保留	D8144	保留
M8145	Y000脉冲输出停止	D8145	DRVI, DRVA执行时的偏置速度[默认100]
M8146	Y001脉冲输出停止	D8146	
M8147	Y000脉冲输出监控	D8147	DRVI, DRVA执行时的最高速度[默认100, 000]
M8148	Y001脉冲输出监控	D8148	DRVI, DRVA执行时加减速时间[默认100]
M8149	Y002脉冲输出监控	D8149	保留
M8150	保留	D8150	
M8151	保留	D8151	PLSY&PLSR输出Y002对应的脉冲个数累积值
M8152	Y002脉冲输出停止	D8152	
M8153	保留	D8153	PLSY&PLSR输出Y003对应的脉冲个数累积值
M8154	保留	D8154	
M8155	保留	D8155	PLSY&PLSR输出Y004对应的脉冲个数累积值
M8156	保留	D8156	Y0端口清零信号定义(ZRN)[默认5=Y005]
M8157	保留	D8157	Y1端口清零信号定义(ZRN)[默认6=Y006]
扩展功能			
M8158	保留	D8158	Y2端口清零信号定义(ZRN)[默认7=Y007]
M8159	保留	D8159	保留
M8160	(XCH)的SWAP功能	D8160	保留
M8161	ASC/RS/ASCII/HEX/CCD的位处理模式	D8161	Y0停止输出中断信号选择
M8162	高速并联连接模式	D8162	Y1停止输出中断信号选择
M8163	保留	D8163	Y2停止输出中断信号选择
M8164	(FROM/TO)传送点数可变模式	D8164	(FROM/TO)传送点数指定模式
M8165	保留	D8165	PLSR, DRVI, DRVA执行时减速时间[默认100] [Y0]通用的由M8135决定是否有效
M8166	M8166为高速计数器禁止X端口电平复位(仅保留沿中断复位)的标志,默认不禁止。	D8166	PLSR, DRVI, DRVA执行时减速时间[默认100] [Y1]通用的由M8136决定是否有效
M8167	(HEY)HEX数据处理功能	D8167	PLSR, DRVI, DRVA执行时减速时间[默认100] [Y2]通用的由M8137决定是否有效
M8168	(SMOV)HEX数据处理功能	D8168	保留
M8169	M8169为高速计数器禁止通过X端口进行复位(无论是电平复位还是沿中断复位)的标志,默认不禁止。	D8169	保留
脉冲捕捉		通讯.链接	
M8170	X000脉冲捕捉	D8170	保留
M8171	X001脉冲捕捉	D8171	保留
M8172	X002脉冲捕捉	D8172	保留
M8173	X003脉冲捕捉	D8173	本站站号设定状态
M8174	X004脉冲捕捉	D8174	通讯子站设定状态
M8175	X005脉冲捕捉	D8175	刷新范围设定状态
M8176	保留	D8176	本站站号设定
M8177	保留	D8177	通讯子站数设定
M8178	保留	D8178	刷新范围设定
M8179	保留	D8179	重试次数设定
M8180	保留	D8180	通信超时设置
通讯.链接		变址寻址	
M8181	保留	D8181	保留
M8182	保留	D8182	位元件地址号No.2/Z1寄存器内容
M8183	数据传送主站出错	D8183	位元件地址号No.3/V1寄存器内容
M8184	数据传送从站1出错	D8184	位元件地址号No.4/Z2寄存器内容
M8185	数据传送从站2出错	D8185	位元件地址号No.5/V2寄存器内容
M8186	数据传送从站3出错	D8186	位元件地址号No.6/Z3寄存器内容
M8187	数据传送从站4出错	D8187	位元件地址号No.7/V3寄存器内容
M8188	数据传送从站5出错	D8188	位元件地址号No.8/Z4寄存器内容
M8189	数据传送从站6出错	D8189	位元件地址号No.9/V4寄存器内容
M8190	数据传送从站7出错	D8190	位元件地址号No.10/Z5寄存器内容
M8191	数据传送进行中	D8191	位元件地址号No.11/V5寄存器内容
M8192	CAN接收超时[N系列]	D8192	位元件地址号No.12/Z6寄存器内容
M8193	自由CAN指令接收状态[N系列]	D8193	位元件地址号No.13/V6寄存器内容
M8194	CAN自由指令发送失败[N系列]	D8194	位元件地址号No.14/Z7寄存器内容
M8195	C251倍频控制	D8195	位元件地址号No.15/V7寄存器内容
M8196	C252倍频控制	D8196	保留

M元件	M元件的描述	D元件	D元件的描述
M8197	C253倍频控制	D8197	保留
M8198	C254倍频控制	D8198	本机作为CANlink远程设备的标识, H2U为10224, H1U为10226
M8199	C255倍频控制	D8199 注2	扩展卡类型号, H2U为自动识别, D8199显示此卡的类型号; H1U不能自动识别, 需要通过D8199设定扩展卡类型。 扩展卡类型如下: 1 RS232扩展卡 2 RS422/485通讯扩展卡 3 AD扩展卡 4 DA扩展卡 5 6A/6B扩展卡 6 高速扩展卡 7 CAN扩展卡
	计数器增/减控制或状态		通讯.链接
M8200	C200控制	D8200	汇川软件版本号XXX.YY, XXX: 非标软件号, YY: 软件版本号。16进制下表示临时版本号
M8201	C201控制	D8201	当前连接扫描时间
M8202	C202控制	D8202	最大连接时间
M8203	C203控制	D8203	主站通讯错误次数
M8204	C204控制	D8204	从站1通讯错误次数
M8205	C205控制	D8205	从站2通讯错误次数
M8206	C206控制	D8206	从站3通讯错误次数
M8207	C207控制	D8207	从站4通讯错误次数
M8208	C208控制	D8208	从站5通讯错误次数
M8209	C209控制	D8209	从站6通讯错误次数
M8210	C210控制	D8210	从站7通讯错误次数
M8211	C211控制	D8211	主站通讯错误代码
M8212	C212控制	D8212	从站1通讯错误代码
M8213	C213控制	D8213	从站2通讯错误代码
M8214	C214控制	D8214	从站3通讯错误代码
M8215	C215控制	D8215	从站4通讯错误代码
M8216	C216控制	D8216	从站5通讯错误代码
M8217	C217控制	D8217	从站6通讯错误代码
M8218	C218控制	D8218	从站7通讯错误代码
M8219	C219控制	D8219	保留
M8220	C220控制	D8220	扩展卡发送缓冲区
M8221	C221控制	D8221	扩展卡发送缓冲区
M8222	C222控制	D8222	扩展卡发送缓冲区
M8223	C223控制	D8223	扩展卡发送缓冲区
M8224	C224控制	D8224	扩展卡发送缓冲区
M8225	C225控制	D8225	扩展卡发送缓冲区
M8226	C226控制	D8226	扩展卡发送缓冲区
M8227	C227控制	D8227	扩展卡发送缓冲区
M8228	C228控制	D8228	扩展卡发送缓冲区
M8229	C229控制	D8229	扩展卡发送缓冲区
M8230	C230控制	D8230	扩展卡接收缓冲区
M8231	C231控制	D8231	扩展卡接收缓冲区
M8232	C232控制	D8232	扩展卡接收缓冲区
M8233	C233控制	D8233	扩展卡接收缓冲区
M8234	C234控制	D8234	扩展卡接收缓冲区
M8235	C235控制	D8235	扩展卡接收缓冲区
M8236	C236控制	D8236	扩展卡接收缓冲区
M8237	C237控制	D8237	扩展卡接收缓冲区
M8238	C238控制	D8238	扩展卡接收缓冲区
M8239	C239控制	D8239	扩展卡通信错误计数器
M8240	C240控制	D8240	CAN功能设置, 参见CAN通信说明[N系列]
M8241	C241控制	D8241	CAN接收超时设定(ms) [N系列]
M8242	C242控制	D8242	CANlink地址设定/显示地址[N系列]
M8243	C243控制	D8243	CAN波特率辅助设定, 或显示拨码设定的波特率[N系列]
M8244	C244控制	D8244	CAN波特率设定[N系列]
M8245	C245控制	D8245	设定CANlink网络设备信息保存起始寄存器[N系列]
M8246	C246状态	D8246	CAN命令及状态[N系列]
M8247	C247状态	D8247	同步时钟计数器L[N系列]

M元件	M元件的描述	D元件	D元件的描述
M8248	C248状态	D8248	同步时钟计数器H[N系列]
M8249	C249状态	D8249	网络设备个数[N系列]
M8250	C250状态	D8250	CAN中断错误[N系列]
M8251	C251状态	D8251	CAN自由指令接收到的数据长度 (MCFL) [N系列]
M8252	C252状态	D8252	CAN自由指令接收到的数据MDLL[N系列]
M8253	C253状态	D8253	CAN自由指令接收到的数据MDLH[N系列]
M8254	C254状态	D8254	CAN自由指令接收到的数据MDHL[N系列]
M8255	C255状态	D8255	CAN自由指令接收到的数据MDHH[N系列]
M8256	保留	D8256	保留
M8257	保留	D8257	保留
M8258	保留	D8258	保留
M8259	保留	D8259	保留
COM2 通讯.链接			
M8260	保留	D8260	通讯格式, 界面配置设定, 默认为0
M8261	发送等待中 (RS2指令)	D8261	站号设置, 界面配置设定, 默认为1
M8262	发送标志 (RS2指令) 指令执行状态 (MODBUS2)	D8262	传送剩余数据数量 (仅对RS2指令)
M8263	接收完成标志 (RS2) 通讯错误标志 (MODBUS2)	D8263	接收到的数据数量 (仅对RS2指令)
M8264	接收中 (仅对RS2指令)	D8264	起始字符STX (仅对RS2指令)
M8265	保留	D8265	终止字符ETX (仅对RS2指令)
M8266	为ON时485BD扩展卡有效	D8266	通讯协议设定, 界面配置设定, 默认为0
M8267	保留	D8267	计算机链接协议接通要求数据起始地址号; SETEX协议: 从站 (PLC) 发送给主站 (显示屏) 缓冲区起始地址, 缓冲区大小为18。
M8268	保留	D8268	计算机链接协议接通要求发送数据数量; SETEX协议: 主站 (显示屏) 发送给从站 (PLC) 缓冲区起始地址, 缓冲区大小为19。
M8269	超时判断	D8269	通讯超时时间判断, 界面配置设定, 默认为10 (100ms)
COM3 通讯.链接			
M8270	保留	D8270	通讯格式, 界面配置设定, 默认为0
M8271	发送等待中 (RS2指令)	D8271	站号设置, 界面配置设定, 默认为1
M8272	发送标志 (RS2指令) 指令执行状态 (MODBUS2)	D8272	传送剩余数据数量 (仅对RS2指令)
M8273	接收完成标志 (RS2) 通讯错误标志 (MODBUS2)	D8273	接收到的数据数量 (仅对RS2指令)
M8274	接收中 (仅对RS2指令)	D8274	起始字符STX (仅对RS2指令)
M8275	保留	D8275	终止字符ETX (仅对RS2指令)
M8276	为ON时485BD扩展卡有效	D8276	通讯协议设定, 界面配置设定, 默认为0
M8277	保留	D8277	计算机链接协议接通要求数据起始地址号; SETEX协议: 从站 (PLC) 发送给主站 (显示屏) 缓冲区起始地址, 缓冲区大小为18。
M8278	保留	D8278	计算机链接协议接通要求发送数据数量; SETEX协议: 主站 (显示屏) 发送给从站 (PLC) 缓冲区起始地址, 缓冲区大小为19。
M8279	超时判断	D8279	通讯超时时间判断, 界面配置设定, 默认为10 (100ms)
CAN 系统特殊软元件			
M8280	CAN协议标志	D8280	当前软件CAN协议版本号 (默认300)
M8281	没有识别到CAN扩展卡, CAN-LINK功能不能使用	D8281	扩展卡类型号, H2U为自动识别, D8280显示此卡的类型号; H1U不能自动识别, 需要通过D8281设定扩展卡类型。 扩展卡类型如下: 1 RS232扩展卡 2 RS422/485通讯扩展卡 3 AD扩展卡 4 DA扩展卡 5 6A/6B/3A扩展卡 6 高速扩展卡 7 CAN扩展卡
M8282	M8282=0: CANLink协议; M8282=1: CAN自由协议;	D8282	保留
M8283	保留	D8283	保留
M8284	0: 拨码设定CAN_LINK地址, D8284显示地址 1: D8284设定CAN_LINK地址	D8284	CAN_LINK地址设定/显示地址

M元件	M元件的描述	D元件	D元件的描述
M8285	0:拨码设定波特率, D8285显示波特率 1:由M8286决定波特率设置	D8285	波特率设定, 或显示拨码及档位设定的波特率
M8286	0:波特率分档设定,由D8286分档设定, D8285显示波特率 1:波特率由D8285,D8286组成的32位寄存器自由设定	D8286	M8285=1, M8286 = 0时 D8286 <10的时候 D8286 = 其它值, 800K D8286 = 6, 600K D8286 = 5, 500K D8286 = 4, 400K D8286 = 3, 300K D8286 = 2, 200K D8286 = 1, 100K D8286 = 0, 10K D8286 >=10的时候 D8286 = 10, 10K D8286 = 20, 20K D8286 = 50, 50K D8286 = 125, 125K D8286 = 250, 250K
M8287	允许保存网络设备信息	D8287	M8287 1, 设定网络设备信息保存起始 元件编号。
M8288	保留	D8288	保留
M8289	重新配置CAN-LINK网络	D8289	保留
M8290	网络启停控制元件	D8290	保留
M8291	同步发送触发元件	D8291	网络设备个数
M8292	0:自由CAN指令接收完成或空闲 1:自由CAN指令接收状态	D8292	CANRX接收到的数据长度 (MCFL)
M8293	CAN自由指令发送失败	D8293	CAN中断寄存器
M8294	CAN自由指令接收成功	D8294	CAN状态寄存器
M8295	CAN状态异常	D8295	CANRX接收数据MDL低16位
M8296	设备地址错误	D8296	CANRX接收数据MDL高16位
M8297	保留	D8297	CANRX接收数据MDH低16位
M8298	M8298=0: CANTX空闲; M8298=1: CANTX正在发送数据;	D8298	CANRX接收数据MDH高16位
M8299	CAN接收超时	D8299	CAN接收超时设定(ms)
M8300	设备通信中断	D8300	设备编号
M8301	网络过载	D8301	本站故障信息 (PLC故障代码:)
M8302	设备发送非法数据 (非CAN-LINK协议数据)	D8302	设备编号
M8303	设备没有正确执行配置信息	D8303	设备编号
M8304	协议版本错误	D8304	设备编号
M8305	网络中有设备告警 (用户程序复位)	D8305	网络告警信息
M8306	启用详细告警记录	D8306	若启用详细告警记录: 设定保存告警/错误起始 元件编号。从该编号开始的64个D元件, 将用来存放0-63号设备的告警信息 (告警码)
M8307	保留	D8307	
M8308	保留	D8308	
M8309	H2UXP 2416/3624点数由默认掉电保存区不可更改改为可以更改的标志	D8309	
M8310	N:N协议数据传送从站8出错	D8310	
M8311	N:N协议数据传送从站9出错	D8311	
M8312	N:N协议数据传送从站10出错	D8312	
M8313	N:N协议数据传送从站11出错	D8313	
M8314	N:N协议数据传送从站12出错	D8314	
M8315	N:N协议数据传送从站13出错	D8315	
M8316	N:N协议数据传送从站14出错	D8316	
M8317	N:N协议数据传送从站15出错	D8317	
M8318	保留	D8318	
M8319	保留	D8319	
M8320	保留	D8320	保留
~		~	
M8419	保留	D8419	保留
高速输入输出			
M8420	保留。	D8420	
M8421	保留。	D8421	
M8422	保留。	D8422	

M元件	M元件的描述	D元件	D元件的描述
M8423	保留。	D8423	
M8424	保留。	D8424	
M8425	保留。	D8425	
M8426	保留。	D8426	
M8427	保留。	D8427	
M8428		D8428	
M8429	DSZR指令执行异常结束标志位	D8429	
M8430		D8430	3G模块与PLC心跳维持（每10s向寄存器写入固定值，用于实现防拆卸功能）---已取消
M8431		D8431	3G模块与服务器心跳维持
M8432		D8432	3G模式下信号强度。信号强度低于10时报警灯常亮，信号强度为0时（某一次没有采集到）只记入日志但不写入autoshop，查看D8432应显示上一次的数值，不能写入0
M8433		D8433	登录的注册码错误或重复时，服务器返回的数值写入
M8434		D8434	网络状态，没有插3G卡也没有以太网连接情况下写入
M8435		D8435	以太网卡版本号
M8436		D8436	以太网IP地址
M8437		D8437	
M8438		D8438	
M8439		D8439	
DSZR指令 Y0脉冲输出端口			
M8440	脉冲输出中监控	D8440	保留
M8441	清零信号输出功能有效	D8441	
M8442	原点回归方向指定	D8442	基底速度 (HZ) [默认200]
M8443	正转极限	D8443	最高速度 (HZ) (32位)
M8444	反转极限	D8444	[默认100000]
M8445	近点信号逻辑反转	D8445	爬行速度 (HZ) [默认1000]
M8446	零点信号逻辑反转	D8446	原点回归速度 (HZ) (32位)
M8447	保留	D8447	[默认50000]
M8448	定位指令驱动中	D8448	加速时间 (ms) [默认100]
M8449	脉冲停止指令	D8449	减速时间 (ms) [默认100]
DSZR指令 Y1脉冲输出端口			
M8450	脉冲输出中监控	D8450	当前值寄存器 (HZ) (32位)
M8451	清零信号输出功能有效	D8451	
M8452	原点回归方向指定	D8452	基底速度 (HZ) [默认200]
M8453	正转极限	D8453	最高速度 (HZ) (32位)
M8454	反转极限	D8454	[默认100000]
M8455	近点信号逻辑反转	D8455	爬行速度 (HZ) [默认1000]
M8456	零点信号逻辑反转	D8456	原点回归速度 (HZ) (32位)
M8457	保留	D8457	[默认50000]
M8458	定位指令驱动中	D8458	加速时间 (ms) [默认100]
M8459	脉冲停止指令	D8459	减速时间 (ms) [默认100]
DSZR指令 Y2脉冲输出端口			
M8460	脉冲输出中监控	D8460	当前值寄存器 (HZ) (32位)
M8461	清零信号输出功能有效	D8461	
M8462	原点回归方向指定	D8462	基底速度 (HZ) [默认200]
M8463	正转极限	D8463	最高速度 (HZ) (32位)
M8464	反转极限	D8464	[默认100000]
M8465	近点信号逻辑反转	D8465	爬行速度 (HZ) [默认1000]
M8466	零点信号逻辑反转	D8466	原点回归速度 (HZ) (32位)
M8467	保留	D8467	[默认50000]
M8468	定位指令驱动中	D8468	加速时间 (ms) [默认100]
M8469	脉冲停止指令	D8469	减速时间 (ms) [默认100]
M8470		D8470	
M8471		D8471	
M8472		D8472	
M8473		D8473	
M8474		D8474	
M8475		D8475	
M8476		D8476	
M8477		D8477	
M8478		D8478	
M8479		D8479	

M元件	M元件的描述	D元件	D元件的描述
M8480		D8480	
M8481		D8481	
M8482		D8482	
M8483		D8483	
M8484		D8484	
M8485		D8485	
M8486		D8486	
M8487		D8487	
M8488			
M8489			
系统错误日志记录功能			
M8490	MEAN指令求和标志位	D8490	错误1发生时间 [小时]
M8491	N:N协议校验取消标志 (默认OFF: 有校验, ON: 取消校验, 兼容三菱的通讯格式)	D8491	错误2发生时间 [小时]
M8492		D8492	错误3发生时间 [小时]
M8493		D8493	错误4发生时间 [小时]
M8494		D8494	错误5发生时间 [小时]
M8495		D8495	错误6发生时间 [小时]
M8496		D8496	错误7发生时间 [小时]
M8497		D8497	错误8发生时间 [小时]
M8498		D8498	错误9发生时间 [小时]
M8499		D8499	错误10发生时间 [小时]
M8500		D8500	错误1错误代码
M8501		D8501	错误2错误代码
M8502		D8502	错误3错误代码
M8503		D8503	错误4错误代码
M8504		D8504	错误5错误代码
M8505		D8505	错误6错误代码
M8506		D8506	错误7错误代码
M8507		D8507	错误8错误代码
M8508		D8508	错误9错误代码
M8509		D8509	错误10错误代码
M8510		D8510	当前系统已运行时间 [单位: 秒]
M8511		D8511	

8.2 系统错误记录功能

8.2.1 错误代码及内容

特殊数据寄存器D8060~D8067，存储的错误代码和内容如下表所示。

类型	出错代码	出错内容	处理方法	
IO范围或 设置出错 M8060(D8060) 继续运行	例1020	没有装I/O起始元件号“1020”时：1=输X(0=输出Y)，020=元件号	还没有装的输入继电器，输出继电器的编号被编入程序。可编程控制器可以继续运行，若是程序员，请进行修改。	
		错误码		错误表示的内容
		1000-1377		X输入信号有误，序号错误或超限
		0000-0377		Y输出信号有误，序号错误或超限
PC硬件出错 M8061(D8061) 停止运行	0000	无异常	检查扩展电线的连接是否正确。	
	6101	RAM出错		
	6102	运算回路错误		
	6103	IO硬件连接错误	运算时间超过D8000的值，检查程序。	
	6104	系统外部24V错误		
	6105	系统监视器错误	PLC重新上下电	
	6106	系统FLASH读写错误		
	6107	系统IO设定错误		
	6108	读取FLASH中FPGA数据次数超限		
6109	FLASH中无FPGA数据	使用Autoshop下载FPGA数据包		
PC/PP通信出错 M8062(D8062) 继续运行	0000	无异常	程序面板(PP)或程序连口连接的设备与可编程控制器(PC)间的连接是否正确。	
	6201	奇偶出错；超过出错；成帧出错		
	6202	通信字符有误		
	6203	通信数据的求和不一致		
	6204	数据格式有误		
	6205	指令有误		
并行连接 通信出错 M8063(D8063) 继续运行	0000	无异常	检查双方的可编程控制器的电源是否为ON，适配器和控制器之间，以及适配器之间连接是否正确。	
	6301	奇偶校验错误，溢出错误，帧错误		
	6302	通讯字符错误		
	6303	通讯数据校验和不一致		
	6304	数据格式错误		
	6305	指令错误		
	6306	监视定时器超时		
	6307~6311	无		
	6312	并联(1:1)协议字符错误		
	6313	并联(1:1)协议校验和错误		
	6314	并联(1:1)协议格式错误		
6315	并联(1:1)协议通讯超时错误			

类型	出错代码	出错内容	处理方法
并行连接 通信出错 M8063(D8063) 继续运行	6316-6329	保留	COM0通讯出错 请检查COM0的通讯电缆是否正确连接。
	6330+10*X	MODBUS从站地址设置错误, 地址大于247;	
	6331+10*X	数据帧长度错误, 帧长度不符合要求, 或帧长小于5;	
	6332+10*X	地址错误, 标准错误帧; 或收发地址不一致;	
	6333+10*X	CRC检验错误	
	6334+10*X	不支持的命令码, 标准错误帧; 或收发命令不一致; 或不支持的命令;	
	6335+10*X	接收超时	
	6336+10*X	数据错误, 标准错误帧;	
	6337+10*X	保留	
	6338+10*X	帧错误, 标准错误帧;	
	6339+10*X	串口协议错误, 使用modbus指令或RS指令时, 未定义成对应的协议;	COM1通讯出错 请检查COM1的通讯电缆是否正确连接; 检查通讯双方通讯格式是否匹配。
	6340	MODBUS从站地址设置错误	
	6341	数据帧长度错误	
	6342	地址错误	
	6343	CRC检验错误	
	6344	不支持的命令码	
	6345	接收超时	
	6346	数据错误	
	6347	缓冲区溢出	COM2通讯出错 请检查COM2的通讯电缆是否正确连接; 检查通讯双方通讯格式是否匹配。
	6348	帧错误	
	6349	串口协议错误	
	6350	MODBUS从站地址设置错误	
	6351	数据帧长度错误	
	6352	地址错误	
	6353	CRC检验错误	
	6354	不支持的命令码	
	6355	接收超时	
	6356	数据错误	
	6357	缓冲区溢出	COM3通讯出错。 请检查COM3的通讯电缆是否正确连接; 检查通讯双方通讯格式是否匹配。
	6358	帧错误	
	6359	串口协议错误	
	6360	MODBUS从站地址设置错误	
	6361	数据帧长度错误	
	6362	地址错误	
	6363	CRC检验错误	
	6364	不支持的命令码	
	6365	接收超时	
6366	数据错误		
6367	缓冲区溢出		
6368	帧错误		
6369	串口协议错误		
6370	保留		
6371	奇偶数出错, 超出出错, 帧出错		
6372	通讯字符错误		
6373	通讯数据校验和不一致		
6374	数据格式错误		
6375	指令错误		
6376	通讯元件范围超限		
6377	通讯端口超限或不存在		

类型	出错代码	出错内容	处理方法
系统参数 设置错误 M8064(D8064) 停止运行	0000	无异常	停止可编程控制器的运行，用参数方式 设定正确值。
	6401	程序和参数不一致	
	6402	程序容量设定不良	
	6403	可变更软元件掉电保持区域设定不良	
	6404	参数区设定不良	
	6405	程序区设定不良	
	6406-6424	保留	
	6425	用户程序校验错误，下载数据有误	
	6426	用户程序，包括运动控制子程序，不完整	
	6427	PLC标识符和用户程序中的标识符不匹配	
	6428	保留	
	6429	程序授权保护标志错误，请确认该标志和用户程序中的程序授权保护标志是否匹配	
	6430-6452	保留	
6453-6465	可变更软元件掉电保持区域设定不良		
用户程序 语法出错 M8065(D8065) 停止运行	0000	无异常	检查编程时对各个指令的使用是否正确？ 产生错误时请用程序模式进行修改。
	6501	保留	
	6502	保留	
	6503	指令参数错误	
	6504	标号定义重复	
	6505	保留	
	6506	使用未定义指令	
	6507	标号P定义不良	
	6508	标号I定义不良	
	6509	保留	
	6510	保留	
6511	高速计数器与中断使用的高速输入端口重复		
用户程序逻辑 回路出错 M8066(D8066) 停止运行	0000	无异常	对整个电路而言，当指令组合不对时， 对指令关系有错时都能产生错误。在程 序中要修改指令的相互关系，使之正确 无误。
	6601~6604	保留	
	6605	在STL中有使用错误的指令	
	6606	有不正确的指令在不正确的位置	
	6607	FOR-NEXT操作错误	
	6608	MC-MCR操作错误	
	6609-6617	保留	
	6618	只能在主程序中使用的指令在其它地方出现	
	6619	在FOR_NEXT之间使用了不能用的指令，如STL、RET、MC、MCR等	
	6620	FOR_NEXT的嵌套超出	
	6621	FOR_NEXT的数量关系不良	
	6622	无NEXT指令	
	6623	无MC指令	
	6624	无MCR指令	
	6625	STL连续使用9次以上	
	6626	STL-RET中有无法使用的指令	
	6627	无RET指令	
	6628	在主程序中无法使用的指令	
	6629	无P或I	
6630	无SRET和IRET指令		
6631	SRET处于无法使用的位置		
6632	FEND处于无法使用的位置		

类型	出错代码	出错内容	处理方法	
指令参数错误& 运行参数错误 M8067 (D8067) 继续运行	0000	没有异常	运算过程中产生错误，以及程序的修改或应用指令的操作数的内容是否有错误。即使语法、电路没有出错，上述原因也可能产生运算错误。（例）T200Z虽没有错但运算结果Z=100时，T=300，这样，元件编号则溢出。	
	6701	CALL&CJ的调用错误		
	6702	CALL运行次数6次以上		
	6704	保留		
	6705	通讯参数区设定不良		
	6706	不存在的元件或元件范围超界		
	6707	数据不合理或超范围		
	6708	FOR&NEXT,MC,MCR,STL,子程序,中断程序关系混乱		
	6709	FROM或TO指令错误		
	6710	主程序中使用了局部变量		
	6711	指令中使用的软元件重复使用错误或使用冲突		
	6712	系统中使用了没有定义的中断		
	6720	CALL指令SRET不配套		
	6721-6729	保留		
	6730	采样时间TS<0;	PID运算停止	产生控制参数的设定值和PID运算中产生数据错误。请检查参数。
	6731	保留		
	6732	输入滤波常数对象不正常;		
	6733	输入比例系数不正常;		
	6734	积分时间不正常;		
	6735	微分增益不正常;		
	6736	微分时间不正常;		
	6737-6739	保留	将运算数据作MAX值，继续运算。	
	6740	采样时间不正常;		
	6741	保留		
	6742	测定值变量溢出		
	6743	偏差不正常;		
	6744	积分项不正常;		
	6745	因微分阀(KP)溢出导致微分值溢出		
	6746	微分项不正常;		
	6747	PID结果不正常;		
	6748	保留		
6749	保留	检查对应的指令，按照错误内容调整指令。		
6750	访问的扩展模块不存在，或模块序号超限			
6751-6759	保留			
6760	高速输入指令运行条数超数量;			
6761-6769	保留			
6770	高速输出指令使用端口重复或冲突			
6771	高速输出指令信号有误			

8

8.2.2 PLC 运行时间 / 异常记录

附录

PLC运行时间/异常设计主要用于记录PLC的运行时间和记录出错信息。方便客户和我们分析PLC在运行过程中出现的问题。

时间记录：寄存器D8511，D8510（32位），时间单位：秒。

错误记录：

错误时间(小时)	错误内容
D8490	D8500
D8491	D8501
D8492	D8502
D8493	D8503
D8494	D8504
D8495	D8505
D8496	D8506

错误时间(小时)	错误内容
D8497	D8507
D8498	D8508
D8499	D8509

记录原则:

- 1) 当D8060--D8067有报告错代码时记录下该错误。
- 2) 出现的错误代码已经在错误记录表中，只更新报错时间，在报错表中的位置不改变。
- 3) 当出错代码记录超过10个时，采用先进先出的原则丢弃原来最先出现的错误代码。
- 4) 用户清除程序，元件存储空间都不能清除错误记录和运行时间。

8.2.3 使用 Autoshop 进行故障诊断和处理

当用户使用Autoshop监控PLC时，可以通过软件界面右下角查看当前PLC运行状态，包括PLC的RUN/STOP状态、故障状态、PLC当前版本、当前扫描时间。



双击故障状态 **故障状态** 即可查看PLC当前错误信息，如下图所示。



当前错误详细信息可通过双击错误信息显示区域或选中后点击错误帮助，弹出错误帮助对话框，如下图所示。



8.3 错误代码寄存器

按下述定时检查，把前项的出错代码存入特殊数据寄存器D8060~D8067。错误代码存储寄存器。

出错项目	电源OFF→ON	电源ON后初次STOP→RUN时	其它
M8060 I/O地址号构成出错	检查	检查	运算中
M8061 PC硬件出错	检查	-	运算中
M8062 PC/PP通信出错	-	-	从PP接收信号时
M8063 连接模块通信出错	-	-	从对方接受信号时
M8064 参数出错 M8065 语法出错 M8066 电路出错	检查	检查	程序变更时(STOP) 程序传送时(STOP)
M8067 运算出错 M8068 运算出错锁存	-	-	运算中(RUN)

D8060~D8067各存一个出错内容，同一出错项目产生多次出错时，每当消除出错原因时，仍存储发生中的出错代码。无出错时存入“0”。

8.4 H2U系列MTQ型与MT型的软件差别

H2u系列的MTQ型PLC，具有6路高速输入，5路高速输出，高速信号处理频率提升，以满足具有多路高速应用的需求。

硬件响应特性比较：

输入端口	H1U/H2U-XP机型	MTQ型
X0	60kHz	100kHz
X1	60kHz	100kHz
X2	10kHz	100kHz
X3	10kHz	100kHz
X4	10kHz	100kHz
X5	10kHz	100kHz
其它X端口	指标相同	
Y0	100kHz	100kHz
Y1	100kHz	100kHz
Y2	100kHz	100kHz
Y3	—	100kHz
Y4	—	100kHz
其它Y端口	指标相同	

AB相高速计数器最高频率指标比较：

AB相计数器	H1U/H2U-XP机型	H2u-3232MTQ
C251~C255	30kHz	50kHz

支持指令比较：

指令	H1U/H2U-XP机型	MTQ支持的输出端口
PLSY	Y0、Y1、Y2	Y0、Y1、Y2、Y3、Y4
PLSR	Y0、Y1、Y2	Y0、Y1、Y2、Y3、Y4
PLSV	Y0、Y1、Y2	Y0、Y1、Y2、Y3、Y4
PWM	Y0、Y1、Y2	Y0、Y1、Y2、Y3、Y4
DRVI	Y0、Y1、Y2	Y0、Y1、Y2、Y3、Y4
DRVA	Y0、Y1、Y2	Y0、Y1、Y2、Y3、Y4
ZRN	Y0、Y1、Y2	Y0、Y1、Y2、Y3、Y4

MT/MTQ型高速脉冲输出时涉及的系统监控寄存器说明:

M8145	Y000脉冲输出停止	D8145	DRVI, DRVA执行时的偏置速度
M8146	Y001脉冲输出停止	D8146	DRVI, DRVA执行时的最高速度[默认100, 000]
M8147	Y000脉冲输出监控	D8147	
M8148	Y001脉冲输出监控	D8148	DRVI, DRVA执行时加减速时间[默认100]
M8149	Y002脉冲输出监控	D8149	保留
M8150	Y003脉冲输出监控	D8150	PLSY&PLSR输出Y002对应的脉冲个数累积值
M8151	Y004脉冲输出监控	D8151	
M8152	Y002脉冲输出停止	D8152	PLSY&PLSR输出Y003对应的脉冲个数累积值
M8153	Y003脉冲输出停止	D8153	
M8154	Y004脉冲输出停止	D8154	PLSY&PLSR输出Y004对应的脉冲个数累积值
M8155	保留	D8155	
M8156	保留	D8156	Y0端口清零信号定义(ZRN)[默认5=Y005]
M8157	保留	D8157	Y1端口清零信号定义(ZRN)[默认6=Y006]
M8158	保留	D8158	Y2端口清零信号定义(ZRN)[默认7=Y007]
M8159	保留	D8159	Y3端口清零信号定义(ZRN)[默认8=Y010]
M8160	(XCH)的SWAP功能	D8160	Y4端口清零信号定义(ZRN)[默认9=Y011]
M8161	ASC/RS/ASCII/HEX/CCD的位处理模式	D8161	保留
M8162	高速并联连接模式	D8162	保留
M8163	保留	D8163	保留
M8164	(FROM/TO)传送点数可变模式	D8164	(FROM/TO)传送点数指定模式
M8165	保留	D8165	PLSR, DRVI, DRVA执行时减速时间[默认100]由M8135决定是否有效[Y0]
M8166	保留	D8166	PLSR, DRVI, DRVA执行时减速时间[默认100]由M8136决定是否有效[Y1]
M8167	(HEY)HEX数据处理功能	D8167	PLSR, DRVI, DRVA执行时减速时间[默认100]由M8137决定是否有效[Y2]

8.5 H2U系列MTP型软件补充说明

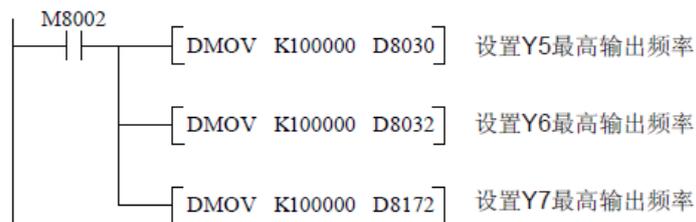
8.5.1 DRVI、DRVA 特殊功能的描述

端口	特殊功能有效		特殊功能无效		
	有效条件		有效条件		
Y000	有效条件	M8135=ON	有效条件	M8135=OFF	
	加速时间	D8104 (默认100ms)	加速时间	D8148 (默认100ms)	
	减速时间	D8165 (默认100ms)	减速时间	D8148 (默认100ms)	
	偏置速度	D8034	偏置速度	D8145	
	最大速度	D8051/D8050 (默认100000)	最大速度	D8146/D8147 (默认100000)	
	运行中改脉冲个数有效		运行中改脉冲个数无效		
	通用功能指定				
	输出停止控制	M8145			
	输出监控	M8147			
	脉冲重新输出	M8085			
	累计脉冲计数	D8140、D8141			
	Y0&Y1累计脉冲	D8136、D8137			
	Y001	有效条件	M8136=ON	有效条件	M8136=OFF
加速时间		D8105 (默认100ms)	加速时间	D8148 (默认100ms)	
减速时间		D8166 (默认100ms)	减速时间	D8148 (默认100ms)	
偏置速度		D8035	偏置速度	D8145	
最大速度		D8053/D8052 (默认100000)	最大速度	D8146/D8147 (默认100000)	
运行中改脉冲个数有效		运行中改脉冲个数无效			
通用功能指定					
输出停止控制		M8146			
输出监控		M8148			
脉冲重新输出		M8086			
累计脉冲计数		D8142、D8143			
Y0&Y1累计脉冲		D8136、D8137			
Y002		有效条件	M8137=ON	有效条件	M8137=OFF
	加速时间	D8106 (默认100ms)	加速时间	D8148 (默认100ms)	
	减速时间	D8167 (默认100ms)	减速时间	D8148 (默认100ms)	
	偏置速度	D8036	偏置速度	D8145	
	最大速度	D8055/D8054 (默认100000)	最大速度	D8146/D8147 (默认100000)	
	运行中改脉冲个数有效		运行中改脉冲个数无效		
	通用功能指定				
	输出停止控制	M8152			
	输出监控	M8149			
	脉冲重新输出	M8087			
	累计脉冲计数	D8150、D8151			
	Y003	有效条件	M8138=ON	有效条件	M8138=OFF
		加速时间	D8107 (默认100ms)	加速时间	D8148 (默认100ms)
减速时间		D8168 (默认100ms)	减速时间	D8148 (默认100ms)	
偏置速度		D8037	偏置速度	D8145	
最大速度		D8057/D8056 (默认100000)	最大速度	D8146/D8147 (默认100000)	
运行中改脉冲个数有效		运行中改脉冲个数无效			
通用功能指定					
输出停止控制		M8153			
输出监控		M8150			
脉冲重新输出		M8088			
累计脉冲计数		D8152、D8153			

端口	特殊功能有效		特殊功能无效	
Y004	有效条件	M8139=ON	有效条件	M8139=OFF
	加速时间	D8108 (默认100ms)	加速时间	D8148 (默认100ms)
	减速时间	D8169 (默认100ms)	减速时间	D8148 (默认100ms)
	偏置速度	D8038	偏置速度	D8145
	最大速度	D8059/D8058 (默认100000)	最大速度	D8146/D8147 (默认100000)
	运行中改脉冲个数有效		运行中改脉冲个数无效	
	通用功能指定			
	输出停止控制	M8154		
	输出监控	M8151		
	脉冲重新输出	M8089		
	累计脉冲计数	D8154、D8155		
Y005	有效条件	M8141=ON	有效条件	M8141=OFF
	加速时间	D8087 (默认100ms)	加速时间	D8148 (默认100ms)
	减速时间	D8090 (默认100ms)	减速时间	D8148 (默认100ms)
	偏置速度	D8093	偏置速度	D8145
	最大速度	D8030/D8031 (需设置100000) 见备注	最大速度	D8146/D8147 (默认100000)
	运行中改脉冲个数有效		运行中改脉冲个数无效	
	通用功能指定			
	输出停止控制	M8155		
	输出监控	M8144		
	脉冲重新输出	M8090		
	累计脉冲计数	D8138、D8139		
Y006	有效条件	M8142=ON	有效条件	M8142=OFF
	加速时间	D8088 (默认100ms)	加速时间	D8148 (默认100ms)
	减速时间	D8091 (默认100ms)	减速时间	D8148 (默认100ms)
	偏置速度	D8094	偏置速度	D8145
	最大速度	D8032/D8033 (需设置100000) 见备注	最大速度	D8146/D8147 (默认100000)
	运行中改脉冲个数有效		运行中改脉冲个数无效	
	通用功能指定			
	输出停止控制	M8156		
	输出监控	M8158		
	脉冲重新输出	M8091		
	累计脉冲计数	D8162、D8163		
Y007	有效条件	M8143=ON	有效条件	M8143=OFF
	加速时间	D8089 (默认100ms)	加速时间	D8148 (默认100ms)
	减速时间	D8092 (默认100ms)	减速时间	D8148 (默认100ms)
	偏置速度	D8095	偏置速度	D8145
	最大速度	D8172/D8173 (需设置100000) 见备注	最大速度	D8146/D8147 (默认100000)
	运行中改脉冲个数有效		运行中改脉冲个数无效	
	通用功能指定			
	输出停止控制	M8157		
	输出监控	M8159		
	脉冲重新输出	M8092		
	累计脉冲计数	D8170、D8171		

备注:

Y5 (M8141)、Y6 (M8142)、Y7 (M8143) 有效条件为ON时, 最大速度需设置, 上电请先执行下面指令, 否则达不到最高速度。



8.5.2 ZRN 指令的描述

端口	特性	对应的软元件
Y000	清除端口使能	M8140=ON
	清除端口号	D8156[默认Y010]
	输出停止控制	M8145
	输出监控	M8147
	累计脉冲计数	D8140、D8141
	Y0&Y1累计脉冲	D8136、D8137
Y001	清除端口使能	M8140=ON
	清除端口号	D8157[默认Y011]
	输出停止控制	M8146
	输出监控	M8148
	累计脉冲计数	D8142、D8143
	Y0&Y1累计脉冲	D8136、D8137
Y002	清除端口使能	M8140=ON
	清除端口号	D8158[默认Y012]
	输出停止控制	M8152
	输出监控	M8149
	累计脉冲计数	D8150、D8151
Y003	清除端口使能	M8140=ON
	清除端口号	D8159[默认Y013]
	输出停止控制	M8153
	输出监控	M8150
	累计脉冲计数	D8152、D8153
Y004	清除端口使能	M8140=ON
	清除端口号	D8160[默认Y014]
	输出停止控制	M8154
	输出监控	M8151
	累计脉冲计数	D8154、D8155
Y005	清除端口使能	M8140=ON
	清除端口号	D8161[默认Y015]
	输出停止控制	M8155
	输出监控	M8144
	累计脉冲计数	D8138、D8139
Y006	清除端口使能	M8140=ON
	清除端口号	D8196[默认Y016]
	输出停止控制	M8156
	输出监控	M8158
	累计脉冲计数	D8162、D8163

端口	特性	对应的软元件
Y007	清除端口使能	M8140=ON
	清除端口号	D8197[默认Y017]
	输出停止控制	M8157
	输出监控	M8159
	累计脉冲计数	D8170、D8171

注意：

H_{2U}-3232MTP是8路高速输出（每路最高100K，互不影响）；

不支持高速计数、外部I/O中断、脉冲捕捉、SPD指令；

PLSR指令可以分别单独设定加减速时间和脉冲个数可在运行中更改；

PLSY指令脉冲个数可在运行中更改；

Y0~Y7不可以作为脉冲输出方向的信号使用。

8.6 H2U系列MTQ/MTP机型不支持的指令一览

分类	指令助记符	功能	是否支持H2U系列MTQ/MTP机型
外设设备	CANTX	CAN数据发送	否
	CANRX	CAN数据接收	否
	RS2	串行数据传送2	否
	MODBUS2	数据传送2	否
浮点数	ECMP	2进制浮点数比较	否
	EZCP	2进制浮点数区间比较	否
	EMOV	2进制浮点数数据移动	否
	EBCD	2进制-10进制浮点数转换	否
	EBIN	10进制-2进制浮点数转换	否
	EADD	2进制浮点数加法	否
	ESUB	2进制浮点数减法	否
	EMUL	2进制浮点数乘法	否
	EDIV	2进制浮点数除法	否
	EXP	2进制浮点数指数运算	否
	LOGE	2进制浮点数自然对数运算	否
	DLOG	2进制浮点数常用对数运算	否
	ESQR	2进制浮点数开方	否
	SIN	浮点数SIN运算	否
	COS	浮点数COS运算	否
	TAN	浮点数TAN运算	否
	ASIN	2进制浮点数ASIN运算	否
	ACOS	2进制浮点数ACOS运算	否
	ATAN	2进制浮点数ATAN运算	否
	RAD	2进制浮点数角度→弧度	否
DEG	2进制浮点数弧度→角	否	
SINH	2进制浮点数SINH运算	否	
COSH	2进制浮点数COSH运算	否	
TANH	2进制浮点数TANH运算	否	
定位	DSZR	带DOG搜索的原点回归	否
	ABS	ABS位置数读取	否

8.7 H2U系列高速处理指令的增强功能说明

H2U系列PLC高速编程指令除了能兼容FX1n/FX2n的指令之外，对其中部分指令功能作了加强，这些加强功能一般通过系统的M变量来配合实现。涉及的高速指令功能如下：

指令	配合使用的M变量	M标志为ON时指令的加强功能
SPD	M8100~M8105分别对应X0~X5端口	新增1分钟脉冲个数（运行速度）计算，该数据是通过实时采样输入端脉冲频率再通过内部运算得出的。
PLSY	M8135~M8137分别对应Y0~Y2	可以在运行中更改输出脉冲个数。
	M8085~M8087分别对应Y0~Y2	在驱动特殊位为ON，可以马上启动下条脉冲输出指令，不需要上条能流无效的处理。
	M8090~M8092分别对应Y0~Y2	可以实现脉冲输出完成后产生一次用户中断。
PLSR DRVI DRVA	M8135~M8137分别对应Y0~Y2	可以在运行中更改输出脉冲个数； 可以更改减速时间（使加速时间不同）。
	M8085~M8087分别对应Y0~Y2	在驱动特殊位为ON，可以马上启动下条脉冲输出指令，不需要上条指令能流无效的处理。
	M8090~M8092分别对应Y0~Y2	可以实现脉冲输出完成后产生一次用户中断。
高速计速器 中断	M8084；D8084~D8086	高速计速器多用户中断（XP不具备此功能）（最大支持24个：I507~I530）

涉及的加强高速指令使用说明如下：

1) SPD指令的描述

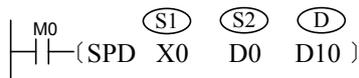
16bit	32bit	<input type="checkbox"/>	FNC56	SPD	脉冲密度，转速，线速度
✓					
7步			指令格式：SPD (S1) (S2) (D)		

X000~X005分别使用了功能使能标志M8100~M8105作为增强功能的生效标志，每个标志可单独设置。

a) 如果功能使能标志为[M8100~M8105]OFF，SPD为基本功能：

- (S1)：脉冲信号输入端口，只能为X0~X5；
- (S2)：为设定的脉冲检测时间长度(ms)1~32767；
- (D)+0：为在S2时间内的脉冲个数，为16位的数据；
- (D)+1：本次时间段对脉冲的计数。
- (D)+2：用于测定剩余时间(ms)。

指令举例：



X0为脉冲信号输入端口；

D0为单位时间(ms)；

D10为D0时间段总脉冲个数；

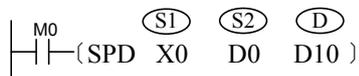
D11为D0时间段正在计的数据；

D12为时间段剩下的时间(MS)；

b) 如果功能使能标志为[M8100~M8105]ON，SPD为加强功能：（XP型不具备）

- (S1) S1：脉冲信号输入端口，只能为X0~X5；
- (S2) S2：为设定的脉冲检测时间长度1~32767(ms)；
- (D)+0：为在S2时间内的脉冲个数，为16位的数据；
- (D)+1, (D)+2：为每分钟内的脉冲个数，为32位的数据；

指令举例：



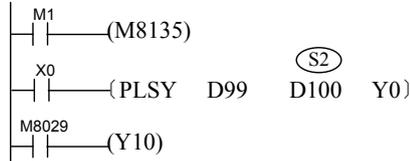
X0为脉冲信号输入端口；D0为单位时间1~32767(ms)；D10为D0时间段总脉冲个数；

D11, D12为运行频率=1分钟脉冲个数*10(0.1为单位)；PLSY指令的特殊说明。

16bit	32bit		FNC57	PLSY	脉冲输出
✓	✓				
7步	13步		指令格式: PLSY (S1) (S2) (D)		

c) 通过使用特殊位M8135~M8137(分别对应Y0~Y2)为ON, 可以实现如下功能:

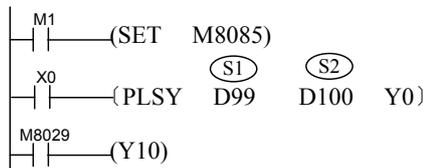
可以在运行中更改输出脉冲个数(可改大, 也可改小)。编程举例:



M1为ON, 该特殊功能有效, 此时X0导通执行PLSY指令, 脉冲发送过程中可以更改D100 (脉冲个数), 脉冲个数可以增加也可以减少; (注意: 更改后的D100的数据要大于该指令已经发送的脉冲个数)

d) 通过使用特殊位M8085~M8087(分别对应Y0~Y2)为ON, 可以实现如下功能:

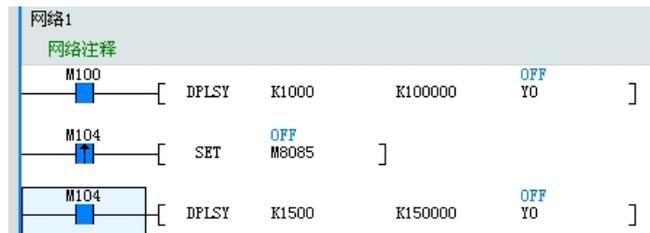
在驱动特殊位为ON, 可以马上启动下条脉冲输出指令, 不需要上条能流无效的处理。编程举例:



当X0=1, PLSY在指令发脉冲过程中, 或脉冲发送完毕, 若M8085被置位, PLSY指令就会重启, 按设定的频率和脉冲数 (S1)、(S2) (即D99、D100的当前值) 重新开始发送脉冲, 而无需指令驱动X0由ON→OFF→ON变化一次。



当M100启动, 正在发脉冲过程中(如上图), 启动M104, 则M100驱动的那一脉冲指令自动清除, 马上执行M104驱动的脉冲输出指令 (如下图), 不需等M100断开。



e) 通过使用特殊位M8090~M8092(分别对应Y0~Y2)为ON, 可以实现如下功能:

可以实现脉冲输出完成后执行一次用户中断; 具体对应如下:

端口号	使用特殊位	对应的用户中断
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504

当特殊功能位 (M8090-M8092) 为ON, 则在指定脉冲个数发送完毕后, 立即执行用户中断I502~I504。

2) PLSR, DRVI, DRVA指令的特殊说明

a) 通过使用特殊位M8135~M8137(分别对应Y0~Y2)为ON, 可以实现如下功能:

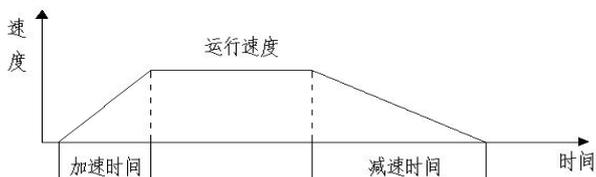
可以在运行中更改输出脉冲个数(可大可小), 同时加速和减速时间分别由如下寄存器定义, 时间单位为ms: (PLSR的加速时间是由PLSR (S1) (S2) (S3) (D) 中的(S3) 决定)

端口号	使用特殊位	修改加速时间用寄存器
Y000	M8135	D8104
Y001	M8136	D8105
Y002	M8137	D8106

端口号	使用特殊位	修改减速时间用寄存器
Y000	M8135	D8165
Y001	M8136	D8166
Y002	M8137	D8167

当特殊位(M8135~M8137)为ON时, PLSR, DRVI, DRVA指令执行过程中, 可以随时更改脉冲总数(注意: 更改后的脉冲总数必须大于当前已经发送的脉冲数量)

加速时间和减速时间可以不同, 但是必须在指令执行之前修改好加减速度时间。



b) 通过使用特殊位M8085~M8087(分别对应Y0~Y2)为ON, 可以实现如下功能:

在驱动特殊位为ON, 可以马上启动下条脉冲输出指令, 不需要上条能流无效的处理。



当M100启动, 正在发脉冲过程中(如上图), 启动M104, 则M100驱动的那一脉冲指令自动清除, 马上执行M104驱动的脉冲输出指令(如下图), 不需等M100断开。



c) 通过使用特殊位M8090~M8092(分别对应Y0~Y2)为ON, 可以实现如下功能:

可以实现脉冲输出完成后执行一次用户中断; , 具体对应如下:

端口号	使用特殊位	对应的用户中断
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504

3) 高速计数器多用户中断(XP不具备此功能)的使用描述

为了满足在高速计数器运行时, 支持多高速自由任务, 实现了高速计数器多用户中断(XP不具备此功能)(最大支持24个, 均为扩展的中断号), 设定和比较用数据表格的方式定义:

标志位	使用描述
M8084	为ON使能高速计数器多用户中断(XP不具备此功能)
D8084	为高速计数器序号235~255

标志位	使用描述
D8085	对应的用户中断个数，最大24个，从I507~I530
D8086	对应多个比较点数据的序号，只能为D元件，且为双字宽度，如200为D200开始的双字

比较点数据存放的例程：

使用高速计数器C235时，D8084=235；使用D200开头的32位寄存器作为数据比较值时，D8086=200；要执行5个中断输出时，D8085=5；最后M8084=ON该功能有效

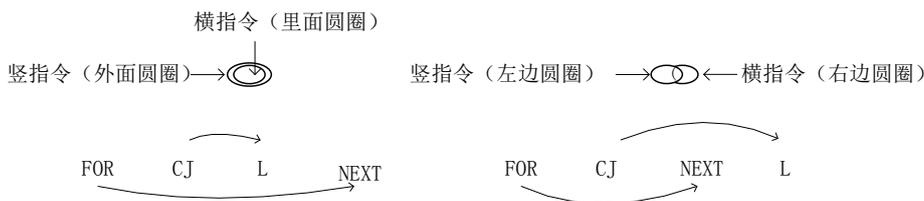
C235的数据	记录单元	存放单元数值	对应的用户中断	D8131的数值
100	D200, D201	=100	I507	0
200	D202, D203	=200	I508	1
300	D204, D205	=300	I509	2
400	D206, D207	=400	I510	3
500	D208, D209	=500	I511	4→0(M8133=ON)

每个中断可以由高速计数器的数值和记录单元的数值产生。

注意：该表格执行时，首先从上到下依次执行，只有每一格数据比较成功后才会继续往下一格执行，表格的最后一格执行完成后M8133=ON，此时数据比较又返回到表格的第1格，建议在返回到第1格时把C235数据清零，这样表格就会循环执行。该功能只有在增计数时才有效！

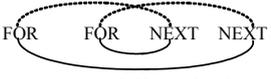
8.8 程序流程控制指令的相互关系

各种程序流程控制指令之间的相互关系如下所示。下表中表示包含关系表示前后区间重复。在AutoShop编程环境中子程序和中断程序单独写，故没有P-SRET、I-IRET、FEND-END、O-FEND、O-END，



竖 \ 横	MC-MCR	CJ-L	EI-DI	FOR-NEXT	STL-RET
MC-MCR	 	 	 	 	 
	 	 	 	 	 
CJ-L	 	 	 	 	 
	 	 	 	 	 
EI-DI	 	 	 	 	 
	 	 	 	 	 
FOR-NEXT	 	 	 	 	 
	 	 	 	 	 
STL-RET	 	 	 	 	 
	 	 	 	 	 
P-SRET	 	 	 	 	 
	 	 	 	 	 
I-IRET	 	 	 	 	 
	 	 	 	 	 
FEND-END	 	 	 	 	 
	 	 	 	 	 
O-FEND	 	 	 	 	 
	 	 	 	 	 
O-END (无FEND)	 	 	 	 	 
	 	 	 	 	 

- 能够没有问题的互相组合使用。
- × 禁止使用的组合，() 之中为错误序号。
- △ 虽然不属于禁止范围，但是可能会导致动作的复杂化，建议尽量避免组合使用。

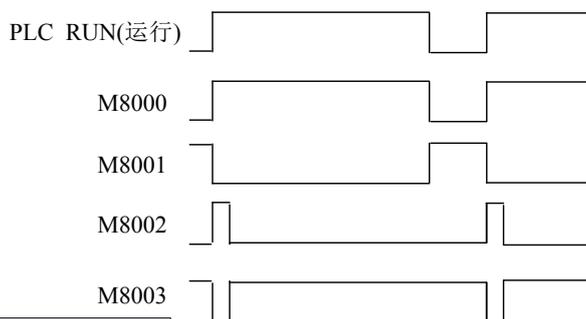
竖 \ 横	P-SRET	I-IRET	FEND-END	备注
MC-MCR	○ × (6608) ○ × (6606)	○ × (6608) ○ × (6606)	○ × (6608) ○ × (6608)	※1为忘记DI的状态。 不属于错误。 ※2  按照实践所示方式动作。 ※3最初的FEND和END有效，不是目的的程序。 不属于错误。 除了部分例外，具有包含关系的指令可以进行组合使用。 但是必须注意以下特例情况。 ①在FOR-NEXT, STL-RET, P-SRET, I-RET中无法使用MC-MCR。 ②在FOR-NEXT, P-SRET, I-RET中无法使用STL-RET。 ③在MC-MCR, FOR-NEXT, P-SRET, I-IRET指令间无法使用I, IRET, SRET, FEND, END等指令进行屏蔽操作。
CJ-L	○ △ ○ △	○ △ ○ △	○ × ○ ○	
EI-DI	○ ○ ○ ○	○ ○ ○ ○	○ ※1 ○ ○	
FOR-NEXT	○ × (6607) ○ × (6701)	○ × (6607) ○ × (6607)	○ × (6607) ○ × (6607)	
STL-RET	○ × (6605) ○ × (6606)	○ × (6605) ○ × (6606)	○ × (6605) ○ × (6605)	
P-SRET	○ × (6606) ○ × (6606)	○ × (6606) ○ × (6606)	○ × (6709) ○ × (6709)	
I-IRET	○ × (6606) ○ × (6606)	○ × (6606) ○ × (6606)	○ × (6606) ○ × (6606)	
FEND-END	○ ○ ○ × (6709)	○ ○ ○ × (6709)	○ ※3 ○ ※3	
0-FEND	○ × (6606) ○ × (6709)	○ × (6606) ○ × (6606)	○ ※3 ○ ※3	
0-END (无FEND)	○ × (6606) ○ × (6709)	○ × (6606) ○ × (6706)	○ ※3 ○ ※3	

8.9 部分特殊继电器和寄存器功能说明

系统运行状态

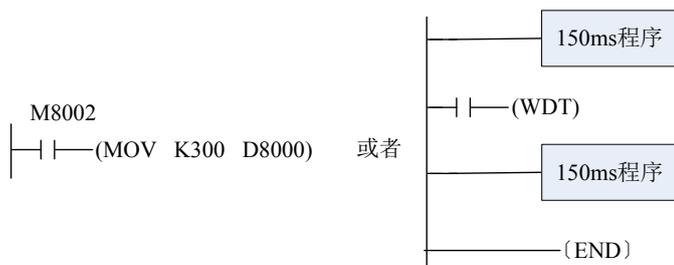
PLC的运行标志 M8000-M8003

- 1) M8000: M8000为RUN中常时ON接点, 即运行监视常开接点(A接点), PLC在RUN的状态下, M8000一直保持为ON。
- 2) M8001: M8001为RUN中常时OFF接点, 即运行监视常闭接点(B接点), PLC在RUN的状态下, M8001一直保持为OFF。
- 3) M8002: PLC开始RUN的第一次扫描ON, 之后保持为OFF。该脉冲的宽度为一次扫描时间, 当要作各种初始设置工作时可以使用本接点。
- 4) M8003: PLC开始RUN的第一次扫描周期OFF, 之后一直ON。即起始负向(RUN的瞬间'OFF')脉冲。



监控定时器 D8000

- 1) 监控定时器专门用来监视PLC的扫描时间, 当扫描时间超过监控定时器的设置时, ERROR红色指示灯长亮, 输出全部变成Off。
- 2) 监控定时器时间的初始值为200ms, 当指令运算过于复杂或者是PLC主机连接众多的特殊模块时都会造成扫描时间过长, 扫描时间是否超过D8000的设置值, 请监视D8010~D8012。此种情况下, 可于程序中使用MOV指令来变更监控定时器的设置值, 将监控定时器的设置值变更为300ms, 也可于PLC程序中加入WDT指令, 当CPU执行至WDT指令时, 内部监控定时器被清除为零, 使得扫描时间不会超过监控定时器的设置时间。



监控定时器最大可设置至32767ms, 但必须注意, 监控定时器设置过大时, 运算异常发生的检出时机将会跟着被拖慢。因此, 若非复杂的运算使得扫描时间超过200ms, 一般的情况下请维持在200ms以下较佳。

单板程序版本 D8001

单板程序版本, 如D8001=24120的含义是: 24为H_{2U}, 120为版本V1.20, 即版本号为V1.20的H_{2U}型PLC; 26为H_{1U}。

程序容量 D8002

程序容量, 4K、8K、16K、24K等, H_{2U}型号PLC的容量是24K, H_{1U}型号PLC的容量是8K。

语法检查信号 M8004、D8004

M8004: 当错误信号M8060~M8067(除了M8062外)中任意一个为ON时, M8004为ON。可用来监控PLC中是否有系统错误。

D8004: M8060~M8067的BCD值, 初始值为0。

电压电池检测M8005~M8009 D8005~D8009 M8030

- 1) 当电池电压D8005的检测值低于D8006（初始值为2.6V）时，M8005有输出；
- 2) 电池电压低报警有出现（M8005为ON）过，M8006就置ON（锁存），PLC掉电再上电时复位，程序无法对其进行复位；
- 3) 当交流电失电5ms后M8007、M8008动作，失电时间在D8008之内PLC程序继续运行、在D8008之外用户程序不执行，M8000为OFF；
- 4) 扩展模块24V掉电时M8009为ON，D8009记录掉电扩展模块的编号；
- 5) M8030为ON时，屏蔽电池电压低的警告。

系统时钟

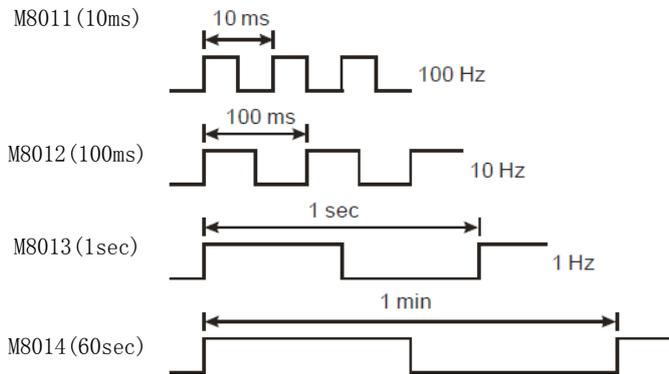
扫描时间监视 D8010~D8012

程序扫描时间的现在值、最小值及最大值被存放在D8010~D8012当中。

- 1) D8010: 扫描时间的现在值；
- 2) D8011: 扫描时间的最小值；
- 3) D8012: 扫描时间的最大值。

周期振荡时钟 M8011~M8014

PLC主机内部均具备下列4种振荡时钟，只要PLC通上电源，这4种振荡时钟就会自动动作。PLC处于STOP状态下，振荡时钟也会动作，所以振荡时钟启动时序及RUN的启动时序并不会同步。



实时时钟 D8013~D8019 M8015~M8019

- 1) M8015为ON时，时钟计时停止；
- 2) M8017每On一次，PLC内部时钟作±30秒校正动作，这里的校正是指当PLC的内部时钟的秒针D8013在1~29时，会被自动归为“0”秒而分针不变、D8013在30~59时，也会被自动归为“0”秒，分针加1分钟。
- 3) 通常的情况下年显示4位数。

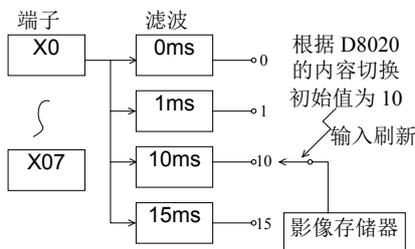
地址号	名称	动作功能
M8015	时钟停止和时间校准	ON时时钟停止，利用ON→OFF的边沿写入时间，执行再动作。
M8016	时间显示停止	ON时时钟显示停止（计时保持动作）
M8017	±30秒修正	利用OFF→ON的边沿修正秒。（当秒数为0~29秒时变为0秒。当秒数为30~59秒时向分进一位，将秒变为0秒）
M8018	安装检测	常时处于ON状态
M8019	RTC错误	校准时间时，当特殊数据寄存器的数据超过设定范围时变为ON。

地址号	名称	设定值的范围	动作功能
D8013	秒	0~59	用于写入校准时间的初始值，或读取当前时间。
D8014	分	0~59	
D8015	时	0~23	
D8016	日	0~31	
D8017	月	0~12	
D8018	年	0~99（公历后两位）	
D8019	星期	0~6（对应日~六）	

指令标志

输入滤波调整 D8020

- 1) X0~X7输入端，可由D8020的内容来设置输入端接收脉冲的反应时间，设置范围0~60ms。默认10ms，
- 2) 当程序中使用高速计数器、中断插入等功能时，则相关端口的滤波时间自动为最短时间，无关的X0~X7端口的滤波时间仍为原设定值D8020。



- 3) PLC电源Off→On变化时，D8020的内容自动变成10
- 4) 若执行以下程序，滤波常数变为0ms。但是实际上该输入端设置了C-R滤波器硬件，即使是将其指定为0，但也不会低于10μs（40点、60点PLC的X2~X5不会低于50μs）。

```
M8000
| | | (MOV K0 D8020)
```

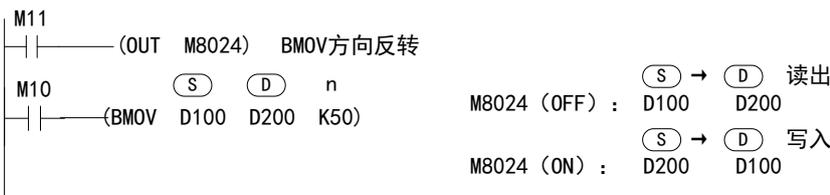
运算标志 M8020~M8022

在所有运算中的各种标志的动作如下。

- 1) 零标志：结果值为0时，M8020=ON
- 2) 借位标志：结果值超出最小处理单位时，M8021=ON
- 3) 进位标志：结果绝对值超出最大使用范围时，M8022=ON

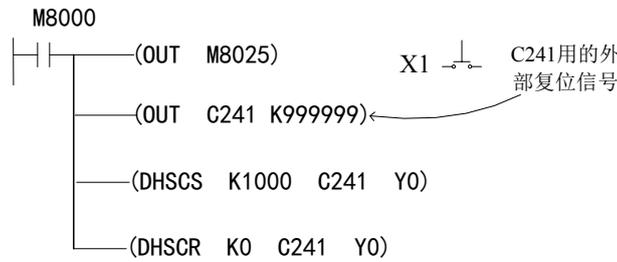
指令赋值换向 M8024

通过控制BMOV指令的反向标记M8024，可用一个程序指令向两个方向传送数据。



HSC指令模式 M8025

高速计数器的输出接点、FNC53（D HSCS）、FNC54（D HSCR）、FNC55(D HSZ)指令中的比较输出，都随计数输入的当前值寄存器的变化而动作。即使通过传送指令DMOV改变当前值（C235~C255），只要没有计数输入，比较输出就不发生变化。这正如前面“注意事项”讲述的那样，关于高速计数器C241等，备有外部复位端子（R），通过复位输入信号的上升沿，执行指令、输出比较结果。详见以下内容。外部复位模式C241用的外部复位端子。



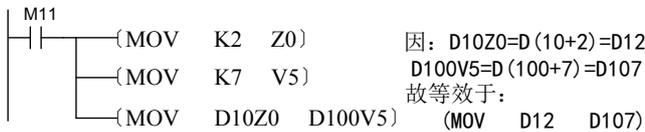
在上例中，当M8025为ON时，C241的现在值假如为1000，此时Y0为ON，如果按下外部复位按钮X1，C241的现在值变为0，即使X0没有计数输入，此时Y0也复位。

执行完毕标志 M8029

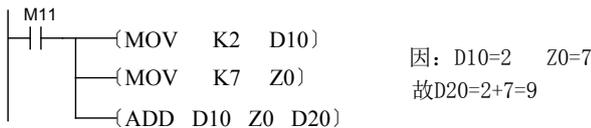
高速脉冲输出、通讯、MTR、HKY、DSW、SEGL、PR等指令执行完毕标志。

变址寄存器 D8028、D8029 D8182~D8195

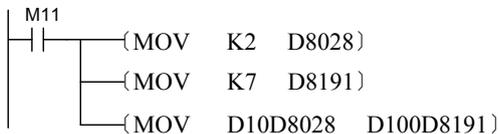
变址寄存器除了和普通的数据寄存器有相同的使用方法外，在应用指令的操作数中，还可以同其他的软元件编号或数值组合使用。但需注意LD，AND，OUT等基本顺控指令或步进梯形图指令的软元件编号不能同变址寄存器组合使用，在程序中我们一般用Z或者V来进行变址修饰：



亦可以将Z、V当普通寄存器使用



Z、V有对应的特殊寄存器，如第一个程序等效于



- D8028为Z0寄存器内容；D8029为V0寄存器内容；
- D8182为Z1寄存器内容；D8183为V1寄存器内容；
- D8184为Z2寄存器内容；D8185为V2寄存器内容；
- D8186为Z3寄存器内容；D8187为V3寄存器内容；
- D8188为Z4寄存器内容；D8189为V4寄存器内容；
- D8190为Z5寄存器内容；D8191为V5寄存器内容；
- D8192为Z6寄存器内容；D8193为V6寄存器内容；
- D8194为Z7寄存器内容；D8195为V7寄存器内容；

位状态 M8031~M8034

- 1) M8031为ON时，清除所有非停电保持的寄存器、继电器等；
- Y、一般用M、一般用S接点状态
 - 一般用T的接点及计时线圈
 - 一般用C的接点及计数线圈及复位线圈
 - 一般用D的现在值寄存器
 - 一般用T的现在值寄存器
 - 一般用C的现在值寄存器

2) M8032为ON时, 清除所有停电保持的寄存器、继电器等;

停电保持用M、S的接点状态

累计型定时器T的接点及计时线圈

停电保持用C及高速计数器C的接点、计数线圈

停电保持用D的现在值寄存器

累计型定时器T的现在值寄存器

停电保持用C及高速计数器C的现在值寄存器

3) M8033为ON时, 停机状态所有的软元件保持运行前的状态不变

所有Y、M、S的接点状态

所有T的接点及计时线圈

所有C及高速计数器C的接点、计数线圈

所有D、T、C的现在值寄存器

4) M8034为ON时, 所有输出点Y变成OFF。

5) 通过强制RUN/STOP操作使M8035(强制RUN模式)和M8036(强制RUN)变为ON, 起动可编程控制器。

6) 将M8037(强制STOP)置于ON, 可以停止可编程控制器的运行。

恒定扫描模式 M8039 D8039

驱动辅助继电器M8039, 将目标扫描时间(以1ms为单位)预先写入数据寄存器D8039中, 则可编程控制器的运算周期不会低于该值。即使运算提早结束时, 也在剩余时间内进行等待, 扫描时间够D8039的值后才返回0步。如果D8039的内容小于实际上程序的扫描时间时, 则以实际上程序的扫描时间为主。



和扫描时间有关的指令RAMP、HKY、SEGL、ARWS、PR, 应用上必须用“固定扫描时间模式”或者是和“定时插入中断”搭配使用。特别是HKY指令, 它是以4×4矩阵方式作16个数字按钮的输入操作, 使用时扫描时间必须固定在20ms以上。D8010~D8012所显示的扫描时间也包括固定的扫描时间。

进步阶梯

步进 M8040~M8049 D8040~D8049

1) 驱动M8040后, 即使具备了转移条件也无法进行状态之间的转移。而且停止状态内的输出将继续其动作。

2) 步进开始, IST指令用标志M8041。

3) 启动脉冲, IST指令用标志M8042。

4) 原点回归状态结束, IST指令用标志M8043。

5) 检测到机械原点动作, IST指令用标志M8044。

6) 所有输出复位禁止, IST指令用标志M8045。

7) 当M8047为ON后, S0~S999中任何一个为ON时, M8046自动接通。用于避免与其他流程同时启动或用作工序的动作标志。

8) 当M8049为ON后, S900~S999中任何一个为ON时, M8048自动接通。用于避免与其他流程同时启动或用作工序的动作标志。

9) D8040~D8047: 将S0~S999的最小动作地址号保存在D8040中, 其他依次。最大的报警地址号保存在D8047中。

10) 当M8049为ON后, D8049保存S900~S999的报警的最小地址号。

中断禁止

禁止中断 M8050~M8059

使能了相对应的中断禁止以后，即使有中断信号，此时也不会产生中断，例如M8050为ON后，即使X0有中断脉冲如入，I00□也不会产生。

各个对应的中断定义如下：

中断允许/禁止设置				
M8050	驱动I00□中断禁止	X输入中断，共有12个中断，分别对应X0~X5端口的上升沿中断、下降沿中断。 中： 1=上升沿中断； 0=下降沿中断	每个标志对应1个外部中断的控制； 当该M标志为OFF时，允许对应的X中断； 当该M标志为ON时，禁止对应的X中断；	
M8051	驱动I10□中断禁止			
M8052	驱动I20□中断禁止			
M8053	驱动I30□中断禁止			
M8054	驱动I40□中断禁止			
M8055	驱动I50□中断禁止			
M8056	驱动I6□口中断禁止			定时中断0
M8057	驱动I7□口中断禁止			定时中断1
M8058	驱动I8□口中断禁止			定时中断2
M8059	驱动计数器中断禁止	高速计数中断，共6个	为ON时，禁止I010~I060的中断	

但X0~X5的脉冲捕捉功能不受中断禁止的限制。

联机功能

并联协议 M8070~M8073 M8162 D8070

并联协议，M8070置位为并联协议主站，M8071置位为并联协议从站，M8070与M8071在一台PLC中不能同时置位，若同时置位并联协议无效，若不存在其它优先协议，可通过D8126设置，D8126=50h为并联协议主站，D8126=5h为并联协议从站。

D8070：判断通讯出错的时间设定，默认为500。

M8162：高速并联连接模式。

	主站发送（从站接收）	从站发送（主站接收）
普通模式 M8162=0	M800~M899 D490~D499	M900~M999 D500~D509
高速模式 M8162=1	D490~D491	D500~D501

具体功能应用设置请看通讯部分并联协议的介绍。

N:N协议 D8076~D8180 M8183~M8191 D8201~D8218

用户只需要设置一台PLC为N：N协议主站，另外多PLC设置为N：N协议从站，且所有使用该协议的PLC串口连接起来，不需要用户程序干预，即可实现多台PLC间互相交换数据。

D8176：站点号，范围0~7，0表示主站点；

D8177：从站点的总数，范围1~7，仅主站需要；

D8178：刷新范围（模式）设置，范围0~2，仅主站需要；

D8179：重试次数设定，仅主站需要；

D8180：通信超时设置，*10ms，仅主站需要；

M8183~M8190：通信出错标志，M8183对应第0号站点（主站），M8184对应第1号站点，依次类推，M8190对应第7号站点；

M8191：正在执行数据传送；

D8201: 监控当前连接扫描时间;

D8202: 监控最大连接时间;

D8203: 监控主站通讯错误次数, 最大计数到10000次就停止计数;

D8204~D8210: 分别监控从站1~7通讯错误次数, 最大计数到10000次就停止计数;

D8211: 主站通讯错误代码;

D8212~D8218: 分别为从站1~7通讯错误代码。D8211~D8218的错误代码含义可查出错信息说明。

具体功能应用设置请看通讯部分N: N协议的介绍。

特殊功能

输出口初始化 M8085~M8089

通过使用特殊位M8085~M8089(分别对应Y0~Y4)为ON, 在PLSY, PLSR, DRVI, DRVA指令中可以实现如下功能:
在驱动特殊位为ON, 可以马上启动下条脉冲输出指令, 不需要上条能流无效的处理;

输出完成中断 M8090~M8094

通过使用特殊位M8090~M8094(分别对应Y0~Y4)为ON, 在PLSY, PLSR, DRVI, DRVA指令中可以实现如下功能:
可以实现脉冲输出完成中断; 具体对应如下:

端口号	使用特殊位	对应的用户中断
Y0	M8090	I502
Y1	M8091	I503
Y2	M8092	I504
Y3	M8093	I505
Y4	M8094	I506

具体功能设置请看H_{2u}特殊功能的介绍。

加减速时间M8135~M8139 D8104~D8108 D8165~D8169

通过使用特殊位M8135~M8139(分别对应Y0~Y4)为ON, 在PLSY, PLSR, DRVI, DRVA指令中可以实现如下功能:
可以在运行中更改输出脉冲个数(可大可小, 只能在加速和匀速的时候变, 在减速的时候无效);
同时在PLSR, DRVI, DRVA指令中减速时间分别由如下寄存器定义:

端口号	使用特殊位	对应的寄存器
Y0	M8135	D8165
Y1	M8136	D8166
Y2	M8137	D8167
Y3	M8138	D8168
Y4	M8139	D8169

在M8135为ON的时候, YO的减速时间由D8165决定, 默认100ms。依次类推, 在DRVI, DRVA指令中加速时间分别由如下寄存器定义:

端口号	使用特殊位	对应的寄存器
Y0	M8135	D8104
Y1	M8136	D8105
Y2	M8137	D8106
Y3	M8138	D8107
Y4	M8139	D8108

在M8135为ON的时候, YO的加速时间由D8104决定, 默认100ms。依次类推。

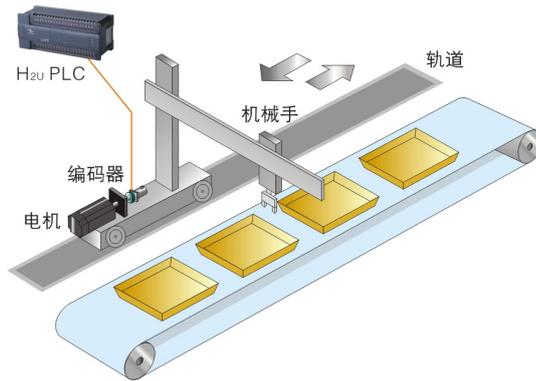
具体功能设置请看H_{2u}特殊功能的介绍。

高速计数器多用户中断M8084 D8084~D8086

为了满足在高速计数器运行时，支持多高速自由任务，可以设置任意一个高速计数器（C235~C255）产生多个中断，实现了如下使用规律：此功能命名为高速计数器多用户中断（XP不具备此功能），最大支持24个：

标志位	描述
M8084	为ON使能高速计数器多用户中断（XP不具备此功能）
D8084	为高速计数器序号C235~C255
D8085	对应的用户中断个数，最大24个，从I507~I530
D8086	对应多个比较点数据的序号，只能为D元件，如200为D200开始的双字

例如：当行车在轨道上执行装卸任务时，要求运行在不同位置时快速执行不同任务，使用高速计数器多用户中断（XP不具备此功能）处理功能，具有响应快操作简单的优势。



具体功能设置请看H2u特殊功能的介绍。

计米器、测速表功能 M8100~M8105

在SPD指令中，如果功能使能标志M8100~M8105(分别对应X0~X5)为ON，则SPD (S1) (S2) (D) 写法中的 (D) 的定义和原指令有改变，

对应的M8100~M8105为OFF时：

(D) +0: 为在 (S2) 时间内的脉冲个数，为16位的数据； (D) +1: 本次时间段对脉冲的计数； (D) +2: 用于测定剩余时间(MS)。

对应的M8100~M8105为ON时：

(D) +0: 为在 (S2) 时间内的脉冲个数，为16位的数据； (D) +1, (D) +2: 为每分钟内的脉冲个数，为32位的数据；

例如指令：



X0为输入端口；
D0为单位时间(ms)；
D10 为D0 时间段总脉冲个数；
D11, D12 为运行频率=1 分钟脉冲个数*10(0.1 为单位)；

具体功能设置请看H2u特殊功能的介绍。

输出端口清零 M8140 D8156~D8160

当M8140为ON的时候，用ZRN指令回到原点D8156~D8160对应的输出点有一个扫描周期的ON输出，用来CLR伺服。

D8156: Y0端口清零信号定义，默认5=Y05

D8157: Y1端口清零信号定义，默认6=Y06

D8158: Y2端口清零信号定义，默认7=Y07

D8159: Y3端口清零信号定义，默认8=Y10

D8160: Y4端口清零信号定义，默认9=Y11

高速环形计数器

高速环形计数器 M8099 D8099

特殊数据寄存器D8099在M8099被驱动后，从下一个扫描周期开始，以0.1ms时钟累计计算。当D8099内容超过32767时，从0开始重新计算。

利用累积型的1ms定时器或特殊数据寄存器D8099（高速环形计数器），可以测定1ms或0.1ms为单位的脉冲的宽度。

通讯、链接

COM0: D8110~D8119

COM0协议	D8116设定	半/全双工模式	通信格式
下载协议/HMI监控协议	01h	全双工	固定为“7E1”
MODBUS-RTU从站/Qlink协议	02h	半双工	由D8120决定
其他协议（含RS指令）		不支持	

COM1: M8120~M8129 D8120~D8129

COM1协议	D8126设定	半/全双工模式	通信格式
RS指令	00h	由D8120的Bit10设定*	由D8120决定
HMI监控协议	01h	半双工	固定
并联协议主站	50h	半双工	固定
并联协议从站	05h	半双工	固定
N: N协议主站	40h	半双工	固定
N: N协议从站	04h	半双工	固定
计算机链接协议	06h	半双工	由D8120决定
MODBUS-RTU从站/Qlink协议	02h	半双工	
MODBUS-ASC从站	03h	半双工	
RS指令	10h	由D8120的Bit10设定*	
MODBUS RTU指令	20h	半双工	
MODBUS-ASC指令	30h	半双工	
MODBUS配置	60h	半双工	

*RS指令半双工/全双工模式由D8120的Bit10设定:

1: 半双工RS485（使用标配端口）

0: 全双工RS232C/RS422（需要使用扩展小板H2U-232BD或H2U-422BD）

8

附录

M8120	保留	D8120	通讯格式，界面配置设定，默认为0
M8121	发送等待中（RS指令）	D8121	站号设置，界面配置设定，默认为1
M8122	发送标志（RS指令） 指令执行状态（MODBUS）	D8122	传送剩余数据数量（仅对RS指令）
M8123	接收完成标志（RS） 通讯错误标志（MODBUS）	D8123	接收到的数据数量（仅对RS指令）
M8124	接收中（仅对RS指令）	D8124	起始字符STX（仅对RS指令）
M8125	保留	D8125	终止字符ETX（仅对RS指令）
M8126	为ON时485BD扩展卡有效	D8126	通讯协议设定，界面配置设定，默认为0
M8127	保留	D8127	计算机链接协议接通要求数据起始地址号

M8128	保留	D8128	计算机链接协议接通要求发送数据数量
M8129	超时判断	D8129	通讯超时时间判断，界面配置设定，默认为10（100ms）

注：COM2和COM3的元件M、D设置请参考《第五章通信功能》。

COM0与COM1、COM2和COM3的功能区别

COM0硬件为标准RS422，接口端子为8孔鼠标头母座；

COM1和COM2硬件为RS485，接口为接线端子；COM3需要通过扩展卡来扩展端口。

COM0的422支持程序下载和HMI监控协议；

COM1、COM2和COM3的485可做主站或者从站，支持RS指令、联机功能、计算机链接协议等。

定位

定位 M8145~M8154 D8136~D8160

M8145：在发脉冲的时候将M8145置ON。则Y0马上停止脉冲输出，PLSY、DRVI等指令在下次驱动能流的时候重新从零发脉冲。DRVA指令在下次驱动能流的时候接着发剩余的脉冲。

M8146：在发脉冲的时候将M8146置ON。则Y1马上停止脉冲输出，PLSY、DRVI等指令在下次驱动能流的时候重新从零发脉冲。DRVA指令在下次驱动能流的时候接着发剩余的脉冲。

M8152：在发脉冲的时候将M8152置ON。则Y2马上停止脉冲输出，PLSY、DRVI等指令在下次驱动能流的时候重新从零发脉冲。DRVA指令在下次驱动能流的时候接着发剩余的脉冲。

M8153：在发脉冲的时候将M8153置ON。则Y3马上停止脉冲输出，PLSY、DRVI等指令在下次驱动能流的时候重新从零发脉冲。DRVA指令在下次驱动能流的时候接着发剩余的脉冲。

M8154：在发脉冲的时候将M8154置ON。则Y4马上停止脉冲输出，PLSY、DRVI等指令在下次驱动能流的时候重新从零发脉冲。DRVA指令在下次驱动能流的时候接着发剩余的脉冲。

M8147：在Y0有脉冲输出的时候，M8147为ON，停止时为OFF，可用来做监控；

M8148：在Y1有脉冲输出的时候，M8148为ON，停止时为OFF，可用来做监控；

M8149：在Y2有脉冲输出的时候，M8149为ON，停止时为OFF，可用来做监控；

M8150：在Y3有脉冲输出的时候，M8150为ON，停止时为OFF，可用来做监控；

M8151：在Y4有脉冲输出的时候，M8151为ON，停止时为OFF，可用来做监控。

D8136: Low word; D8137: High word

作为Y0和Y1输出定位指令的当前值数据累加寄存器使用，用DRVI、DRVA指令时，对应旋转方向增减当前值，另外，由于PLSY、PLSR只有脉冲输出没有方向信号的指令亦使用相同的当前值寄存器，因此当执行这些指令时，当前值为脉冲输出数的累加值；

D8140: Low word; D8141: High word

作为Y0输出定位指令的当前值数据寄存器使用，用DRVI、DRVA指令时，对应旋转方向增减当前值，另外，由于PLSY、PLSR只有脉冲输出没有方向信号的指令亦使用相同的当前值寄存器，因此当执行这些指令时，当前值为脉冲输出数的累加值；

D8142: Low word; D8143: High word

作为Y1输出定位指令的当前值数据寄存器使用，用DRVI、DRVA指令时，对应旋转方向增减当前值，另外，由于PLSY、PLSR只有脉冲输出没有方向信号的指令亦使用相同的当前值寄存器，因此当执行这些指令时，当前值为脉冲输出数的累加值；

D8150: Low word; D8151: High word

作为Y2输出定位指令的当前值数据寄存器使用，用DRVI、DRVA指令时，对应旋转方向增减当前值，另外，由于PLSY、PLSR只有脉冲输出没有方向信号的指令亦使用相同的当前值寄存器，因此当执行这些指令时，当前值为脉冲输出数的累加值；

D8152: Low word; D8153: High word

作为Y3输出定位指令的当前值数据寄存器使用，用DRVI、DRVA指令时，对应旋转方向增减当前值，另外，由于PLSY、PLSR只有脉冲输出没有方向信号的指令亦使用相同的当前值寄存器，因此当执行这些指令时，当前值为脉冲输出数的累加值；

D8154: Low word; D8155: High word

作为Y4输出定位指令的当前值数据寄存器使用，用DRVI、DRVA指令时，对应旋转方向增减当前值，另外，由于PLSY、PLSR只有脉冲输出没有方向信号的指令亦使用相同的当前值寄存器，因此当执行这些指令时，当前值为脉冲输出数的累加值；

D8145: 执行DRVI、DRVA等指令时的基底速度。控制步进电机时，设定速度时需考虑步进电机的共振区域和自动启动频率。设定范围：最高速度（D8147，D8146）/10以下。超过该范围时，自动降为最高速度的1/10数值运行。

D8146: Low word; D8147: High word

执行DRVI、DRVA等指令时的最高速度。高速输出口输出脉冲的频率必须小于该最高速度。设定范围：10~100,000 (Hz)

D8148: DRVI、DRVA、PLSR指令时的加减速时间。加减速时间表示到达最高速度（D8147，D8146）所需的时间。因此，当输出脉冲频率低于最高速度（D8147，D8146）时，实际加减速时间会缩短。设定范围：50~5,000(ms);

M8135~M8139:(D8165~D8169)

M8135: Y0端口更改脉冲和减速时间有效标志位；通过使用该特殊位可以实现在运行中更改输出的脉冲个数同时减速时间可以单独修改。D8165对应Y0的减速时间。

M8136: Y1端口更改脉冲和减速时间有效标志位；通过使用该特殊位可以实现在运行中更改输出的脉冲个数同时减速时间可以单独修改。D8166对应Y0的减速时间。

M8137: Y2端口更改脉冲和减速时间有效标志位；通过使用该特殊位可以实现在运行中更改输出的脉冲个数同时减速时间可以单独修改。D8167对应Y0的减速时间。

M8138: Y3端口更改脉冲和减速时间有效标志位；通过使用该特殊位可以实现在运行中更改输出的脉冲个数同时减速时间可以单独修改。D8168对应Y0的减速时间。

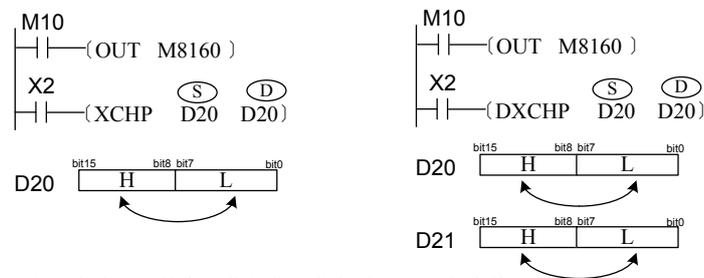
M8139: Y4端口更改脉冲和减速时间有效标志位；通过使用该特殊位可以实现在运行中更改输出的脉冲个数同时减速时间可以单独修改。D8169对应Y0的减速时间。

M8135~M8139只适用于以下指令：PLSY/PLSR/DRVI/DRVA。

数据处理

XCH的SWAP功能 M8160

当特殊变量M8160=1时，且 **(D)** 与 **(S)** 为同一地址，完成的操作将是高8位与低8位的交换，32位的指令也一样，完成的操作将是两个寄存器的高8位与低8位的值各自进行互相交换。相当于SWAP指令的操作，一般用SWAP指令来实现。



左图中将D20的高8位与低8位的值进行互相交换
右图中将D20的高8位与低8位的值进行互相交换，
D21的高8位与低8位的值进行互相交换，

8

附录

8位与16位模式 M8161

M8161标志决定了变量的宽度模式，当M8161=OFF时，为16bit模式；当M8161=ON时，为8bit模式，因此实际使用变量区域的长度增加。

在ASC/RS/ASCII/HEX/CCD中应用。

传送点数可变模式 M8164 D8164

FROM/T0指令的传送点数可变模式：若M8164=ON，执行FROM/T0指令时，特殊数据寄存器D8164（FROM/T0指令的传送点数指定寄存器）的内容作为传送点数n进行处理；

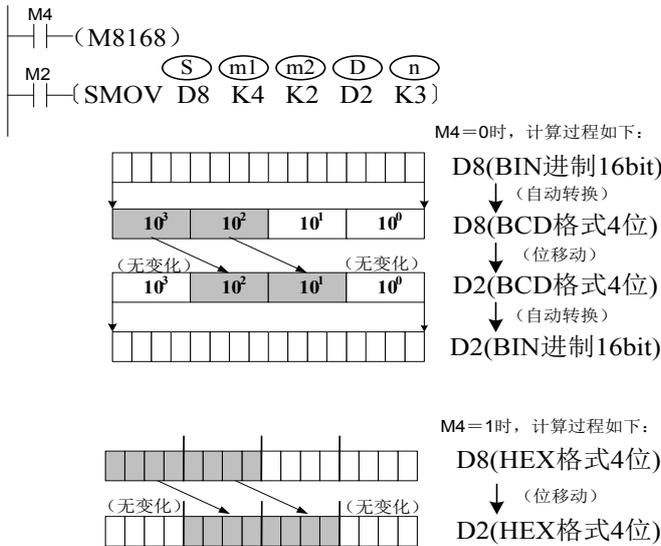
位切换字功能 M8167

在HEY指令中将M8167置为ON，HEY指令将①~? 按键按16进制数据进行存储，保存到 **(D2)** 单元。

例：[123BF]输入后，**(D2)**中以BIN形式存储[123BF]，即将A~F的功能改变，详细介绍请参考HEY指令。

SMOV十进制与十六进制切换功能 M8168

在SMOV指令中，当M8168为OFF时是BCD模式（十进制的位），当M8168为ON时是BIN模式，在BIN模式下以4个位作为一个单位作传送（十六进制的位）。



假设D8=K1234，D2=K5678，，则当M8168为OFF时（BCD模式），将M2置ON，则D2的值变为K5128；
 当M8168为ON时（BIN模式），此时D8=H04D2=K1234，D2=H162E=K5678，将M2置ON，则D2=H104E=K4174。

脉冲捕捉

脉冲捕捉功能 M8170~M8175

- M8170 X00脉冲捕捉
- M8171 X01脉冲捕捉
- M8172 X02脉冲捕捉
- M8173 X03脉冲捕捉
- M8174 X04脉冲捕捉
- M8175 X05脉冲捕捉

若需要对出现在X0~X5端口的瞬间脉冲信号作出反应，但对反应动作时间没有特别要求，就可以使用“脉冲捕捉”功能，PLC会将出现在X0~X5端口的上升沿信号保存在M8170~M8175单元，主程序中可作为判断处理的依据，响应处理完毕，可人为将之清除。

执行中断允许EI指令后，当输入点X000~X005OFF→ON变化时，特殊辅助继电器M8170~M8175置位进行中断处理。为了再次获得输入，必须利用程序对设定的元件进行复位操作。脉冲捕捉动作同个别中断禁止用辅助继电器M8050~M8055的动作无关，M8050~M8055的置ON不能禁止此捕捉功能。

高速输入

表格高速比较模式 M8130 M8131 D8130

在表格高速比较模式中，第一行的数据一致时，表格计数器D8130变为1，并进入第二行操作。以下同样动作，最后一行操作完毕时，完毕标志M8131动作，回到初始行重复动作。

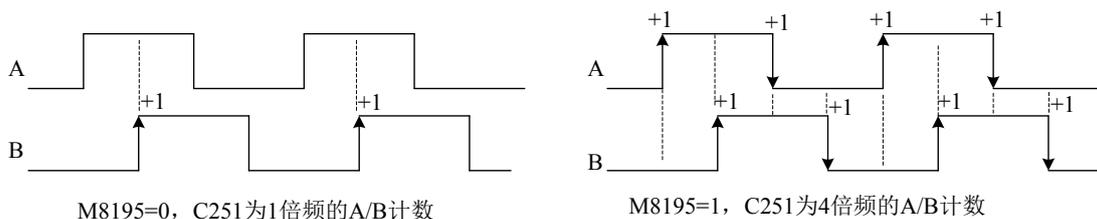
具体应用请参照：HSZ指令。

倍频控制 M8195~M8199

A/B相高速计数器T251~T255有1倍频和4倍频两种频率模式，分别由特殊寄存器M8195~M8199设定。AB相高速计数器的信号，占用两个脉冲输入口，对PLC的等效脉冲数影响按2倍计算，若C251~C255的A/B输入1倍频模式时，最大高速输入频率为

50kHz。若C251~C255的A/B输入4倍频模式时，为软件计数模式，最大高速输入频率降为25kHz。

当M8195为OFF的时候，C251的AB相输入为一倍频；当M8195为ON的时候，C251的AB相输入为四倍频；
如下图：

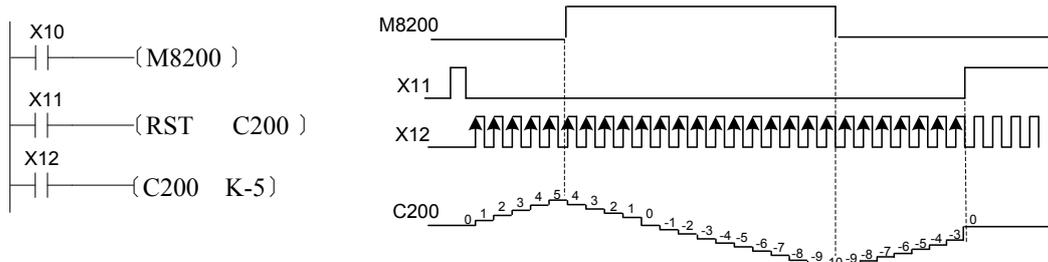


和C251一样：

当M8196为OFF的时候，C252的AB相输入为一倍频；当M8196为ON的时候，C252的AB相输入为四倍频；
当M8197为OFF的时候，C253的AB相输入为一倍频；当M8197为ON的时候，C253的AB相输入为四倍频；
当M8198为OFF的时候，C254的AB相输入为一倍频；当M8198为ON的时候，C254的AB相输入为四倍频；
当M8199为OFF的时候，C255的AB相输入为一倍频；当M8199为ON的时候，C255的AB相输入为四倍频；

32位计数增减控制 M8200~M8234

对于普通的32位增减计数，可利用特殊的辅助继电器M8200~M8234指定增计数 / 减计数的方向，如对C△△△驱动M8△△△，则为减计数，不驱动时，则为增计数，如下例：



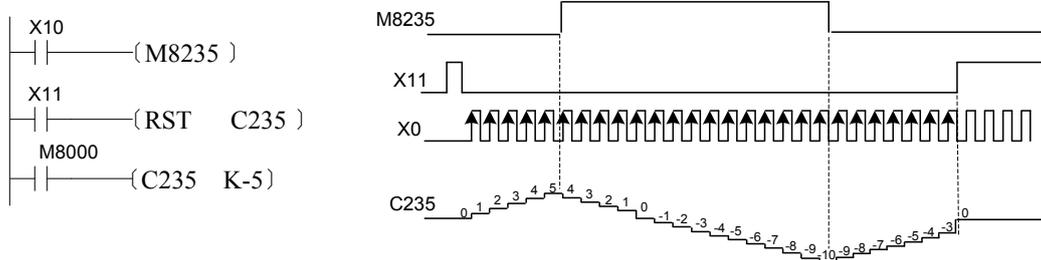
当M8200为OFF时，C200为增计数；当M8200为ON时，C200为减计数
和C200一样：

- 当M8201为OFF时，C201为增计数；当M8201为ON时，C201为减计数
- 当M8202为OFF时，C202为增计数；当M8202为ON时，C202为减计数
- 当M8203为OFF时，C203为增计数；当M8203为ON时，C203为减计数
- 当M8204为OFF时，C204为增计数；当M8204为ON时，C204为减计数
- 当M8205为OFF时，C205为增计数；当M8205为ON时，C205为减计数
- 当M8206为OFF时，C206为增计数；当M8206为ON时，C206为减计数
- 当M8207为OFF时，C207为增计数；当M8207为ON时，C207为减计数
- 当M8208为OFF时，C208为增计数；当M8208为ON时，C208为减计数
- 当M8209为OFF时，C209为增计数；当M8209为ON时，C209为减计数
- 当M8210为OFF时，C210为增计数；当M8210为ON时，C210为减计数
- 当M8211为OFF时，C211为增计数；当M8211为ON时，C211为减计数
- 当M8212为OFF时，C212为增计数；当M8212为ON时，C212为减计数
- 当M8213为OFF时，C213为增计数；当M8213为ON时，C213为减计数
- 当M8214为OFF时，C214为增计数；当M8214为ON时，C214为减计数
- 当M8215为OFF时，C215为增计数；当M8215为ON时，C215为减计数
- 当M8216为OFF时，C216为增计数；当M8216为ON时，C216为减计数

当M8217为OFF时，C217为增计数；当M8217为ON时，C217为减计数
 当M8218为OFF时，C218为增计数；当M8218为ON时，C218为减计数
 当M8219为OFF时，C219为增计数；当M8219为ON时，C219为减计数
 当M8220为OFF时，C220为增计数；当M8220为ON时，C220为减计数
 当M8221为OFF时，C221为增计数；当M8221为ON时，C221为减计数
 当M8222为OFF时，C222为增计数；当M8222为ON时，C222为减计数
 当M8223为OFF时，C223为增计数；当M8223为ON时，C223为减计数
 当M8224为OFF时，C224为增计数；当M8224为ON时，C224为减计数
 当M8225为OFF时，C225为增计数；当M8225为ON时，C225为减计数
 当M8226为OFF时，C226为增计数；当M8226为ON时，C226为减计数
 当M8227为OFF时，C227为增计数；当M8227为ON时，C227为减计数
 当M8228为OFF时，C228为增计数；当M8228为ON时，C228为减计数
 当M8229为OFF时，C229为增计数；当M8229为ON时，C229为减计数
 当M8230为OFF时，C230为增计数；当M8230为ON时，C230为减计数
 当M8231为OFF时，C231为增计数；当M8231为ON时，C231为减计数
 当M8232为OFF时，C232为增计数；当M8232为ON时，C232为减计数
 当M8233为OFF时，C233为增计数；当M8233为ON时，C233为减计数
 当M8234为OFF时，C234为增计数；当M8234为ON时，C234为减计数

高速单相单计数增减控制 M8235~M8245

单相单计数型高速计数输入，只有1个计数脉冲信号输入端，其计数的增减由程序利用对应的特殊M寄存器决定为增计数或减计数。



当M8235为OFF时，C235为增计数；当M8235为ON时，C235为减计数和C235一样：
 当M8236为OFF时，C236为增计数；当M8236为ON时，C236为减计数
 当M8237为OFF时，C237为增计数；当M8237为ON时，C237为减计数
 当M8238为OFF时，C238为增计数；当M8238为ON时，C238为减计数
 当M8239为OFF时，C239为增计数；当M8239为ON时，C239为减计数
 当M8240为OFF时，C240为增计数；当M8240为ON时，C240为减计数
 当M8241为OFF时，C241为增计数；当M8241为ON时，C241为减计数
 当M8242为OFF时，C242为增计数；当M8242为ON时，C242为减计数
 当M8243为OFF时，C243为增计数；当M8243为ON时，C243为减计数
 当M8244为OFF时，C244为增计数；当M8244为ON时，C244为减计数
 当M8245为OFF时，C245为增计数；当M8245为ON时，C245为减计数
 C241~C245具有硬件复位输入功能，部分具有硬件起停输入功能。

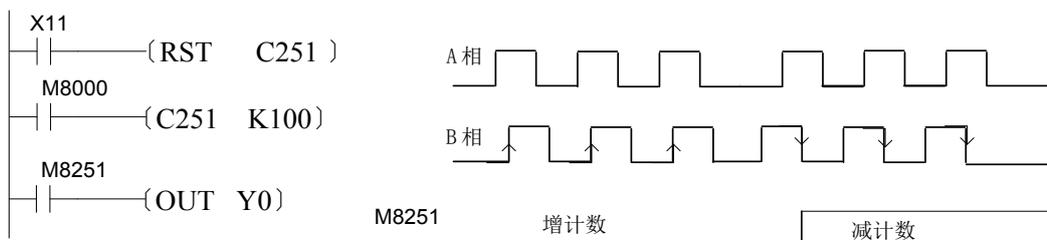
高速计数增减监控 M8246~M8255

单相2输入高速计数和AB相输入高速计数利用外部端口的输入，可自动增计数或减计数。

单相2计数型，有2个计数脉冲信号输入端，分别为增计数脉冲输入端和减计数脉冲输入端；部分计数器还具有硬件复位、起停的信号输入端口：

AB相计数型，是根据AB两相的相位决定计数的方向，计数方法是：当A脉冲为高电平时，B相的脉冲上升沿作加计数，B相的脉冲下降沿作减计数。

通过读取M8246-M8255的状态，可监控C246-C255的增计数 / 减计数状态。如下图C251的监控：



当C251为增计数时，M8251为OFF；当C251为减计数时，M8251为ON。通过监控M8251可以知道C251是增计数还是减计数，和C251一样：

当C246为增计数时，M8246为OFF；当C246为减计数时，M8246为ON。通过监控M8246可以知道C246是增计数还是减计数

当C247为增计数时，M8247为OFF；当C247为减计数时，M8247为ON。通过监控M8247可以知道C247是增计数还是减计数

当C248为增计数时，M8248为OFF；当C248为减计数时，M8248为ON。通过监控M8248可以知道C248是增计数还是减计数

当C249为增计数时，M8249为OFF；当C249为减计数时，M8249为ON。通过监控M8249可以知道C249是增计数还是减计数

当C250为增计数时，M8250为OFF；当C250为减计数时，M8250为ON。通过监控M8250可以知道C250是增计数还是减计数

当C252为增计数时，M8252为OFF；当C252为减计数时，M8252为ON。通过监控M8252可以知道C252是增计数还是减计数

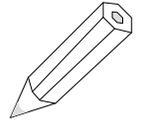
当C253为增计数时，M8253为OFF；当C253为减计数时，M8253为ON。通过监控M8253可以知道C253是增计数还是减计数

当C254为增计数时，M8254为OFF；当C254为减计数时，M8254为ON。通过监控M8254可以知道C254是增计数还是减计数

当C255为增计数时，M8255为OFF；当C255为减计数时，M8255为ON。通过监控M8255可以知道C255是增计数还是减计数。

Memo NO. _____

Date / /



A series of horizontal lines for writing, consisting of 25 evenly spaced lines that fill the majority of the page below the header information.

创变·精彩



官方微信



服务与技术支持APP

深圳市汇川技术股份有限公司

Shenzhen Inovance Technology Co., Ltd.

地址：深圳市宝安区宝城70区留仙二路鸿威工业区E栋

总机：(0755)2979 9595

传真：(0755)2961 9897

客服：400-777-1260

<http://www.inovance.com>

苏州汇川技术有限公司

Suzhou Inovance Technology Co., Ltd.

地址：苏州市吴中区越溪友翔路16号

总机：(0512)6637 6666

传真：(0512)6285 6720

客服：400-777-1260

<http://www.inovance.com>

销售服务联络地址

由于本公司持续的产品升级造成的内容变更，恕不另行通知
版权所有 © 深圳市汇川技术股份有限公司
Copyright © Shenzhen Inovance Technology Co., Ltd.