1 MEMOBUS 通信的构成

使用MEMOBUS 协议，可与MEMOCON 系列等可编程控制器（以下简称PLC）进行串行传输。

MEMOBUS 通信由1 台主站（PLC）和最多31 台从站构成。主站和从站的通信（串行通信）通常以主站开始通信、从站响应的方式进行。

主站同时和1 台从站间进行信号通信。因此，对各个从站预先设定地址编号，主站指定该编号进行信号通信。接到主站指令的从站执行指定的功能，对主站作出响应。

（**注意**：不是所有的主机询问帧，从机都会回答。比如主机广播，从机就不会响应）

2 通信规格

MEMOBUS 通信的规格如下表所示。

3、通讯错误处理

具体错误码，详见参数表

4、通讯超时（Over time）检出

此参数设定串联通讯通讯超时的检出时间。当在此参数设定时间内，无任何资料传输，即表是通讯超时，具体时间，详见参数表 F09.02。

5、BIT 流格式

MODBUS 通讯分为 RTU 和 ASCII 两种编码方式，此处编码按 RTU 方式直接传送，字符结构： 11 位，可以是下列 3 种格式任意之一。具体选择方式，请看参数表。

说明：H　表示十六进制.　　B　表示二进制

项目	规格
接口	RS-422、RS-485
通信参数	波特率：可从1200、2400、4800、9600、19200、38400、57600、76800、115200bps 中选择(F09.00 设置)
	数据长度：8 位固定
	校验：可从偶数/ 奇数/ 无中选择
	停止位：1 位固定
通信协议	MEMOBUS 基准（仅限RTU 模式）
可连接台数	最多31 台（使用RS-485 时）

6、通信资料结构（RTU 模式）

modbus 的 RTU 模式，数据格式：

10ms 间隔　+ **从站地址　+　功能码 + 具体数据 + CRC CHK** + 10ms 间隔

```
|                |            |            |
|                |            |            |
|                |            |            |____长度：2 字节
|                |            |_____________长度：多字节
|                |_____________________________长度：1 字节
|_____________________________________________长度：1 字节
```

(1) 功能码：

码能码	实现的功能
03H	读出寄存器内容
06H	写入一个字节的资料到寄存器
10H	写入多笔资料到寄存器

MEMOBUS 功能寄存器表：

F00.00	0000	F00.08	0008	F00.16	0010
F00.01	0001	F00.09	0009	F00.17	0011
F00.02	0002	F00.10	000A	F00.18	0012
F00.03	0003	F00.11	000B	F00.19	0013
F00.04	0004	F00.12	000C	F00.20	0014
F00.05	0005	F00.13	000D	F00.21	0015
F0006	0006	F0014	000E		
F0007	0007	F0015	000F		

以上是 F00 项的参数例表。其它类推

MEMOBUS 监控参数寄存器表

Pd.00	0d00	Pd.08	0d08	Pd.16	0d10
Pd.01	0d01	Pd.09	0d09	Pd.17	0d11
Pd.02	0d02	Pd.10	0d0A	Pd.18	0d12
Pd.03	0d03	Pd.11	0d0B	Pd.19	0d13
Pd.04	0d04	Pd.12	0d0C	Pd.20	0d14
Pd05	0d05	Pd13	0d0D	Pd.21	0d15
Pd06	0d06	Pd14	0d0E		
Pd07	0d07	Pd15	0d0F		

读故障代码寄存器为 0E00

运行停止寄存器：3000H

频率给定命令寄存器:3001

7、功能码对应的通讯举例：

（1）、 03H：读功能表内容

正常询问：

从站地址	01H
功能码	03H
寄存器	00H
	01H
资料量(word)	00H
	01H
CRC 检验码高位	D5H
CRC 检验码低位	CAH

正常回应：

从站地址	01H
功能码	03H
寄存器	00H
	01H
数据(word)	00H
	00H
CRC 检验码高位	14H
CRC 检验码低位	0AH

（2）、06H：更改一组数据到功能表中

正常询问：

从站地址	01H
功能码	06H
寄存器	00H
	02H
数据(word)	13H
	88H
CRC 检验码高位	E9H
CRC 检验码低位	5CH

正常**回应：**

从站地址	01H
功能码	06H
寄存器	00H
	02H
数据(word)	13H
	88H
CRC 检验码高位	E9H
CRC 检验码低位	5CH

说明：修改运行频率数字设定为 50.00Hz,如果写的数据超出了上下限，功能码的最高位(bit7)置 1(06H 变成 86H)，更改无效。数据原样返回.通信更改的数据掉电不保存。

（3）、08H:回路测试:

正常询问:

从站地址	01H
功能码	08H
寄存器	00H
	00H
资料(word)	00H
	00H
CRC 检验码高位	E0H
CRC 检验码低位	0BH

正常回应:

从站地址	01H
功能码	08H
寄存器	00H
	00H
资料(word)	00H
	00H
CRC 检验码高位	E0H
CRC 检验码低位	0BH

说明：只是测试通信是否正常，发什么数据就应什么数据.

（4）、10H：连续写入两笔资料

正常询问:

从站地址	01H
功能码	10H
寄存器起始地址	00H
	03H
资料量（word）	00H 02H
资料量（Byte）	04H
起始数据	00H 00H
下一数据	00H 01H
CRC 检验码高位	72H
CRC 检验码低位	7AH

正常回应：

从站地址	01H
功能码	10H
寄存器起始地址	00H
	03H
资料量（word）	00H 02H
CRC 检验码高位	B1H
CRC 检验码低位	C8H

说明：更改运行命令通道(0000)为键盘操作运行命令通道，运转方向（0001）为反转
最多只可以同时修改两个参数,如果数据超出上下限,将按照正常回应中的功能码最高位(bit7)置 1(10H 变成 80H)，如果第一个数据正确，第二个数据超出上下限，也会报错。但第一个数据将修改成功。通信更改的数据掉电不保存.

（5）、运行频率控制指令：

寄存器编号	内容	
3000H	Bit0～1	00B:无功能
		01B:停止
		10B:启动
		11B:JOG 启动
	Bit2～3	00B:运行
		保留
	Bit4～5	00B:无功能
		01B:正方向指令
		10B:反方向指令
		11B: 无功能
	Bit6～7	01:故障复位
	Bit8～11	保留

表 9-04

正常询问:

从站地址	01H
功能码	10H
寄存器	30H
	00H
资料量（word）	00H 02H
资料量(word)	04H
运行控制命令	00H 12H
给定 频率值	13H 88H
CRC 检验码高位	0AH
CRC 检验码低位	FDH

从站地址	01H
功能码	10H
寄存器	30H
	00H
资料量（word）	00H 02H
CRC 检验码高位	4EH
CRC 检验码低位	C8H

说明：先设置功能参数表 F00.00 为频率给定通道为通讯给定,运行命令通道 F00.02 为通讯命令给定.
运行频率为 50.00Hz,正转.
注意：运行命令和频率给定是相互独立的。

停止命令:

正常询问：

从站地址	01H
功能码	06H
寄存器	30H
	00H
控制指令	00H
	01H
CRC 检验码高位	47H
CRC 检验码低位	0AH

正常回应：

从站地址	01H
功能码	06H
寄存器	30H
	00H
控制指令	00H
	01H
CRC 检验码高位	47H
CRC 检验码低位	0AH

（5）、读取监控参数

例如：从驱动器地址为 1FH 的内部设定参数

为 0D00H（F0D.00）中读取监控参数值：

正常询问：

从站地址	01H
功能码	03H
寄存器	0DH
	00H
保留数据	00H
	00H
CRC 检验码高位	47H
CRC 检验码低位	66H

正常回应:

从站地址	1FH
功能码	03H
寄存器	0DH
	00H
读的数据	13H
	88H
CRC 检验码高位	4AH
CRC 检验码低位	30H

说明：0D00H（F0D：00）就是菜单号。读取的就是监控参数第一个菜单运行在 50.00Hz 时内容。

如果读数据帧出错,功能码最高位(bit7)置 1(03H 变成 83H).

（6）、读取电机运行状态

正常询问：

从站地址	01H
功能码	03H
寄存器	0EH
	00H
保留数据	00H
	00H
CRC 检验码高位	47H
CRC 检验码低位	22H

正常回应:

从站地址	01H
功能码	03H
寄存器	0EH
	00H
电机运行状态	01H
	00H
CRC 检验码高位	46H
CRC 检验码低位	B2H

故障时回应:

从站地址	01H
功能码	03H
寄存器	0EH
	80H
电机运行状态	00H
	0EH
CRC 检验码高位	46H
CRC 检验码低位	B2H

说明：在读电机运行状态时，如果有故障,寄存器的第七位(bit7)就会为 1，且电机运行状态的内容就会成为故障代码. 如果读数据帧出错,功能码最高位(bit7)置 1.

故障状态	1	加速运行中过流
	2	减速运行中过流
	3	匀速运行中过流
	4	加速运行中过压
	5	减速运行中过压
	6	匀速运行中过压
	7	停机时过压
	8	运行中欠压
	9	输入缺相
	10	功率模块故障
	11	散热器过热
	12	变频器过载
	13	电机过载
	14	外部设备故障
	15	RS485 通讯故障
	16	保留
	17	电流检测错误
	18	键盘通讯故障
故障代码 代码	19	控制板故障

说明:为 0 时没有故障

电机状态	0	保留
	1	保留
	2	加速中
	3	减速中
	4	保留
	5	保留
	6	保留
	7	保留
	8	运行准备就绪
	9	电机转向 0-正 1-反
	10	保留
	11	指令方向 0-正 1-反
	12	运行中
	13	保留
	14	保留
	15	保留

（7）、RTU 模式的检查码（CRC Check）

检查码由 Address 到 Data content 结束。

其运算规则如下：

步骤 1：令 16-bit 暂存器 (CRC 暂存器) = FFFFH.

步骤 2：Exclusive OR 第一个 8-bit byte 的讯息指令与低位元 16-bit CRC 暂存器, 做 Exclusive OR ，将结果存入 CRC 暂存器内。

步骤 3：右移一位 CRC 暂存器，将 0 填入高位元處。

步骤 4:检查右移的值,如果是 0, 将步骤 3 的新值存入 CRC 暂存器内, 否则 Exclusive OR A001H 与 CRC 暂存器，将结果存入 CRC 暂存器内。

步骤 5：重复步骤 3~步骤 4，将 8-bit 全部运算完成。

步骤 6：重复步骤 2~步骤 5，取下一个 8-bit 的讯息指令，直到所有讯息指令运算完成。最后，得到的 CRC 暂存器的值，即是 CRC 的检查码。值得注意的是 CRC 的检查码必须交换放置于讯息指令的检查码中。

以下为用 C 语言所写的 CRC 检查码运算范例：

```
unsigned char* data     // 讯息指令指标
unsigned char length    // 讯息指令的长度
unsigned int crc_chk(unsigned char* data, unsigned char length)
{
int j;
unsigned int reg_crc=0xffff;
while(length--)
{
   reg_crc ^= *data++;
   for(j=0;j<8;j++)
   {
     if(reg_crc & 0x01)
      { /* LSB(b0)=1 */
        reg_crc=(reg_crc>>1) ^ 0Xa001;
      }
       Else
       {
       reg_crc=reg_crc >>1;
       }
    }
}
return reg_crc;// 最后回传 CRC 暂存器的值
}
```