● 通讯协议

- ➢ 采用 MODBUS 协议，主机询问，处于主动状态；从机回答，处于被动状态。

  （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）

● 通讯地址

- ➢ 设定范围： 01~31
- ➢ 当系统使用 RS-485 串联通讯介面控制或监控时，每一台驱动器必须设定其通讯地址且每一个连结网中每个地址均为”唯一”不可重复。
- ➢ 出厂设定值：01

● 通讯传送速度 Baud Rate

- ➢ 波特率大小，详见参数表

● 通讯错误处理

- ➢ 具体错误码，详见参数表

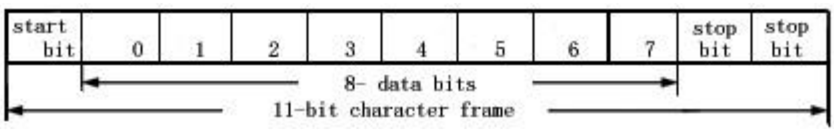
● 通讯超时（Over time）检出

- ➢ 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。

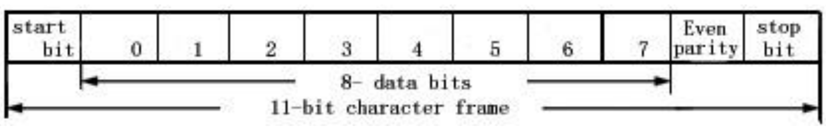
● BIT 流格式

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。

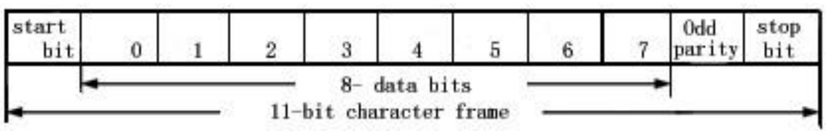
- ➢ (资料格式 8，N，2)



- ➢ (资料格式 8，E，1)



- ➢ (资料格式 8，O，1)



● 通信资料结构（RTU 模式）

任何一帧 modbus 的 RTU 模式，数据格式如下：

```
10ms 间隔   + 从机地址   +   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
```

下面列表也许更直观，但含义不变：

STX	保持无输入讯号大于等于 10ms
Address	通信地址：8-bit 二进制地址
Function	功能码：8-bit 二进制地址
DATA(n-1)	资料内容： n×8-bit 资料，n<＝2（2 笔 16bit 资料）
……	
DATA 0	
CRC CHK Low	CRC 检查码： 16-bit CRC 检查码由 2 个 8-bit 二進制組合
CRC CHK High	
END	保持无输入讯号大于等于 10ms

表格中各项具体含义如下：

- Address：通讯的地址，范围 0 ~ 31 （十进制）
  - 00H：所有驱动器广播（Broadcast）,广播帧从机不回应。
    （想想，如果从机都同时回应，主机怎样接收？会造成数据风暴。）
  - 01H ~ 1FH：对第具体某一地址驱动器。

- Function：功能码，也叫命令字节，有 4 种可能：
  - **03H**：读出寄存器内容。
  - **06H**：写入一笔资料到寄存器
  - **08H**：回路侦测
  - **10H**：写入多笔资料到寄存器
- DATA(n-1)：具体数据，下面会有应用实例。

- RTU 模式的检查码（CRC Check），详细介绍在最后一页。

● 功能码对应的通讯帧举例：

◆ 03H：读出寄存器内容

例如：从驱动器地址为 1FH 的内部设定参数为 0006H（F006）中读取参数值：

询问讯息帧格式：

Address	1FH
Function	03H
Starting data address	00H
	06H
Sizes	00H
	01H
CRC CHK Low	67H
CRC CHK High	B5H

回应讯息帧格式：

Address	1FH
Function	03H
data address	00H
	06H
Data content	10H
	88H
CRC CHK Low	ABH
CRC CHK High	D3H

询问帧：1FH+03H+00H+06H+00H+01H+67H+B5H

具体含义如下：

- Address ： 1FH ---- 该设备 ID 是 1FH。
- Function ： 03H ---- 读出寄存器内容。
- Starting data address：0006H ----寄存器地址为 0x0006，表示从该寄存器读取参数。
- Sizes ： 0001H ----读取 1 个地址的数据。
- CRC CHK：参考最后一页的 RTU 模式的检查码（CRC Check）获取方法。

回应帧：1FH+03H+00H+06H+10H+88H+ABH+D3H

具体含义如下：

- Address ： 1FH ---- 该设备 ID 是 1FH。
- Function ： 03H ---- 读出寄存器内容。
- data address ： 0006H ----寄存器地址为 0x0006，表示已从该寄存器读取。
- Data content : 1088H----表示读出的内容。

- CRC CHK：参考最后一页的 RTU 模式的检查码（CRC Check）获取方法。

◆ 06H：写入一笔资料到寄存器

例如：对驱动器地址 1FH，写入 5000（1388H）到驱动器内部设定参数 0006H。

询问讯息帧格式：

Address	1FH
Function	06H
Data address	00H
	06H
Data content	13H
	88H
CRC CHK Low	67H
CRC CHK High	23H

回应讯息格式：

Address	1FH
Function	06H
Data address	00H
	06H
Data content	13H
	88H
CRC CHK Low	67H
CRC CHK High	23H

询问帧：1FH+06H+00H+06H+13H+88H+67H+23H

具体含义如下：

- Address ： 1FH ---- 该设备 ID 是 1FH。
- Function ： 06H ---- 写寄存器内容。
- Data address ： 0006H ----寄存器地址为 0x0006，表示写内容到寄存器。
- Data content ： 1388H ----写的内容，在 0x0006 写入 1388H。
- CRC CHK：参考最后一页的 RTU 模式的检查码（CRC Check）获取方法。

回应帧：1FH+06H+00H+06H+13H+88H+67H+23H

具体含义如下：

- Address ： 1FH ---- 该设备 ID 是 1FH。
- Function ： 06H ----写寄存器内容。
- data address ： 0006H ----寄存器地址为 0x0006，表示从该寄存器读取参数。
- Data content : 1388H----表示写进寄存器的内容。
- CRC CHK：参考最后一页的 RTU 模式的检查码（CRC Check）获取方法。

◆ 10H：连续写入数笔资料（只能连续修改两个参数）

例如，变更驱动器（地址 1FH）的上下限频率设定 00-06=50.00(1388H),00-07=00.01(0001H)

询问帧：

Address	1FH
Function	10H
资料 起始地址	00H 06H
资料量（word）	00H 02H
资料量（Byte）	04H
第一笔资料	13H 88H
第二笔资料	00H 01H
CRC CHK Low	56H
CRC CHK High	C3H

回应帧：

Address	1FH
Function	10H
资料 起始地址	00H 06H
资料量（word）	00H 02H
CRC CHK Low	A2H
CRC CHK High	77H

询问帧：1FH+10H+00H+06H+00H+02H+04H+13H+88H+00H+01H+56H+C3H

具体含义如下：

- Address ：1FH ---- 该设备 ID 是 1FH
- Function ：10H ---- 写寄存器内容
- 起始地址 ：0006H ----寄存器起始地址为 0x0006，表示写内容到 0x0006，0x0007。
- 资料量（word） ：0002H ----写的内容的字量。
- 资料量（Byte） ：04------写的内容的字节量。
- 第一笔资料 ：1388H 写的第一笔内容。
- 第二笔资料 ：0001H 写的第二笔内容。
- CRC CHK：参考最后一页的 RTU 模式的检查码（CRC Check）获取方法。

回应帧：1FH+10H+00H+06H+00H+02H+A2H+77H

具体含义如下：

- Address ：1FH ---- 该设备 ID 是 1FH
- Function ：10H ---- 写寄存器内容
- 起始地址 ：0006H ----寄存器起始地址为 0x0006，表示写内容到 0x0006，0x0007。
- 资料量（word）：0002H ----写的内容的字量。
- CRC CHK：参考最后一页的 RTU 模式的检查码（CRC Check）获取方法。

请特别注意，回应帧只返回询问帧前面的六个字节，其 CRC CHK 是这六个字节的 CRC CHK。

- **通信协议的参数字址定义，如下表：**

定 义	参数字址	功 能 说 明	
驱动器内部设定参数	GGnnH	GG 表示参数群，nn 表示参数号码。例如：04-01 由 0401H 来表示。	
对驱动器的命令	2000H	Bit0～1	00B:无功能
			01B:停止
			10B:启动
			11B:JOG 启动
		Bit2～3	00B:一次运行
			11B:连续运行
		Bit4～5	00B:无功能
			01B:正方向指令
			10B:反方向指令
			11B: 无功能
		Bit6～7	保留
		Bit8～11	保留
	2001H	频率命令	

写入启动正转命令及其运转频率给定举例(连续运行)：

询问帧：

Address	1FH
Function	10H
参数字址	20H
	00H
资料量（word）	00H 02H
资料量（Byte）	04H
驱动命令	00H 1EH
给定 频率值	10H 88H
CRC CHK Low	67H
CRC CHK High	E6H

回应帧：

Address	1FH
Function	10H
参数字址	20H
	00H
资料量（word）	00H 02H
CRC CHK Low	49H
CRC CHK High	B6H

写入启动正转命令及其运转频率给定举例(一次运行)：

询问帧：

Address	1FH
Function	10H
参数字址	20H
	00H
资料量（word）	00H 02H
资料量（Byte）	04H
驱动命令	00H 12H
给定 频率值	10H 88H
CRC CHK Low	A7H
CRC CHK High	E5H

回应帧：

Address	1FH
Function	10H
参数字址	20H
	00H
资料量（word）	00H 02H
CRC CHK Low	49H
CRC CHK High	B6H

写入启动反转命令及其运转频率给定举例(连续运行)：

询问帧：

Address	1FH
Function	10H
参数字址	20H
	00H
资料量（word）	00H 02H
资料量（Byte）	04H
驱动命令	00H 2EH
给定 频率值	10H 88H
CRC CHK Low	67H
CRC CHK High	E9H

回应帧：

Address	1FH
Function	10H
参数字址	20H
	00H
资料量（word）	00H 02H
CRC CHK Low	49H
CRC CHK High	B6H

写入停止命令举例：

询问帧：

Address	1FH
Function	06H
参数字址	20H
	00H
驱动命令	00H
	01H
CRC CHK Low	40H
CRC CHK High	74H

回应帧：

Address	1FH
Function	06H
参数字址	20H
	00H
驱动命令	00H
	01H
CRC CHK Low	40H
CRC CHK High	74H

● 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（FD00）中读取监控参数值：

询问讯息帧格式：

Address	1FH
Function	03H
data address	0DH
	00H
保留数据	00H
	00H
CRC CHK Low	44H
CRC CHK High	D8H

回应讯息帧格式：

Address	1FH
Function	03H
data address	0DH
	00H
B0B1	10H
	84H
B2B3	41H
	48H
CRC CHK Low	47H
CRC CHK High	D5H

其中，询问帧中的保留数据对读取结果没有影响，可以任意设计。而 B0B1,B2B3 则参考下面的表。(请注意，读取地址超过 0D28 无效)

B0~B3 参数表：

B0 B1		监控 1 参数值
B2 B3	0	负 1 位小数,显示最后点
	1	0 位小数
	2	1 位小数
	3	2 位小数
	4	3 位小数
	5	V
	6	Hz
	7	A
	8	参数正常
	9	保留参数
	10	*
	11	*
	12	输入端子格式
	13	故障格式
	14	停止时，显示 0
	15	?

- **读取故障代码**

例如：从驱动器地址为 1FH 的内部设定参数为 0E01H（FE01）中读取故障代码：

询问讯息帧格式：

Address	1FH
Function	03H
data address	0EH
	01H
保留数据	00H
	00H
CRC CHK Low	15H
CRC CHK High	5CH

回应讯息帧格式：

Address	1FH
Function	03H
data address	0EH
	01H
B4B5	FFH
	FFH
B6B7	01H
	48H
CRC CHK Low	0FH
CRC CHK High	2BH

其中，询问帧中的保留数据对读取结果没有影响，可以任意设计。如果没有故障发生，则 B4B5 返回 FFFFH，否则返回的 B4B5B6B7 参考下面的表。

B4~B7 参数表：

B4 B5		没有故障显示 FFFF, 如有故障则(5~11 位)为故障代码	
B6 B7	0	电压正常	
	1	电机转向 0-正 1-反	
	2	输出相序 0-正 1-反	
	3	指令方向 0-正 1-反	
	4	运行中	
	5	故障中，故障代码在 B4B5 的 5~11 位中	
	6	频率通道为 0（数字设定）	
	7		
	8		
	9		
	10	加速中	
	11	减速中	
	12		
	13		
	14		
	15		

● 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, 否則 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
}
```