

# HPLA+ Modbus 标准通讯协议格式

## ASCII 格式

### 通信资料格式

STX “:” (3AH)	ADDRESS	FUNC	DataH	DataL	LRC	END CR(0DH) LF(0AH)
---------------------	---------	------	-------	-------	-----	---------------------------

1) STX: 起始单元 “:” (3AH)

2) ADDR 通讯位置 8-bit 位置包含了 2 个 ASCII 码

00: 广播方式为 MODBUS

01-250: 相应变频器地址

3) FUNC: 命令码 8-bit 命令

#### 01 读线圈

上位机发送数据格式:

“:” ADDRESS 01 ADDRH ADDRL NUMH NUML LRC 0X0D  
0X0A

注: ADDR: 0000 --- FFFF; NUM: 0001----0020 (16 进制数) (NUM 为要读的线圈的数量)

正确时变频器返回数据格式:

“:” ADDRESS 01 BYTECOUNT DATA1 DATA2 DATA3 DATAN  
LRC 0X0D 0X0A

注: BYTECOUNT= NUM / 8 或 BYTECOUNT = NUM / 8 + 1 (取整或取整加一)

错误时变频器返回数据格式:

“:” ADDRESS 0X81 01 or 02 or 03 or 04 LRC 0X0D 0X0A

如: 要检测变频器运行方向(假如变频器反转运行)(CD160=CD161=1, CD162=0)

应发送数据: “:” 01 01 00 05 00 01 F9 0D 0A (16 进制), 转为 ASCII 码

格式为: 3A 30 31 30 31 30 30 30 35 30 30 30 31 46 39 0D 0A (ASCII 码)

变频器返回数据: “:” 01 01 01 01 FC 0D 0A (16 进制), 转为 ASCII 码

格式为: 3A 30 31 30 31 30 31 30 31 46 43 0D 0A (ASCII)

返回的数据位为 “01” (16 进制) 化为 2 进制数为 “0000 0001” 末位表示运行状态为 “1” 表示变频器正在反转运行 (如是正转或停止状态则为 “0” 详见线圈说明)

#### 03 读保持寄存器

上位机发送数据格式:

“:” ADDRESS 03 ADDRH ADDRL NUMH NUML LRC 0X0D  
0X0A

注: ADDR: 0 --- 0xFFFF; NUM: 0001---0004 (NUM 为要读的保持寄存器的数量)

正确时变频器返回数据格式:

“:” ADDRESS 03 BYTECOUNT DATA1 DATA2 DATA3 DATAN  
LRC 0D 0A

**注: BYTECOUNT = 2 \* NUM**

错误时变频器返回数据格式:

“:” ADDRESS 0X83 01 or 02 or 03 or 04 LRC 0X0D 0X0A

如: 要读变频器功能码 CD000 中的设定值 (假如为 30.00HZ) (CD160=CD161=1 CD162=0)

应发送数据: “:” 03 00 00 00 01 FC 0D 0A (16 进制), 转为 ASCII 码格式为  
3A 30 33 30 30 30 30 30 30 30 31 46 43 0D 0A (ASCII)

变频器返回数据: “:” 01 03 02 0B B8 37 0D 0A (16 进制), 转为 ASCII 码格式  
为:

3A 30 31 30 33 30 32 30 42 42 38 33 37 0D DA

返回的数据位为 “0BB8” (16 进制) 转换为 10 进制数为 3000 表示 CD000 设置值为  
30.00

#### 04 读取输入寄存器

上位机发送数据格式:

“:” ADDRESS 04 ADDRH ADDR L NUMH NUM L LRC 0X0D  
0X0A

**注: ADDR: 0 ---- 0xFFFF; NUM: 0001 ---- 0004 (NUM 为要读的输入寄存器的数量)**

正确时变频器返回数据格式:

“:” ADDRESS 04 BYTECOUNT DATAH1 DATA L1 ---- DATAH N  
DATA L N

**注: BYTECOUNT = NUM \* 2**

错误时变频器返回数据格式:

“:” ADDRESS 0X84 01 or 02 or 03 or 04 LRC 0X0D 0X0A

如: 要读变频器温度显示值 (假如温度为 36.2) (CD160=CD161=1 CD162=0)

应发送数据: “:” 01 04 00 06 00 01 F4 0D 0A (16 进制), 转为 ASCII 格式  
为:

3A 30 31 30 34 30 30 30 36 30 30 30 31 46 34 0D 0A (ASCII)

变频器返回数据: “:” 01 04 02 01 6A 8E 0D 0A (16 进制), 转为 ASCII 码格式  
为

3A 30 31 30 34 30 32 30 31 36 41 38 45 0D 0A (ASCII)

返回的数据位为 “016A” (16 进制) 转换为 10 进制为 “362” 表示温度为 36.2

#### 05 写单个线圈状态

上位机发送数据格式:

“:” ADDRESS 05 ADDRH ADDR L DATAH DATA L LRC 0X0D  
0X0A

**注: ADDR: 0 ---- 0xFFFF DATA: 0X0000 or 0XFF00 (16 进制数)**

正确时变频器返回数据格式:

“:” ADDRESS 05 ADDRH ADDR L DATAH DATA L LRC 0X0D  
0X0A

错误时变频器返回数据格式:

“:” ADDRESS 0X85 01 or 02 or 03 or 04 LRC 0X0D 0X0A

## 06 写单个保持寄存器值

上位机发送数据格式:

“ : “ ADDRESS 06 ADDRH ADDRL DATAH DATAL LRC 0X0D  
0X0A

**注:ADDR: 0 ---- 0xFFFF**

正确时变频器返回数据格式:

“ : “ ADDRESS 06 ADDRH ADDRL DATAH DATAL LRC 0X0D  
0X0A

错误时变频器返回数据:

“ : ” ADDRESS 0X86 01 or 02 or 03 or 04 LRC 0X0D 0X0A

如: 要对变频器功能码 CD000 写入 30.00 (CD160=CD161=1 CD162=0),

应发送数据: “ : ” 01 06 00 00 0B B8 36 0D 0A (16 进制), 转为 ASCII 格

式为

3A 30 31 30 36 30 30 30 30 30 30 42 42 38 33 36 0D 0A (ASCII)

变频器返回数据: “ : ” 01 06 00 00 0B B8 36 0D 0A (16 进制), 转为 ASCII

格式为

3A 30 31 30 36 30 30 30 30 30 30 42 42 38 33 36 0D 0A (ASCII)

返回的数据位为 “0BB8” (16 进制) 转换为 10 进制为 “3000” 表示 CD000 设置值为 30.00

## 0F 写多个线圈状态

上位机发送数据格式

“ : “ ADDRESS 0F ADDRH ADDRL NUMH NUML COUNT  
DATAH1 DATAL1 DATA2H DATA2L ----- DATANH DATANL LRC  
0X0D 0X0A

**注:ADDR: 0 ----- 0xFFFF ; NUM: 1 ----- 1968 / 0X7B0 ; COUNT: NUM / 8 或  
NUM/ 8 + 1 (NUM 为要写的线圈的数量)**

正确时变频器返回数据格式:

“ : “ ADDRESS 0F ADDRH ADDRL NUMH NUML LRC 0X0D  
0X0A

错误时变频器返回数据格式:

“ : ” ADDRESS 0X8F 01 or 02 or 03 or 04 LRC 0X0D 0X0A

如: 要使变频器正转 (CD160=CD161=1 CD162=0),

应发送数据: “ : ” 01 0F 00 48 00 08 01 02 LRC 0D 0A (16 进制), 转为 ASCII

格式为

3A 30 31 30 46 30 30 34 38 30 30 30 38 30 31 30 30 30 32 LRC 0D

0A (ASCII)

“0002” 即 0000 0010; 即线圈 0x49 置为 1, 给变频器发正转命令。

变频器返回数据: “ : ” 01 0F 00 48 00 08 LRC 0D 0A (16 进制), 转为 ASCII

格式为

3A 30 31 30 46 30 30 34 38 30 30 30 38 LRC 0D 0A (ASCII)

## 10 写多个保持寄存器

上位机发送数据格式

“ : “ ADDRESS 0x10 ADDRH ADDRL NUMH NUML  
BYTECOUNT DATAH1 DATAL1 DATA2H DATA2L ----- DATANH

DATANL LRC 0X0D 0X0A

注:ADDR: 0-----0XFFFF; NUM: 0-125 (NUM 为要写的保持寄存器的数量)

正确时变频器返回数据格式

“:” ADDRESS 0x10 ADDRH ADDRL NUMH NUML LRC 0X0D 0X0A

COUNT= NUM \* 2

错误时变频器返回数据格式:

“:” ADDRESS 0X90 01 or 02 or 03 0r 04 LRC 0X0D 0X0A

如要同时对 CD000 写入 30.00, 对 CD001 写入 110.0, (CD160=CD161=1 CD162=0)

应发送数据:“:”01 10 00 00 00 02 04 0B B8 04 4C D6 0D 0A (16 进制), 转为 ASCII 格式:

3A 30 31 31 30 30 30 30 30 30 30 30 32 30 34 30 42 42 38 30 34 34 43 44 36 0D 0A (ASCII)

变频器返回数据:“:” 01 10 00 00 00 02 45 44 0D 0A (16 进制), 转为 ASCII 格式为:

3A 30 31 31 30 30 30 30 30 30 30 30 32 34 35 34 34 0D 0A (ASCII)

发送的要写入 16 进制数据 “0BB8” 和 “04 4C” 转换为 10 进制数分别为 3000 和 1100 表示对 CD000 和 CD001 写入的数据分别为 30.00 和 110.0

通信错误代码说明:

- 01 非法的功能码
- 02 非法数据地址
- 03 非法数据值
- 04 设备失败

功能码说明

线圈功能码说明

线圈地址	名称	R/W	说明
0000	保留	R	
0001	保留	R	
0002	保留	R	
0003	运行	R	0 – 停止 1 – 运行
0004	点动	R	0 – 无效 1 – 点动
0005	正/反转	R	0 – 正转 1 – 反转
0006	制动	R	0 – 无效 1 – 制动 P02-00 = 0
0007	频率跟踪	R	0 – 无效 1 – 频率跟踪 P02-00 = 1
0008	IGBT 短路	R	0 – 无效 1 – 错误 ‘OC’
0009	CT 检测过流 high	R	0 – 无效 1 – 错误 ‘Oc’
000A	CT 检测过流 low	R	0 – 无效 1 – 错误 “oc”
000B	对地短路	R	0 – 无效 1 – 错误 ‘GF’
000C	过压	R	0 – 无效 1 – 错误 ‘OU’
000D	保险丝熔断	R	0 – 无效 1 – 错误 ‘FB’
000E	低压	R	0 – 无效 1 – 错误 ‘Lu’

000F	变频器过热	R	0 – 无效 1 – 错误 ‘OH’
0010	变频器过载	R	0 – 无效 1 – 错误 ‘OL’
0011	马达过载	R	0 – 无效 1 – 错误 ‘OA’
0012	马达过转矩	R	0 – 无效 1 – 错误 ‘OT’
0013	接触器坏掉	R	0 – 无效 1 – 错误 ‘LU’
0014	制动管坏	R	0 – 无效 1 – 错误 ‘BT’
0015	CPU 故障	R	0 – 无效 1 – 错误 ‘FE’
0016	存储器坏掉	R	0 – 无效 1 – 错误 ‘BE’
0017	总故障指示	R	0 – 无效 1 – 错误 ‘KE’
0018	变频器过载	R	0 – 无效 1 – 报警 ‘OL’
0019	马达过载	R	0 – 无效 1 – 报警 ‘OA’
001A	马达过转矩	R	0 – 无效 1 – 报警 ‘OT’
001B	温度偏高	R	0 – 无效 1 – 报警 ‘OH’
001C	急停	R	0 – 无效 1 – 报警 ‘ES’
001E	4- 20MA 断线	R	0 – 无效 1 – 报警 ‘20’
001F	参数设置错误	R	0 – 无效 1 – 报警 ‘PR’
0020	外部开关量输入 FB	R	0—无效 1—有效
0021	外部开关量输入 MCS	R	0—无效 1—有效
0022	外部开关量输入 FOR	R	0—无效 1—有效
0023	外部开关量输入 REV	R	0—无效 1—有效
0024	外部开关量输入 SPL	R	0—无效 1—有效
0025	外部开关量输入 SPM	R	0—无效 1—有效
0026	外部开关量输入 SPH	R	0—无效 1—有效
0027	外部开关量输入 RST	R	0—无效 1—有效
0028-002F	状态输出 MLTIOUT1	R	0—无效 1—有效
0030-0037	状态输出 MLTIOUT2	R	0—无效 1—有效
0038-003F	状态输出 MLTIOUT3	R	0—无效 1—有效
0040-0047	状态输出 MLTIOUT4	R	0—无效 1—有效
0048	RUN	W	0—无效 1—有效
0049	FOR	W	0—无效 1—有效
004A	REV	W	0—无效 1—有效
004B	STOP	W	0—无效 1—有效
004C	F/R	W	0—无效 1—有效
004D	JOG	W	0—无效 1—有效
004E	JOGF	W	0—无效 1—有效
004F	JOGR	W	0—无效 1—有效

#### 保持寄存器功能码说明

功能码地址与键盘的功能码参数数值显示一样

#### 输入寄存器功能码地址说明

输入寄存器地址	名称	R/W	说明
---------	----	-----	----

0000	输出频率	R	
0001	设定频率	R	
0002	输出电流	R	
0003	输出转速	R	
0004	直流电压	R	
0005	交流电压	R	
0006	温度	R	
0007	计数器	R	
0008	反馈值	R	
0009	目标值	R	
000A	当前运行时间	R	
000B	总运行时间		

#### 4) DATA: 资料内容 $n \times 8\text{-bit}$ 资料

#### 5) LRC: 侦误值

ASCII 模式，采用 LRC (Longitudinal Redundancy Check) 侦误值。

LRC 侦误值乃是将 ADDR 至最后一个资料内容加总。得到结果以 256 单位，超出部分去除（如结果为 11128H 则取 1128H）然后计算二次反补后得到结果即为 LRC 侦误值。

### RTU 模式

静音	ADDR	FUNC	DATAH	DATAL	CRCH	CRCL	静音
> 50ms							>50ms

### 功能码数据同 HOLLIT 标准协议中 ASCII 格式中的功能码

- 校验码 CRC 算法同 HOLLIT 普通协议中 RTU 格式中的 CRC 算法

### 范例：

#### 01 读线圈

如：要检测变频器运行方向（假如变频器反转运行）（CD160=CD161=1，CD162=3）

应发送数据： 01 01 00 05 00 01 CRC（16 进制），

变频器返回数据： 01 01 01 01 CRC（16 进制）

返回的数据位为“01”（16 进制）化为 2 进制数为“0000 0001”末位表示运行状态为“1”表示变频器正在反转运行（如是正转或停止状态则为“0”详见线圈说明）

#### 03 读保持寄存器

如：要读变频器功能码 CD000 中的设定值（假如为 30.00HZ）（CD160=CD161=1 CD162=3）

应发送数据： 03 00 00 00 01 CRC（16 进制）

变频器返回数据： 01 03 02 0B B8 CRC（16 进制）

返回的数据位为“0BB8”（16 进制）转换为 10 进制数为 3000 表示 CD000 设置值为 30.00

#### 04 读取输入寄存器

如：要读变频器温度显示值（假如温度为 36.2）（CD160=CD161=1 CD162=3）

应发送数据： 01 04 00 06 00 01 CRC（16 进制）

变频器返回数据： 01 04 02 01 6A CRC（16 进制）

返回的数据位为“016A”（16 进制）转换为 10 进制为 “362”表示温度为 36.2

#### 05 写单个线圈状态

如：要使变频器的正转无效，反转有效（CD160=CD161=1 CD162=3）

应发送数据： 01 05 00 49 00 00 FF 00 CRC（16 进制）

变频器返回数据： 01 05 00 49 00 00 FF 00 CRC（16 进制）

“0000”代表无效;“FF00”代表有效

#### 06 写单个保持寄存器值

如：要对变频器功能码 CD000 写入 30.00（CD160=CD161=1 CD162=3），

应发送数据： 01 06 00 00 0B B8 36 0D 0A（16 进制）

变频器返回数据： 01 06 00 00 0B B8 36 0D 0A（16 进制）

返回的数据位为“0BB8”（16 进制）转换为 10 进制为 “3000”表示 CD000 设置值为 30.00

#### 0F 写多个线圈状态

如：要使变频器的正转无效（CD160=CD161=1 CD162=3）

应发送数据： 01 0F 00 48 00 08 01 00 02 CRC（16 进制）

变频器返回数据： 01 0F 00 48 00 08 CRC（16 进制）

“0002”即 0000 0010; 即线圈 0x49 置为 1，给变频器发正转命令。

#### 10 写多个保持寄存器

如要同时对 CD000 写入 30.00，对 CD001 写入 110.0，（CD160=CD161=1 CD162=3）

应发送数据： 01 10 00 00 00 02 04 0B B8 04 4C CRC（16 进制）

变频器返回数据： 01 10 00 00 00 02 45CRC（16 进制）

发送的要写入 16 进制数据“0BB8”和“04 4C”转换为 10 进制数分别为 3000 和 1100 表示对 CD000 和 CD001 写入的数据分别为 30.00 和 110.0

HLP-A.C+海利 MODBUS 通讯规约

在使用 RS485 通讯界面时，每一台变频器必须设定其通讯位置，电脑便根据每一台变频器个别位置实施控制。

- 1: 本通讯协议有二种方式:
- 1) RTU 方式 (Remote Terminal Unit) 模式
  - 2) ASCII 方式 (American Standard Code for information interchange) 模式

RTU 模式:

每个 8-bit 资料由两个 4-bit 十六进位字元组成，如：64H

ASCII 模式:

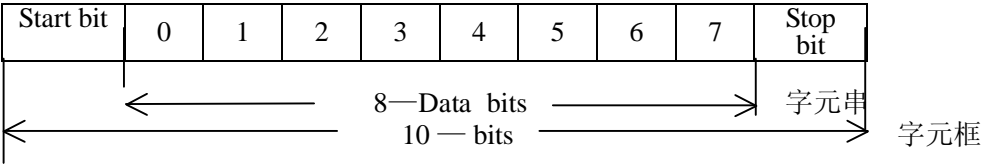
每个 8-bit 资料由两个 ASCII 字元组成，如：一个 1-bit 资料 64H (十六进位) 以 ASCII “64” 表示，包含 6 (36H) 和 4 (34H)

字元符号	0	1	2	3	4	5	6	7
ASCII 码	30H	31H	32H	33H	34H	35H	36H	37H

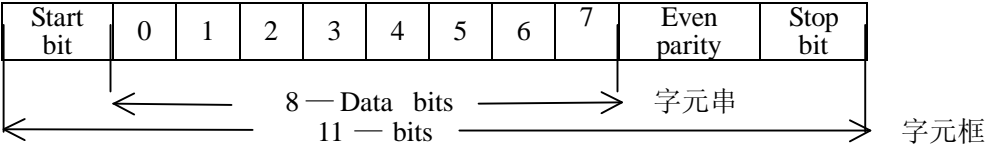
字元符号	8	9	A	B	C	D	E	F
ASCII 码	38H	39H	41H	42H	43H	44H	45H	46H

2: 通讯资料方式

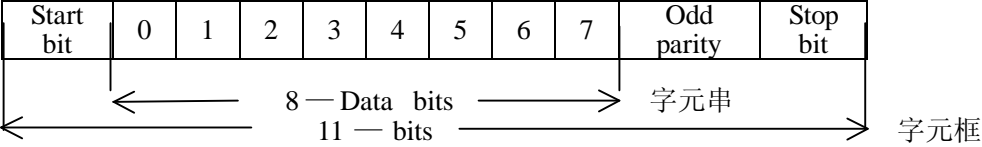
1) 8N1 For ASCII CD162=0



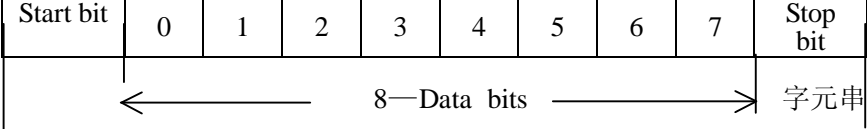
2) 8E1 For ASCII CD162=1



3) 8O1 For ASCII CD162=2



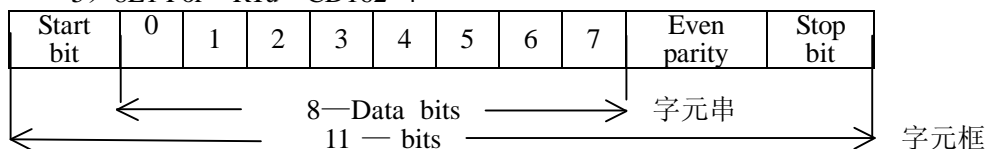
4) 8N1 For RTu CD162=3



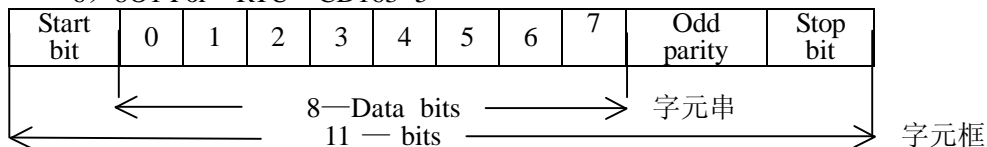


← 11 — bits → 字节框

5) 8E1 For RTu CD162=4



6) 8O1 For RTU CD163=5



### 3: 通讯资料格式

#### 3.1 ASCII 模式

##### 通讯资料格式

STX “:” (3AH)	ADDR	FUNC	LEN	DATE <sub>(n-1)</sub> ...DATA <sub>0</sub>	LRC	END CR(0DH) LF(0AH)
---------------------	------	------	-----	--	-----	---------------------------

1) STX: 启始单元 “:” (3AH)

2) ADDR 通讯位置 8-bit 位置包含了 2 个 ASCII 码

00: 广播方式为 MODBUS

01—250 相应变频器地址

3) FUNC: 功能码 8-bit 位置包含了 2 个 ASCII 码

01: FUNC READ 读取功能码数据

02: FUNC WRIT 功能码设定

03: 控制命令

04: 控制状态读取

05: 变频器串口频率设定

06: 保留

07: 保留

08: 回路检测

#### a. 读取功能码数据 (01)

发送格式: ADDR 01 LEN Data

ADDR=0 时, 无返回数据

ADDR≠0 且符合变频器地址时有响应

当返回为正常时, 格式如下:

ADDR 01 LEN Data

如返回为一个字时 LEN=3 一个字节时 LEN=2

当无此功能或无效时返回为:

ADDR 81H 01

例如：读取 CD000 的参数

发送： 3A 30 31 30 31 30 31 30 30 LRC 0D 0A

“30 30”即 00。代表 CD000。

接收： 3A 30 31 30 31 30 33 30 30 31 33 38 38 LRC 0D 0A

说明：CD000 = 01388H = 5000,即 50Hz.

## b.功能码设定 (02)

发送格式：ADDR 02 LEN Data

ADDR=0 作广播用，可以设定，但无应答

ADDR≠0 时可以设定 同时有回应讯号

当返回为正常时，格式如下：

ADDR 02 LEN Data

当设定不正确时，或无此功能，返回参数为：

ADDR 82H 01

例如：将 CD000 参数值改为 60.00Hz

发送： 3A 30 31 30 32 30 33 30 30 31 37 37 30 46 44 0D 0A

“30 30”即 00。代表 CD000。”31373730”即 1770 (Hex),转换为十进制数为 6000。

接收： 3A 30 31 30 32 30 33 30 30 31 37 37 30 46 44 0D 0A

## c.控制命令 (03)

发送格式：ADDR 03 01 CNTR

ADDR=0 为广播，无返回响应

ADDR≠0 时有响应，返回

CNTR

7	6	5	4	3	2	1	0
jogr	jogf	jog	r/f	stop	Rev	for	Run

当设定正确时，返回当前控制状态 格式：ADDR 03 01 CNST

CNST

7	6	5	4	3	2	1	0
跟踪启动	制动	r/f	jogin g	Runing	r/f	jog	Run

检测不正确时

ADDR 83H 01 CNST

例如：通讯控制变频器运行时即 CD033 = 2 时

发运行命令

发送：3A 30 31 30 33 30 31 30 31 LRC 0D 0A

“ 3031 ” 即 01 ， 01 为数据，代表运行命令。

接收：3A 30 31 30 33 30 31 30 39 LRC 0D 0A

“ 3039 ” 即 09 ， 09 为数据，代表变频器处于运行状态。

注意： 因通讯实时性，变频器返回状态滞后性，所以发命令之后返回的数据不能正确反应变频器当前状态。

#### d:读取状态值 (04)

格式: ADDR 04 01 CFG

ADDR=0 时无返回

ADDR≠0 时有返回

CFG=0—8 时返回单个状态

0: Set F 1: Out F 2: Out A 3: RoTT 4: DCV

5: ACV 6: Cont 7: Tmp 8: Error 和 CNST

例 1: 读取设定频率

发送: 3A 30 31 30 34 30 33 30 30 LRC 0D 0A

返回: 3A 30 31 30 34 30 33 31 33 38 38 LRC 0D 0A

其中 “31333838” 即 1388, 13 88 为数据, 13 为高位, 88 为低位。

例 2: 读取错误代码和当前变频器状态

发送: 3A 30 31 30 34 30 33 30 38 LRC 0D 0A

返回: 3A 30 31 30 34 30 33 30 38 30 30 30 39 LRC 0D 0A

其中 “3030” 即 00。00 为数据, 代表无故障。

“3039” 即 09。09 为数据, 代表当前变频器为运行状态。

#### e:变频器串口频率设定 (05)

格式: ADDR 05 02 Data

ADDR=0 时无返回

ADDR≠0 时有返回

例 变频器频率设定为 50.00Hz

发送: 3A 30 31 30 35 30 32 31 33 38 38 LRC 0D 0A

返回: 3A 30 31 30 35 30 32 31 33 38 38 LRC 0D 0A

其中 “31333838” 即 1388, 13 88 为数据, 13 为高位, 88 为低位。(通过串口设定频率时, 须先将 CD034 设为 2。)

4) LEN: 资料长度 指  $D_{(n-1)} \sim D_{(0)}$  的长度, 长度设定: 1 个 Word 时 LEN=3, 1 个 Byte 时或 <1 byte 时 LEN=2

5) DATA: (Data characters) 资料内容, 2n 个 ASCII 组合成 n 个 bytes, 最多有 50 个 ASCII。

6) LRC: 侦误值

ASCII 模式, 采用 LRC (Longitudinal Redundancy Check) 侦误值。LRC 侦误值乃是将 ADDR 至最后一个资料内容加总。得到结果以 256 为单位, 超出部分去除 (如结果为 128H 则取 28H) 然后计算二次反补后得到结果即为 LRC 侦误值。

7) 例如: 对 01 变频器写入 30.00Hz (写入 CD000)

STX	ADDR	FUNC	LEN	DATA	LRC	END
“.”	“0” “1”	“0” “2”	“0” “3”	“0” “0” “0” “B” “B” “8”	“3” “7”	“CR” “LF”
3AH	30H 31H	30H 32H	30H 33H	30H 30H 30H 42H 42H 38H	33H 37H	0DH 0AH

LRC 值计算:

$$01H + 02H + 03H + 00H + 0BH + B8H = C9H$$

C9H 二次反补为 37H

所以传送数据内容为:

3AH 30H 31H 30H 32H 30H 33H 30H 30H 30H 42H 42H 38H 33H 37H  
0DH 0AH

### 3.2 RTU 模式

静音	ADDR	FUNC	LEN	D <sub>(n-1)</sub> ~ D <sub>(0)</sub>	CRC	静音
>50ms						>50ms

- 1) 静音：表示 50ms 时间以上无串口中断
- 2) ADDR：通讯位置 8-bit 位置
- 3) FUNC：命令码 8-bit 命令，具体内容参见 3.1 章节中命令码中的详细说明
- 4) LEN：资料长度 指 D<sub>(n-1)</sub> ~ D<sub>(0)</sub> 的长度
- 5) DATA：资料内容 n×8-bit 资料
- 6) CRC：侦误值

RTU 模式采用 CRC (cyclical Redundancy Check) 侦误值。CRC 侦误值以下列步骤计算。

- 1：载入一个内容为 FFFFH 的 16-bit 暂存器（称 CRC 暂存器）。
- 2：将命令讯息第一个位元组与 CRC16-bitCRC 暂存器的低次位元组进行 Exclusive OR 运算，并将结果存回 CRC 暂存器。
- 3：将 CRC 暂存器内容右移 1bit，最左 bit 填入 0，检查 CRC 暂存器最低位元的值。
- 4：若 CRC 暂存器最低位元为 0，则重复步骤 3；否则将 CRC 暂存器与 A001H 进行 Exclusive OR 运算。
- 5：重复 3 及 4，直到 CRC 暂存器的内容右移 8bits，此时，该位元组已完全处理。
- 6：对命令讯息下一个位元组重复步骤 2 与 5 直到所有位元组皆完成处理，CRC 暂存器的最后内容即是 CRC 值。当在命令讯息中传送 CRC 值时，低位元组须与高位元组交换顺序，即低位元组将先被传送。

7：范例：

#### a.读取功能码数据 (01)

发送格式： ADDR 01 LEN Data

ADDR=0 时，无返回数据

ADDR≠0 且符合变频器地址时有响应

当返回为正常时，格式如下：

ADDR 01 LEN Data

如返回为一个字时 LEN=3 一个字节时 LEN=2

当无此功能或无效时返回为：

ADDR 81H 01

例如：读取 CD000 的参数

发送： 01 01 01 00 CRC

00 为数据。代表 CD000.

接收： 01 01 03 00 13 88 CRC

说明：CD000 = 01388H = 5000,即 50Hz.

#### b.功能码设定 (02)

发送格式： ADDR 02 LEN Data

ADDR=0 作广播用，可以设定，但无应答

ADDR≠0 时可以设定 同时有回应讯号

当返回为正常时，格式如下：

ADDR 02 LEN Data

当设定不正确时，或无此功能，返回参数为：

ADDR 82H 01

例如：将 CD000 参数值改为 60.00Hz

发送：01 02 03 00 17 70 CRC

00 为数据。代表 CD000. 1770（Hex），转换为十进制数为 6000。

接收：01 02 03 00 17 70 CRC

**c.控制命令（03）**

发送格式：ADDR 03 01 CNTR

ADDR=0 为广播，无返回响应

ADDR≠0 时有响应，返回

CNTR

7	6	5	4	3	2	1	0
jogr	jogf	jog	r/f	stop	Rev	for	Run

当设定正确时，返回当前控制状态 格式：ADDR 03 01 CNST

CNST

7	6	5	4	3	2	1	0
跟踪启动	制动	r/f	joging	Runing	r/f	jog	Run

检测不正确时 ADDR 83H 01 CNST

例如：通讯控制变频器运行时即 CD033 = 2 时

发运行命令

发送：01 03 01 01 CRC

01 为数据，代表运行命令。

接收：01 03 01 09 CRC

09 为数据，代表变频器处于运行状态。

注意：因通讯实时性，变频器返回状态滞后性，所以发命令之后返回的数据不能正确反应变频器当前状态。

**d:读取状态值（04）**

格式：ADDR 04 01 CFG

ADDR=0 时无返回

ADDR≠0 时有返回

CFG=0—8 时返回单个状态

0: Set F 1: Out F 2: Out A 3: RoTT 4: DCV  
5: ACV 6: Cont 7: Tmp 8: Error 和 CNST

例 1： 读取设定频率

发送：01 04 03 00 CRC

返回：01 04 03 13 38 CRC

其中 13 88 为数据，13 为高位，88 为低位。

例 2: 读取错误代码和当前变频器状态

发送: 01 04 03 08 CRC

返回: 01 04 03 08 00 09 CRC

其中 00 为数据, 代表无故障。

09 为数据, 代表当前变频器为运行状态。

#### e:变频器串口频率设定 (05)

格式: ADDR 05 02 Data

ADDR=0 时无返回

ADDR≠0 时有返回

例 变频器频率设定为 50.00Hz

发送: 01 05 02 13 88 CRC

返回: 01 05 02 13 88 CRC

其中 13 88 为数据, 13 为高位, 88 为低位。(通过串口设定频率时, 须先将 CD034 设为 2。)

#### 8: 范例 2:

下列以 C 语言产生 CRC 值。此函数需要两个参数

Unsigned char data←指讯息缓冲区的指标

Unsigned char length←讯息缓冲区中的位元组数目

此函数将传回 unsigned integer 型态之 CRC 值

```
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 rge_crc;
}
```