

10. 通讯功能（UART）

1.1 串口描述

串口配置：（9600, 8, N, 1）

波特率：9600

数据位：8 位

校验位：无

停止位：1

通讯功能：

串口通讯功能分为 APP 功能和 CFG 功能两部分。APP 功能为正常应用功能，包含芯片主动发送状态消息，和接收外部控制命令的功能。CFG 功能为高级配置功能，主要实现芯片的工作模式配置、参数校准等功能。APP 功能通常应用在逆变系统工作时，而 CFG 功能通常应用在逆变系统停机状态下。通过 CFG 功能配置的参数，会存储在芯片内部的 FLASH 空间中，在芯片上电时自动加载。

1.2 APP 功能

APP 功能为正常应用功能，包含芯片主动发送状态消息，和接收外部控制命令的功能。APP 功能通常应用在逆变系统工作时，持续向外发送状态消息，并实时接收外部命令，执行相应控制操作。

1.2.1 APP 消息发送

芯片上电后，会间隔 200ms 周期持续向外部发送状态消息，长度为 16 个字节。

状态消息：

状态消息（200ms 周期发送）		
BYTE0	报头	0x55
BYTE1	输出电压高字节	输出电压： 电压数据为 2 个字节表示，最小分辨率是 0.1V。 举例：[0x08, 0xCF] 用两个 16 进制表示电压，0x08 转换到 10 进制是 8，0xCF 转换到 10 进制是 207，数据的十进制值为 $8*256+207=2255$ ，得出的电压 $V=2255*0.1V=225.5V$ 。
BYTE2	输出电压低字节	
BYTE3	输出电流高字节	输出电流： 电流数据为 2 个字节表示，最小分辨率是 0.01A。 举例：[0x02, 0xCF] 用两个 16 进制表示电压，0x02 转换到 10 进制是 2，0xCF 转换到 10 进制是 207，数据的十进制值为 $2*256+207=719$ ，得出的电流 $I=719*0.01A=7.19A$ 。
BYTE4	输出电流低字节	
BYTE5	输入电压高字节	输入电压： 电压数据为 2 个字节表示，最小分辨率是 0.1V。 举例：[0x0E, 0x83] 用两个 16 进制表示电压，0x0E 转换到 10 进制是 14，0x83 转换到 10 进制是 131，数据的十进制值为 $14*256+131=3715$ ，得出的电压 $V=3715*0.1V=371.5V$ 。
BYTE6	输入电压低字节	

BYTE7	保留	保留
BYTE8	保留	保留
BYTE9	IGBT 温度	IGBT 温度： 温度数据为 1 个字节，有符号数，最小分辨率是 1℃。 举例：[0x16]转换为 10 进制是 20，温度=20℃ [0xF0]转换为 10 进制是-16，温度=-16℃
BYTE10	故障码	错误代码： 0x00：未定义 0x01：过载保护 0x02：短路保护 0x03：输出欠压保护 0x04：直流母线电压过压保护 0x05：直流母线电压过低保护 0x06：相位错误 0x07：温度保护 0x08：过流保护 0x09：频率不匹配 0x0A：发动机转速太低 0x0C：关闭输出
BYTE11	环境温度	逆变器温度： 温度数据为 1 个字节，有符号数，最小分辨率是 1℃。 举例：[0x16]转换为 10 进制是 20，温度=20℃ [0xF0]转换为 10 进制是-16，温度=-16℃
BYTE12	输出功率高字节	输出功率： 功率数据为 2 个字节表示，最小分辨率是 1W。 举例：0x06. 0x40 用两个 16 进制表示电压，0x06 转换到 10 进制是 6，0x40 转换到 10 进制是 64，数据的十进制值为 6*256+64=1600，得出的功率 P=1600*1W=1600W。
BYTE13	输出功率低字节	
BYTE14	CRC 校验高字节	循环冗余校验是 $CRC16=f(X^{16}+X^{15}+X^2+1)$ 对前 14 个字节 BYTE0-BYTE13 执行 CRC16 运算，BYTE14=校验结果高字节，BYTE15=校验结果低字节。
BYTE15	CRC 校验低字节	

1.2.2 APP 消息接收

芯片可接收的 APP 消息共有 2 条。

逆变关闭：收到逆变关闭消息后，关闭逆变输出。

逆变开启：收到逆变开启消息后，清除故障状态，启动逆变输出。

消息长度同样为 16 字节，超时 50ms 接收，即外部发送数据时，两个字节之间的时间间隔应小于 50ms，如超过 50ms，则判断为当前消息结束，为提高通讯效率，两个字节之间的时间间隔可以尽量小。两组消息之间时间间隔应大于 50ms，为避免接收乱帧，推荐两组消息之间时间间隔大于 100ms。

逆变关闭消息：

逆变关闭（超时 50ms 接收）		
BYTE0	命令字段 1	0x0F
BYTE1	命令字段 2	0xF0
BYTE2	命令字段 3	0x5A
BYTE3	命令字段 4	0x36
BYTE4	保留	0x00
BYTE5	保留	0x00
BYTE6	保留	0x00
BYTE7	保留	0x00
BYTE8	保留	0x00
BYTE9	保留	0x00
BYTE10	保留	0x00
BYTE11	保留	0x00
BYTE12	保留	0x00
BYTE13	保留	0x00
BYTE14	CRC 校验高字节	循环冗余校验是 $CRC16=f(X^{16}+X^{15}+X^2+1)$ 对前 14 个字节 BYTE0-BYTE13 执行 CRC16 运算，BYTE14=校验结果高字节， BYTE15=校验结果低字节。
BYTE15	CRC 校验低字节	

逆变开启消息：

逆变开启（超时 50ms 接收）		
BYTE0	命令字段 1	0x7D
BYTE1	命令字段 2	0xD7
BYTE2	命令字段 3	0xFE
BYTE3	命令字段 4	0xDA
BYTE4	保留	0x00
BYTE5	保留	0x00
BYTE6	保留	0x00
BYTE7	保留	0x00
BYTE8	保留	0x00
BYTE9	保留	0x00
BYTE10	保留	0x00
BYTE11	保留	0x00
BYTE12	保留	0x00
BYTE13	保留	0x00
BYTE14	CRC 校验高字节	循环冗余校验是 $CRC16=f(X^{16}+X^{15}+X^2+1)$ 对前 14 个字节 BYTE0-BYTE13 执行 CRC16 运算，BYTE14=校验结果高字节， BYTE15=校验结果低字节。
BYTE15	CRC 校验低字节	

1.3 CFG 功能

CFG 功能为高级配置功能，主要实现芯片的工作模式配置、参数校准等功能。CFG 功能通常应用在逆变系统停机状态下。通过 CFG 功能配置参数，会存储在芯片内部的 FLASH 空间中，在芯片上电时自动加载。

CFG 功能需要外部发送请求消息，芯片响应请求服务并回复应答消息。

发送和接收均采用 16 字节固定长度，消息以 ASCII 码 'E'、'G' 开头，CRC16 结尾。为区分 APP 消息和 CFG 消息，CRC 校验结果稍有不同，APP 消息的 CRC 校验结果 $=f(X^{16}+X^{15}+X^2+1)$ 。而 CFG 消息的 CRC 校验结果相当于在 APP 校验基础上加 1，即 CFG 消息的校验结果 $=f(X^{16}+X^{15}+X^2+1)+1$ 。

1.3.1 CFG 请求消息

CFG 请求消息格式：

CFG 请求消息（超时 50ms 接收）		
BYTE0	报头 1	0x45 - 'E'
BYTE1	报头 2	0x47 - 'G'
BYTE2	服务编码(SID)	主机请求的服务内容
BYTE3	子功能(sfund)/地址(addr)	当前服务下的子功能或地址
BYTE4	请求数据 1	
BYTE5	请求数据 2	
BYTE6	请求数据 3	
BYTE7	请求数据 4	
BYTE8	请求数据 5	
BYTE9	请求数据 6	
BYTE10	请求数据 7	
BYTE11	请求数据 8	
BYTE12	请求数据 9	
BYTE13	请求数据 10	
BYTE14	CRC 校验高字节	循环冗余校验是 $CRC16=f(X^{16}+X^{15}+X^2+1)+1$ 对前 14 个字节 BYTE0-BYTE13 执行 CRC16 运算，BYTE14=校验结果高字节， BYTE15=校验结果低字节。
BYTE15	CRC 校验低字节	

1.3.2 CFG 应答消息

CFG 应答消息格式：

CFG 请求消息（超时 50ms 接收）		
BYTE0	报头 1	0x45 - 'E'
BYTE1	报头 2	0x47 - 'G'
BYTE2	服务编码(SID)	主机请求的 CFG 服务编码
BYTE3	子功能(sfund)/地址(addr)	当前服务下的子功能或地址

BYTE4	应答数据 1	
BYTE5	应答数据 2	
BYTE6	应答数据 3	
BYTE7	应答数据 4	
BYTE8	应答数据 5	
BYTE9	应答数据 6	
BYTE10	应答数据 7	
BYTE11	应答数据 8	
BYTE12	应答数据 9	
BYTE13	应答数据 10	
BYTE14	CRC 校验高字节	循环冗余校验是 $CRC16=f(X16+X15+X2+1)+1$ 对前 14 个字节 BYTE0-BYTE13 执行 CRC16 运算, BYTE14=校验结果高字节, BYTE15=校验结果低字节。
BYTE15	CRC 校验低字节	

1.3.3 0x22 服务-读 DID

1.3.4 0x2E 服务-写 DID

0x2E 服务是写 DID 服务, 通过请求 0x2E 服务, 主机可以将配置参数和版本信息等内容写入芯片。

主机 0x2E 请求消息:

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12	Byte13	Byte14	Byte15
0x45	0x47	0x2E	addr	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	CRC16	

addr 是 DID 的地址, 不同的地址存储不同的 DID 信息。

从机回复:

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12	Byte13	Byte14	Byte15
0x45	0x47	0x2E	addr	resp	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16	

resp = 1 : 写入成功

resp = 0 : 写入失败

DID 信息表:

DID 信息表					
ADDR	DID	W/R	LEN	TYPE	描述
0x09	ProductDate	w/r	4	BCD	此 DID 可以写入整机生产日期, 如 2019 年 10 月 15 日: d1 = 0x20 d2 = 0x19 d3 = 0x10 d4 = 0x15
0x0A	SerialNo	w/r	10	ASCII	此 DID 可以写入整机序列号, 如 HS 厂生产的 2019 年第 21 周第 0001 号产品: "HS19210001" d1 = 'H'

					d2 = 'S' d3 = '1' d4 = '9' d5 = '2' d6 = '1' d7 = '0' d8 = '0' d9 = '0' d10 = '1'
0x0B	PartNo	r	10	ASCII	此 DID 表示芯片型号： "EG8025"
0x0C	ChipID	r	10	hex	此 DID 包含芯片 ID，每颗芯片唯一： 如：0x0123456789ABCDEF0123 d1 = 0x01 d2 = 0x23 d3 = 0x45 d4 = 0x67 d5 = 0x89 d6 = 0xAB d7 = 0xCD d8 = 0xEF d9 = 0x01 d10 = 0x23
0x0D	UsartVer	r	10	ASCII	串口通讯协议版本号，如 V1.0 版本 19 年 10 月 15 日发行： "1.0.191015"
0x0E	SoftwareVer	r	10	ASCII	芯片固件版本号，如 V2.1 版本 19 年 10 月 15 日发行： "2.1.191015"
0x0F	HardwareVer	r	10	ASCII	芯片硬件版本号，如 V3.2 版本 19 年 10 月 15 日发行： "3.2.191015"

1.3.1 0x21 服务-读 CFG

0x21 服务是读 CFG 服务，逆变运行、保护参数，通过请求 0x21 服务，主机可以读取芯片的逆变运行、保护参数等内容。

主机 0x21 请求消息：

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12	Byte13	Byte14	Byte15
0x45	0x47	0x21	addr	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00		CRC16

addr 是 CFG 的地址，不同的地址存储不同的 CFG 信息。

从机回复：

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12	Byte13	Byte14	Byte15
0x45	0x47	0x21	addr	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10		CRC16

若从机回复 d1~d10 全为 0xFF，那么表示读取 CFG 失败。

1.3.2 0x2D 服务-写 CFG

0x2D 服务是写 CFG 服务，通过请求 0x2D 服务，主机可以将配置参数和版本信息等内容写入芯片。

主机 0x2D 请求消息：

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12	Byte13	Byte14	Byte15
0x45	0x47	0x2D	addr	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	CRC16	

addr 是 CFG 的地址，不同的地址存储不同的 CFG 信息。

从机回复：

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12	Byte13	Byte14	Byte15
0x45	0x47	0x2D	addr	resp	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16	

resp = 1 : 写入成功

resp = 0 : 写入失败

CFG 信息表：

功能	addr	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	CRCH	CRCL
逆变参数保存	0x00	-	-	-	-	-	-	-	-	-	-		
输出电压	0x01	Vset*10		-	-	-	-	-	-	-	-		
输出频率	0x02	Fset*100		-	-	-	-	-	-	-	-		
恒功率&恒流	0x03	Iset*100		Pset		Iset_max*100		Pset_max		-	-		
母线电压高保护	0x10	Vdch2*10		Vdch1*10		Delay*0.02		-	-	-	-		
母线电压低保护	0x11	Vdcl2*10		Vdcl1*10		Delay*0.02		-	-	-	-		
输出电压高保护	0x12	Vouth*10		Delay*0.02		-	-	-	-	-	-		
输出电压低保护	0x13	Voutl*10		Delay*0.02		-	-	-	-	-	-		
一级过载保护	0x14	Iout1*100		Pout1		Delay*0.02		-	-	-	-		
二级过载保护	0x15	Iout2*100		Pout2		Step		-	-	-	-		
过载闪灯设定	0x16	Iled*100		Pled		-	-	-	-	-	-		
温度保护	0x17	Tboard		Tight		Delay*0.02		-	-	-	-		
短路保护	0x18	SCvol*10		Delay*0.02		Reset*0.02		RstDelay*0.02		-	-		
软起动	0x19	SST_Vin*10		SST_Iout*100		SST_Vout*10		Delay*0.02		Step			

1.3.3 0x2F 服务-IO 控制

0x2F 服务是 IO 控制服务，通过请求 0x2F 服务，主机可以控制芯片工作在特殊的状态，如测试模式等。

主机 0x2F 请求消息：

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12	Byte13	Byte14	Byte15
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------

0x45	0x47	0x2F	sfun	ctl	d2	d3	d4	d5	d6	d7	d8	d9	d10	CRC16
------	------	------	------	-----	----	----	----	----	----	----	----	----	-----	-------

sfun 是不同的 IO 服务子功能，ctl 是当前子功能下的控制字。

从机回复：

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12	Byte13	Byte14	Byte15
0x45	0x47	0x2F	sfun	resp	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CRC16	

resp = ctl : 请求成功

resp = 0xFF : 请求失败

IO 控制功能表：

IO 控制功能表		
0x02	0x00	逆变电路闭环输出
	0x03	逆变电路开环 SPWM 输出