

## DDSU666 型单相电表通信规程（总结篇）

电表 A（RS485 通信）序列号：23 12 05 02 99 63，  
BCD 码为：35 D4 E4 C8 4B，低位在前：63 99 02 05 12 23

电表 B（Modbus 协议）序列号：23 11 17 10 66 43，  
BCD 码为：35 C FA7 2D D3，低位在前：43 66 10 17 11 23

以下操作均针对电表 A，按照 DL/T645-2007《多功能电能表通信协议》规约进行的测试。

（已设置串口参数：波特率 9600bps，8 位数据位，偶校验，1 位停止位）

（1）读电能表通信地址

1.1 请求帧，读地址方式：

[发送]FE FE 68 AA AA AA AA AA 68 13 00 DF 16 // DF 为校验码，下同。

[接收]FE FE FE FE 68 63 99 02 05 12 23 68 93 06 96 CC 35 38 45 56 0B 16  
96 CC 35 38 45 56 还原 BCD 码（减去 33H）后地址为：63 99 02 05 12 23，倒序为 23 12 05 02 99 63

1.2 请求帧读数据方式：

[发送]FE FE 68 AA AA AA AA AA 68 11 04 34 37 33 37 B6 16

[接收]FE FE FE FE 68 63 99 02 05 12 23 68 91 0A 34 37 33 37 96 CC 35 38 45 56 E2 16  
96 CC 35 38 45 56 还原 BCD 码（减去 33H）后地址为：63 99 02 05 12 23，倒序为 23 12 05 02 99 63

（2）更改通信速率：

[发送]FE FE 68 63 99 02 05 12 23 68 17 01 20 40 16 将速率更改为 9600BPS，特征字 20 错误

[接收]FE FE FE FE 68 63 99 02 05 12 23 68 D7 01 3B 1B 16 从站异常应答 3B 对应的波特率为 2400

[发送]FE FE 68 63 99 02 05 12 23 68 17 01 43 63 16 特征字 43 将速率更改为 4800BPS

[接收]FE FE FE FE 68 63 99 02 05 12 23 68 97 01 43 E3 16 从站正常应答，波特率设置为 4800

[发送]FE FE 68 63 99 02 05 12 23 68 17 01 53 73 16 将速率更改为 9600BPS

[接收]FE FE FE FE 68 63 99 02 05 12 23 68 97 01 53 F3 16 从站正常应答，波特率设置为 9600

（3）读取电网频率：

[发送]FE FE 68 63 99 02 05 12 23 68 11 04 35 33 B3 35 6D 16

[接收]FE FE FE FE 68 63 99 02 05 12 23 68 91 06 35 33 B3 35 CC 7C 37 16

其中，CC 7C 还可能是：C8 7C，C9 7C，C6 7C，36 83，37 83 等

CC 7C 还原 BCD 码（减去 33H）后值为：99 49，倒序为 49 99，因分辨率为 0.01Hz，即为 49.99Hz

36 83 还原 BCD 码（减去 33H）后值为：03 50，倒序为 50 03，因分辨率为 0.01Hz，即为 50.03Hz

校验码计算方法：68+63+99+02+05+12+23+68+11+04+35+33+B3+35=36D，取最低 2 位，即 6D
--

（4）读取总电能：

[发送]FE FE 68 63 99 02 05 12 23 68 11 04 33 33 33 33 E9 16

[接收]FE FE FE FE 68 63 99 02 05 12 23 68 91 08 33 33 33 37 33 33 33 3D 16

其中，37 33 33 33 还原 BCD 码（减去 33H）后值为：04 00 00 00，

倒序为 00 00 00 04，值为 4，总电能分辨率为 0.01kWh，即表示 0.04kWh

（5）读取当前电压：

[发送]FE FE 68 63 99 02 05 12 23 68 11 04 33 34 34 35 ED 16



以下操作均针对电表 B，按照 MODBUS-RTU 通讯协议进行的测试。  
(已设置串口参数：波特率 9600bps，8 位数据位，无校验，1 位停止位)

开 始	地址码	功能码	数据区	CRC 校验码	结 束
大于 3.5 个字符 停顿时间	1 字节	1 字节	n 字节	2 字节	大于 3.5 个字符 停顿时间

先设置好从机地址码为 01H，后进行以下操作：

(1) 读寄存器操作-编程参数

1.1 主机要读取从机地址为 01H，起始寄存器地址为 0CH 的 1 个寄存器数据：

[发送]01 03 00 0C 00 01 44 09 //000CH 地址单元存放的是波特率，长度占 1 个字，44 09 为 CRC-16 校验码

[接收]01 01 82 80 A0 98 95 51 //串口助手设置偶校验，但电表中为无校验，故返回消息为非法的功能码 01

[接收]01 03 02 00 03 F8 45 //将串口助手的校验位设置与电表一致后，返回消息正常

各数据说明：

01：从机地址码； 03：从机功能码；

02：返回数据字节数；2 个数为 00 03

00 03：为从机的波特率代号，即 9600bps

F8 45：CRC-16 校验码，低位在前

1.2 读协议切换模式字及 MODBUS-RTU 的地址：

[发送]01 03 00 05 00 02 D4 0A //0005H 寄存器地址单元中为协议切换模式，后一个单元为 Addr

[接收]01 03 04 00 05 00 01 2B F2

各数据说明：

01：从机地址码； 03：从机功能码；

04：返回数据字节数；4 个数为 00 05 00 01

00 05：为从机的协议切换模式字为 5，表示 Modbus-RTU 协议，8n1，表示 8 个数据位无校验，1 个停止位；

00 01：为 0006H 寄存器地址单元中，从机的地址为 01。

1.3 读寄存器全部地址中的编程参数数据：

[发送]01 03 00 00 00 10 44 06 //读从 0000H 地址单元开始 16 个地址单元中的数据

[接收]01 03 20 02 BD 00 D0 00 00 00 05 00 00 00 05 00 01 00 0A 00 01 00 05 00 41  
00 24 00 03 D3 57 00 01 00 6C BF D7

各数据说明：

01：从机地址码； 03：从机功能码；

20：返回该字段之后的 32 个字节数据 (02 BD 00 D0 .....00 01 00 6C)

02 BD：编程密码 codE； 00 D0：软件版本；

00 00：电能清零； 00 05 00 00：保留参数；

00 05：协议切换模式； 00 01：Addr；

00 0A 00 01 00 05 00 41 00 24：地址单元 0007H-000BH，保留参数

00 03：为从机的波特率代号，即 9600bps

D3 57 00 01 00 6C：地址单元 000EH-0010H，保留参数

BF D7：CRC-16 校验码，低位在前

(2) 读寄存器操作-二次侧全部电量数据：

[发送]01 03 20 00 00 10 4F C6 //读取寄存器 2000H 单元中的 16 个字的电压值

[接收]01 03 20 43 5C 4C CD 3D 50 E5 60 3C 3A C7 11 00 00 00 00 3C 3A C7 11 3F  
80 00 00 00 00 00 00 42 47 EB 85 82 62

各数据说明：

01：从机地址码； 03：从机功能码；

20：返回该字段之后的 32 个字节数据（43 5C 4C CD ..... 42 47 EB 85）

43 5C 4C CD：A 相电压数据，待解析； 3D 50 E5 60：A 相电流数据，待解析；

3C 3A C7 11：瞬时总有功功率数据，待解析； 00 00 00 00：瞬时总无功功率数据；

3C 3A C7 11：瞬时总视在功率数据，待解析； 3F 80 00 00：总功率因数，待解析；

00 00 00 00：地址单元 200CH-000DH，保留参数； 42 47 EB 85：电网频率，待解析；

82 62：CRC-16 校验码，低位在前

### （3）读取当前电压：

[发送]01 03 20 00 00 02 CF CB //读取寄存器 2000H 单元中的 2 个字的电压值

[接收]01 03 04 43 5C 4C CD DA F0

接收数据各字段说明：

01 03 04：含义与前相同，其后 4 个字节为当前 A 相电流数据；

43 5C 4C CD：当前 A 相电压数据；倒序为 00 00 53，电流分辨率为 0.001A，故为 0.053A

DA F0：CRC-16 校验码，低位在前

倒序为 21 85，电压分辨率为 0.1V，故为 218.5V

### （4）读取当前电流：

[发送]01 03 20 02 00 02 6E 0B //读取寄存器 2002H 单元中的 2 个字的电流值

[接收]01 03 04 3D 54 FD F4 F7 58

接收数据各字段说明：

01 03 04：含义与前相同，其后 4 个字节为当前 A 相电流数据；

3D 54 FD F4：当前 A 相电流数据；倒序为 00 00 53，电流分辨率为 0.001A，故为 0.053A

F7 58：CRC-16 校验码，低位在前

### （5）读取功率

#### 5.1 读取瞬时总有功功率：

[发送]01 03 20 04 00 02 8E 0A //读取寄存器 2004H 单元中的 2 个字的瞬时总有功率值

[接收]01 03 04 3C 3A C7 11 44 52

01 03 04：含义与前相同，其后 4 个字节为总有功功率的数据；

3C 3A C7 11：总有功功率的数据；倒序为 00 01 20，功率分辨率为 0.0001kW，故为 0.0120kW，即 12.0W

44 52：CRC-16 校验码，低位在前

#### 5.2 读取瞬时总视在功率：

[发送]01 03 20 08 00 02 4E 09 //读取寄存器 2008H 单元中的 2 个字的瞬时总视在功率值

[接收]01 03 04 3C 3C 6A 7F 59 2F

01 03 04：含义与前相同，其后 4 个字节为总有功功率的数据；

3C 3A 6A 7F：瞬时总视在功率的数据；倒序为 00 01 20，功率分辨率为 0.0001kW，故为 0.0120kW，即 12.0W

59 2F：CRC-16 校验码，低位在前

### （6）读取功率因数：

[发送]01 03 20 0A 00 02 EF C9 //读取寄存器 200AH 单元中的 2 个字的功率因数值

[接收]01 03 04 3F 80 00 00 F7 CF

01 03 04：含义与前相同，其后 4 个字节为总有功功率的数据；

3F 80 00 00：功率因数的数据； F7 CF：CRC-16 校验码，低位在前

(7) 读取频率: (注: 从机地址更改为 **0x02**, 偶校验)

[发送]01 03 20 0E 00 02 AE 08 //读取寄存器 200EH 单元中的 2 个字的功率因数数值

[发送]02 03 20 0E 00 02 AE 3B //读取寄存器 200EH 单元中的 2 个字的功率因数数值

[接收]02 03 04 42 48 1E B8 54 8F

02 03 04: 含义与前相同, 其后 4 个字节为总有功功率的数据;

42 48 1E B8 : 从机信号频率的数据;

54 8F: CRC-16 校验码, 低位在前

(8) 读取电能相关参数:

8.1 读取有功总电能:

[发送]01 03 40 00 00 02 D1 CB 或 //从机地址为 0x01 时

[发送]02 03 40 00 00 02 D1 F8 //从机地址为 0x02 时

[接收]01 03 04 3D 23 D7 0A D9 A2 或 02 03 04 00 00 00 00 C9 33 电能被清零过

接收数据各字段说明:

01 03 04: 含义与前相同, 其后 4 个字节为当前 A 相电流数据;

3D 23 D7 0A: 当前有功总电能; 倒序为 00 00 00 04, 值为 4, 总电能分辨率为 0.01kWh, 即表示 0.04kWh

D9 A2: CRC-16 校验码, 低位在前

8.2 读取正向有功总电能:

[发送]01 03 40 0A 00 02 F1 C9

[接收]01 03 04 3D 23 D7 0A D9 A2

接收数据各字段说明:

01 03 04: 含义与前相同, 其后 4 个字节为当前 A 相电流数据;

3D 23 D7 0A: 当前正向有功总电能; 倒序为 00 00 00 04, 值为 4, 总电能分辨率为 0.01kWh, 即表示 0.04kWh

D9 A2: CRC-16 校验码, 低位在前

8.3 读取反向有功总电能:

[发送]01 03 40 14 00 02 91 CF

[接收]01 03 04 00 00 00 00 FA 33

接收数据各字段说明:

01 03 04: 含义与前相同, 其后 4 个字节为当前 A 相电流数据;

00 00 00 00: 当前反向有功总电能; 倒序为 00 00 00 04, 值为 4, 总电能分辨率为 0.01kWh, 即表示 0.04kWh

FA 33: CRC-16 校验码, 低位在前

(9) 写寄存器相关编程参数:

9.1 修改数据协议切换设置

9.1.1 将数据协议切换设置改为偶校验

[发送]01 10 00 05 00 01 02 00 06 26 07 或 //从机地址为 0x01 时

[发送]02 10 00 05 00 01 02 00 06 32 F7 //从机地址为 0x02 时

发送数据各字段说明:

01: 从机地址码; 10: 写寄存器操作;

00 05: 往此地址开始的寄存器中写入数据; 00 01: 写入寄存器数量为 1;

02: 写入数据 2 个字节; 00 06: 写入寄存器的数据值;

26 07: CRC-16 校验码, 低位在前

[接收]01 10 00 05 00 01 11 C8 或 02 10 00 05 00 01 11 C8

接收数据各字段说明:

01: 从机地址码; 10: 写寄存器操作;  
00 05: 往此地址开始的寄存器中写入数据; 00 01: 写入寄存器数量为 1;  
11 C8: CRC-16 校验码, 低位在前

9.1.2 将数据协议切换设置改回无校验

[发送]01 10 00 05 00 01 02 00 05 66 06 //00 05 即为写入的数据, 8n1

[接收]01 10 00 05 00 01 11 C8

发送及接收各字段含义参考前面说明。

备注: ChangeProtocol 协议切换模式字:  
数据为 1 时为 DL/T 645-2007 协议, 8E1, 表示 8 个数据位偶校验, 1 个停止位;  
数据为 2 时 Modbus-RTU 协议, 8n2, 表示 8 个数据位无校验, 2 个停止位;  
数据为 5 时 Modbus-RTU 协议, 8n1, 表示 8 个数据位无校验, 1 个停止位;  
数据为 6 时 Modbus-RTU 协议, 8E1, 表示 8 个数据位偶校验, 1 个停止位;  
数据为 7 时 Modbus-RTU 协议, 801, 表示 8 个数据位奇校验, 1 个停止位;

9.2 写寄存器修改从机地址

9.2.1 将从机地址 0x01 修改为 0x02

[发送]01 10 00 06 00 01 02 00 02 27 F7

发送数据各字段说明:

01: 从机地址码; 10: 写寄存器操作;  
00 06: 往此地址开始的寄存器中写入数据; 00 01: 写入寄存器数量为 1;  
02: 写入数据 2 个字节; 00 02: 写入寄存器的数据值, 即修改从机地址为 02;  
27 F7: CRC-16 校验码, 低位在前

[接收]01 10 00 06 00 01 E1 C8

接收数据各字段说明:

01: 从机地址码; 10: 写寄存器操作;  
00 06: 往此地址开始的寄存器中写入数据; 00 01: 写入寄存器数量为 1;  
E1 C8: CRC-16 校验码, 低位在前

9.2.2 将从机地址 02 改回为 0x01

[发送]02 10 00 06 00 01 02 00 01 73 06

发送数据各字段说明:

02: 从机当前地址码; 10: 写寄存器操作;  
00 06: 往此地址开始的寄存器中写入数据; 00 01: 写入寄存器数量为 1;  
02: 写入数据 2 个字节; 00 01: 写入寄存器的数据值, 即修改从机地址为 01;  
73 06: CRC-16 校验码, 低位在前

[接收]02 10 00 06 00 01 E1 FB

接收数据各字段说明:

01: 从机地址码; 10: 写寄存器操作;  
00 06: 往此地址开始的寄存器中写入数据; 00 01: 写入寄存器数量为 1;  
11 C8: CRC-16 校验码, 低位在前

9.3 修改从机的波特率

BAud 波特率: 0:1200bps; 1:2400bps; 2:4800bps; 3:9600bps;

9.3.1 修改从机的波特率为 4800bps

[发送]01 10 00 0C 00 01 02 00 02 27 5D

发送数据各字段说明：

02：从机当前地址码； 10：写寄存器操作；  
00 0C：往此地址开始的寄存器中写入数据； 00 01：写入寄存器数量为 1；  
02：写入数据 2 个字节； 00 02：写入寄存器的数据值，即修改从机的通信速率为 4800bps；  
27 5D：CRC-16 校验码，低位在前

[接收]01 10 00 0C 00 01 C1 CA

接收数据各字段说明：

01：从机地址码； 10：写寄存器操作；  
00 0C：往此地址开始的寄存器中写入数据； 00 01：写入寄存器数量为 1；  
C1 CA：CRC-16 校验码，低位在前

### 9.3.2 将从机的波特率改回 9600bps

[发送]01 10 00 0C 00 01 02 00 03 E6 9D

发送数据各字段说明：

02：从机当前地址码； 10：写寄存器操作；  
00 0C：往此地址开始的寄存器中写入数据； 00 01：写入寄存器数量为 1；  
02：写入数据 2 个字节； 00 03：写入寄存器的数据值，即修改从机的通信速率为 9600bps；  
E6 9D：CRC-16 校验码，低位在前

[接收]01 10 00 0C 00 01 C1 CA

接收数据各字段说明同上。

### 9.4 写寄存器将从机电量清零（慎用!!!!!!）

[发送]01 10 00 02 00 01 02 00 01 66 72

发送数据各字段说明：

01：从机地址码； 10：写寄存器操作；  
00 02：往此地址开始的寄存器中写入数据； 00 01：写入寄存器数量为 1；  
02：写入数据 2 个字节； 00 01：写入寄存器的数据值 1，即进行电量清零操作；  
66 72：CRC-16 校验码，低位在前

[接收]01 10 00 02 00 01 A0 09

接收数据各字段说明：

01：从机地址码； 10：写寄存器操作；  
00 02：从机被写入数据的寄存器地址； 00 01：写入寄存器数量为 1；  
A0 09：CRC-16 校验码，低位在前

查询当前电量情况：

[发送]01 03 40 00 00 02 D1 CB

[接收]01 03 04 00 00 00 00 FA 33 //4 个数据位全为 0

BA D2: CRC-16 校验码, 低位在前



附 1：寄存器各参数地址信息表

表 9 通讯参数信息

参数地址	参数代号	参数说明	数据类型	数据长度Word	读写属性
编程参数					
0000H	UcodE	编程密码 codE	int	1	R/W
0001H	REV.	软件版本	int	1	R
0002H	CLrE	电能清零 CLr.E	int	1	R/W
0003H	RESERVED	保留	int	1	
0004H	RESERVED	保留	int	1	
0005H	ChangeProtocol	协议切换设置	int	1	R/W
0006H	Addr	此地址只有 Modbus-RTU 时有效	int	1	R/W
0007H	RESERVED	保留	int	1	
0008H	RESERVED	保留	int	1	
0009H	RESERVED	保留	int	1	
000AH	RESERVED	保留	int	1	
000BH	RESERVED	保留	int	1	
000CH	BAud	波特率	int	1	R/W
000DH	Switch Out	开关量输出	int		R/W
000EH	RESERVED	保留	int	1	
000FH	RESERVED	保留	int	1	
0010H	RESERVED	保留	int	1	
二次侧电量数据					
2000H	U	A 相电压	float	2	R
2002H	I	A 相电流	float	2	R
2004H	P	瞬时总有功功率	float	2	R
2006H	Q	瞬时总无功功率	float	2	R
2008H	S	瞬时总视在功率	float	2	R
200AH	PF	总功率因数	float	2	R
200CH	RESERVED	保留	float	2	R
200EH	Freq	电网频率	float	2	R
2010H	RESERVED	保留	long	2	R
电 能 二 次 侧 数 据					
4000H	Ep	有功总电能	float	2	R
400AH	Imp	正向有功电能	float	2	R
4014H	Exp	反向有功电能	float	2	R

附 2：CRC-16（MODBUS）校验在线计算网址：

<http://www.ip33.com/crc.html>

操作界面截图如下：

CRC（循环冗余校验）在线计算

Hex

Ascii

校验文件

01 10 00 02 00 01 02 00 01

需要校验的数据:

输入的数据为16进制，例如： 31 32 33 34

参数模型 NAME: CRC-16/MODBUS x16+x15+x2+1

宽度 WIDTH: 16

多项式 POLY (Hex): 8005 例如: 3D65

初始值 INIT (Hex): FFFF 例如: FFFF

结果异或值 XOROUT (Hex): 0000 例如: 0000

☐输入数据反转 (REFIN)

☐输出数据反转 (REFOUT)

计算

清空

校验计算结果 (Hex): 7266

复制

校验计算结果 (Bin): 0111001001100110

复制

高位在左低位在右，使用时请注意高低位顺序!!!

附 3:16 进制转单精度浮点数在线计算网址：

<https://www.5axxw.com/tools/web/jinzhi.html>

操作界面截图如下：

