



OEL85 系列

Modbus 通讯协议手册



官方微信，一扫即得

如需资料下载，请登录：www.owon.com.cn/download

2025.06 版本 V1.0.0

©福建利利普光电科技有限公司版权所有，保留所有权利。

owon[®] 产品受专利权的保护，包括已取得的和正在申请的专利。本文中的信息将取代所有以前出版资料中的信息。

本手册信息在印刷时是正确的。然而，福建利利普光电科技有限公司将继续改进产品并且保留在任何时候不经通知的情况下变动规格的权利。

owon[®] 是福建利利普光电科技有限公司的注册商标。

福建利利普光电科技有限公司

福建漳州市蓝田工业开发区鹤鸣路（原横三路）19 号利利普光电科技楼

Tel: 4006-909-365

Fax: 0596-2109272

Web: www.owon.com.cn

E-mail: info@owon.com.cn

目 录

1. 概述	2
1.1 配置通讯波特率	2
1.2 选择通讯协议	2
1.3 设置通讯地址	3
2. Modbus 说明	4
2.1 帧格式	4
2.1.1 Modbus RTU	4
2.2 功能码	4
2.2.1 03 (0x03) 读寄存器	4
2.2.2 06 (0x06) 写单个寄存器	6
2.2.3 16 (0x10) 写寄存器	7
2.2.4 04 (0x04) 读输入寄存器	8
3. 地址列表	10
3.1 寄存器地址	10
4. 编程参考	14
4.1 通讯模型	14
4.2 数值转换	14
4.3 触摸屏通信配置	15

通告

本手册版权归本公司所有。手册中包含的信息，用于客户二次开发或搭建测试系统。

对本手册可能包含的错误或由提供、执行和使用本手册所造成的损害，本公司恕不负责。

版本修订记录

日期	版本	修订之章节
2025 年 6 月	1.0.0	完成本手册

1. 概述

为满足用户使用组态软件或其它软件包开发上位机的需求，本公司在原有设计基础上开发标准 Modbus 通讯协议。因此，OEL8500 具有 SCPI 与 Modbus 两种通讯协议。用户在使用前，应在显示面板上选择合适的协议。

本文档介绍 OEL8500 的 Modbus 通讯协议。

1.1 配置通讯波特率

OEL8500 系列负载支持 USB(VCP)、RS232 等通讯接口。

接口	说明
USB	USB 转虚拟串口（VCP）
RS232	标准串口

通讯参数的配置方法如下：

1. 在前面板上，按 **config** 键，通过旋钮选择“系统设置”，并按确认键进入系统参数设置界面。



2. 转动旋钮使光标移动至“波特率”的项目，按确认键进入编辑状态。
3. 转动旋钮选择相关的参数。
4. 按确认键使通讯参数生效。

1.2 选择通讯协议

OEL8500 系列负载支持 SCPI 与 Modbus 两种通讯协议，系统默认为 SCPI 协

议，若需要切换至 Modbus 协议，请按下面的步骤操作：

1. 在前面板上，按 **config** 键，通过旋钮选择“系统设置”，并按确认键进入系统参数设置界面。



2. 转动旋钮使光标移动至“通讯协议”项目，按确认键进入编辑状态。
3. 转动旋钮或按方向键选择“Modbus”。
4. 按确认键使通讯协议生效。

1.3 设置通讯地址

当用户选择通讯协议为 Modbus 时，还需进一步设置 Modbus 的通讯地址，请按下面的步骤操作：

1. 在前面板上，按 **config** 键，通过旋钮选择“系统设置”，并按确认键进入系统参数设置界面。



2. 转动旋钮使光标移动至“设备地址”项，按确定键进入编辑状态。
3. 转动旋钮或按数字键输入新的地址（地址设置范围：1~9）。
4. 按确认键使设备地址生效。

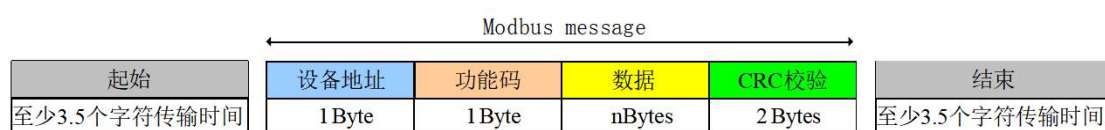
2. Modbus 说明

OEL8500 系列负载支持标准 Modbus RTU 协议。

2.1 帧格式

2.1.1 Modbus RTU

对于串口通讯（例如 RS232），采用 RTU 模式传输 Modbus 消息，其格式如下图所示。



- **设备地址：**Modbus 通讯地址，范围 1～9。
- **功能码：**请求执行某种操作，例如读寄存器、写寄存器、读块数据等。
- **数据：**描述操作具体信息，例如读寄存器的起始地址，读寄存器个数。
- **CRC 校验：**计算“设备地址”、“功能码”、“数据”的 CRC 校验结果。

2.2 功能码

OEL8500 系列的 Modbus 协议仅用到《标准 MODBUS 协议》的部分功能码，具体如下表。

功能码	含义	备注
0x03	读保持寄存器	按 16 位字读取
0x04	读输入寄存器	按 16 位字读取
0x06	写一个保持寄存器	按 16 位字写
0x10	写多个保持寄存器	按 16 位字写

2.2.1 03（0x03）读寄存器

该功能码用于读取连续多个寄存器的内容。

请求

功能码	1 Byte	0x03
起始地址	2 Bytes	0x0000~0xFFFF
寄存器数目	2 Bytes	

应答

功能码	1 Byte	0x03
返回字节数目	1 Byte	2xN
寄存器数值	N x 2 Bytes	value

错误

错误码	1 Byte	0x83
异常码	1 Byte	01/02/03/04

异常码说明

错误码	0x01	0x02	0x03	0x04
说明	不支持的功能码	寄存器地址 超出范围	寄存器个数 超出范围	读失败

举例

Request: 01 03 03 E8 00 02 44 7B

Response: 01 03 04 00 01 00 00 AB F3

Request		Response	
Field	Hex	Field	Hex
设备地址	01	设备地址	01
功能码	03	功能码	03
寄存器地址（高字节）	03	字节个数	04
寄存器地址（低字节）	E8	寄存器数值（高字节）	00
寄存器个数（高字节）	00	寄存器数值	01
寄存器个数（低字节）	02	寄存器数值	00
CRC 校验（低字节）	44	寄存器数值（低字节）	00
CRC 校验（高字节）	7B	CRC 校验（低字节）	AB
		CRC 校验（高字节）	F3

2.2.2 06（0x06）写单个寄存器

该功能码用于写单个寄存器。

请求

功能码	1 Byte	0x06
起始地址	2 Bytes	0x0000~0xFFFF
寄存器数值	2 Bytes	value

应答

功能码	1 Byte	0x10
寄存器地址	2 Bytes	0x0000~0xFFFF
寄存器数值	2 Bytes	value

错误

错误码	1 Byte	0x86
异常码	1 Byte	01/02/03/04

异常码说明

错误码	0x01	0x02	0x03	0x04
说明	不支持的功能码	寄存器地址超出范围	寄存器个数超出范围或字节个数不对或寄存器数值不对	写失败

举例

Request: 01 06 03 E8 00 01 C8 7A

Response: 01 06 03 E8 00 01 C8 7A

Request		Response	
Filed	Hex	Field	Hex
设备地址	01	设备地址	01
功能码	06	功能码	06
寄存器地址（高字节）	03	寄存器地址（高字节）	03
寄存器地址（低字节）	E8	寄存器地址（低字节）	E8
寄存器个数（高字节）	00	寄存器个数（高字节）	00
寄存器个数（低字节）	01	寄存器个数（低字节）	01

CRC 校验（低字节）	C8	CRC 校验（低字节）	CB
CRC 校验（高字节）	7A	CRC 校验（高字节）	7A

2.2.3 16（0x10）写寄存器

该功能码用于写连续多个寄存器。

请求

功能码	1 Byte	0x10
起始地址	2 Bytes	0x0000~0xFFFF
寄存器数目	2 Bytes	2
字节数目	1 Byte	4
寄存器数值	4 Bytes	value

应答

功能码	1 Byte	0x10
起始地址	2 Bytes	0x0000~0xFFFF
寄存器数值	2 Bytes	2

错误

错误码	1 Byte	0x90
异常码	1 Byte	01/02/03/04

异常码说明

错误码	0x01	0x02	0x03	0x04
说明	不支持的功能码	寄存器地址超出范围	寄存器个数超出范围或字节个数不对或寄存器数值不对	写失败

举例

Request: 01 10 03 F4 00 02 04 00 00 44 7A 5B 0B

Response: 01 10 03 F4 00 02 00 7E

Request		Response	
Filed	Hex	Field	Hex
设备地址	01	设备地址	01

功能码	10	功能码	10
寄存器地址（高字节）	03	寄存器地址（高字节）	03
寄存器地址（低字节）	F4	寄存器地址（低字节）	F4
寄存器个数（高字节）	00	寄存器个数（高字节）	00
寄存器个数（低字节）	02	寄存器个数（低字节）	02
字节数目	04	CRC 校验（低字节）	00
寄存器数值（高字节）	00	CRC 校验（高字节）	7E
寄存器数值	00		
寄存器数值	44		
寄存器数值（低字节）	7A		
CRC 校验（低字节）	5B		
CRC 校验（高字节）	0B		

2.2.4 04（0x04）读输入寄存器

该功能码用于读取连续多个保持寄存器的内容。

请求

功能码	1 Byte	0x04
起始地址	2 Bytes	0x0000~0xFFFF
寄存器数目	2 Bytes	2

应答

功能码	1 Byte	0x04
起始地址	1 Byte	2xN
寄存器数值	N x 2 Bytes	value

错误

错误码	1 Byte	0x83
异常码	1 Byte	01/02/03/04

异常码说明

错误码	0x01	0x02	0x03	0x04
-----	------	------	------	------

说明	不支持的功能码	寄存器地址 超出范围	寄存器个数 超出范围	读失败
----	---------	---------------	---------------	-----

举例

Request: 01 04 00 96 00 02 91 E7

Response: 01 04 04 AA C6 3D 1B 6A FA

Request		Response	
Filed	Hex	Field	Hex
设备地址	01	设备地址	01
功能码	04	功能码	04
寄存器地址（高字节）	00	字节个数	04
寄存器地址（低字节）	96	寄存器数值（高字节）	AA
寄存器个数（高字节）	00	寄存器数值	C6
寄存器个数（低字节）	02	寄存器数值	3D
CRC 校验（低字节）	91	寄存器数值（低字节）	1B
CRC 校验（高字节）	E7	CRC 校验（低字节）	6A
		CRC 校验（高字节）	FA

3. 地址列表

3.1 寄存器地址

负载寄存器如表 3-1 保持寄存器、表 3-2 输入寄存器所示。

表 3-1 保持寄存器

地址	功能	数据类型	地址增量
1000	输出控制(ON/OFF)	uint16_t	1
1001	reserve	uint16_t	1
1002	输出电压设置	float	2
1004	输出电流设置	float	2
1006	电流上升斜率	float	2
1008	电流下降斜率	float	2
1010	输出功率设置	float	2
1012	输出电阻设置	float	2
1014	电压量程选择	uint16_t	1
1015	电流量程选择	uint16_t	1
1016	电压上升斜率	float	2
1018	电压下降斜率	float	2
1020	输出模式	int16_t	1
1021	短路模式	uint16_t	1
1022	带载电压	float	2
1024	电流限制	float	2
1026	动态的方式	uint16_t	1
1027	cc 动态低电流值	float	2
1029	cc 动态低电流时间值	float	2
1031	cc 动态高电流值	float	2
1033	cc 动态高电流时间值	float	2
1035	cc 动态电流上升斜率	float	2

1037	cc 动态电流下降斜率	float	2
1039	远端补偿设置	uint16_t	1
1040	RS485 通信地址设置	uint16_t	1
1041	触发信号	uint16_t	1
1042	电池测试——放电截止电压	float	2
1044	电池测试——放电容量	float	2
1046	电池测试——放电时间	float	2
1048	电池测试——测试条件掩码	uint16_t	1
1049	电池测试——测试启停控制	uint16_t	1
1050	串/并模式选择	uint16_t	1
1051	串/并联合数	uint16_t	1
1052	主从机设置	uint16_t	1
1053	本机地址	uint16_t	1
1054	并机控制	uint16_t	1
1055	保留	uint16_t	1
1100	OCP 启动电压	float	2
1102	OCP 初始电流	float	2
1104	OCP 步进电流	float	2
1106	OCP 步进延时	float	2
1108	OCP 保护电压	float	2
1110	OCP 测试最大电流	float	2
1112	预留	float	2
1114	预留	float	2
1116	OPP 启动电压	float	2
1118	OPP 初始功率	float	2
1120	OPP 步进功率	float	2
1122	OPP 步进延时	float	2
1124	OPP 保护电压	float	2

1126	OPP 最大功率	float	2
1128	OPP 最小功率	float	2
1130	预留	float	2
1200	电压限制	float	2
1202	电流限制	float	2
1204	OVP	float	2
1206	OCP	float	2
1208	OPP	float	2
1210	触发方式选择	uint16_t	1

表 3-2 输入寄存器

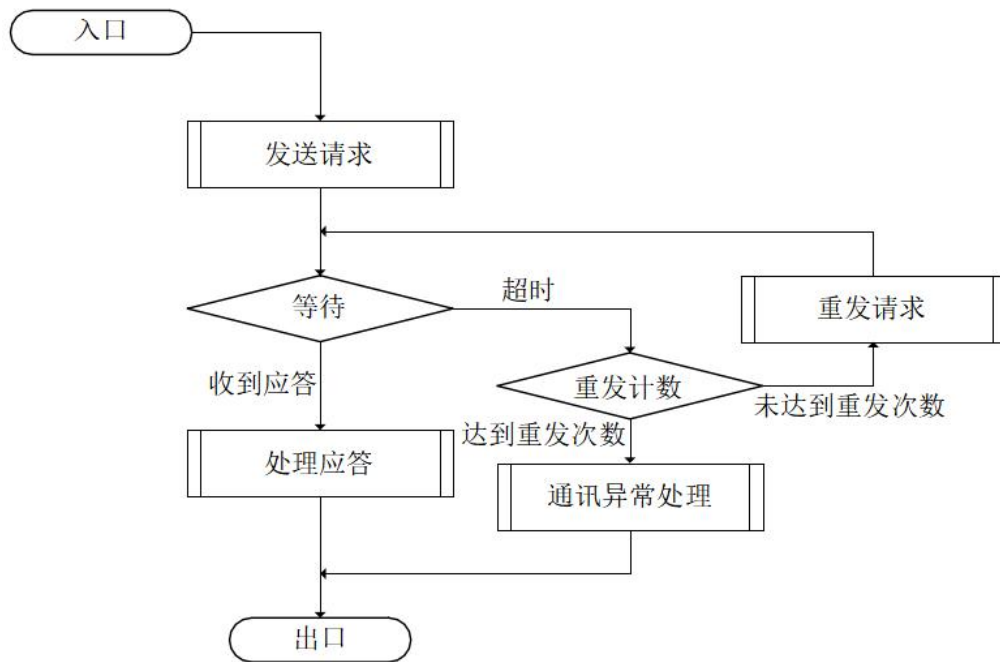
地址	功能	数据类型	地址增量
150	输出电压（本机）	float	2
152	输出电流（本机）	float	2
154	输出功率（本机）	float	2
156	放电容量 Ah	float	2
158	放电电量 Wh	float	2
160	放电时间	float	2
162	测试状态	uint16_t	1
163	预留	uint16_t	1
164	OCP 测试数值	float	2
166	OCP 测试状态	uint16_t	1
167	预留	uint16_t	1
168	OPP 测试数值	float	2
170	OPP 测试状态	uint16_t	1
171	预留	uint16_t	1
250	运行状态	uint16_t	1
251	系统运行标志	uint16_t	1
252	系统故障状态字	uint16_t	1
253	市电输入故障字	uint16_t	1

254	内部故障字 1	uint16_t	1
255	内部故障字 2	uint16_t	1
256	输出故障字	uint16_t	1
257	并机故障字	uint16_t	1
258	告警字	uint16_t	1
259	预留	uint16_t	1
260	预留	uint16_t	1
261	系统运行标志位（锁存）	uint16_t	1
262	预留	uint16_t	1
263	市电故障字（锁存）	uint16_t	1
264	内部故障字 1（锁存）	uint16_t	1
265	内部故障字 2（锁存）	uint16_t	1
266	输出故障字（锁存）	uint16_t	1
267	并机故障字（锁存）	uint16_t	1
268	告警字	uint16_t	1

4. 编程参考

4.1 通讯模型

超时时间建议设为 50~500ms，视应答帧长度而定。应答帧越长，超时时间也应该越长。重发次数建议设为 3~5 次。



4.2 数值转换

Modbus 数据传输采用大端格式。如果控制设备（PC、PLC、组态软件、用户 APP 等）为小端格式，那么在使用数据前，需要对数据进行大小端格式转换处理。

```
typedef union
```

```
{
```

```
    int reg_int;           //32 位带符号整型
```

```
    float reg_float;       //单精度浮点型
```

```
    UCHAR reg_bytes[4];    //字节序列
```

```
}unReg32Type;
```

```
UCHAR buff_modbus[256];    //Modbus 数据缓存
```

unReg32Type data_reg; //复合寄存器数据

在小端格式设备中

data_reg.reg_bytes[3] = buff_modbus[0];

data_reg.reg_bytes[2] = buff_modbus[1];

data_reg.reg_bytes[1] = buff_modbus[2];

data_reg.reg_bytes[0] = buff_modbus[3];

在大端格式设备中

data_reg.reg_bytes[0] = buff_modbus[0];

data_reg.reg_bytes[1] = buff_modbus[1];

data_reg.reg_bytes[2] = buff_modbus[2];

data_reg.reg_bytes[3] = buff_modbus[3];

以上代码仅供参考。用户在通讯应用中，应根据控制设备的实际情况正确处理数据。

4.3 触摸屏通信配置

设备属性名	设备属性值
采集优化	1-优化
设备名称	设备0
设备注释	ModbusRTU_串口
初始工作状态	1 - 启动
最小采集周期(ms)	100
设备地址	1
通讯等待时间	200
16位整数字节序	0 - 12
32位整数字节序	2 - 3412
32位浮点字节序	2 - 3412
字符串字节序	0 - 21
字符串编码格式	0 - ASCII
校验数据字节序	0 - LH[低字节,高字节]
功能码校验	0 - 校验功能码
分块采集方式	0 - 按最大长度分块
0区写功能码	2 - 默认值
4区写功能码	2 - 默认值
1区最大块长	120
0区最大块长	120