

串口修改参数及硬件 TCP/IP 协议栈

嵌入式设备联网解决方案

版权©2008 上海卓岚信息科技有限公司保留所有权力

ZL DUI 20090825.1.0



版权©2008 上海卓岚信息科技有限公司保留所有权力

版本信息

对该文档有如下的修改：

修改记录			
日期	版本	文档编号	修改内容
2009-8-25	Rev.1	ZL DUI 20090825.1.0	发布版本
2010-4-26	Rev.2	ZL DUI 20090825.2.0	增加命令方式的说明
2013-2-25	Rev.3	ZL DUI 20090825.3.0	去掉已经不再使用的“硬件模式”和不常用的“硬件协议栈库函数
2018-3-13	Rev.4	ZL DUI 20090825.3.0	增加二代串口指令

所有权信息

未经版权所有者同意，不得将本文档的全部或者部分以纸面或者电子文档的形式重新发布。

本文档只用于辅助读者使用产品，上海卓岚公司不对使用该文档中的信息而引起的损失或者错误负责。本文档描述的产品和文本正在不断地开发和完善中。上海卓岚信息科技有限公司有权利在未通知用户的情况下修改本文档。

目 录

1. 概述	5
1.1. 硬件协议栈与软件协议栈.....	5
1.2. 硬件协议栈原理	5
2. 命令模式	6
2.1. 命令格式	6
2.2. 二代命令格式	7
2.3. 参数格式	10
2.4. 读参数步骤	14
2.5. 写参数步骤	16
2.6. 注意事项	16
3. 典型应用	17
3.1. 读取连接状态	17
3.2. 控制联网产品连接.....	17
3.3. 读取本地 IP 地址.....	18
3.4. 修改 dns 服务器.....	18
3.5. 查看系统是否初始化完毕.....	18
3.6. 一次设置方法	18
3.7. 串口重启设备	19
3.8. 指定 DNS 服务器	20
3.9. 读取设备名称	20
3.10. 写入设备名称	20
3.11. 读取设备 ID	20
3.12. 修改目的 IP.....	20
3.13. 修改 UDP 模式的目的端口.....	21
3.14. 读取设备版本号	21
3.15. MDIP 器件的多目标设置.....	21
3.16. 用户参数空间使用.....	22
3.17. 带界面的自定义参数.....	23

3.18.	注册包和心跳包的串口写入.....	23
4.	售后服务和技术支持	24

1. 概述

1.1. 硬件协议栈与软件协议栈

硬件 TCP/IP 协议栈是相对于软件 TCP/IP 协议栈而言的。软件 TCP/IP 一般提供 socket 接口，通过调用库函数实现连接、监听、发送、接收等操作，例如 Windows 的 socket API 函数有 connect、listen、send、recv 等。硬件协议栈是一种较新的概念，硬件协议栈存在于 ZLSN2000 联网模块内部，用户 MCU 通过串口给 ZLSN2000 发送命令，控制 ZLSN2000 运行 TCP/IP 协议栈，达到连接、监听、发送、接收等网络功能，其功能和直接调用软件 TCP/IP 协议栈类似。

嵌入式系统使用硬件 TCP/IP 协议栈的优点：

1. 与使用软件 TCP/IP 协议栈相比，硬件协议栈不占用用户 CPU、无需 RAM，减轻了用户 MCU 的负担，而且硬件协议栈是成熟产品，具有较强的稳定性。
2. 与不具有硬件 TCP/IP 协议栈的联网模块相比，硬件 TCP/IP 使得用户 MCU 具有更强的灵活性，联网模块可以进行二次开发，基本上可以实现软件协议栈的所有控制功能。

1.2. 硬件协议栈原理

ZLSN2000 的硬件协议栈的实现实际是通过修改 ZLSN2000 内部参数实现的，例如用户修改了目的域名或 IP 则 ZLSN2000 自动向新的目的发起连接，从而实现 connect()函数。

于是硬件协议栈的功能其实也提供用户设备端修改模块参数的方法。这为用户在设备端通过键盘、触摸屏等方法修改设备 IP、模式等参数提供了方法。

ZLSN2000 提供两种模式修改设备参数：命令模式、硬件模式。

命令模式：用户通过串口向 ZLSN2000 发送一个固定的命令识别流，让 ZLSN2000 参数修改。

硬件模式：用户的 MCU 的 IO 控制引脚和 ZLSN2000 的 SPR、SPA 连接，通过硬件引脚之间的时序控制达到参数修改的目的。

两种模式的比较：

1. 命令模式无需增加硬件连线，对硬件影响小，特别是对于通过 DB9 串口线和设备连接的情况，增加硬件连线比较困难。

2. 硬件模式的控制时序较为复杂，用户开发需要的时间较长。而命令模式操作相对简单，且可以通过计算机串口调试工具模拟发送命令字。
3. 硬件模式的唯一优点是不存在命令识别流和数据流干扰的顾虑，即用户发送的数据流中出现命令识别流。但是 ZLSN2000 的命令识别流有 10 个字节，共有 1.2×10^{24} 种可能，即 1.2 亿亿亿种可能，如果以最高的 115200bps 的波特率不断发送数据，那么平均来说，需要 366040 亿年才出现一次。那么这种可能性已经可以到了忽略的程度。
- 建议：建议用户采用命令模式。

2. 命令模式

2.1. 命令格式

命令模式读取、写入参数的步骤如下：在任何时刻，向 ZLSN2000 串口写入如图 1 所示命令，即可完成参数的读写。

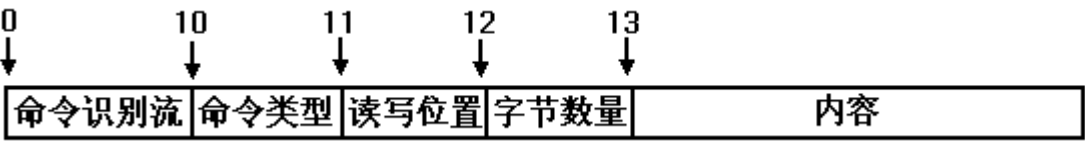


图 1 命令格式

其中命令识别流为 10 字节，表示命令的开始，必须为如下数据：ed f2 a3 56 ca db 91 84 b0 d7

命令类型为 1 字节，表示命令的类型。第 0 位为 1 表示写参数，为 0 表示读参数。

表 1. 常用命令类型

命令字	经常使用	功能
0x00	是	读参数。
0x01	是	写参数。断电不保存。但是本地 IP、子网掩码、网关、DHCP 模式、DNS 服务器 IP 修改后会重启，并且保存参数。

表 2. 专用命令类型

命令字	经常使用	功能
-----	------	----

0x02	否	用于 MDIP 型号产品：不关闭之前的 TCP 连接，再发起第二个 TCP 连接。
0x03	否	写参数。并保存参数，写入 ZLSN2000 的 Flash 存储器进行保存。
0x04	否	加密狗命令。
0x05	否	写参数。一定保存参数，且一定重启模块，且不关闭之前的 TCP 连接。
0x06	否	读网页（ZLSN2030EX 专用）。
0x07	否	和 0x03 命令一样，但是一定重启模块。
0x08	否	写网页（ZLSN2030EX 专用）。
0x09	否	基于参数的通信，参考《向中心服务器发送模块参数功能.pdf》
0x0a	否	Uniform control 控件触发（ZLSN2030EX 专用）。
0x0b	否	保存参数的基于参数的通信，参考《向中心服务器发送模块参数功能.pdf》
0x0c	否	TXET control 控件提交数据（ZLSN2030EX 专用）。
0x0d	否	ZLAN7142 专用的配置指令。

读写位置为 1 字节，表示从参数的第几个字节开始读取。参考图 4 参数格式。

字节数量为 1 字节，表示此次需要读取或写入的数据量。

内容，在写入操作时，内部部分为需要写入的内容，其长度应该和字节数量字段定义一致。

2.2. 二代命令格式

为了保证发送的串口指令的正确性，现在可以支持二代命令格式如下：

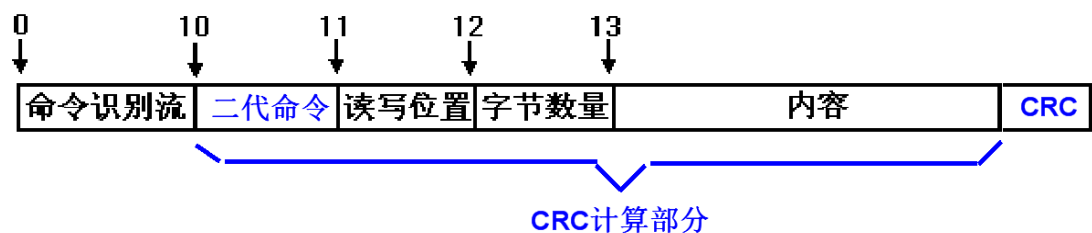


图 2 二代命令格式

如图所示：二代命令和一代命令的差别在于

1. 将命令类型改为二代命令。二代命令是一代命令加上 0x80 得到，比如原来的写命令为 0x01，现在变为 0x81。
2. 如果是二代命令，则必须在命令的结尾增加 CRC16 的 2 个字节的校验。校

验的内容是不含命令识别流（不含 CRC 本身），如图所示。如果 CRC 错误则命令不会被执行。这个保证了意外写入数据不会被执行。

3. 二代命令都会有反馈，如果执行成功则返回正确指令，如果错误则返回错误指令。

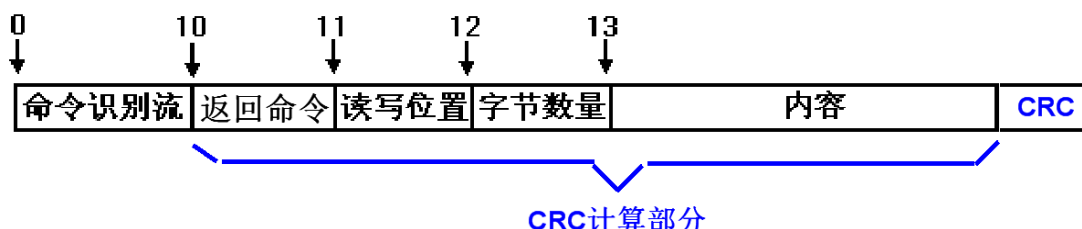


图 3 返回指令格式

可以看到反馈的命令的格式和写入的命令格式是一样的，这样可以方便用户计算 CRC 和命令识别。“返回命令”有 4 种：（1）0x0e 表示指令格式错误，包括 CRC 错误，此时读写位置为 0，字节数量为 0，内容为空，CRC 为 10 03。（2）0x10 表示指令格式和 CRC 正确，但是数值范围可能越界或者写入的 MAC 地址和原 MAC 地址不相符（MAC 无法修改）。注意该返回错误只存在于需要写 flash 的 0x03 等指令，0x01 指令是不会返回这个错误的。（3）0x0f 表示指令正确执行，此时读写位置为 0，字节数量为 0，内容为空，CRC 为 41 c3。（4）0x80 表示二代读指令的返回值数据，此时读写位置、字节数量为发送的指令一样的读写位置、字节数量，内容为读取的数据的内容，CRC 为图示指令部分的校验和。

任何模块支持二代命令的都同时支持一代命令。建议支持二代的采用二代命令。

以 1.558(2003)版本的固件测试如下：

比如原来读取连接状态的命令为：ed f2 a3 56 ca db 91 84 b0 d7 00 3d 01，其返回值只是 00 或者 01，容易和通讯数据搞混。改为二代命令时，首先将 00 改为 80，然后增加“80 3d 01”的 CRC 校验为 a1 78。所以整个命令为 ed f2 a3 56 ca db 91 84 b0 d7 80 3d 01 a1 78。发送后收到的返回指令为：ed f2 a3 56 ca db 91 84 b0 d7 80 3d 01 00 b9 b8。这里 ed f2 a3 56 ca db 91 84 b0 d7 是命令识别流，80 是读指令返回指令，3d 是发送的位置，01 表示后面数据内容为 1 个字节，接着是一个字节的内容 00，然后是内容 80 3d 01 00 的 CRC 校验 b9 b8。如果发送的指令 CRC 错误将会收到 ed f2 a3 56 ca db 91 84 b0 d7 0e 00 00 10 03。

再比如原来修改模式为客户端的命令为 ed f2 a3 56 ca db 91 84 b0 d7 01 14 01

01。改为二代命令为 ed f2 a3 56 ca db 91 84 b0 d7 **81 14 01 01 a8 4c**。如果没有返回说明波特率错误、模块没有连接；如果返回 ed f2 a3 56 ca db 91 84 b0 d7 **0f 00 00 41 c3** 则表明命令执行正确；如果返回 ed f2 a3 56 ca db 91 84 b0 d7 **0e 00 00 10 03** 则表明命令解析错误（包括 CRC 错误）。

还有一种可能是写入的数值范围越界或者试图修改 MAC 地址，比如发送 ed f2 a3 56 ca db 91 84 b0 d7 **83 14 01 04 69 f7**。这里注意是 0x83 也就是写 flash 指令而不是 0x81 写内存指令，另外将工作模式设置为 4，实际上模式的范围是 0~3，4 是不合法的。此时返回指令为 ed f2 a3 56 ca db 91 84 b0 d7 **10 00 00 70 05**。这里的 0x10 表示数据内容越界，导致 flash 无法写入。另外注意如果之前使用 0x81 指令将数据写入到内存，最后使用 0x83 写入 flash 则只要之前任何一条指令内容越界都会导致 0x83 指令返回 10 错误代码。

另外比如指令 ed f2 a3 56 ca db 91 84 b0 d7 **83 1f 01 ff 59 b6**，这里的 1f 开始的位置是 mac 地址，是无法修改的，这里用户试图写 flash 修改。则也会返回 10 指令内容无法写入的错误。

10 指令的返回值有两个条件：（1）必须有写 flash 的动作。如果是没有使用 0x83 指令而是 0x81 指令，实际越界的数据会写到内存中，但是由于未写入 flash 也就没有被系统保存而使用，所以也是无效的。（2）写入的内容无法被正确写入 flash。

注意无论是写还是读指令，相邻指令之间建议有 100ms 的间隔。

2.3. 参数格式

	↓ 0 Byte	↓ 2 byte	↓ 4 byte	
0 (00H)	Local IP Addr			
4 (04H)	NetMask			
8 (08H)	GateWay			
12 (0CH)	Destination IP Addr			
16 (10H)	Local IP Port		Destination IP Port	
20 (14H)	Work mode	Pad		
24 (18H)	Pad			
28 (1CH)	Pad		Pad2	
32 (20H)	Pad2			
36 (24H)	Pad2	Baundrate	Device Name	
40 (28H)	Device Name			
44 (2CH)	Device Name			
48 (30H)	Parity	Gap Time	Packing length	
52 (34H)	F_end en	F_end byte	F_start en	F_start byte
56 (38H)	DHCP en	Flow_Ctrl	Dest_Mode	DataSize
60 (3CH)	AppPro	status	DnsServerIP	
64 (40H)	DnsServerIP		Dest string	
68 (44H)	Dest string			
72 (48H)	Dest string			
76 (4CH)	Dest string			
80 (50H)	Dest string			
84 (54H)	Dest string			
88 (58H)	Dest string			
92 (5CH)	Dest string			
96 (60H)	recon_time	keep_alive	web_port	
100 (64H)	UDPF_pos	UDPF_code	UDPF_mask	ver
104 (68H)	func_sel	Group_IP		
108 (6CH)	Group_IP	io_set	func_en	sm_param_t
112(70H)	func_sel2	reserve 2 bytes		user param
	user param(52 bytes)			

图 4 参数格式

如图 4 所示为模块参数格式，发送和接收参数时，第 1 字节先发送。上图

中的字节采用大端模式 (big-endian)，例如 Local IP Addr 的高字节在左边。各参数的含义请参考《ZLSN2000 数据手册》。

1. Local IP Addr (本地 IP 地址): 4 字节。
2. Net Mask(子网掩码): 4 字节。
3. GateWay (网关): 4 字节。
4. Dest IP (目的 IP): 4 字节。具有 DNS 功能的联网产品, 该字段的作用被 DestString 字段代替, 该字段无效; 不具有 DNS 功能的联网产品, 该字段使用 4 字节表示的目的 IP 地址。您的产品是否具有 DNS 功能请咨询卓岚公司。
5. Local IP Port (本地端口): 2 字节。
6. Dest Port (目的端口): 2 字节。
7. Work mode (工作模式): 1 字节。数值 0、1、2、3 分别对应: 服务器模式 (TCP Server)、客户端模式 (TCP Client)、UDP 模式、UDP 组播模式。
8. Pad (填充区): 10 字节, 应该全部为 0。
9. Pad2 (填充区 2, 也即 DevID): 6 字节。当写入参数含有这个字段时, 应该在这里填写设备的正确 ID, 否则如果写入的 ID 不正确, 那么设备会丢弃所有其它的写入参数。
10. Baundrate (波特率): 1 字节。从 0~13 (13 表示 460800) 分别对应: 1200、2400、4800、7200、9600、14400、19200、28800、38400、57600、76800、115200、230400、460800。
11. Device Name (设备名称): 10 字节。必须是以 0 结尾的可见字符串。
12. Parity (奇偶位): 1 字节。0~4 分别对应: None、Odd、Even、Mark、Space 五种方式。
13. Gap Time (间隔时间): 1 字节。
14. Packing length (包长度): 2 字节, 数值范围 1~1400。
15. F_end en (帧尾字符有效位): 1 字节, 0、1 分别表示帧尾规则不起作用、起作用。V1.472 版本开始这个字节不再有效。
16. F_end byte (帧尾字符): 1 字节。V1.472 版本开始这个字节不再有效。
17. F_start en (帧首字符有效位): 1 字节, 0、1 分别表示帧首规则不起作用、起作用。V1.472 版本开始这个字节不再有效。
18. F_start byte (帧首字符): 1 字节。V1.472 版本开始这个字节不再有效。
19. DHCP en (DHCP 有效位): 1 字节。0、1 分别表示使用静态 IP、使用 DHCP

获得 IP。

20. Flow Control (流控方式): 1 表示采用 CTS、RTS 流控; 0 表示不采用流控。
21. Dest_Mode(目的模式): 1 字节。0、1 分别表示静态模式和动态模式。
22. DataSize (串口位数): 8~5bit, 分别对应 0、1、2、3。
23. AppPro (转化协议): 0 表示透明传输, 1 表示 Modbus TCP 和 Modbus RTU 之间的转化, 2 表示 RealCom。
24. Status (修改模式): 读取该参数时, Status 的 bit0=1 表示当前 TCP 连接已建立或者处于 UDP 状态, 否则 bit0=0。这个字段提供了读取联网模块当前状态的一个方法。
25. DnsServerIP (DNS 服务器 IP): 设置为 DNS 服务器 IP, 4 字节。
26. Dest string (目的地址): 具有 DNS 的联网产品, 该字段表示目的 IP 字符串, 例如写入“192.168.0.3”这个字符串为十六进制: 0x31, 0x39, 0x32, 0x2e, 0x31, 0x36, 0x38, 0x2e, 0x30, 0x2e, 0x33, 0x00。注意最后添加一个字符串末尾 0。不具有 DNS 功能的联网产品该字段无效。
27. Recon_Time (断线重连时间): 1 字节, 范围 0~255。
28. Keep_Alive (保活定时时间): 1 字节, 范围 0~255。
29. Web_port(网页访问端口): 2 字节, 通过浏览器访问网页的端口。
30. UDPF_pos、UDPF_code、UDP_mask (UDP 应用层滤波参数): 共 3 个字节, 一般都设置为 0, 即可。该参数目前已经无效。
31. ver (模块版本): 1 个字节, 不可修改。ver=0 是表示版本号为 1.383。实际的版本号为 383+ver, 例如 ver=117, 表示版本为 1.500。
32. func_sel (模块支持的功能): 1 个字节, 无法修改。每位的具体含义如下:
bit0=1 表示支持网页下载; bit1=1 表示支持 DNS 域名系统; bit2=1 表示支持 REAL_COM 协议; bit3=1 表示支持 Modbus TCP 转 RTU; bit4=1 表示支持串口修改参数; bit5=1 表示支持自动获取 IP (DHCP); bit6=1 表示支持存储扩展 EX 功能; bit7=1 表示支持多 TCP 连接。
33. Group_IP (UDP 组播地址): 4 个字节, 范围 224.0.0.0 到 239.255.255.255。在整个包中的位置如下。

ff	ff	d3	08	04	44	00	b2	c3	26	5a	4c	02	c0	a8	01
de	ff	ff	ff	00	c0	a8	01	01	c0	a8	01	03	10	64	10
64	00	38	38	38	38	38	38	38	38	38	38	28	4f	98	22
cd	a1	0b	6c	69	31	00	00	00	00	00	00	00	00	03	05
14	00	00	00	00	00	00	01	00	00	00	08	08	04	04	31
39	32	2e	31	36	38	2e	31	2e	33	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	0c	3c	00
50	00	00	00	9c	be	e6	5a	4c	01	00	10	05	27	00	00
09	04	00	00	00	00	0a	04	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

34. io_set (IO 端口设置): IO 端口控制字, 请参考 IO 控制相关文档。
35. func_en (功能选择): 功能选择/使能控制字。bit0=1 打开“数据重启功能”, bit1=1 打开“向中心服务器发送模块参数功能”。bit2=1 打开“修改参数需密码功能”。bit3=1 打开“UDP 进制接收广播包功能”。请参考相关文档。bit4=1 打开“P2P 功能”。
36. sm_param_t (sm 参数时间): 向中心服务器发送模块参数功能发送间隔时间, 单位为分钟。
37. func_sel2 (模块支持的功能 2): 该模块支持的高级功能。bit0=1 表示支持 IO 配置功能。bit1=1 表示支持 UDP 组播功能。bit2=1 表示支持多目标 IP 功能。
38. 保留 54 字节。该保留的 54 字节留作将来升级使用。参数总字节数为 167 字节 (对于 UDP 管理端口协议, 需加上头部 2 个标识和 1 个命令类型, 总长度为 170 字节)。

参数格式的 C 描述为:

```
typedef unsigned char  zl_u8;
typedef char           zl_s8;
typedef unsigned short zl_u16;
typedef short          zl_s16;
typedef unsigned long  zl_u32;
typedef bit            zl_bool;
typedef signed   long  zl_s32;

typedef zl_u32 IP_ADDR;

#define MAX_KEY_LEN           10
#define MAX_DEV_NAME_LEN     10
#define ETHER_ADDR_LEN       6
#define DNS_NAME_MAX_LEN     30

struct SSServerParam
{
    IP_ADDR param_local_ip;
```

```
IP_ADDR param_net_mask;
IP_ADDR param_gate_way;
IP_ADDR param_dest_ip;
zl_u16 param_local_port;
zl_u16 param_dest_port;
zl_u8 param_work_mode;
zl_s8 param_key[MAX_KEY_LEN];
zl_u8 ether_addr[ETHER_ADDR_LEN];
zl_u8 baudrate_index;
zl_s8 dev_name[MAX_DEV_NAME_LEN];
zl_u8 param_parity;
zl_u8 param_max_no_s_data_interval;
zl_u16 param_max_data_len;
zl_u8 param_fram_end_en;
zl_u8 param_fram_end_byte;
zl_u8 param_fram_start_en;
zl_u8 param_fram_start_byte;
zl_u8 param_ip_mode;
zl_u8 param_flow_control;
zl_u8 param_dest_dynamic;
zl_u8 param_data_bits;
zl_u8 app_protocol;
zl_u8 status;
IP_ADDR dns_server_ip;
zl_s8 dns_name[DNS_NAME_MAX_LEN];
zl_u8 keep_alive_time;
zl_u8 reconnect_time;
zl_u16 web_port;
zl_u8 udp_filter_pos, udp_filter_code, udp_filter_mask;
zl_u8 ver;
zl_u8 func_sel;
IP_ADDR udp_group_ip;
zl_u8 io_set;
zl_u8 func_en;
zl_u8 server_mode_param_t;
zl_u8 func_sel2;
zl_u8 var1[2];
zl_u8 var2[52];
};
```

2.4. 读参数步骤

将图 1 所示命令以最快地、连续地发送给 ZLSN2000。即每个字节之间的停顿不应该长于 Gap Time（间隔时间），请参考 ZLSN2000 说明书关于 Gap Time 的说明，对于 115200bps，默认 Gap Time 为 3ms。发送的波特率必须是模块当前的波特率。发送命令后在串口即可接收到指定内容的参数。

以读取目的域名或 IP 字符串为例，在图 4 中查到 Dest String 的范围为 42H

到 60H, 字节数为 1EH。那么命令应该为: ed f2 a3 56 ca db 91 84 b0 d7 **00 42 1E**。其中 00 表示读命令, 不写 Flash, 42 为参数开始位置, 1E 为读取的字节数。

以 16 进制用 ZLComDebug 发送以上命令的结果如图 5, 可以看到此时在串口接收到目的 IP 字符串。

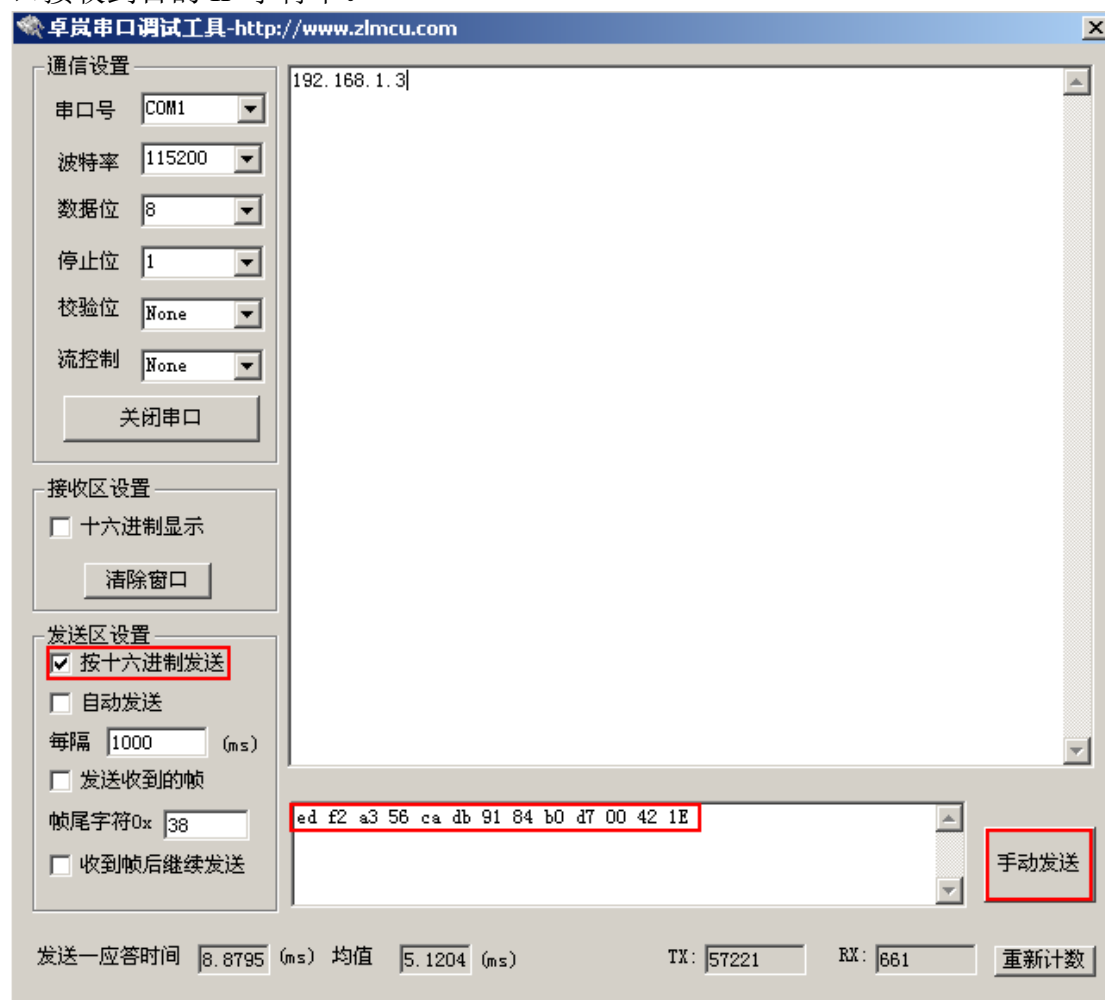


图 5 读取参数演示

使用技巧: 在通信中读取参数时, 可能返回的数据可通信数据冲突。比如查询连接状态为: ed f2 a3 56 ca db 91 84 b0 d7 00 3d 01, 返回 00 表示未连接 01 表示已经连接。但是这个可能和通信数据冲突。为此可以查询改为: ed f2 a3 56 ca db 91 84 b0 d7 00 3d 05, 这样返回的就是 00 08 08 04 04, 由图 4 知后面的 08 08 04 04 是 DNS, 这个 DNS 一般是固定的 08080404, 这样这个相当于一个标志性的尾缀, 为用户区分是通信数据和查询数据提供了区分。

1. 用户可以通过 ZLVircom 程序查看参数，以检查参数是否被成功修改。
2. 有的时候读取参数需要 2-3 秒钟的原因。这是因为你将模块至于 TCP 客户端，在 TCP 客户端模式时，如果连接未建立（LINK 灯未亮），那么有时候读取、写入参数会有延迟。原因是 TCP 客户端在进行连接操作时，外部命令响应是暂停的。如果处于 TCP 服务端或者 UDP 模式，或者连接已建立则不会有此问题。
3. 读取参数时的断续情况。使用参数读取命令读取参数时，模块会不断输出参数内容，但是整个输出内容可能会在中间打断几个毫秒。这是因为参数的输出和网口到串口数据的输出采用不同的机制，后者不会发生断续的情况。
4. 对于新的版本模块，那么如果一次写入的参数含有设备 ID 的，那么必须写入和设备一样的 ID，否则所有写入的参数都将被丢弃。如果不知道先读取 ID，填写在 ID 区域。

3. 典型应用

3.1. 读取连接状态

向模块的串口发送如下命令

```
ed f2 a3 56 ca db 91 84 b0 d7 00 3d 01
```

模块会返回一个字节的**状态字**，如果状态字 0x00 表示未连接（在串口调试助手中请使用 hex 方式显示该数据）。如果为 0x01 则表示已连接。所谓连接就是 TCP 连接已经建立或者处于 UDP 模式，已连接也等价于 LINK（一般为绿色）灯点亮（LINK 引脚电平为 0）。注意 UDP 模式查询出来的都是 1。

如果模块出于繁忙工作状态中，则可能暂时不会响应以上命令，以上命令会被简单忽略。所以用户的接收程序需要有一个等待超时机制。繁忙的工作状态包括：

1. 模块正在向目的 IP 连接过程中，如果无法建立 tcp 连接，这个连接持续时间可能最长为 5 秒钟。
2. 模块正在初始化启动中，根据是否使用 DHCP 的 IP 模式，启动时间有所不同。

以上发送的串口命令不会被和卓岚模块连接的另一端（例如 PC 机）接收，这保证了发送以上命令不会影响通信协议。

以上命令发送后到收到状态字的时间间隔平均为 7ms 左右，所以等待超时时间为 10ms 应该是可以的。

对于处于 TCP 客户端的联网模块，用户可以每隔 100ms（根据用户需要选择，在允许情况下尽量大一些）发送以上命令，检测连接是否建立。一旦连接建立则可以发送用户协议数据。

可以查询改为：ed f2 a3 56 ca db 91 84 b0 d7 00 3d 05，这样返回的就是 00 08 08 04 04，由图 4 知后面的 08 08 04 04 是 DNS，这个 DNS 一般是固定的 08080404，这样这个相当于一个标志性的尾缀，为用户区分是通信数据和查询数据提供了区分。

3.2. 控制联网产品连接

问：是否可以实现通过 MCU 发送控制命令来让联网产品发起客户端连接？
发送命令让让联网产品断开连接？

答：是可以的。具体的做法是让联网产品平常置于 TCP 服务器模式，需要发起连接的时候就发送命令让联网产品置于 TCP 客户端模式，此后联网产品即可以自动连接。

发起连接的命令如下：

```
ed f2 a3 56 ca db 91 84 b0 d7 01 14 01 01
```

断开连接的命令为：

```
ed f2 a3 56 ca db 91 84 b0 d7 01 14 01 00
```

3.3. 读取本地 IP 地址

发送指令：

```
ed f2 a3 56 ca db 91 84 b0 d7 00 00 04
```

返回数据为 ca a8 01 c8 这种十六进制的 IP 地址。

3.4. 修改 dns 服务器

首先要将 IP 模式改为静态，否则无法手动设置 dns 服务器 IP。

```
ed f2 a3 56 ca db 91 84 b0 d7 01 38 01 00
```

之后设置新的 dns 服务器 IP。注意两个命令之间应该有 1 秒钟时间间隔。

```
ed f2 a3 56 ca db 91 84 b0 d7 01 3e 04 08 08 08 08
```

3.5. 查看系统是否初始化完毕

当 DHCP 被使能的时候，系统的初始化时间是不定的。为了确认系统是否已经初始化完成。可以发送一个串口命令，如果有应答则说明系统初始化完毕。例如发送命令：ed f2 a3 56 ca db 91 84 b0 d7 00 03 01。如果返回一个字节的 IP 末尾字节，则说明系统初始化完毕。但是用户不能太频繁发送命令，每隔 1 秒钟发送一次命令是建议值，因为太频繁的发送命令本身将会减缓系统的启动时间。

3.6. 一次设置方法

实际上当系统启动的时候，用户可以选择通过串口发送一次性的参数设置命令设置用户需的所有参数，例如从串口发送如下命令：

```
ed f2 a3 56 ca db 91 84 b0 d7 01 00 68 c0 a8 01 c8 ff ff ff 00 c0 a8 01 01 c0 a8  
01 03 10 64 00 50 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0b 5a 4c
```

```
44 45 56 30 30 30 31 00 00 03 05 14 00 00 00 00 00 00 01 00 00 00 ca 60 d1 85
77 77 77 2e 62 61 69 64 75 2e 63 6f 6d 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 0c 3c 00 50 00 00 00 12
```

命令解释如下：ed f2 a3 56 ca db 91 84 b0 d7（识别流） 01（写参数，且不保存参数） 00（从 00 位置开始写） 68（共写 0x68 个字节） **c0 a8 01 c8**（IP 地址 **192.168.1.200**） **ff ff ff 00**（子网掩码 **255.255.255.0**） **c0 a8 01 01**（网关 **192.168.1.1**） c0 a8 01 03（未定义） 10 64（端口 4196） 00 50（目的端口 80） **01**（工作模式 **TCP 客户端**） 00 00 00 00 00 00 00 00 00 00 5a 4d 01 02 03 04（设备 ID，必须为正确的对应的 IP，如果不知道先读取 ID，填写在这个区域） 0b（波特率为 115200） 5a 4c 44 45 56 30 30 30 31 00（设备名称） 00（校验位为无） 03（间隔时间为 3ms） 05 14（包长度） 00 00 00 00（帧首尾字符） **00**（静态还是动态 IP） 00（流控方式） 01（目的模式） 00（串口位数） 00（转化协议） 00（未定义） **ca 60 d1 85**（DNS） **77 77 77 2e 62 61 69 64 75 2e 63 6f 6d 00**（目的域名 www.baidu.com 的十六进制） 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0c（重连时间） 3c（保活定时时间） 00 50（网页端口） 00 00 00（UDP 过滤） 12（版本）

最新版本的参数为 169 字节，也就是改为 ed f2 a3 56 ca db 91 84 b0 d7 03 00 a9 ...

3.7. 串口重启设备

方法 1

固件版本 V506 及其以上版本请使用 07 命令来重启模块（参考表 2. 专用命令类型）。07 命令和 03 命令是类似的，只不过 07 命令会重启模块。例如发送：ed f2 a3 56 ca db 91 84 b0 d7 07 1f 01 00。这个命令试图将设备 ID 的第一个直接改写为 00，但是由于 ID 是无法修改的，这个命令实际的效果只是重启了模块。

方法 2

对于固件版本为 V506 之前的，采用如下方法重启模块。交替发送 ed f2 a3 56 ca db 91 84 b0 d7 01 3e 04 08 08 08 08 或者 ed f2 a3 56 ca db 91 84 b0 d7 01 3e 04 08 08 04 04。说明：如果上一次重启发送的是第一个命令，那么这次让模块

重启发送第二个命令，反之发送第一个命令。每发送一次后模块就会重启。

3.8. 指定 DNS 服务器

讲述如何在动态获取 IP 的方式下手动指定 DNS 服务器。在动态获取 IP（IP 模式为动态或 DHCP）方式下，DNS 服务器也是自动获取的。如果既要动态获取 IP，又要手动设置 DNS 服务器，那么实现方法的思路是：先用动态 IP 模式启动模块，然后修改 DNS 服务器。这样模块会自动保留自动获取的 IP，同时又让用户可以手动设置 IP。

1. 将模块设置为动态 IP 方式。发送：ed f2 a3 56 ca db 91 84 b0 d7 **01 38 01 01**
2. 检查系统启动完毕：即每隔一秒发送：ed f2 a3 56 ca db 91 84 b0 d7 00 03 01。如果收到一个字节的应答，则说明启动启动完毕。
3. 改为静态 IP 模式。发送指令 ed f2 a3 56 ca db 91 84 b0 d7 **01 38 01 00**
4. 同样用第二步方法检查系统启动完毕。
5. 发送指令 ed f2 a3 56 ca db 91 84 b0 d7 **01 3e 04 08 08 08 08**。将 DNS 服务器 IP 设置为 8.8.8.8，注意最后的 4 个字节即为 DNS 服务器 IP 的十六进制表示。

3.9. 读取设备名称

ed f2 a3 56 ca db 91 84 b0 d7 00 26 0a

3.10. 写入设备名称

ed f2 a3 56 ca db 91 84 b0 d7 03 26 0a 61 62 63 64 65 66 67 68 69 00

3.11. 读取设备 ID

V1.512 以后的版本可以支持 ID 的读出，读取的 ID 为 6 字节。读取指令为：
ed f2 a3 56 ca db 91 84 b0 d7 00 1f 06

3.12. 修改目的 IP

对于支持 DNS 的模块请修改 Dest string（目的地址）这个参数，不支持 DNS 的模块请修改 Dest IP（目的 IP）字段。

ed f2 a3 56 ca db 91 84 b0 d7 02 73 size 01 01 00 07 (size-6) (4byte IP) (2byte

port) (4byte IP) (2byte port) 00

在这个指令中 ed f2 a3 56 ca db 91 84 b0 d7 为识别流。02 是专门针对 MDIP 产品的命令字。size 就是后续数据的总长度。01 01 00 07 是固定数据。(size-6) 是总长度减 6 字节。接下来是 4 字节 IP+2 字节端口，都是大端格式，最多有 7 个组合，即 7 个目标 IP。也可以为 1~7 个任意的长度。最后有一个 00 字符。

例如命令：

```
ed f2 a3 56 ca db 91 84 b0 d7 02 73 12 01 01 00 07 0c c0 a8 01 58 04 01 c0 a8
01 58 04 02 00
```

这里 size 为 18 字节。并设置 2 个目的 IP 和端口，目的 IP 为：192.168.1.88:1025 和 192.168.1.88:1026。

发送这个命令后模块立即向新增加的 192.168.1.88 的 1026 端口发起连接。

注意：

1. 这里的 02 命令本身不保存目的 IP，即掉电后串口写入的目的 IP 端口会丢失。
2. 可以用 07 命令代替 02 来保存写入的 IP 地址。但是也会关闭正在通信的其它 TCP 连接。只有 02 命令可以不关闭当前老的 TCP 连接。
3. 还有一个特性的是 02 命令只能够增加新的 TCP 连接，不能够删除（关闭）老的 TCP 连接。如果需要关闭则请使用 07 命令。

3.16.用户参数空间使用

从图 4 可以看到最后部分是 User Param，这部分空间从位置 115 开始的共 52 个字节是可以被用户使用的。例如用于：用户暂存数据，用于将数据存于参数空间中，然后通过“定时向服务器发送数据”和“参数获取”向服务器传输数据。

但是注意存在如下功能的模块 User Param 部分已经使用，用户不应该使用该部分区域，这些功能包括：无线功能、VLAN 功能、多目标 IP 的 MDIP 器件、代理服务器功能、注册包心跳包。

写入数据的方法是，向模块的串口发送如下指令：

```
ed f2 a3 56 ca db 91 84 b0 d7 02 73 (size) ff (size-3) (content) 00
```

其中 size 是一个大小，需要小于 52，表示 size 后面还有多少数据。content 是写入的数据。例如 size 为 11 的时候的指令为：

```
ed f2 a3 56 ca db 91 84 b0 d7 02 73 0b ff 08 d1 d2 d3 d4 d5 d6 d7 d8 00
```

其中 d1~d8 是用户写入的任何数据。实际写入 User Param 的数据为 ff 08 d1 d2 d3 d4 d5 d6 d7 d8 00。开头的 ff 和末尾的 00 是为了后续格式兼容而做的附加数据。

以上写入 User Param 的数据不一定会保存到模块 Flash 中（掉电可能丢失），而是暂时存在内存中。但是也有可能被手动参数保存操作时，保存起来。以上指令不会影响（例如关闭）当前模块的 TCP 连接。

以上的命令类型为 02。02 这个命令用户设置 MDIP 型号的产品的多目的 IP 是很好用的。但是一般的话用 03 命令（保存参数的写指令）即可。但是如果用户想保存参数然后立即重启的用 07 命令代替 03 命令。

3.17.带界面的自定义参数

和“用户参数空间使用”不同的是，这里的自定义参数是可以用 zlvircom 查看并且编辑的。使用的参数是类似“MDIP 器件的多目的设置”的多目的 IP 区域，只不过由于器件不是多目的的 MDIP 器件，所以这个区域是可以用户自己设置但是不会被认为是一个目的 IP。

例如

ed f2 a3 56 ca db 91 84 b0 d7 01 73 0c 01 01 00 07 06 00 00 00 09 00 00 00

将 IP 为：0.0.0.9，端口为：0，写入设备。之后可以用 zlvircom 的高级参数的第一个多目的进行查看和编辑。

3.18.注册包和心跳包的串口写入

假如一个设备的注册包为“123456”，心跳包为“abcdef”，则在更多高级选项中设置如下，参考《卓岚定制心跳包使用说明》

图 6 注册包心跳包例子

这个参数是被设置到用户参数空间的。

使用 ed f2 a3 56 ca db 91 84 b0 d7 00 73 25 指令发送到模块的串口可以获取设置了以上参数的参数空间内容：“00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 31 32 33 34 35 36 00 00 00 61 62 63 64 65 66 00 00 00 00 00 00 00”。可见 0x82 位置开始的是注册包的字符串“31 32 33 34 35 36”，末尾再跟 3 个 00；接着就是

注意请将 0x73 到 0x82 的空间也设置为全 00，例如只设置心跳包为“hello”的方法为，需要将“00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 68 65 6c 6c 6f 00 00 00 68 65 6c 6c 6f 00 00 00”写入到参数区 0x73 开始的位置，这个数据长度为 0x1F。这里的“00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00”是 0x73 到 0x82 需要设置为 00 的部分；这里的“68 65 6c 6c 6f 00 00 00”是 hello 的注册包（注意只有心跳包时，注册包也要设置为和心跳包一样；后面的“68 65 6c 6c 6f 00 00 00”是心跳包。注意只有注册包的时候，后面的心跳包部分参数就不需要设置。

```
ed f2 a3 56 ca db 91 84 b0 d7 03 73 1f 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 68 65 6c 6c 6f 00 00 00 68 65 6c 6c 6f 00 00 00
```

**ed f2 a3 56 ca db 91 84 b0 d7 07 73 1f 00 00 00 00 00 00 00 00 00 00 00 00
00 00 68 65 6c 6c 6f 00 00 00 68 65 6c 6c 6f 00 00 00**

邮箱: support@zlmcu.com