



## 【EC04-AT&JSON 指令集】



**成都亿佰特电子科技有限公司**  
Chengdu Ebyte Electronic Technology Co.,Ltd.

# 目录

第 1 章 AT 指令部分.....	2
1.1 进入 AT 指令.....	3
1.2 退出 AT 指令.....	3
1.3 重启设备.....	3
1.4 恢复出厂设置.....	3
1.5 查询固件版本.....	4
1.6 查询设备 ID.....	4
1.7 查询 IMEI.....	4
1.8 查询 ICCID.....	5
1.9 查询卡状态.....	5
1.10 查询信号强度.....	5
1.11 查询网络状态.....	6
1.12 基站定位.....	6
1.13 GPS 定位（支持 GPS 定位设备独有）.....	6
1.14 查询/配置串口参数.....	7
1.15 查询/设置网络协议参数.....	7
1.16 查询/设置链路使能.....	8
1.17 查询链路状态.....	8
1.18 查询/设置心跳包.....	9
1.19 查询/设置注册包.....	9
1.20 查询/设置 MQTT 连接信息.....	10
1.21 查询/设置订阅主题.....	11
1.22 查询/设置发布主题.....	11
1.23 查询/设置 HTTP 模式.....	12
1.24 查询/设置多链路协议分发模式.....	12
1.25 查询/设置边缘采集.....	13
1.26 查询/配置 Modbus TCP 转 Modbus RTU 使能.....	13
1.27 网络升级.....	14
第 2 章 JSON 指令.....	14
1.28 获取/配置设备信息.....	14
1.29 获取/配置串口配置信息.....	15
1.30 获取/配置 Socket 信息.....	16
1.31 设备重启.....	21
1.32 固件升级.....	21
第 3 章 短信指令说明.....	21
修订历史.....	23
关于我们.....	23

## 第 1 章 AT 指令部分

- 1、所有指令必须在配置模式下才有效。
- 2、所有的 AT 的格式都是 “\r\n” 结尾。
- 3、所有指令的应答都是 “\r\n” 开头，“\r\n” 结尾的格式。
- 4、本文所有 “\r”，“\n” 是指回车和换行的转译字符，而是字符串。

错误码表：

错误码	说明
-1	无效的命令格式
-2	无效的命令
-3	暂未定义
-4	无效的参数
-5	暂未定义

快速指令表：

指令	说明
AT+EXAT	退出 AT
AT+REBT	重启设备
AT+ RESTORE	恢复出厂配置
AT+VER	查询固件版本
AT+SN	查询设备 ID
AT+IMEI	查询 IMEI
AT+ICCID	查询 ICCID
AT+CPIN	查询卡状态
AT+CREG	查询网络状态
AT+LBS	基站定位
AT+GPS	GPS 定位（支持 GPS 定位设备独有）
AT+UART	查询串口参数
AT+SOCK=0,"base"?	查询网络协议参数
AT+SOCK=0,"link/sw"?	查询链路使能
AT+SOCK=0,"link/status"?	查询链路状态
AT+SOCK=0,"link/alive"?	查询心跳包
AT+SOCK=0,"regist"?	查询注册包
AT+SOCK=0,"mqtt/connnet"?	查询 MQTT 连接信息
AT+SOCK=0,"mqtt/sub"?	查询订阅信息
AT+SOCK=0,"mqtt/pub"?	查询发布信息
AT+SOCK=0,"http"?	查询 HTTP 参数
AT+POTOCOL	查询协议分发模式
AT+COLLECTER=0?	查询边缘采集
AT+MODBUS_COVERTO	查询 Modbus 使能状态

## 1.1 进入 AT 指令

发送+++不用换行，收到\r\n+OK\r\n后 5S 内发送一条正确的 AT 指令即可进入配置模式。

### 【举例】

先发送+++

收到\r\n+OK\r\n

再发送 AT\r\n

收到\r\n+OK\r\n

## 1.2 退出 AT 指令

指令	EXAT
功能	退出 AT 指令模式
发送	AT+EXAT<CR><LF>
返回	<CR><LF>+OK<CR><LF>

### 【举例】

发送： AT+EXAT\r\n

收到： \r\n+OK\r\n

## 1.3 重启设备

指令	REBT
功能	重启设备
发送	AT+REBT<CR><LF>
返回	<CR><LF>+OK<CR><LF>

### 【举例】

发送： AT+REBT\r\n

收到： \r\n +OK\r\n

## 1.4 恢复出厂设置

指令	RESTORE
功能	恢复出厂设置
发送	AT+RESTORE<CR><LF>
返回	<CR><LF>+OK<CR><LF>

### 【举例】

发送： AT+RESTORE\r\n

收到: \r\n +OK\r\n  
等待设备重启。

### 1.5 查询固件版本

指令	VER
功能	查询固件版本
发送	AT+VER<CR><LF>
返回	<CR><LF>+OK=<verString><CR><LF>

**【举例】**

发送: AT+VER\r\n

收到: \r\n +OK=9024-0-10\r\n

### 1.6 查询设备 ID

指令	SN
功能	查询设备 ID
发送	AT+SN<CR><LF>
返回	<CR><LF>+OK=<snString><CR><LF>

**【举例】**

发送: AT+SN\r\n

收到: \r\n +OK=xxxxxxxxxxxxxx\r\n

每台设备的 SN 码都不同。

### 1.7 查询 IMEI

指令	IMEI
功能	查询 IMEI
发送	AT+IMEI<CR><LF>
返回	<CR><LF>+OK=<imeiString><CR><LF>

**【举例】**

发送: AT+IMEI\r\n

收到: \r\n +OK =xxxxxxxxxxxxxx\r\n

每台设备的 IMEI 码不同。

## 1.8 查询 ICCID

指令	ICCID
功能	查询 ICCID
发送	AT+ICCID<CR><LF>
返回	<CR><LF>+OK=<iccidString><CR><LF>

**【举例】**

发送：AT+ICCID\r\n

收到：\r\n+OK=xxxxxxxxxxxx\r\n

## 1.9 查询卡状态

指令	CPIN
功能	查询卡的状态
发送	AT+CPIN<CR><LF>
返回	<CR><LF>+OK=<status><CR><LF>
备注	Status, 1 (正常), 0 (无卡或者卡异常)

**【举例】**

发送：AT+CPIN\r\n

收到：\r\n+OK=1\r\n

## 1.10 查询信号强度

指令	CSQ
功能	查询信号强度
发送	AT+CSQ<CR><LF>
返回	<CR><LF>+OK=<status><CR><LF>
备注	Status, 信号强度

**【举例】**

发送：AT+CSQ\r\n

收到：\r\n+OK=22\r\n

### 1.11 查询网络状态

指令	CREG
功能	查询网络状态
发送	AT+CREG<CR><LF>
返回	<CR><LF>+OK=<status><CR><LF>
备注	Status, 1 (注册网络成功), 其他正常连接基站或者连接失败

**【举例】**

发送: AT+CSQ\r\n

收到: \r\n+OK=1\r\n

### 1.12 基站定位

指令	LBS
功能	基站定位
发送	AT+LBS<CR><LF>
返回	<CR><LF>+OK=<Lng, Lat><CR><LF>
备注	Lng: 经度 Lat: 纬度

**【举例】**

发送: AT+CSQ\r\n

收到: \r\n+OK=103.936800,30.767124\r\n

### 1.13 GPS 定位 (支持 GPS 定位设备独有)

指令	GPS
功能	基站定位
发送	AT+GPS<CR><LF>
返回	<CR><LF>+OK=<Lat, lng><CR><LF>
备注	Lat: 纬度 Lng: 经度

**【举例】**

发送: AT+GPS\r\n

收到: \r\n+OK=30.767124N,103.936800E\r\n

### 1.14 查询/配置串口参数

指令	UART
功能	查询/配置串口参数
发送（查询）	AT+UART<CR><LF>
返回（查询）	<CR><LF>+OK<baudrate, uartData, uartStop, parity, uartFlow, packLen, packTime><CR><LF>
发送（配置）	AT+UART=<baudrate, uartData, uartStop, parity, uartFlow, packLen, packTime><CR><LF>
返回（配置）	<CR><LF>+OK<CR><LF>
备注	Baudrate, 波特率: 1200、2400、4800、9600、19200、38400、57600、115200、230400、460800; Uartdata, 数据位: 8、7; Uartstop, 停止位: 1、2; Parity, 校验位: NONE、ODD、EVEN; Uartflow, 流控: 0 为关闭; Packlen, 打包长度: 50~1024, 默认 1024; Packtime, 打包时间: 2~2000, 单位 5ms, 默认 10 * 5ms;

**【举例】**

查询:

发送: AT+UART\r\n

收到: \r\n+OK=115200,8,1,NONE,0,1024,10\r\n

设置:

配置波特率为 9600, 数据位为 8, 停止位为 1, 无校验无流控, 打包长度为 1024、打包时间 10ms。

发送: AT+UART=9600,8,1,NONE,0,1024,10\r\n

收到: \r\n+OK\r\n

### 1.15 查询/设置网络协议参数

指令	AT+SOCK=<SockID," base" >
功能	设置/查询网络协议参数格式
发送（查询）	AT+SOCK=<SockID," base" ? >
返回（查询）	<CR><LF>+OK=<SockID,"base",sw,workMode,sockType,shortLinkTime,DesIP,desPort><CR><LF>
发送（设置）	AT+SOCK=<SockID,"base",sw,workMode,sockType,shortLinkTime,DesIP,desPort><CR><LF>
返回（设置）	<CR><LF>+OK<CR><LF>
备注	SockID, socket 号: 0、1、2、3、4 "base", 关键字 Sw, 链路开关: 0 (关闭)、1 (开启) Workmode, 工作模式: 0 (透传), 1 (HTTP), 2 (MQTT) Socktype, 工作协议: 0 (UDPC)、1 (TCPC)、2 (UDPS)、3 (TCPS) Shortlinktime, 短连接时间 0 (关闭)、最大可配置 65535 DesIP, 目标 IP, 最大长度 64, 支持域名解析 DesPort, 目标端口, 0~65535



**【举例】**

查询：

发送：AT+SOCK=0,"base"?\r\n

收到：\r\n+OK=0,"base",0,0,1,0,test.ebyte.com,10687\r\n

设置：

发送：AT+SOCK=0,"base",0,0,1,0,test.ebyte.com,10687\r\n

收到：\r\n+OK\r\n

**1.16 查询/设置链路使能**

指令	AT+SOCK=<SockID," link/sw" >
功能	查询/设置链路使能
发送（查询）	AT+SOCK=<SockID," link/sw" ? >
返回（查询）	<CR><LF>+OK=< SockID," link/sw" ,sw><CR><LF>
发送（设置）	AT+SOCK=<SockID," link/sw" ,sw>
返回（设置）	<CR><LF>+OK<CR><LF>
备注	SockID, socket 号：0、1、2、3 "link/sw", 关键字 Sw, 链路开关：0（关闭）、1（开启）

**【举例】**

查询：

发送：AT+SOCK=0,"link/sw"?\r\n

收到：\r\n+OK=0,"link/sw",1\r\n

设置：

发送：AT+SOCK=0,"link/sw",1\r\n

收到：\r\n+OK\r\n

**1.17 查询链路状态**

指令	AT+SOCK=<SockID," link/status" ? >
功能	查询链路状态
发送	AT+SOCK=<SockID," link/status" ? ><CR><LF>
返回	<CR><LF>+OK=<SockID," link/status" ,status><CR><LF>
备注	SockID, socket 号：0、1、2、3 "link/status", 关键字 status, 链路状态：0（断开）、1（连接中）、2（连接成功）

**【举例】**

查询：

发送：AT+SOCK=0,"link/status"?\r\n

收到：\r\n+OK=0,"link/status",0\r\n

### 1.18 查询/设置心跳包

指令	AT+SOCK=<SockID," keep/alive" >
功能	查询网络链接状态
发送(查询)	AT+SOCK=<SockID," keep/alive" ? >
返回(查询)	<CR><LF>+OK=<SockID,"keep/alive",sw,keepAliveType,infoType,defaultData,time><CR><LF>
发送(设置)	AT+SOCK=<SockID,"keep/alive",sw,keepAliveType,infoType,defaultData,time><CR><LF>
返回(设置)	<CR><LF>+OK<CR><LF>
备注	<p>SockID, socket 号: 0、1、2、3、4                      "keepalive", 关键字                      sw, 开关: 0 (关闭), 1 (开启)                      Keepalivetype, 心跳包模式, 0 (IMEI)、1 (ICCID)、2 (SN)、3 (位置信息)、4 (自定义)                      infoType, 数据格式: 0 (ASCII)、1 (HEX)                      defaultData, 自定义心跳包内容: 最大配置 64 字节数                      Time, 心跳时间: 最大可配时间 65535s</p>

**【举例】**

查询:

发送: AT+SOCK=<SockID," keep/alive" ? >\r\n

收到: \r\n+OK=0,"keep/alive",1,0,0,KeepAlive,60\r\n

设置:

发送: AT+SOCK=0,"keep/alive",1,0,0,KeepAlive,60\r\n

收到: \r\n+OK\r\n

### 1.19 查询/设置注册包

指令	AT+SOCK=<SockID," regist" >
功能	查询/设置注册包
发送(查询)	AT+SOCK=<SockID," regist" ?>
返回(查询)	<CR><LF>+OK=<SockID,"regist",sw,registType,infoType,defaultData,way><CR><LF>
发送(设置)	AT+SOCK=<SockID,"regist",sw,registType,infoType,defaultData,way><CR><LF>
返回(设置)	<CR><LF>+OK<CR><LF>
备注	<p>SockID, socket 号: 0、1、2、3                      "regist", 关键字                      Sw, 开关: 0 (关闭), 1 (开启)                      registtype, 注册包模式, 0 (IMEI)、1 (ICCID)、2 (SN)、3 (自定义)                      infoType, 数据格式: 0 (ASCII)、1 (HEX)                      defaultData, 自定义心跳包内容: 最大配置 64 字节数                      Way: 0 (连接发送一次)、1 (每包数据前)</p>

**【举例】**

查询:

发送: AT+SOCK=0,"regist"?\r\n

收到: \r\n+OK=0,"regist",0,0,0,Regist,0\r\n

设置:

发送: AT+SOCK==0,"regist",0,0,0,Regist,0\r\n

收到: \r\n+OK\r\n

## 1.20 查询/设置 MQTT 连接信息

指令	AT+SOCK=<SockID," mqtt/connet" >
功能	查询/设置 MQTT 连接信息
发送 (查询)	AT+SOCK=<SockID," mqtt/connet" ? >
返回 (查询)	<CR><LF>+OK=<sockID,"mqtt/connet",sw,mode,productKey,deviceName,deviceSecret><CR><LF>
发送 (设置)	AT+SOCK=<sockID,"mqtt/connet",sw,mode,productKey,deviceName,deviceSecret><CR><LF>
返回 (设置)	<CR><LF>+OK<CR><LF>
备注	<p>SockID, socket 号: 0、1、2、3                      "mqtt/connet", 关键字                      Sw, 开关: 0 (关闭), 1 (开启)                      Mode, 云平台: 0 (阿里云), 1 (ONENET)、2 (百度云) 3 (3.1 标准 MQTT)                      productKey: 阿里产品密钥、百度设备密钥、ONENET 设备 ID                      deviceName: 阿里设备名称、百度用户名、ONENET 产品 ID                      deviceSecret: 阿里设备密钥、百度密码、ONENET 鉴权信息</p>

**【举例】**

查询:

发送: AT+SOCK=0,"mqtt/connect"?\r\n

收到: \r\n+OK=0,"mqtt/connet",1,0,a1GlhuTU1yN,DEV01,7e0e9087838525\r\n

设置:

发送: AT+SOCK=0,"mqtt/connet",1,0,a1GlhuTU1yN,DEV01,7e0e9087838525\r\n

收到: \r\n+OK\r\n

### 1.21 查询/设置订阅主题

指令	AT+SOCK=<SockID," mqtt/sub" >
功能	查询/设置订阅主题
发送（查询）	AT+SOCK=<SockID," mqtt/sub" ? >
返回（查询）	<CR><LF>+OK=<sockID,"mqtt/sub",sw,topicName,qos><CR><LF>
发送（设置）	AT+SOCK=<sockID,"mqtt/sub",sw,topicName,qos><CR><LF>
返回（设置）	<CR><LF>+OK<CR><LF>
备注	SockID, socket 号：0、1、2、3 "mqtt/sub", 关键字 Sw, 开关：0（关闭），1（开启） TopicName, 订阅主题 Qos, 服务等级，0、1

**【举例】**

查询：

发送：AT+SOCK=0,"mqtt/sub"?\r\n

收到：\r\n+OK=0,"mqtt/sub",1,/a1GlhuTU1yN/DEV01/user/1234,0\r\n

设置：

发送：AT+SOCK=0,"mqtt/sub",1,/a1GlhuTU1yN/DEV01/user/1234,0\r\n

收到：\r\n+OK\r\n

### 1.22 查询/设置发布主题

指令	AT+SOCK=<SockID," mqtt/pub" >
功能	查询/设置发布主题
发送（查询）	AT+SOCK=<SockID," mqtt/pub" ? >
返回（查询）	<CR><LF>+OK=<sockID,"mqtt/pub",sw,topicName,qos><CR><LF>
发送（设置）	AT+SOCK=<sockID,"mqtt/pub",sw,topicName,qos><CR><LF>
返回（设置）	<CR><LF>+OK<CR><LF>
备注	SockID, socket 号：0、1、2、3 "mqtt/pub", 关键字 Sw, 开关：0（关闭），1（开启） TopicName, 订阅主题 Qos, 服务等级，0、1

**【举例】**

查询：

发送：AT+SOCK=0,"mqtt/pub"?\r\n

收到：\r\n+OK=0,"mqtt/pub",1,/a1GlhuTU1yN/DEV01/user/1234,0\r\n

设置：

发送：AT+SOCK=0,"mqtt/pub",1,/a1GlhuTU1yN/DEV01/user/1234,0\r\n

收到：\r\n+OK\r\n

### 1.23 查询/设置 HTTP 模式

指令	AT+SOCK=<SockID," http" >
功能	查询/设置 HTTP 模式
发送（查询）	AT+SOCK=<SockID," http" ?><CR><LF>
返回（查询）	<CR><LF>+OK=<sockID,"http",method,url,head><CR><LF>
发送（设置）	AT+SOCK==<sockID,"http",method,url,head><CR><LF>
返回（设置）	<CR><LF>+OK<CR><LF>
备注	SockID, socket 号: 0、1、2、3 "http", 关键字 method, 方法: 0 (GET), 1 (POST) url, 资源路径, 最大可配置 64Bit head, 包头, 最大可配置 128Bit

**【举例】**

查询:

发送: AT+SOCK=0,"http"?\r\n

收到: \r\n+OK=0,"http",0,/url, Host:www.ebyte.com\r\n

设置:

发送: AT+SOCK=0,"http",0,/url, Host:www.ebyte.com\r\n

收到: \r\n+OK\r\n

### 1.24 查询/设置多链路协议分发模式

指令	AT+POTOCOL
功能	查询/设置多链路协议分发模式
发送（查询）	AT+ POTOCOL<CR><LF>
返回（查询）	<CR><LF>+OK=<Potocol><CR><LF>
发送（设置）	AT+POTCOL=<Potocol><CR><LF>（限制 2-255s, 0 为关闭）
返回（设置）	<CR><LF>+OK<CR><LF>
备注	Potocol, 开关: 0（关闭）、1（开启）

**【举例】**

查询:

发送: AT+POTOCOL\r\n

收到: \r\n+OK=0\r\n

设置:

发送: AT+POTOCOL=1\r\n

收到: \r\n+OK\r\n

### 1.25 查询/设置边缘采集

指令	AT+COLLECTER
功能	查询/设置边缘采集
发送（查询）	AT+COLLECTER=<ch>?<CR><LF>
返回（查询）	<CR><LF>+OK=<ch,sw,cmd,infoType,delayTime,crcSw><CR><LF>
发送（设置）	AT+COLLECTER=<ch,sw,cmd,infoType,delayTime,crcSw><CR><LF>
返回（设置）	<CR><LF>+OK<CR><LF>
备注	Ch, 通道: 0~9 Sw, 开关: 0 (关闭), 1 (打开) Cmd, 采集命令: 最大可配置字节数 64Bit infoType, 数据格式: 0 (ASCII)、1 (HEX) DelayTime, 时间间隔: 范围 0~65535 crcSw, CRC 校验使能: 0 (关闭)、1 (打开)

**【举例】**

查询:

发送: AT+COLLECTER=0?\r\n

收到: \r\n+OK=0,0,0,0,0,0\r\n

设置:

发送: AT+COLLECTER=0,0,0,0,0,0\r\n

收到: \r\n+OK\r\n

### 1.26 查询/配置 Modbus TCP 转 Modbus RTU 使能

指令	AT+MODBUS_COVERTOR
功能	查询/设置 Modbus TCP 转 Modbus RTU 使能
发送（查询）	AT+MODBUS_COVERTOR<CR><LF>
返回（查询）	<CR><LF>+OK=<sw,mtcpid><CR><LF>
发送（设置）	AT+MODBUS_COVERTOR=<sw,mtcpid><CR><LF>
返回（设置）	<CR><LF>+OK<CR><LF>
备注	Sw, 开关 Mtcpid, 指定转换的 modbus 地址: 0 表示任意地址, 1~255

**【举例】**

查询:

发送: AT+MODBUS\_COVERTOR \r\n

收到: \r\n +OK=1,0 \r\n

设置:

发送: AT+MODBUS\_COVERTOR=1,0\r\n

收到: \r\n+OK\r\n

## 1.27 网络升级

指令	FOTA
功能	网络升级
发送	AT+FOTA=<url,username,password><CR><LF>
返回	<CR><LF>+OK=<status><CR><LF>
备注	url: 资源路径 username: 用户名 password: 密码

## 第 2 章 JSON 指令

支持网络、串口配置如下：



## 1.28 获取/配置设备信息

字段	说明
type	消息类型，读：getStatus，返回：getStatusRes
id	设备编号 例如 W020923P00004CFA7
imei	4G 模组入网编号
version	固件版本号
iccid	物联网卡 ID

csq	信号强度 0-99
latitude	纬度
longitude	经度
linkNum	设备链路总数(不包括平台使用链路数, 仅用户可使用配置的链路总数)
flow	流量

读取参数请求:

```
{"type": "getStatus"}
```

设备返回

```
{
  "type": "getStatusRes",
  "id": "2020060200000023c061",
  "imei": "869516059780150",
  "iccid": "898602B6102080006958",
  "latitude": "",
  "longitude": "",
  "csq": 24,
  "version": "9024-0-10",
  "linkNum": 4,
  "flow": 620
}
```

### 1.29 获取/配置串口配置信息

字段	说明
type	消息类型, 读/写: getUart/setUart, 返回: getUartRes
id	设备编号 例如 W020923P00004CFA7
baud	波特率, 支持 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800
parity	校验位, 支持 NONE, EVEN, ODD
stop	停止位, 支持 1, 2
DataBits	数字位, 5, 6, 7, 8 几个选项

读取参数请求:

```
{"type": "getUart"}
```

设备返回:

```
{
```



```

    "type": "getUartRes",
    "id": "2020060200000023c061",
    "baud": 115200,
    "DataBits": 8,
    "parity": "NONE",
    "stop": 1
}

```

配置参数请求:

```

{"type":"setUart","baud":115200,"DataBits":8,"parity":"NONE","stop":1}

```

设备返回:

```

{
    "type": "getUartRes",
    "id": "2020060200000023c061",
    "baud": 115200,
    "DataBits": 8,
    "parity": "NONE",
    "stop": 1
}

```

### 1.30 获取/配置 Socket 信息

字段	说明
type	消息类型, 读/写: getSocket/setSocket, 返回: getSocketRes
id	设备编号, 例如 W020923P00004CFA7
socketID	socketID
sw	链路开关, 0 关闭, 1 开启
proc	工作模式, 0: 网络透传, 1: http, 2: mqtt
socket_type	sock 协议, 0: UDPC,1: TCPC, 2: UDPS,3: TCPS
addr	服务器地址, 64 字节
desPort	端口号
short_link	短链接时间, 0: 表示长连接
status	链路状态, 0: 未连接, 1: 连接
keepAlive (心跳数据)	
keepAlive->sw	心跳使能, 0: 关闭, 1: 开启

keepAlive->time	心跳时间
keepAlive->dataType	心跳数据类型, 0: IMEI, 1: ICCID, 2: SN, 3: 自定义
keepAlive->dataFormat	数据类型 0/1 ascii/hex
keepAlive->data	自定义数据, 64 字节
Regist (注册数据)	
regist->sw	注册包使能, 0: 关闭, 1: 开启
regist->dataType	注册包数据类型, 0: IMEI, 1: ICCID, 2: SN 3: 自定义
regist->way	发送方式, 0: 每次入网发送 1 次, 1: 每次入网发送, 每包数据前面加上注册包
regist->dataFormat	数据类型 0/1 ascii/hex
regist->data	自定义数据, 64 字节
Mqtt (MQTT 参数)	
mqtt->sw	使能, 0: 关闭 1: 打开
mqtt->mode	mqtt 模式: 0: 阿里云, 1: oneNET, 2: 百度云, 3: 3.1 版本标准 mqtt
mqtt->deviceName	mqtt 设备名称, 64 字节
mqtt->productKey	产品密钥, 64 字节
mqtt->deviceSecret	设备密钥, 64 字节
mqtt->clientId	clientId, 64 字节
mqtt->username	用户名, 64 字节
mqtt->password	密码, 64 字节
mqtt->subTopic	订阅主题
mqtt->subSw	订阅使能, 0: 不订阅 1, 订阅
mqtt->subQos	订阅主题消息等级, 0, 1, 2

mqtt->pubTopic	发布主题
mqtt->pubSw	发布使能, 0: 不发布 1, 发布
mqtt->pubQos	发布主题消息等级, 0, 1

读取参数请求:

```
{"type": "getSocket","socketID": 0}
```

设备返回:

```
{
  "type": "getSocketRes",
  "id": "2020060200000023c061",
  "socketID": 0,
  "sw": 1,
  "proc": 0,
  "socket_type": 1,
  "short_link": 0,
  "addr": "112.54.89.224",
  "desPort":4000,
  "status": 2,
  "keepAlive": {
    "sw": 0,
    "dataType": 4,
    "dataFormat": 0,
    "data": "{\"keepalive\": \"9600\"}",
    "time": 60
  },
  "regist": {
    "sw": 0,
    "dataType": 3,
    "dataFormat": 0,
    "data": "{\"register\": \"EC04G-02\"}",
    "way": 0
  },
  "mqtt": {
    "sw": 1,
    "mode": 0,
    "deviceName": "",
    "productKey": "",
    "deviceSecret": "",
    "subSw": 1,
    "subTopic": "",
    "subQos": 0,
    "pubSw": 1,
  }
}
```

```

        "pubTopic": "",
        "pubQos": 0
    }
}
配置参数请求:
{
    "type": "setSocket",
    "id": "2020060200000049cb8d",
    "socketID": 1,
    "sw": 1,
    "proc": 2,
    "socket_type": 0,
    "short_link": 0,
    "addr": "a1GlhuTU1yN.iot-as-mqtt.cn-shanghai.aliyuncs.com",
    "desPort":1883,
    "status": 2,
    "keepAlive": {
        "sw": 0,
        "dataType": 0,
        "dataFormat": 0,
        "data": "KeepAlive",
        "time": 60
    },
    "regist": {
        "sw": 0,
        "dataType": 0,
        "dataFormat": 0,
        "data": "Regist",
        "way": 0
    },
    "mqtt": {
        "sw": 1,
        "mode": 0,
        "deviceName": "DEV01",
        "productKey": "a1GlhuTU1yN",
        "deviceSecret": "",
        "subSw": 1,
        "subTopic": "/a1GlhuTU1yN/DEV01/user/1234",
        "subQos":0,
        "pubSw": 1,
        "pubTopic": "/a1GlhuTU1yN/DEV01/user/1234",
        "pubQos": 0
    }
}

```

设备返回:

```
{
  "type": "getSocketRes",
  "id": "2020060200000023c061",
  "socketID": 1,
  "sw": 1,
  "proc": 2,
  "socket_type": 0,
  "short_link": 0,
  "addr": "a1GIhuTU1yN.iot-as-mqtt.cn-shanghai.aliyuncs.com",
  "desPort":1883,
  "status": 2,
  "keepAlive": {
    "sw": 1,
    "dataType": 0,
    "dataFormat": 0,
    "data": "KeepAlive",
    "time": 60
  },
  "regist": {
    "sw": 0,
    "dataType": 0,
    "dataFormat": 0,
    "data": "Regist",
    "way": 0
  },
  "mqtt": {
    "sw": 0,
    "mode": 0,
    "deviceName": "DEV01",
    "productKey": "a1GIhuTU1yN",
    "deviceSecret": "",
    "subSw": 1,
    "subTopic": "/a1GIhuTU1yN/DEV01/user/1234",
    "subQos":0,
    "pubSw": 1,
    "pubTopic": "/a1GIhuTU1yN/DEV01/user/1234",
    "pubQos": 0
  }
}
```

### 1.31 设备重启

字段	说明
type	消息类型 reload
id	设备编号 例如 W020923P00004CFA7

发送请求:

```
{
  "type": "reload"
  "id": "88888888000000054b9e"
}
```

### 1.32 固件升级

字段	说明
type	消息类型, upgrade
id	设备编号 例如 W020923P00004CFA7
url	固件下载地址
username	用户名
password	密码
taskId	任务 ID

```
{
  "type": "settingVerison",
  "id": "W910308000005CBA8",
  "version": " http://cloudadmin.ebyte.com/uploads/version/9024-0-10.bin "
}
```

http://cloudadmin.ebyte.com/uploads/version/9024-0-10.bin,NULL,NULL

## 第 3 章 短信指令说明

短信指令支持 AT 的全部，当需要使用以下格式发送，绿色部分可替换为其他 AT 指令，同时支持使用串口以及网络发送该指令格式。



手机发送:

`{"type": "at", "message": "AT+VER"}`

【注】带转义字符需按照以下格式发送: `{"type": "at", "message": "AT+SOCK=1, \"base\?"}`

收到

+OK=9024-0-10(返回数据参考 AT 指令返回)

最终解释权归成都亿佰特电子科技有限公司所有。

## 修订历史

版本	修订日期	修订说明	维护人
1.0	2021-10-19	初始版本	LC

## 关于我们



销售热线：4000-330-990

公司电话：028-61399028

技术支持：[support@cdebyte.com](mailto:support@cdebyte.com)

官方网站：[www.ebyte.com](http://www.ebyte.com)

公司地址：四川省成都市高新西区西区大道 199 号 B5 栋


**成都亿佰特电子科技有限公司**  
**EBYTE Chengdu Ebyte Electronic Technology Co.,Ltd.**