



【AT 指令集】NA611-S&SA

第 1 章 AT 指令

- 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（除进入指令外）。
- 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，不是字符串。

错误 AT 返回 \r\n+ERR=-1\r\n

1.1 进入 AT 指令

发送+++不用换行，收到\r\n+OK\r\n后进入配置模式。

【举例】

先发送+++

收到\r\n+OK\r\n

1.2 退出 AT 指令

指令	AT+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	AT+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	AT+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	AT+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=FW-9076-x-xx\r\n

1.6 查询 MAC 地址指令

指令	AT+MAC
功能	查询设备 MAC 地址
发送	AT+MAC<CR><LF>
返回	<CR><LF>+OK=<MAC_ADDR><CR><LF>

【举例】

发送：AT+MAC\r\n

收到：\r\n+OK=XXXXXXXXXXXX\r\n

1.7 查询本地 IP 地址指令

指令	AT+LOCAL_IP
功能	查询设备本地地址
发送	AT+LOCAL_IP<CR><LF>
返回	<CR><LF>+OK=<local_ip><CR><LF>

【举例】

发送: AT+LOCAL_IP\r\n
收到: \r\n +OK=192.168.4.1\r\n

1.8 查询/配置网络 AT 头

指令	AT+NET_AT_HEAD
功能	查询/配置网络 AT 头
发送（查询）	AT+NET_AT_HEAD<CR><LF>
返回（查询）	<CR><LF>+OK=<NET_AT_HEAD><CR><LF>
发送（配置）	AT+NET_AT_HEAD=<NET_AT_HEAD><CR><LF>
返回（配置）	<CR><LF>+OK<CR><LF>
备注	NET_AT_HEAD :网络 AT 头，网络收发的数据如果是带有这个头的，将会被视为 AT 指令，最大长度 24 字符，最小 3 字符

【举例】

查询：
发送: AT+NET_AT_HEAD\r\n
收到: \r\n+OK=NETEbyte\r\n
设置：
最大长度不能超过 24 位
发送: AT+NET_AT_HEAD = NETEbyte\r\n
收到: \r\n+OK\r\n

1.9 查询/配置空闲重启

指令	AT+FREE_RESTART_TIME
功能	查询/配置空闲重启参数
发送（查询）	AT+FREE_RESTART_TIME<CR><LF>
返回（查询）	<CR><LF>+OK=<FREE_RESTART_SW, FREE_RESTART_TIME><CR><LF>
发送（配置）	AT+FREE_RESTART_TIME=<FREE_RESTART_SW, FREE_RESTART_TIME><CR><LF>
返回（配置）	<CR><LF>+OK<CR><LF>
备注	FREE_RESTART_SW: 空闲重启开关，如果为 0，代表空闲重启关闭，如果为 1 代表空闲重启打开 FREE_RESTART_TIME: 空闲重启时间，单位分钟，当设备长时间没有数据交互时会触发空闲重启设备。0 代表关闭空闲重启，最短 1 分钟，最长可设置 60*24*7 分钟（一周）

【举例】

查询：
发送: AT+FREE_RESTART_TIME\r\n
收到: \r\n+OK=1,120\r\n 代表空闲重启打开，时间为 120 分钟
设置：
发送: AT+FREE_RESTART_TIME =1,180 \r\n 代表设置空闲重启打开，时间设置为 180 分

钟

收到: \r\n+OK\r\n

1.10 查询/配置串口参数

指令	AT+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, 上位机上有 7 位但是暂不支持 Uartstop, 停止位: 1、2 Parity, 校验位: NONE、ODD、EVEN Uartflow, 流控: 0 为关闭, 保留, 暂时无效, 用户使用填零即可。 packLen, packTime: 分别是打包长度 (64-1536) 和打包时间 (15-150), 没有特殊使用无需修改

【举例】

查询:

发送: AT+UART\r\n

收到: \r\n+OK=115200,8,1,NONE,0,1024,30\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 30ms。

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

收到: \r\n+OK\r\n

1.11 查询网络状态

指令	AT+NET
功能	查询 WIFI 网络状态
发送	AT+NET<CR><LF>
返回	<CR><LF>+OK=<status><CR><LF>
备注	STA: Status, 1 (网络连接成功), 0 (网络未连接) AP: Status 为 1

【举例】

发送: AT+NET\r\n

收到: \r\n +OK=1\r\n

1.12 查询/配置 WIFI 参数

指令	AT+WIFI
功能	查询/配置 WIFI 参数
发送（查询）	AT+WIFI<CR><LF>
返回（查询）	<CR><LF>+OK<wifi_mode, ssid, pass><CR><LF>
发送（配置）	AT+WIFI=<wifi_mode, ssid, pass><CR><LF>
返回（配置）	<CR><LF>+OK<CR><LF>
备注	<p>wifi_mode, wifi 模式: 0:AP 模式(热点模式), 1: STA 模式(终端模式, 连接热点)</p> <p>ssid, wifi 的账号名: 如果处于 AP 模式就是热点的账户名, 如果处于 STA 模式就是 STA 的账户名, 避免使用中文, 用英文字符以及数字, 最大长度为 32</p> <p>pass, wifi 的密码: 如果处于 AP 模式那就是热点本身的密码, 如果处于 STA 模式就是目标路由的密码, 避免使用中文, 用英文字符以及数字, 最大长度为 16</p> <p>注: 如果想要设置为 STA 连接无密码 WiFi(开放式 WiFi), 需要将 WiFi 的 pass 设置为 x(小写 x)即可, 如果设置为开放式 AP 热点模式, 也是将 pass 设置为 x 即可, 不可以填写空</p>

【举例】

查询:

发送: AT+WIFI\r\n

收到: \r\n+OK=0,EBYTE_TEST,12345678\r\n

设置:

配置 WIFI 模式为 AP 模式, 热点名称是 EBYTE_TEST, 密码是 12345678

发送: AT+WIFI=0,EBYTE_TEST,12345678\r\n

收到: \r\n+OK\r\n

1.13 查询/配置 WIFI IP 设置参数

指令	AT+WIFI_IP
功能	查询/配置 WIFI 设置参数
发送（查询）	AT+WIFI_IP<CR><LF>
返回（查询）	<CR><LF>+OK=<ip_mode, static_ip, net_mask, get_way_ip, dns_addr><CR><LF>
发送（配置）	AT+WIFI_IP=<ip_mode, static_ip, net_mask, gateway_ip, dns_addr><CR><LF>
返回（配置）	<CR><LF>+OK<CR><LF>
备注	<p>ip_mode: ip 的模式 0: 静态 ip 模式 1: DHCP 模式</p> <p>static_ip: 当处于静态 ip 模式下的 ip</p> <p>net_mask: 子网掩码</p>

gateway_ip: 路由地址 dns_addr: dns 服务器地址

【举例】

查询:

发送: AT+WIFI_IP\r\n

收到: \r\n+OK=1,192.168.10.155,255.255.255.0,192.168.10.1,114.114.114.114\r\n

设置:

配置 IP 为静态模式, 热点名称是 EBYTE_TEST, 密码是 12345678

发送: AT+WIFI_IP=0,192.168.10.155,255.255.255.0,192.168.10.1,114.114.114.114\r\n

收到: \r\n+OK\r\n

1.14 查询/配置协议分发设置参数

指令	AT+POTOCOL
功能	查询/配置协议分发设置参数
发送（查询）	AT+POTOCOL<CR><LF>
返回（查询）	<CR><LF>+OK<SW><CR><LF>
发送（配置）	AT+POTOCOL=<SW><CR><LF>
返回（配置）	<CR><LF>+OK<CR><LF>
备注	SW: 协议分发的开关 0: 协议分发关闭 1: 协议分发打开

【举例】

查询:

发送: AT+POTOCOL\r\n

收到: \r\n+OK=0\r\n

设置:

禁用多协议转换:

发送: AT+POTOCOL=0\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 "base", 关键字 Sw, 链路开关: 0 (关闭)、1 (开启) Workmode, 工作模式: 0 (普通模式), 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, 最小 4 秒(如果设置 1-4 直接按 4 秒运行) 默认 0, 只在 TCP Client 模式下有效 DesIP, 目标 IP, 最大长度 72, 支持域名解析, 如果是 server 模式, 填写本机 IP 地址 (如果不知道 IP 地址就填写 192.168.1.1, 如果设备 ip 地址不是这个, 会在读取参数的时候更新, 如果已经知道 IP 地址就直接填写本机 IP 地址) DesPort, 目标端口, 0~65535, 如果是 server 模式下, 这个端口就是 server 开设的端口
----	--

【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,time,defaultData><CR><LF>
发送 (设置)	AT+SOCK=<SockID,"keep/alive",sw , keepAliveType,infoType,time,defaultData ><CR><LF>
返回 (设置)	<CR><LF>+OK<CR><LF>
备注	SockID, socket 号: 0,1,2,3 "keepalive", 关键字 sw, 开关: 0 (关闭), 1 (开启), 注意只在 TCP Client, UDP Client 起作用, 其他模式下无心跳包(MQTT 心跳包固定 60s 一次保活) keepAliveType: 心跳包模式 0 (MAC, 请设置非 hex 模式) 1 (自定义), 默认自定义 infoType, 数据格式: 0 (ASCII)、1 (HEX) Time, 心跳时间: 最大可配时间 65535s defaultData, 自定义心跳包内容: 最大配置 72 字节数

【举例】

查询:

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

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

设置:

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

收到: \r\n+OK\r\n

1.19 查询/设置串口心跳包

指令	AT+UART_KEEPAIVE
功能	查询网络链接状态
发送 (查询)	AT+UART_KEEPAIVE
返回 (查询)	<CR><LF>+OK=<sw,msg_mode,time,data><CR><LF>
发送 (设置)	AT+UART_KEEPAIVE=< sw,msg_mode,time,data ><CR><LF>

返回（设置）	<CR><LF>+OK<CR><LF>
备注	sw: 串口心跳包开关 msg_mode: 消息类型 (0: ascii 类型, 1: hex 类型) time: 心跳包时间间隔, 0-65535(填 0 直接默认一秒一次) data: 心跳包内容, 最大 48 字节

【举例】

查询:
发送: AT+UART_KEEPAKIVE\r\n
收到: \r\n+OK= \r\n
设置:
禁用串口心跳:
发送: AT+UART_KEEPAKIVE=0,0,0,test\r\n
收到: \r\n+OK\r\n

1.20 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	SockID, socket 号: 0,1,2,3 "regist", 关键字 Sw, 开关: 0 (关闭), 1 (开启), 注意只在 TCP Client, UDP Client 模式下本参数起作用 registtype: 注册包模式: 0 (MAC, 设置 MAC 形式后无视 infoType, 只能是 HEX 格式, 因为 MAC 包头一般给服务器识别用, 都是 16 进制) 1 (自定义) 默认自定义 infoType, 数据格式: 0 (ASCII)、1 (HEX) defaultData, 自定义心跳包内容: 最大配置 72 字节数 Way: 0 (连接发送一次)、1 (每包数据前)

【举例】

查询:
发送: AT+SOCK=0,"regist"?\r\n
收到: \r\n+OK=0,"regist",0,1,0,Regist,0\r\n
设置:
发送: AT+SOCK=0,"regist",0,1,0,Regist,0\r\n
收到: \r\n+OK\r\n

1.21 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: 1、2、3、4</p> <p>"mqtt/connet", 关键字</p> <p>Sw, 开关: 0 (关闭), 1 (开启)</p> <p>Mode, 云平台: 0 (阿里云), 1 (ONENET)、2 (百度云) 3 (3.1.1 标准 MQTT)</p> <p>productKey: 阿里产品密钥、百度设备密钥、ONENET 设备 ID, 最大可配置长度为 64 字节</p> <p>deviceName: 阿里设备名称、百度用户名、ONENET 产品 ID, 最大可配置长度为 64 字节</p> <p>deviceSecret: 阿里设备密钥、百度密码、ONENET 鉴权信息, 最大可配置长度为 96 字节</p>

1.22 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	<p>SockID, socket 号: 1、2、3、4</p> <p>"mqtt/sub", 关键字</p> <p>Sw, 开关: 0 (关闭), 1 (开启)</p> <p>TopicName, 订阅主题名, 最大长度可配置 128 字节</p> <p>Qos, 服务等级, 0、1、2</p>

1.23 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：1、2、3、4 "mqtt/pub", 关键字 Sw, 开关：0（关闭），1（开启） TopicName, 订阅主题名，最大可配置长度 128 字节 Qos, 服务等级，0、1、2

1.24 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,data_head><CR><LF>
发送（设置）	AT+SOCK=<sockID,"http",method,url,head,data_head ><CR><LF>
返回（设置）	<CR><LF>+OK<CR><LF>
备注	SockID, socket 号：1、2、3、4 "http", 关键字 method, 方法：0（GET），1（POST） url, 资源路径，最大可配置 128Bit head, 包头，最大可配置 128Bit data_head, 接受到的信息发送出来的时候是否带有包头，0，不带包头，1 带有包头。默认值为 1

1.25 查询/设置 Modbus 工作模式以及指令超时时间

指令	AT+MOD_WORK_MODE
功能	查询/设置 Modbus 工作模式以及指令超时时间
发送（查询）	AT+MOD_WORK_MODE=?<CR><LF>
返回（查询）	<CR><LF>+OK=<mode,time><CR><LF>
发送（设置）	AT+MOD_WORK_MODE=<mode,time><CR><LF>
返回（设置）	<CR><LF>+OK<CR><LF>
备注	Mdoe:modbus 工作模式(0(默认)：关闭状态，1：简单协议转换，2：多主机模式，3：存储型网关，4：配置型网关，5：自动上传) Time: 指令超时时间（200-65535，默认 1500，单位 ms）

1.26 查询/设置 Modbus TCP 与 Modbus RTU 的协议转换开关

指令	AT+MOD_PTCL
功能	查询/设置 Modbus TCP 与 Modbus RTU 的协议转换开关
发送（查询）	AT+MOD_PTCL=?<CR><LF>
返回（查询）	<CR><LF>+OK=<sw><CR><LF>
发送（设置）	AT+MOD_PTCL=<sw><CR><LF>
返回（设置）	<CR><LF>+OK<CR><LF>
备注	Sw: 开关 0: 关闭 1: 打开

1.27 查询/设置 Modbus 配置型网关的预存指令

指令	AT+MOD_CMD_EDIT
功能	查询/设置 Modbus 配置型网关的预存指令
发送（查询）	AT+MOD_CMD_EDIT=?<CR><LF>
返回（查询）	<CR><LF>+OK=<指令列表><CR><LF>
发送（设置）	AT+MOD_CMD_EDIT="method", 010300000003 ><CR><LF>
返回（设置）	<CR><LF>+OK<CR><LF>
备注	method: 指令操作方法, ADD/DLE 添加和删除 010300000003: 想要操作的指令, 按照标准 6 位 modbus 指令, 所以上命令部分长度固定为 12, 不按规定填写不予存储

1.28 查询/设置 Modbus 指令存储时间与指令查询间隔

指令	AT+MOD_GETWAY_TIME
功能	查询/设置 Modbus 工作模式以及指令超时时间
发送（查询）	AT+MOD_GETWAY_TIME =?<CR><LF>
返回（查询）	<CR><LF>+OK=<time, interval_time><CR><LF>
发送（设置）	AT+MOD_GETWAY_TIME =<time, interval_time><CR><LF>
返回（设置）	<CR><LF>+OK<CR><LF>
备注	time: 指令存储时间 (1-255, 默认 5, 单位 s) interval_time: 指令间隔时间 (10-65535, 默认 100ms, 单位 ms)

修订历史

版本	修订日期	修订说明	维护人
1.0	2021-04-06	初始版本	ken

关于我们



销售热线：4000-330-990

公司电话：028-61399028

技术支持：support@cdebyte.com

官方网站：

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.