

1. 文档介绍

1.1 文档范围

本手册详细介绍了UC8088 WIOTA AP模块提供的AT指令集。

1.2 命令语法

1.2.1 命令格式

本手册中所有命令行必须以“AT”或“at”作为开头，以回车作为结尾。响应通常紧随命令之后，且通常以“<回车><换行><响应内容><回车><换行>”的形式出现。在命令介绍时，“<回车><换行>”通常被省略了。

1.2.2 命令类型

通常命令可以有如下表所示的四种类型中的一种或多种形式。

类型	格式	说明
测试命令	AT+<cmd>=?	用于查询设置命令或内部程序设置的参数及其取值范围
查询命令	AT+<cmd>?	用于返回参数的当前值
设置命令	AT+<cmd>=<...>	用于设置用户自定义的参数值
执行命令	AT+<cmd>	用于读取只读参数或不需要额外参数的情况

1.2.3 参数类型

命令参数虽然多种多样，但是都可以简单地归结为整数类型和字符串类型（包括不带双引号的字符串和带双引号的字符串）这两种基本的类型，如下表所示。

类型	示例
整数类型	123
字符串类型	abc
	"hellow ,world"

1.2.4 注意事项

- AT串口输入时不支持回删键(backspace)功能。
- AT指令不区分大小写。

2. 基础 AT指令详细说明

2.1 AT

AT测试指令。

Command	Possible response(s)
AT	OK
	ERROR

实例说明如下表：

输入AT指令	输出结果	输出结果说明
AT	OK	AT指令功能可正常使用
	ERROR	AT指令功能无法正常使用

2.2 AT+RST 重启

系统重启，会同时重启ap8288和ap8088。

Command	Possible response(s)
+RST	OK
	ERROR

实例说明如下表：

输入AT指令	输出结果	输出结果说明
AT+RST	OK +CHOOSEMODEM +SYSTEM:START	系统重启成功
	ERROR	指令执行失败

2.3 ATE 回显

AT指令回显功能。

Command	Possible response(s)
ATE<value>	OK
	ERROR

- <value>：默认AT回显关闭。
0：关闭回显。
1：打开回显。

实例说明如下表：

输入AT指令	输出结果	输出结果说明
ATE0	OK	关闭回显成功
	ERROR	指令执行失败
ATE1	OK	开启回显成功
	ERROR	指令执行失败

2.4 AT&L 查询AT指令列表

查询所有支持的AT指令列表。

Command	Possible response(s)
AT&L	OK
	ERROR

实例说明如下表：

输入AT指令	输出结果	输出结果说明
AT&L	Commands list : AT&L AT+RST OK	输出所有的AT指令
	ERROR	指令执行失败

2.5 AT+UART AT串口配置

AT默认使用UART0，故该命令可配置UART0的属性，一般不建议更改。

Command	Possible response(s)
AT+UART=<baud_rate>,<data_bits>,<stop_bits>,<parity>,<flow_control>	OK
	ERROR
AT+UART?	+UART=<baud_rate>,<data_bits>,<stop_bits>,<parity>,<flow_control>
	ERROR

- <baudrate>：波特率，默认UART0的波特率配置为115200，最大支持的波特率921600。
- <databits>：有效数据长度。
- <stopbits>：停止位。
- <parity>：奇偶检验。
- <flow_control>：流控，默认值为1，不支持配置。

实例说明如下表：

输入AT指令	输出结果	输出结果说明
AT+UART=115200,8,0,0,1	OK	设置UART0属性成功
	ERROR	指令执行失败
AT+UART?	+UART=115200,8,0,0,1 OK	输出UART0的配置信息
	ERROR	指令执行失败

2.6 AT+YMODEM 进入Ymodem刷机模式

进入Ymodem串口刷机模式。

Command	Possible response(s)
+YMODEM=<type>	OK
	ERROR

- <type>：刷新的设备。
8088：刷新UC8088的FLASH；
8288：刷新UC8288（基带）的FLASH。

2.7 系统上报

Command	Mean
+SYSTEM:START	启动RT-THREAD系统

2.8 AT+REFLASH 基站侧基带恢复

恢复成UC8088中保存的镜像，重刷一次UC8288（基带）的FLASH。通过烧写器烧写UC8288（基带）后，需要执行一次该指令，确保UC8088与UC8288版本一致。

Command	Possible response(s)
+REFLASH	OK
	ERROR

3. WITOA AT命令详细说明

3.1 AT+WIOTAINIT 初始化

初始化WIoTat协议栈。

Command	Possible response(s)
+WIOTAINIT	OK
	ERROR

实例说明如下表：

输入AT指令	输出结果	输出结果说明
AT+WIOTAINIT	OK	WIoTa协议栈初始化成功
	ERROR	指令执行失败

3.2 AT+WIOTAFREQ 锁频

设置频点，终端和AP需要设置相同频点才能同步。

Command	Possible response(s)
+WIOTAFREQ=<freq_idx>	OK
	ERROR
+WIOTAFREQ?	+WIOTAFREQ=<freq_idx>
	ERROR

- <freq_idx>: 频点idx，范围0~200

实例说明如下表：

输入AT指令	输出结果	输出结果说明
AT+WIOTAFREQ=145	OK	设置频点为145
	ERROR	指令执行失败
AT+WIOTAFREQ?	+WIOTAFREQ=145	查询到配置的频点为145
	ERROR	指令执行失败

3.3 AT+WIOTAACTIVETIME 设置连接态保持时间

设置AP连接态保持时间，必须和终端保持一致，否则会导致无法正常进行上下行业务。

Command	Possible response(s)
+WIOTAACTIVETIME = <active_time>	OK
	ERROR
+WIOTAACTIVETIME ?	+WIOTAACTIVETIME =<active_time>
	ERROR

- <active_time>: 该值不可设置太大，否则可能导致其他终端无法接入，范围0~0xffffffff，单位：秒
该值有默认值在128配置下为2秒，256配置下为3秒。512配置下为4秒，1024配置下为8秒。非特殊需求不建议更改连接态保持时间。

实例说明如下表：

输入AT指令	输出结果	输出结果说明
AT+WIOTAACTIVETIME=5	OK	设置连接态保持时间为5秒成功
	ERROR	指令执行失败
AT+WIOTAACTIVETIME?	+WIOTAACTIVETIME=5	查询到配置的连接态保持时间为5秒
	ERROR	指令执行失败

3.4 AT+WIOTACONFIG 系统配置

设置系统配置。

Command	Possible response(s)
+WIOTACONFIG=<id_len>,<symbol_len>,<dlul_ratio>,<bt_value>,<group_num>,<ap_max_pow>,<specturm_idx>,<system_id>,<subsystem_id>	OK
	ERROR
+WIOTACONFIG?	+WIOTASYSTEMCONFIG=<id_len>,<symbol_len>,<dlul_ratio>,<bt_value>,<group_num>,<ap_max_pow>,<specturm_idx>,<system_id>,<subsystem_id>
	ERROR

- <id_len>: user id长度，取值0,1,2,3代表2,4,6,8字节。默认四字节，IOTE该变量需要与AP保持一致，现在只支持设置为1，即四字节。
- <symbol_len>: 帧配置，取值0,1,2,3代表128,256,512,1024。
- <dlul_ratio>: 帧配置，该值代表一帧里面上下行的比例，取值0,1代表1:1和1:2。
- <bt_value>: 该值和调制信号的滤波器带宽对应，BT越大，信号带宽越大，取值0,1代表BT配置为1.2和BT配置为0.3，bt_value为0时，代表使用的是低阶mcs组，即低码率传输组。bt_value为1时，代表使用的是高mcs组，即高码率传输组。
- <group_num>: 帧配置，取值0,1,2,3代表一帧里包含1,2,4,8个上行group数量。
- <ap_max_pow>: ap射频最大功率，默认22dbm，由于该值可能为负数，但at暂不支持负数解析，tag0.09版本之后，实际需要设置的功率加20则为输入值，例如想要设置功率-10dbm，则at+wiotapow=0,10，想要设置功率20dbm，则 at+wiotapow=0,40。
- <specturm_idx>: 频谱序列号，默认为3，即470-510M，现在仅支持设置为3。
- <system_id>: 系统id，每个id是0-0xFFFFFFFF，16进制格式输入，不需要0x。此值为预留值，必须设置，但是不起作用。
- <subsystem_id>: 子系统id，每个id是0-0xFFFFFFFF，16进制格式输入，不需要0x。（子系统的识别码，终端IOTE如果要连接该子系统（AP），需要将config配置里的子系统ID参数配置成该ID）

实例说明如下表：

输入AT指令	输出结果	输出结果说明
AT+WIOTACONFIG=1,1,0,1,0,47,3,11223344,21456981	OK	配置系统信息成功
	ERROR	指令执行失败
AT+WIOTACONFIG?	+WIOTACONFIG=1,1,0,1,0,47,3,11223344,21456981	输出当前系统配置信息
	ERROR	指令执行失败

3.5 AT+WIOTARUN 启动/关闭WIoTa协议栈

启动WIoTa系统。

启动WIoTa后，收到数据会主动上报串口，数据最长为1024字节。

格式是：+WIOTARECV:<type>,<user_id>,<data_len>,<data>。

Command	Possible response(s)
+WIOTARUN=<state>	OK
	ERROR

- <state>:
0：退出协议栈。
1：启动协议栈。

实例说明如下表：

输入AT指令	输出结果	输出结果说明
AT+WIOTARUN=1	OK	WIoTa协议栈启动成功
	ERROR	指令执行失败
AT+WIOTARUN=0	OK	WIoTa协议栈关闭成功
	ERROR	指令执行失败

3.6 +WIOTARECV WIoTa数据上报

接受到上行数据主动上报（不需要AT命令设置，系统直接上报接收到的数据到串口）。

Command	Possible response(s)
无	+WIOTARECV=<type>,<user_id>,<data_len>,<recv_data> OK

- <type>:
0：普通上行数据。
1：连续数据包数据。
- <user_id>：上报数据的终端id。
- <data_len>：上报数据的长度。
- <recv_data>：上报数据的内容。

实例说明如下表：

输入AT指令	输出结果	输出结果说明
无	+WIOTARECV=0,0x12345678,13,Hello WIoTa OK	接收到id为12345678的终端13个字节的普通上行数据“Hello WIoTa”

3.7 AT+WIOTABLACKLIST 添加或移除黑名单

添加一个user_id到黑名单或从黑名单中移除一个user_id。

Command	Possible response(s)
+WIOTABLACKLIST=<user_id>,<mode>	OK
	ERROR
+WIOTABLACKLIST?	+WIOTABLACKLIST=<user_id>,<blacklist_num>
	ERROR

- <userid>: 要添加或移除的user_id (0~0xffffffff)
- <mode>:
0: 添加黑名单。
1: 移除黑名单。

实例说明如下表：

输入AT指令	输出结果	输出结果说明
AT+WIOTABLACKLIST=12345678,0	OK	添加id为0x12345678的终端到黑名单成功
	ERROR	指令执行失败
AT+WIOTABLACKLIST=12345678,1	OK	从黑名单中移除id为0x12345678的终端成功
	ERROR	指令执行失败
AT+WIOTABLACKLIST?	+WIOTABLACKLIST=0x12345678,1 OK	输出查询结果，有一个id为0x12345678的终端在黑名单中
	ERROR	指令执行失败

3.8 AT+WIOTAIOTEINFO 查询iote信息

查询当前时刻下在线的iote信息和离线的iote信息，iote_status为0时表示离线，为1时表示在线，con_num表示在线的iote数量，discon_num表示离线的iote数量。

Command	Possible response(s)
+WIOTAIOTEINFO?	+WIOTAIOTEINFO=<user_id>,<iote_status>,<con_num>,<discon_num>
	ERROR

- <userid>: 终端id。
- <iote_status>: 终端状态。
 - 0: 离线状态。
 - 1: 在线状态。
- <con_num>: 在线的终端个数。
- <discon_num>: 离线的终端个数。

实例说明如下表：

输入AT指令	输出结果	输出结果说明
AT+WIOTAIOTEINFO?	+WIOTAIOTEINFO=0x12345678,1,1,0 OK	查询到一个id为0x12345678的终端处于在线状态，目前共有1个在线终端，0个离线终端
	ERROR	指令执行失败

3.9 AT+WIOTABC 发送广播数据

发送广播数据，执行该命令后，需在10秒内在串口工具的发送区输入长度为len的数据，并点击发送，不然会获取字符失败。

Command	Possible response(s)
+WIOTABC=<len>,<mode>,<timeout>	OK
	ERROR

- <len>: 要发送的广播数据长度，注意该长度包含了回车字符的长度，范围0~1024byte
- <mode>:
 - 0: 普通广播数据。
 - 1: OTA广播数据。
- <timeout>: 发送数据超时时间，范围0~0xffffffff。

实例说明如下表：

输入AT指令	输出结果	输出结果说明
AT+WIOTABC=11,0,10000\r\nAP ready!	send bc suc OK	发送普通广播数据“AP ready!”成功
	ERROR	指令执行失败
AT+WIOTABC=11,1,10000\r\nOTA test data	send bc suc OK	发送OTA广播数据“OTA test data”成功
	ERROR	指令执行失败

3.10 AT+WIOTASEND 发送数据

AP发送下行数据，执行该命令后，需在10秒内在串口工具的发送区输入长度为len的数据，并点击发送，不然会获取字符失败。

Command	Possible response(s)
+WIOTASEND=<len>,<user_id>,<timeout>	OK
	ERROR

- <len>: 要发送的广播数据长度，注意该长度包含了回车字符的长度，范围0~310byte。
- <user_id>: 终端id，范围0~0xffffffff。
- <timeout>: 发送数据超时时间，范围0~0xffffffff。

实例说明如下表：

输入AT指令	输出结果	输出结果说明
AT+WIOTASEND=14,12345678,10000 \\r\\nHello WIoTa!	send pdu suc OK	发送数据“Hello WIoTa!” 到id为0x12345678的终端成功
	ERROR	指令执行失败

3.11 AT+WIOTASCANFREQ 扫频

AP端扫频命令，可扫一组频点和全扫，返回扫频结果，执行该命令后需要在窗口工具的发送区输入长度为dataLen（dataLen只能大于或等于输入的字符串长度，不能小于否则会获取字符串失败），个数为freqNum的字符串，并点击发送。

Command	Possible response(s)
+WIOTASCANFREQ=<timeout>,<data_len>,<freq_num>	+WIOTASCANFREQ=<freq_idx>,<rss_i>,<snr>,<is_synced>
	ERROR

- <timeout>: 扫频的总超时时间，为0默认为永久等待，范围0~0xffffffff。
- <data_len>: 发送字符串的总长度+\\r\\n，比如要扫描的频点为1,2,3,4,5这五个频点。
 - 1) 执行at命令AT+WIOTASCANFREQ=10000,11,5。
 - 2) 当出现>时十秒钟内在串口工具的发送区内输入字符串1,2,3,4,5。
 - 3) 点击发送。
 - 4) 等待扫频结果返回，结果会通过串口打印出来。
- <freq_num>: 频点个数，范围0~200，该参数为0时为全扫。

实例说明如下表：

输入AT指令	输出结果	输出结果说明
AT+WIOTASCANFREQ=60000,70,20 \\n5,15,25,35,45,55,65,75,85,95,105,115,125,135,145,155,165,175,185,195	+WIOTASCANFREQ: freq_idx=5,snr=0,rssi=-58, is_synced=0 freq_idx=195,snr=0,rssi=-40, is_synced=1 OK	扫描20个频点成功，并输出个频点的snr、rssi和is_synced信息
	ERROR	指令执行失败
AT+WIOTASCANFREQ=0,0,0	+WIOTASCANFREQ: freq_idx=0,snr=0,rssi=-58, is_synced=0 freq_idx=100,snr=0,rssi=-40, is_synced=1 freq_idx=200,snr=25,rssi=-35, is_synced=0 OK	全扫，扫描201个频点成功，并输出个频点的snr、rssi和is_synced信息
	ERROR	指令执行失败

3.12 AT+WIOTATEMP 读取ap8288芯片实时温度

读取ap8288芯片内部实时温度，无任务是两帧内完成结果上报，有任务会直接返回读取失败。

Command	Possible response(s)
+WIOTATEMP?	+WIOTATEMP=<temp> OK
	ERROR

- <temp>: 读取到的UC8288温度，单位：摄氏度。

实例说明如下表：

输入AT指令	输出结果	输出结果说明
AT+WIOTATEMP?	+WIOTATEMP=47 OK	读取到基带芯片温度为47摄氏度
	ERROR	指令执行失败

3.13 AT+WIOTAPOW 设置ap8288射频功率

设置ap8288射频功率，设置范围为：0 - 29dbm。

Command	Possible response(s)
+WIOTAPOW=<power>	OK
	ERROR
+WIOTAPOW?	+WIOTAPOW=<power> OK
	ERROR

- <power>: 最大功率 (0-29) dbm, tag0.09版本之后, 实际需要设置的功率加20则为输入值, 例如想要设置功率-10dbm, 则 at+wiotapow=10, 想要设置功率20dbm, 则 at+wiotapow=40。

实例说明如下表:

输入AT指令	输出结果	输出结果说明
AT+WIOTAPOW=47	OK	设置射频功率为27 (47-20) dbm成功
	ERROR	指令执行失败或设置射频功率失败
AT+WIOTAPOW?	+WIOTAPOW=27 OK	查询到射频功率为27dbm
	ERROR	指令执行失败

3.14 AT+WIOTAVERSION 查询当前软件版本

查询当前WIoTa库的版本号、git 信息、编译生成库的时间, 如果未开启协议栈只能查到ap8088的信息, ap8288的信息需要启动WIoTa协议栈 ([3.5 AT+WIOTARUN 启动WIoTa协议栈](#)) 后方可查询到。

Command	Possible response(s)
+WIOTAVERSION?	+WIOTAVERSION:<version_8088>,<version_8288> +GITINFO:<gitInfo_8088>,<gitInfo_8288> +TIME:<make_time_8288>,<make_time_8288> +CCEVERSION:<cce_version> OK
	ERROR

- version_8088: 当前UC8088的版本号。
- version_8288: 当前UC8288的版本号。
- gitInfo_8088: 当前UC8088的git信息。
- gitInfo_8288: 当前UC8288的git信息。
- make_time_8288: 当前UC8088的创建时间。
- make_time_8288: 当前UC8088的创建时间。
- cce_version: 当前CCE的版本号。

实例说明如下表:

输入AT指令	输出结果	输出结果说明
AT+WIOTAVERSION?	+WIOTAVERSION:v0.10_ap8088, v0.10_ap8288 +GITINFO:Fri Apr 15 14:16:23 2022,Fri Apr 15 10:00:12 2022 +TIME:Apr 18 2022 09:34:56, Apr 15 2022 14:37:36 +CCEVERSION:b7a380 OK	输出查询到的版本信息
	ERROR	指令执行失败

3.15 AT+WIOTAHOPPING 设置跳频

设置跳频频点或模式。

Command	Possible response(s)
+WIOTAHOPPING=<type>,<value>, <value1>	OK
	ERROR

- <type>:
0: 设置跳频频点。
1: 设置跳频模式。
- <value>: type为0时表示跳频频点（0-200）；type为1时表示在原频点工作的帧数（0-255）。
- <value1>: type为0时无意义设为0；type为1时表示在跳频频点工作的帧数（0-255）。

实例说明如下表：

输入AT指令	输出结果	输出结果说明
AT+WIOTAHOPPING=0,147,0	OK	设置跳频频点为147
	ERROR	指令执行失败
AT+WIOTAHOPPING=1,10,20	OK	表示在原频点工作10帧后 在跳频频点工作20帧，如此循环
	ERROR	指令执行失败

3.16 AT+WIOTAIOTENUM 设置连接态终端数量

设置同一个子帧上连接态终端的最大数量。

Command	Possible response(s)
+WIOTAIOTENUM=<max_num>	OK
	ERROR

- <max_num>: 最大数量，默认4个，最多8个

实例说明如下表：

输入AT指令	输出结果	输出结果说明
AT+WIOTAIOTENUM=5	OK	设置同一个子帧上连接态终端的最大数量为5个成功
	ERROR	指令执行失败

3.17 AT+WIOTABCMCS设置广播传输速率

设置广播的mcs。

Command	Possible response(s)
+WIOTABCMCS=<bc_mcs>	OK
	ERROR

- <bc_mcs>: mcs等级，在不同symbol length和不同MCS下，对应每帧传输的应用数据量（byte）会有差别，NA表示不支持，见下表：

symbol length	mcs0	mcs1	mcs2	mcs3	mcs4	mcs5	mcs6	mcs7
128	5	7	50	64	78	NA	NA	NA
256	5	13	20	50	106	155	190	NA
512	5	13	29	40	71	134	253	295
1024	5	13	29	61	106	218	449	617

实例说明如下表：

输入AT指令	输出结果	输出结果说明
AT+WIOTABCMCS=3	OK	设置广播的mcs为3成功
	ERROR	指令执行失败

3.18 AT+WIOTARATE设置传输模式和速率

三种模式：

第一种基本模式，是基本速率设置。

在第一种模式的基础上，在系统配置中dlul_ratio为1:2时，才能打开第二种模式，打开该模式能够提高该帧结构情况下两倍速率，默认第二种模式开启状态。

在第一种模式的基础上，打开第三种模式，能够提升（8*(1 << group_number)）倍单终端的速率，但是会影响网络中其他终端的上行，建议在大数据量快速传输需求时使用。

备注：group_number为系统配置中的参数。

Command	Possible response(s)
+WIOTARATE =<rate_mode>, <rate_value>	OK
	ERROR

- <rate_mode>:
 - 0: 当rate_mode为UC_RATE_NORMAL时, rate_value为UC_MCS_LEVEL。
 - 1: 当rate_mode为UC_RATE_MID时, rate_value为0或1, 表示关闭或打开, 必须和终端的状态保持一致。
 - 2: 当rate_mode为UC_RATE_HIGH时, rate_value为0, 表示关闭, rate_value为其他值, 表示当实际发送数据量 (byte) 大于等于该值时才会真正开启该模式, 常用建议设置rate_value为100, 可单独开启, 建议最好和终端状态保持一致。

实例说明如下表:

输入AT指令	输出结果	输出结果说明
AT+WIOTARATE=1,1	OK	当dlul_ratio为1:2时, 开启模式二
	ERROR	指令执行失败
AT+WIOTARATE=1,0	OK	关闭模式二
	ERROR	指令执行失败
AT+WIOTARATE=2,0	OK	关闭模式三
	ERROR	指令执行失败
AT+WIOTARATE=2,100	OK	当数据长度大于等于100byte时, 开启模式三
	ERROR	指令执行失败

3.19 AT+WIOTALOG设置WIoTa log

设置WIoTa AP log模式, 可根据情况开关log和切换log串口。

Command	Possible response(s)
+WIOTALOG=<mode>	OK
	ERROR

- <mode>:
 - 0: 关闭UART log。
 - 1: 开启UART log。
 - 2: UART log使用UART0, 如果从UART1切换到UART0, 会把UART0的波特率改为460800, 此时AT的波特率也是用该值。
 - 3: UART log使用UART1, 如果从UART0切换到UART1, 会把UART0的波特率恢复为115200。
 - 4: 关SPI log。
 - 5: 开SPI log。
- 注意: 默认状态下, UART log使用UART1, 波特率460800, AT使用UART0, 波特率115200, 在UART log的串口切换后 (AT会自动换到另外一个串口上), 需要特别注意串口工具使用的波特率是否对应, 如果AT的波特率不对时, 发送AT CMD会直接导致AT挂住!

实例说明如下表:

输入AT指令	输出结果	输出结果说明
AT+WIOTALOG=0	OK	关闭串口log
	ERROR	指令执行失败
AT+WIOTALOG=1	OK	开启串口log
	ERROR	指令执行失败
AT+WIOTALOG=2	OK	串口log使用UART0
	ERROR	指令执行失败
AT+WIOTALOG=3	OK	串口log使用UART1
	ERROR	指令执行失败
AT+WIOTALOG=4	OK	关闭SPI log
	ERROR	指令执行失败
AT+WIOTALOG=5	OK	开启SPI log
	ERROR	指令执行失败

3.20 AT+WIOTACRC设置wiota crc校验

设置WIoTaap crc校验长度限制和关闭crc，默认开启，校验长度为大于等于100。

Command	Possible response(s)
+WIOTACRC=<crc_limit>	OK
	ERROR
+WIOTACRC?	+WIOTACRC=<crc_limit> OK
	ERROR

- <crc_limit>:
0：关闭CRC校验。
大于0：CRC校验的数据长度。 如，crc_limit为100，则大于等于100字节的数据会在末尾自动加CRC16的校验。

实例说明如下表：

输入AT指令	输出结果	输出结果说明
AT+WIOTACRC=0	OK	关闭CRC16校验
	ERROR	指令执行失败
AT+WIOTACRC=100	OK	当发送数据大于等于100byte时，开启CRC16校验
	ERROR	指令执行失败
AT+WIOTACRC?	+WIOTACRC=100 OK	查询到CRC16的长度为100byte
	ERROR	指令执行失败

3.21 AT+WIOTASTATE查询ap上下行状态

查询AP端上下行状态，可查询单个终端的单个状态，单个终端的所有状态，全部终端的所有状态。

Command	Possible response(s)
+WIOTASTATE=<get_or_reset>,<user_id>,<state_type>	+WIOTASTATE=<user_id>,<ul_rcv_len>,<ul_rcv_suc>,<dl_send_len>,<dl_send_suc>,<dl_send_fail>
	+WIOTASTATE=<user_id>,<state>
	OK
	ERROR

- <get_or_reset>:
 - 0: 查询状态。
 - 1: 重置状态。
- <user_id>: 终端的user_id。
- <state_type>: 查询的状态类型。
 - 1: 查询上行接受成功的数据总长度。
 - 2: 查询上行成功接受的次数。
 - 3: 查询下行成功的总长度。
 - 4: 查询下行成功的次数。
 - 5: 查询下行失败的次数。

注：上行失败的AP端不做统计。

如： id不带0x。

 - 1) AT+WIOTASTATE=0,0,0为查询所有终端的所有状态。
 - 2) AT+WIOTASTATE=0,12345678,0为查询0x12345678的所有状态。
 - 3) AT+WIOTASTATE=0,12345678,1为查询0x12345678的上行接受成功的数据总长度。
 - 4) AT+WIOTASTATE=1,0,0为重置所有终端的所有状态。
 - 5) AT+WIOTASTATE=1,12345678,0为重置0x12345678的所有状态。
 - 6) AT+WIOTASTATE=1,12345678,1为重置0x12345678的上行接受成功的数据总长度。

实例说明如下表：

输入AT指令	输出结果	输出结果说明
AT+WIOTASTATE=0,0,0	+WIOTASTATE=0x12345678, 100,1,100,1,0 OK	查询到id为 0x12345678的终端, 成功收到长度为 100byte的上行数据, 成功接收到1次完整的 上行数据, 成功发送长 度为100byte的下行数 据, 成功发送1次完整 的下行数据, 下行发送 失败的次数为0
	ERROR	指令执行失败
AT+WIOTASTATE=0,12345678,0	OK	查询到id为 0x12345678的终端, 成功收到长度为 100byte的上行数据, 成功接收到1次完整的 上行数据, 成功发送长 度为100byte的下行数 据, 成功发送1次完整 的下行数据, 下行发送 失败的次数为0
	ERROR	指令执行失败
AT+WIOTASTATE=0,12345678,1	+WIOTASTATE=0x12345678,100 OK	查询到id为 0x12345678的终端, 成功接收到长度为 100byte的上行数据
	ERROR	指令执行失败
AT+WIOTASTATE=1,0,0	OK	重置所有终端的所有状 态成功
	ERROR	指令执行失败
AT+WIOTASTATE=1,12345678,0	OK	重置id为0x12345678 的终端的所有状态成功
	ERROR	指令执行失败
AT+WIOTASTATE=1,12345678,1	OK	重置id为0x12345678 的终端的上行接受数据 长度成功
	ERROR	指令执行失败

3.22 AT+WIOTAREADREG 查询某地址的内容

查询AP或基带某个地址或寄存器的内容。

Command	Possible response(s)
+WIOTAREADREG =<type>,<addr>	+WIOTAREADREG=<value> OK
	ERROR

- <type>:
 - 0: 查询ap8088内存地址内容。
 - 1: 查询ap8288内存地址内容。
- <addr>: 要查询的地址, 如3b0014, 注意地址不带0x。
- <value>: 从输入地址查询到的前四个字节的內容。

实例说明如下表:

输入AT指令	输出结果	输出结果说明
AT+WIOTAREADREG=0,2fe000	+WIOTAREADREG=0xffffffff OK	查询UC8088内存地址为0x2fe000的四个字节内容为0xffffffff
	ERROR	指令执行失败
AT+WIOTAREADREG=1,3b1014	+WIOTAREADREG=0x2b47e49c OK	查询UC8288内存地址为0x3b1014的四个字节内容为0x2b47e49c
	ERROR	指令执行失败

4. 正常启动流程

启动流程	输出结果	输出结果说明
AT+WIOTAINIT	OK	初始化WIoTa协议栈
AT+WIOTAFREQ=145	OK	设置AP频点为145
AT+WIOTARUN=1	OK	启动WIoTa协议栈

以上三个命令便可启动默认配置的WIoTa协议栈。