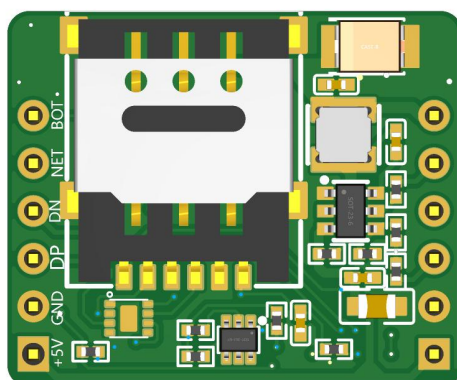


DTU 指令手册



修订历史

版 本	日 期	作 者	描 述
V0.001	2023-12-01	张 工	初 版

目 录

指令说明	4
1. 串口配置指令	4
2. APN 配置指令	5
3. 配置保存指令	6
4. 服务器配置指令	7
5. 基站定位信息查询指令	7
6. 网络时间查询指令	9
7. 信号质量查询指令	10
8. SIM 卡 ICCID 号查询指令	11
9. 模组 IMEI 查询指令	12
10. 版本号查询指令	13
11. GNSS 定位信息查询指令	14
12. MQTT 服务器参数配置	16
13. MQTT 会话设置	17
14. MQTT 用户信息设置	18
15. MQTT 遗嘱设置	19
16. 设置订阅主题	20
17. 设置发布主题	21
18. 查询 MQTT 服务器配置参数	22
19. 清除所有订阅主题	22
20. 查询网络连接状态	24
21. 配置应答模式	25
22. 获取 MQTT 账户参数	26
23. 获取发布主题参数	27
24. 获取订阅主题参数	28
25. 查询 MQTT 登录状态	29
26. 发布消息和接收消息	30
27. 注册包开关功能	31
28. 注册包内容	31
状态码列表	33

指令说明

1. 串口配置指令

指令名称: @config@baund@value###

指令描述:

字 段	值	描 述
@	无	指令标识符
config	config	配置指令
baund	baund	波特率标识
value	9600/115200	波特率值, 取值范围 9600、115200、921600
###	无	指令结尾标识符

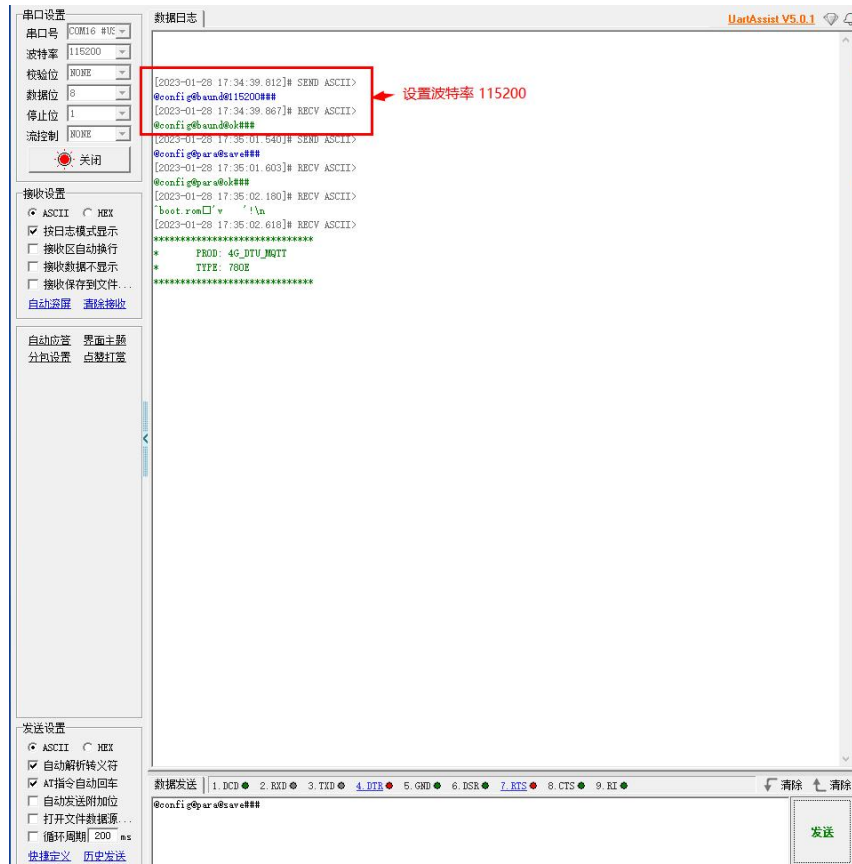
➤ 发送指令: @config@baund@115200###

➤ 返回值

成功: @config@baund@ok###

失败: @config@baund@error###

使用示例:



2. APN 配置指令

指令名称: @config@apn@name###

指令描述:

字段	值	描述
@	无	指令标识符
config	config	配置指令
apn	apn	指令标识
name	name	Apn 名称
###	无	指令结尾标识符

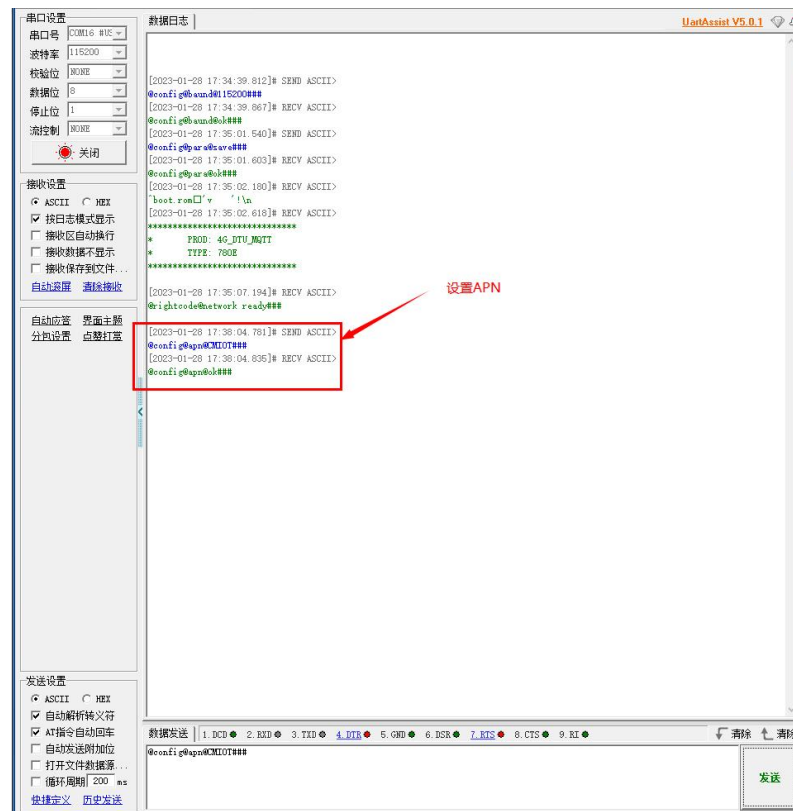
➤ 发送指令: @config@apn@CMIOT###

➤ 返回值

成功: @config@apn@ok###

失败: @config@apn@error###

使用示例:



3. 配置保存指令

指令名称: @config@para@save###

指令描述:

字段	值	描述
@	无	指令标识符
config	config	配置指令
para	para	参数标识
save	save	保存指令
###	无	指令结尾标识符

提示: 一旦发送该指令, 新配置的参数立马生效, 且模块自动重启。

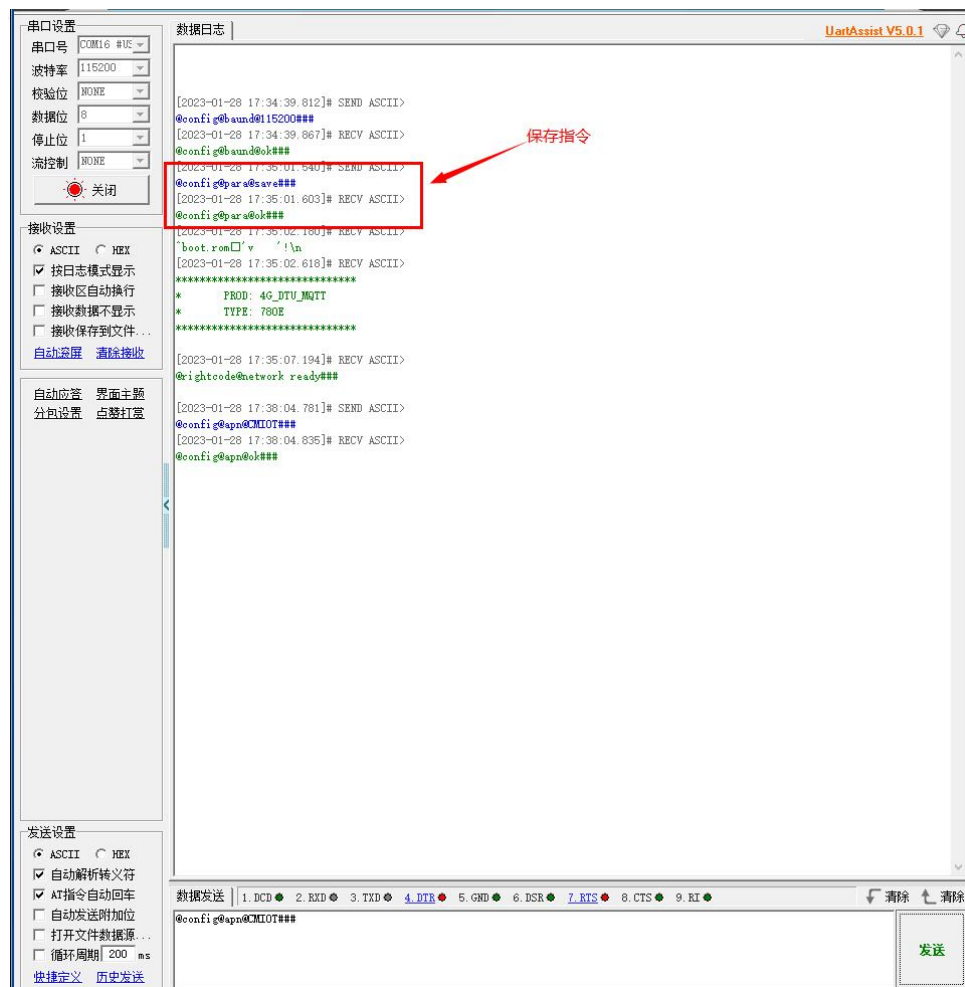
➤ 发送指令: @config@para@save###

➤ 返回值

成功: @config@para@ok###

失败: @config@para@error###

使用示例:



4. 服务器配置指令

指令名称: @config@server@tcp,ping,time,ip,port###

指令描述:

字 段	值	描 述
@	无	指令标识
config	config	配置指令
server	server	服务器标识
agreement	tcp/udp	网络通信协议 TCP 或 UDP
heartbeat	字符串	心跳名称, 字节长度不超过
time	时间	心跳间隔时间, 单位秒, 默认值
ip	地址	服务器 IP 地址, 255.255.255.255
port	端口号	服务器端口号
###	无	指令结尾标识符

➤ 发送指令: @config@server@tcp,ping,15,120.78.235.2,32972###

➤ 返回值

成功: @config@server@ok###

失败: @config@server@error###

使用示例:

5. 基站定位信息查询指令

指令名称: @get@position@

指令描述:

字 段	值	描 述
@	无	指令标识
get	get	获取指令
position	position	基站定位

➤ 发送指令: @get@position@

➤ 返回值:

成功: @get@position@113.3851809,23.1598592###

失败: @get@position@error###

使用示例：



6. 网络时间查询指令

指令名称: @get@time@

指令描述:

字 段	值	描 述
@	无	指令标识
get	get	获取指令
time	time	网络时间

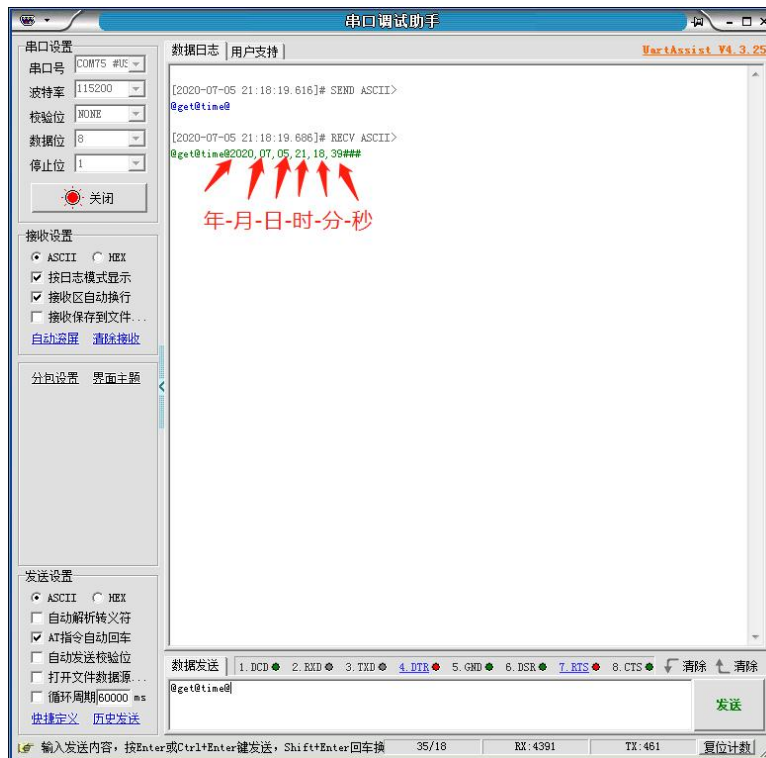
➤ 发送指令: @get@time@

➤ 返回值:

成功: @get@time@2020,05,20,08,30,20###

失败: @get@time@error###

使用示例:



7. 信号质量查询指令

指令名称: @get@csq@

指令描述:

字段	值	描述
@	无	指令标识
get	get	获取指令
csq	csq	信号质量

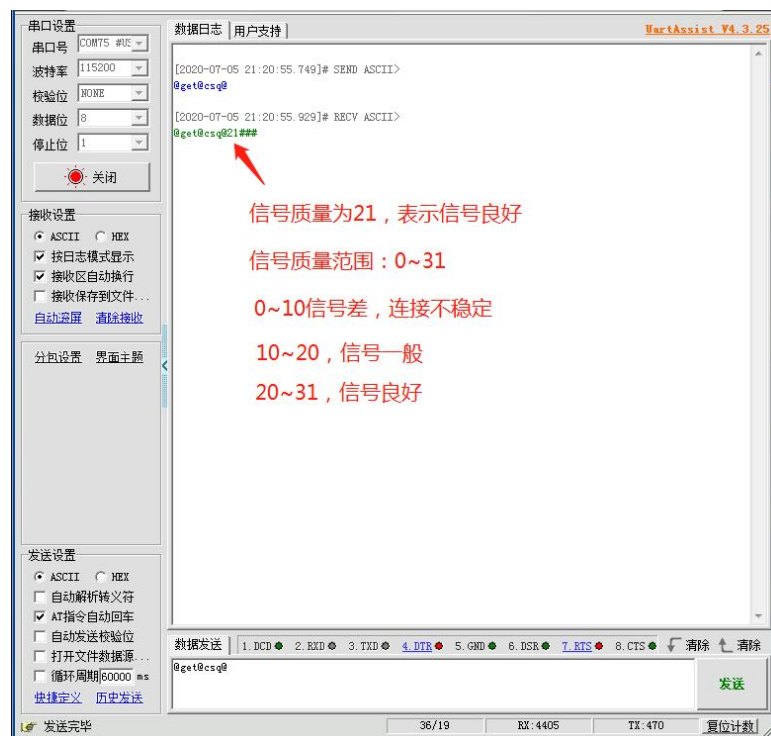
➤ 发送指令: @get@csq@

➤ 返回值:

成功: @get@csq@27###

失败: @get@csq@error###

使用示例:



8. SIM 卡 ICCID 号查询指令

指令名称: @get@iccid@

指令描述:

字 段	值	描 述
@	无	指令标识
get	get	获取指令
iccid	iccid	SIM 卡 ICCID 号

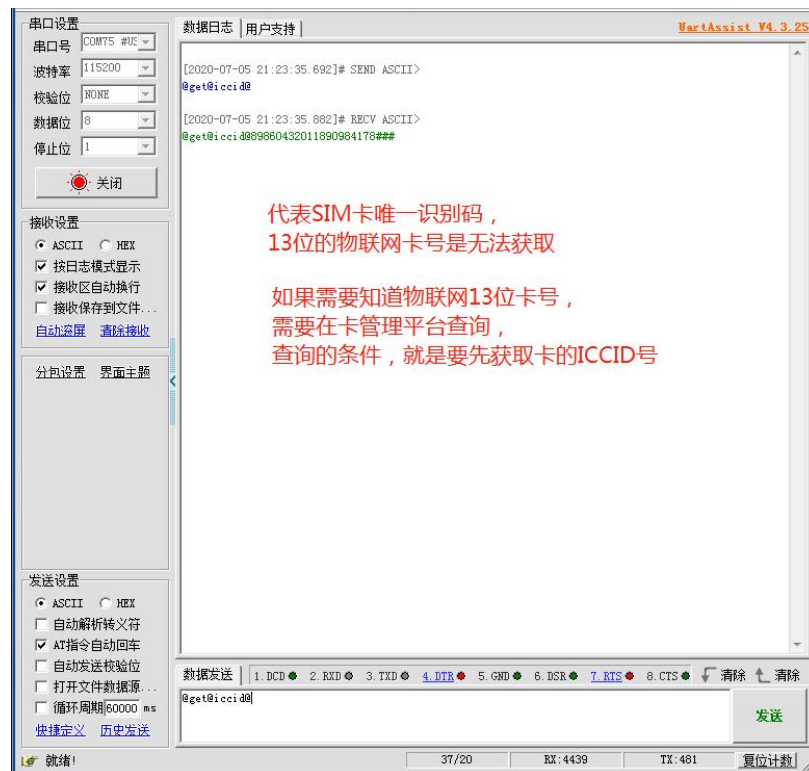
➤ 发送指令: @get@iccid@

➤ 返回值

成功: @get@iccid@6756861193947764369###

失败: @get@iccid@error###

使用示例:



9. 模组 IMEI 查询指令

指令名称: @get@imei@

指令描述:

字段	值	描述
@	无	指令标识
get	get	获取指令
imei	imei	IMEI 号

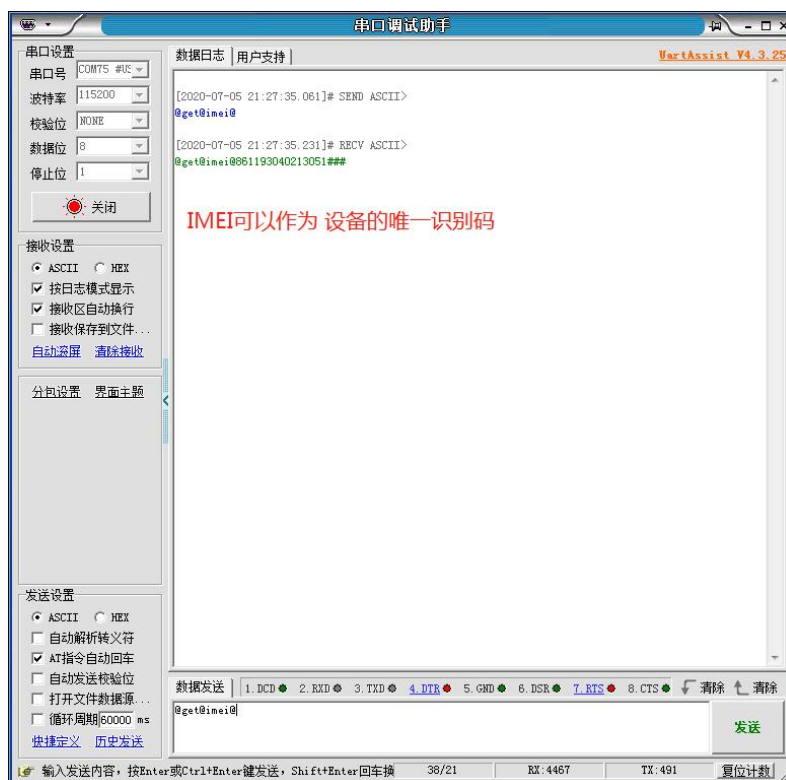
➤ 发送指令: @get@imei@

➤ 返回值:

成功: @get@imei@861193948764321###

失败: @get@imei@error###

使用示例:



10. 版本号查询指令

指令名称: @get@version@

指令描述:

字段	值	描述
@	无	指令标识
get	get	获取指令
version	version	版本号

➤ 发送指令: @get@version@

➤ 返回值

成功: @get@version@###

失败: @get@version@error###

使用示例:



11. GNSS 定位信息查询指令

定位信息第一种获取方式：

指令名称：@get@gnss@gnss_decode@

指令描述：

字段	值	描述
@	无	指令标识
get	get	获取指令
gnss_decode	gnss_decode	获取经纬度信息

➤ 发送指令：@get@gnss_decode@

➤ 返回值

成功：@get@gnss_decode@114.594185,27.388342,0.01###

失败：@get@gnss_decode@NoFix###

使用示例：



定位信息第二种获取方式:

指令名称: @get@gnss@gnss_raw@

指令描述:

字 段	值	描 述
@	无	指令标识
get	get	获取指令
gnss_raw	gnss_raw	获取源始定位信息

- 发送指令: @get@gnss@gnss_raw@
- 返回值

成功: @get@gnss_decode@114.594185,27.388342,0.01###

失败: @get@gnss_decode@NoFix###

使用示例:



12. MQTT 服务器参数配置

指令名称: @config@server@host,port,cliend_id,keepAlive###

指令描述:

字段	值	描述
@	无	指令标识
config	config	设置指令
server	server	服务器标识
host	地址	MQTT 服务器 IP 地址或域名
port	数值	MQTT 服务器端口号
cliend_id	字符串	MQTT_cliend_ID
keepAlive	数值	连接保活时间, 单位秒, 推荐值 60
###	无	指令结尾标识符

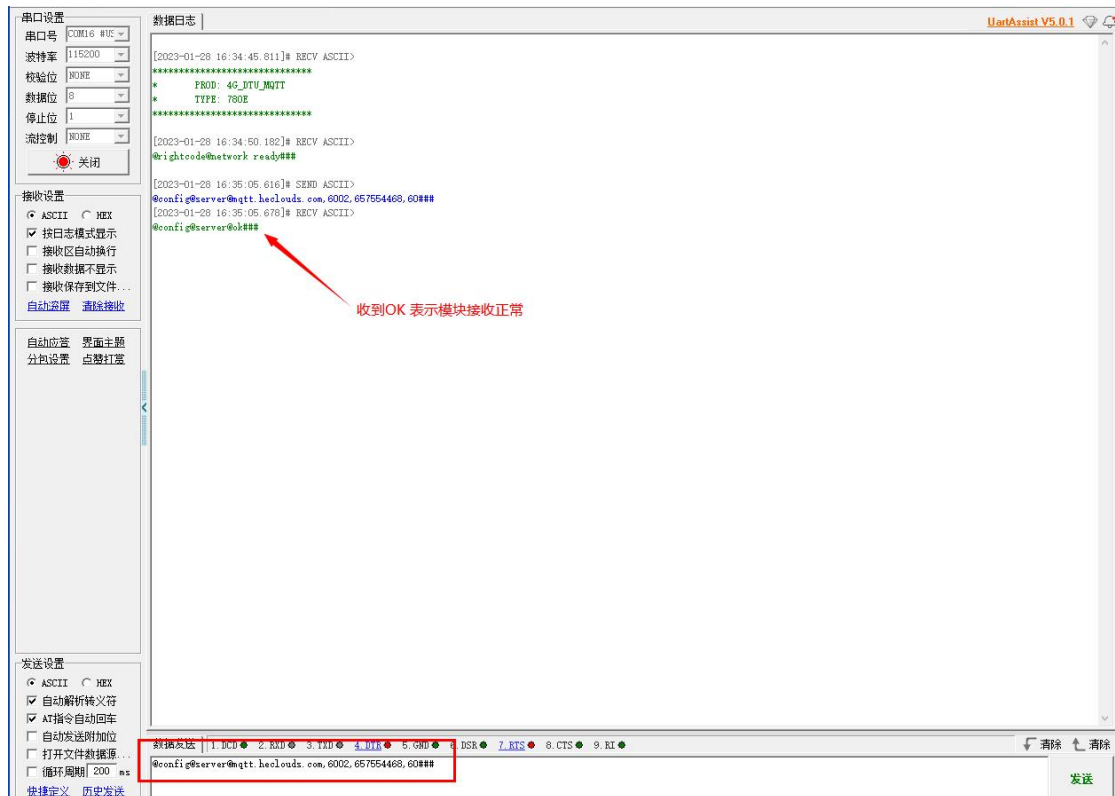
➤ 发送指令: @config@server@mq.tlink.io,1883,OK4ZS40G93619F5Y,60###

➤ 返回值

成功: @config@server@ok###

失败: @config@server@error###

使用示例:



13. MQTT 会话设置

指令名称: @config@remain@clean,version###

指令描述:

字段	值	描述
@	无	指令标识
config	config	设置指令
remain	remain	MQTT 会话
clean	0-1	1, 取消会话; 0, 不取消
version	2018-10-31	MQTT 协议版本默认 2018-10-31
###	无	指令结尾标识符

➤ 发送指令: @config@remain@1,2018-10-31###

➤ 返回值

成功: @config@remain@ok###

失败: @config@remain@error###

使用示例:



14. MQTT 用户信息设置

指令名称: @config@user@username,password,encrypt###

指令描述:

字段	值	描述
@	无	指令标识
config	config	设置指令
user	字符串	用户信息标识
username	字符串	MQTT 用户名 ID
password	字符串	MQTT 用户密码
encrypt	0-1	用户信息加密; 1, 加密; 0, 不加密
###	无	指令结尾标识符

➤ 发送指令: @config@user@220520,LBWIoT,0###

➤ 返回值

成功: @config@user@ok###

失败: @config@user@error###

使用示例:



15. MQTT 遗嘱设置

指令名称: @config@will@willFlag,willRetain,willQos,will-topic,will-mesg###

指令描述:

字段	值	描述
@	无	指令标识
config	config	设置指令
will	will	遗嘱标志
willFlag	0-1	遗嘱开启, 0 关闭, 1 开启
willRetain,	0-1	遗嘱消息是否保存; 0 关闭, 1 开启
willQos	0-2	遗嘱等级 0/1/2
will-topic	字符串	遗嘱主题, 最大 512 字节
will-mesg	字符串	遗嘱消息, 最大 512 字节
###	无	指令结尾标识符

➤ 发送指令: @config@will@0,0,0,0,0###

➤ 返回值

成功: @config@will@ok###

失败: @config@will@error###

使用示例:



16. 设置订阅主题

指令名称: @config@sub@topic,qos###

指令描述:

字段	值	描述
@	无	指令标识
config	config	设置指令
sub	sub	订阅指令标志
topic	字符串	MQTT 服务器上的主题
qos	0-2	主题等级 0/1/2
###	无	指令结尾标识符

➤ 发送指令: @config@sub@IOT_LBW,0###

➤ 返回值

成功: @config@sub@ok###

失败: @config@sub@error###

使用示例: 若报重复订阅相同主题, 则无需再设置。

串口设置

串口号: COM16 #US

波特率: 115200

校验位: NONE

数据位: 8

停止位: 1

流控制: NONE

接收设置

☒ ASCII ☐ HEX

☒ 按日志模式显示

☐ 接收区自动换行

☐ 接收数据不显示

☐ 接收保存到文件...

自动应答 清除接收

自动应答 界面主题

分包设置 点阵打宽

发送设置

☒ ASCII ☐ HEX

☒ 自动解析转义符

☒ AT指令自动回车

☐ 自动发送附加位

☐ 打开文件数据源...

☐ 循环周期: 200 ms

快捷定义 历史发送

数据日志

UartAssist V5.0.1

[2023-01-28 16:34:45.811]# RECV ASCII>

* PROD: 4G_DTU_MQTT

* TYPE: 780E

[2023-01-28 16:34:50.182]# RECV ASCII>

@rightcode@network ready###

[2023-01-28 16:35:05.616]# SEND ASCII>

@config@server@mqtt.heclouids.com,6002,657554468,60###

[2023-01-28 16:35:05.678]# RECV ASCII>

@config@server@ok###

[2023-01-28 16:43:21.669]# SEND ASCII>

@config@remain@1,2018-10-31###

[2023-01-28 16:43:21.727]# RECV ASCII>

@config@remain@ok###

[2023-01-28 16:44:51.860]# SEND ASCII>

@config@user@220520,LBW,IOT,0###

[2023-01-28 16:44:51.919]# RECV ASCII>

@config@user@ok###

[2023-01-28 16:48:09.902]# SEND ASCII>

@config@will@0,0,0,0###

[2023-01-28 16:48:09.969]# RECV ASCII>

@config@will@ok###

[2023-01-28 16:50:12.651]# SEND ASCII>

@config@sub@IOT_LBW,0###

[2023-01-28 16:50:12.704]# RECV ASCII>

@config@sub@ok###

[2023-01-28 16:50:14.535]# RECV ASCII>

@config@sub@repeat###

首次设置订阅主题

如果是重复设置相同主题, 则会报重复订阅

数据发送

1. DCD 2. RXD 3. TXD 4. DTR 5. GND 6. DSR 7. RTS 8. CTS 9. RI

@config@sub@IOT_LBW,0###

发送

17. 设置发布主题

指令名称: @config@pub@topic,qos,retain,dup###

指令描述:

字段	值	描述
@	无	指令标识
config	config	设置指令
pub	pub	发布指令标志
topic	字符串	MQTT 服务器上的主题
qos	0-2	主题等级 0/1/2
retain	0-1	服务器是否存储消息，并下发新的订阅者，0 否；1 是
dup	0-1	消息是否重发
###	无	指令结尾标识符

➤ 发送指令: @config@pub@IOT_LBW,0,0,0###

➤ 返回值

成功: @config@pub@ok###

失败: @config@pub@error###

使用示例:

The screenshot shows the UartAssist V5.0.1 interface. On the left, the '串口设置' (Serial Port Settings) panel is visible, showing 'COM16 #UC' as the port and '115200' as the baud rate. The '接收设置' (Receive Settings) panel shows 'ASCII' selected. The '发送设置' (Send Settings) panel shows 'ASCII' selected. The main '数据日志' (Data Log) window displays a series of received and sent messages. A red box highlights the message '@config@pub@IOT_LBW,0,0,0###' in the log, and a red arrow points to it with the text '设置发布主题' (Set publish topic). The bottom of the window shows the '数据发送' (Data Send) area with the same command entered and a '发送' (Send) button.

18. 查询 MQTT 服务器配置参数

指令名称: @get@server@

指令描述:

字段	值	描述
@	无	指令标识
get	get	查询指令
server	server	服务器指令标志
###	无	指令结尾标识符

➤ 发送指令: @get@server@

➤ 返回值

成功: @set@server@mq.tlink.io,1883,OK4ZS40G93619F5Y,60###

失败: @set@server@error###

使用示例:



19. 清除所有订阅主题

指令名称: @clean@sub@

指令描述：

字 段	值	描 述
@	无	指令标识
clean	clean	清除标志
sub	sub	订阅的主题

➤ 发送指令：@clean@sub@

➤ 返回值

成功：@clean@sub@ok###

失败：@clean@sub@error###

使用示例：模块存储的旧平台的订阅主题需要清除掉

串口设置

串口号 COM26 #UC

波特率 115200

校验位 NONE

数据位 8

停止位 1

流控制 NONE

接收设置

ASCII HEX

按日志模式显示

接收区自动换行

接收数据不显示

接收保存到文件...

自动滚屏 清除接收

发送设置

ASCII HEX

自动解析转义符

AT指令自动回车

自动发送附加位

打开文件数据源...

循环周期 200 ms

快捷定义 历史发送

数据日志

UartAssist V5.0.1

[2023-01-31 19:11:09.570]# 设备已插入

[2023-01-31 19:11:31.606]# SEND ASCII>

@clean@sub@

[2023-01-31 19:11:31.669]# RECV ASCII>

@clean@sub@ok###

清除被存储的订阅主题

[2023-01-31 19:11:41.519]# SEND ASCII>

@config@para@save###

[2023-01-31 19:11:41.580]# RECV ASCII>

@config@para@ok###

需要发保存指令，才会生效

[2023-01-31 19:11:42.169]# RECV ASCII>

boot.rom v '!\n

[2023-01-31 19:11:42.626]# RECV ASCII>

* PROD: 4G_DTU_MQTT

* TYPE: 780E

[2023-01-31 19:11:43.696]# RECV ASCII>

@rightcode@network ready###

数据发送

1. DCD 2. RXD 3. TXD 4. DTR 5. GND 6. DSR 7. RTS 8. CTS 9. RI

@config@para@save###

清除 清除

发送

20. 查询网络连接状态

指令名称: @get@netstate@

指令描述:

字 段	值	描 述
@	无	指令标识
get	get	查询指令
netstate	netstate	服务器指令标志

➤ 发送指令: @get@netstate@

➤ 返回值

成功: @get@netstate@ok###

失败: @get@netstate@fail###

使用示例:



21. 配置应答模式

指令名称: @config@mqtt_ack_mode@value###

指令描述: 模块出厂默认方式是数据透传模式

字段	值	描述
@	无	指令标识
config	config	配置指令
mqtt_ack_mode	mqtt_ack_mode	应答模式指令标志
value	1/0	0: 应答数据透传模式 1: 应答数据含头部信息

➤ 发送指令: @config@mqtt_ack_mode@1###

➤ 返回值

成功: @config@mqtt_ack_mode@ok###

失败: @config@mqtt_ack_mode@error###

串口设置

串口号: COM26 #US

波特率: 115200

校验位: NONE

数据位: 8

停止位: 1

流控制: NONE

接收设置

ASCII HEX

按日志模式显示

接收区自动换行

接收数据不显示

接收保存到文件...

自动滚屏 清除接收

自动应答 页面主题

分包设置 点灯打靶

发送设置

ASCII HEX

自动解析转义符

AT指令自动回车

自动发送附加位

打开文件数据源...

循环周期: 200 ms

快捷定义 历史发送

数据日志

UartAssist V5.0.1

多主题订阅工作模式下: 应答数据含头部信息

设置此模式, 则模块返回的消息中含有主题来源信息

消息头部信息

消息内容

数据发送

1. DCD 2. RXD 3. TXD 4. DTR 5. GND 6. DSR 7. RTS 8. CTS 9. RI

清除 清除

发送

22. 获取 MQTT 账户参数

指令名称: @get@user@

指令描述:

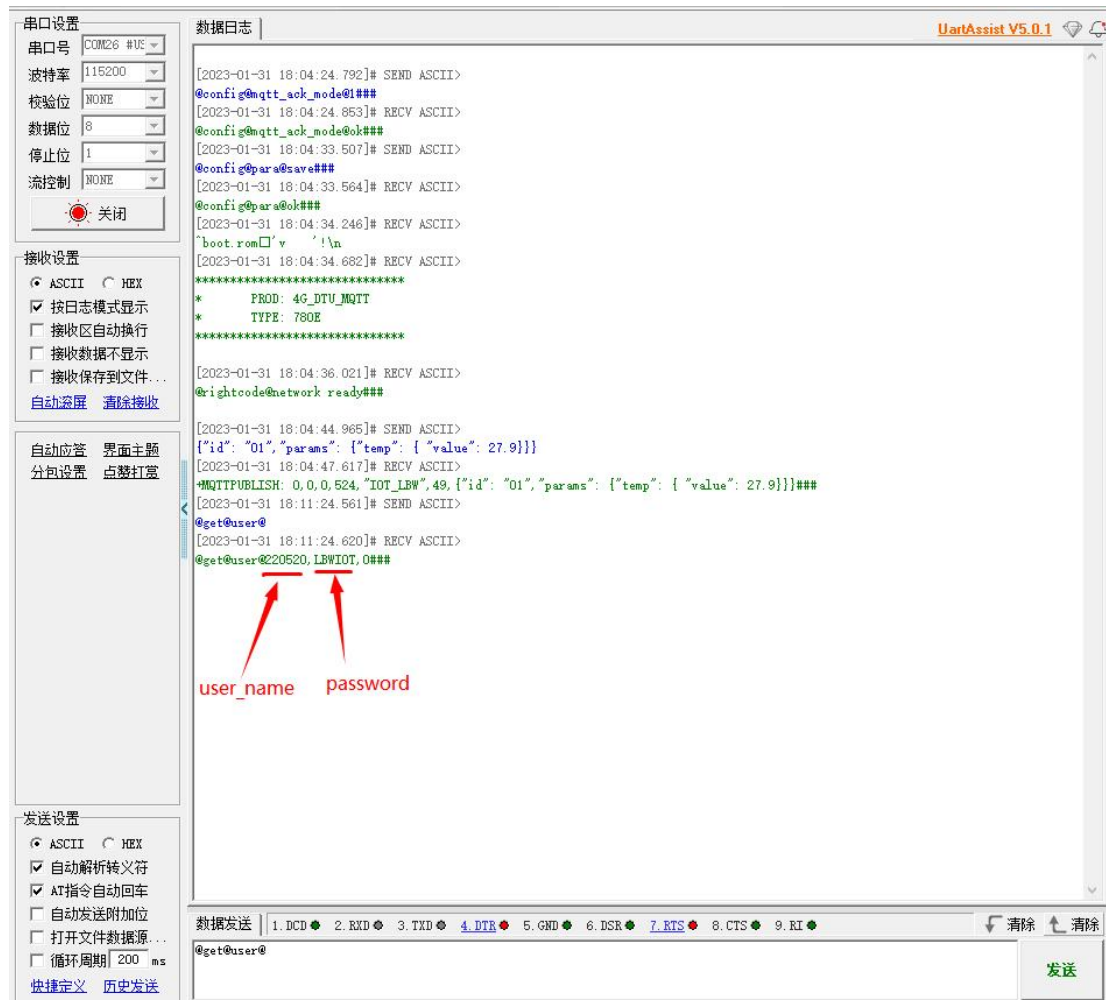
字段	值	描述
@	无	指令标识
get	get	配置指令
user	user	MQTT 用户信息指令标志

➤ 发送指令: @get@user@

➤ 返回值

成功: @get@user@username,password,encrypt###

失败: @get@user@error###



The screenshot shows the UartAssist V5.0.1 interface. On the left, the '串口设置' (Serial Port Settings) panel is visible, showing 'COM26 #US' as the selected port. The '数据日志' (Data Log) panel on the right displays the following log entries:

```

[2023-01-31 18:04:24.792]# SEND ASCII>
@config@mqtt_ack_mode@###
[2023-01-31 18:04:24.853]# RECV ASCII>
@config@mqtt_ack_mode@ok###
[2023-01-31 18:04:33.507]# SEND ASCII>
@config@para@save###
[2023-01-31 18:04:33.564]# RECV ASCII>
@config@para@ok###
[2023-01-31 18:04:34.246]# RECV ASCII>
"boot.rom" v "!\n
[2023-01-31 18:04:34.682]# RECV ASCII>
*****
*   PROD: 4G_DTU_MQTT
*   TYPE: 780E
*****
[2023-01-31 18:04:36.021]# RECV ASCII>
@rightcode@network_ready###
[2023-01-31 18:04:44.965]# SEND ASCII>
{"id": "01", "params": [{"temp": {"value": 27.9}}]}
[2023-01-31 18:04:47.617]# RECV ASCII>
+MQTTPUBLISH: 0,0,0,524,"IOT_LBW",49,{"id": "01", "params": [{"temp": {"value": 27.9}}]}###
[2023-01-31 18:11:24.561]# SEND ASCII>
@get@user@
[2023-01-31 18:11:24.620]# RECV ASCII>
@get@user@20520,LBW10T.0###
  
```

Red arrows point to the values '20520' and 'LBW10T.0' in the log, which are labeled 'user_name' and 'password' respectively.

23. 获取发布主题参数

指令名称: @get@pub@

指令描述:

字段	值	描述
@	无	指令标识
get	get	配置指令
pub	pub	发布主题指令标志

➤ 发送指令: @get@pub@

➤ 返回值

成功: @get@pub@<topic>###

失败: @get@pub@error###



24. 获取订阅主题参数

指令名称: @get@sub@

指令描述:

字段	值	描述
@	无	指令标识
get	get	配置指令
sub	sub	发布主题指令标志

➤ 发送指令: @get@sub@

➤ 返回值

成功: @get@sub@<topic1>,<topic2>,<topic3>###

失败: @get@sub@error###

使用示例:

串口设置: COM26 #115200 NONE 8 1 NONE 关闭

接收设置: ASCII HEX 换日志模式显示 接收数据不显示 接收数据不显示 接收数据不显示 自动滚屏 清除接收

发送设置: ASCII HEX 自动解析转义符 AT指令自动回车 自动发送附加位 打开文件数据源... 循环周期 200 ms 快捷定义 历史发送

数据日志: [2023-01-31 19:11:09.570]# 设备已插入
[2023-01-31 19:11:31.606]# SEND ASCII>
@clean@sub@
[2023-01-31 19:11:31.669]# RECV ASCII>
@clean@sub@ok###
[2023-01-31 19:11:41.519]# SEND ASCII>
@config@para@save###
[2023-01-31 19:11:41.580]# RECV ASCII>
@config@para@ok###
[2023-01-31 19:11:42.189]# RECV ASCII>
boot_rom v '!\n
[2023-01-31 19:11:42.628]# RECV ASCII>

* FRID: 4G_DTU_MQTT
* TYPE: 780E

[2023-01-31 19:11:43.696]# RECV ASCII>
@rightode@network ready###
[2023-01-31 19:16:41.497]# SEND ASCII>
@get@sub@
[2023-01-31 19:16:41.552]# RECV ASCII>
@get@sub@,###
[2023-01-31 19:19:17.151]# SEND ASCII>
@config@sub@IOT_LBW,0###
[2023-01-31 19:19:17.205]# RECV ASCII>
@config@sub@ok###
[2023-01-31 19:19:24.626]# SEND ASCII>
@config@para@save###
[2023-01-31 19:19:24.682]# RECV ASCII>
@config@para@ok###
[2023-01-31 19:19:25.270]# RECV ASCII>
boot_rom v '!\n
[2023-01-31 19:19:25.724]# RECV ASCII>

* FRID: 4G_DTU_MQTT
* TYPE: 780E

[2023-01-31 19:19:26.917]# RECV ASCII>
@rightode@network ready###
[2023-01-31 19:19:45.401]# SEND ASCII>
@get@sub@
[2023-01-31 19:19:45.441]# RECV ASCII>
@get@sub@IOT_LBW,,###

由于清除了订阅主题，所以获取到的主题为空

重新再订阅新的主题

模组最大存储订阅主题个数是三个，分别用逗号隔开

查询存储的订阅主题，显示只有一个订阅主题

数据发送: 1.DCD 2.RXD 3.TXD 4.DTR 5.GND 6.DSR 7.RTS 8.CTS 9.RI 清除 清除

发送: @get@sub@ 发送

就绪! 57/14 RX:2840 TX:295 复位计数

25. 查询 MQTT 登录状态

指令名称: @get@platform connect state@

指令描述:

字段	值	描述
@	无	指令标识
get	get	查询指令
platform connect state	platform connect state	服务器状态标志

➤ 发送指令: @get@platform connect state@

➤ 返回值

成功: @get@platform connect state@succeed###

失败: @get@platform connect state@fail###

使用示例:

The screenshot shows the UartAssist V5.0.1 interface. On the left, the '串口设置' (Serial Port Settings) panel is visible with 'COM16 #US' selected. The '数据日志' (Data Log) panel on the right displays a series of communication logs. A red box highlights the command '@get@platform connect state@' and its response '@get@platform connect state@succeed###'. A red arrow points to this response with the label '查询登录MQTT平台状态' (Query MQTT platform login status). Below the log, the '数据发送' (Data Send) panel shows the command being sent to the device.

```

[2023-01-28 17:08:38.582]# SEND ASCII>
@onfig@server@mqtt.heclouds.com,6002,657554468,60###
[2023-01-28 17:08:38.650]# RECV ASCII>
@onfig@server@ok###
[2023-01-28 17:08:48.558]# SEND ASCII>
@onfig@user@220520,LEWIoT,0###
[2023-01-28 17:08:48.626]# RECV ASCII>
@onfig@user@ok###
[2023-01-28 17:08:56.082]# SEND ASCII>
@onfig@remain@01,2018-10-31###
[2023-01-28 17:08:56.137]# RECV ASCII>
@onfig@remain@ok###
[2023-01-28 17:09:05.502]# SEND ASCII>
@onfig@will@0,0,0,0###
[2023-01-28 17:09:05.570]# RECV ASCII>
@onfig@will@ok###
[2023-01-28 17:09:12.911]# SEND ASCII>
@onfig@sub@IoT_LW,0###
[2023-01-28 17:09:12.979]# RECV ASCII>
@onfig@sub@ok###
[2023-01-28 17:09:20.795]# SEND ASCII>
@onfig@pub@IoT_LW,0,0,0###
[2023-01-28 17:09:20.850]# RECV ASCII>
@onfig@pub@ok###
[2023-01-28 17:09:34.345]# SEND ASCII>
@onfig@para@save###
[2023-01-28 17:09:34.460]# RECV ASCII>
@onfig@para@ok###
[2023-01-28 17:09:34.985]# RECV ASCII>
'boot.rom'v '!\n
[2023-01-28 17:09:35.420]# RECV ASCII>
*****
*   PROD: 4G_DTU_MQTT
*   TYPE: 780E
*****
[2023-01-28 17:09:39.957]# RECV ASCII>
@rightcode@network ready###
[2023-01-28 17:09:45.410]# SEND ASCII>
@get@platform connect state@
[2023-01-28 17:09:45.472]# RECV ASCII>
@get@platform connect state@succeed###
[2023-01-28 17:10:29.652]# SEND ASCII>
{"id": "01", "params": {"temp": {"value": 27.9}}}
[2023-01-28 17:10:32.274]# RECV ASCII>
{"id": "01", "params": {"temp": {"value": 27.9}}}
  
```

26. 发布消息和接收消息

MQTT 透传功能：单个主题模式下，直接数据透传。

示例：



27. 注册包开关功能

仅 TCP 模式支持注册包功能，当 TCP 连接成功，自动向服务器发送注册包。

指令名称：@config@login_switch@parameter###

指令描述：

字段	值	描述
@	无	指令标识
config	config	配置指令
login_switch	login_switch	注册包开关指令
parameter	parameter	开关值，1 开启，0 关闭
###	###	指令结束符

示例：

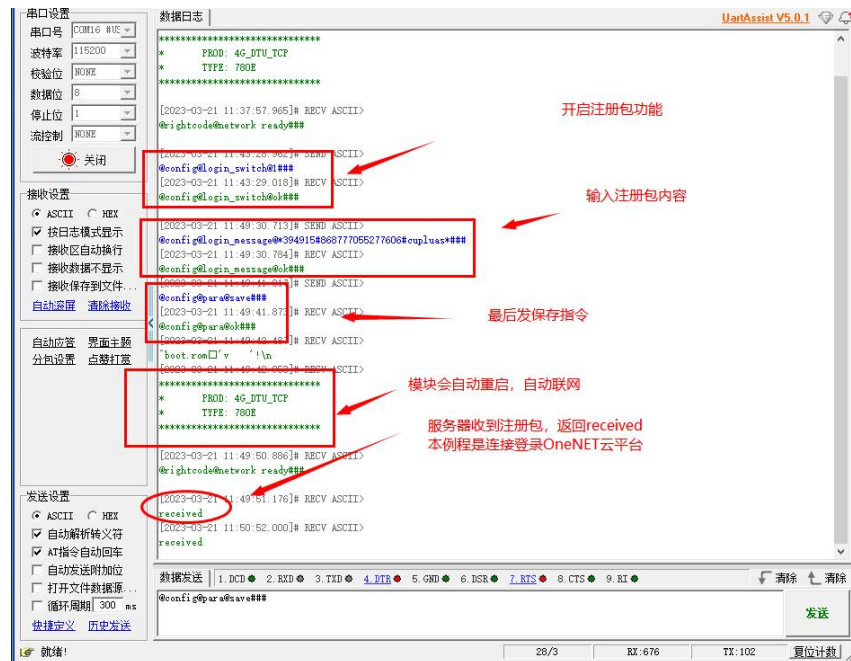
➤ 发送指令：@config@login_switch@1###

➤ 返回值

成功：@config@login_switch@ok###

失败：@config@login_switch@error###

使用示例：



28. 注册包内容

注册包内容可自定义。

指令名称: @config@login_switch@parameter###

指令描述:

字段	值	描述
@	无	指令标识
config	config	配置指令
login_message	login_message	注册包开关指令
parameter	parameter	注册包内容, 可自定义填写
###	###	指令结束符

示例: 注册包内容: *394915#868777055277606#cupluas*

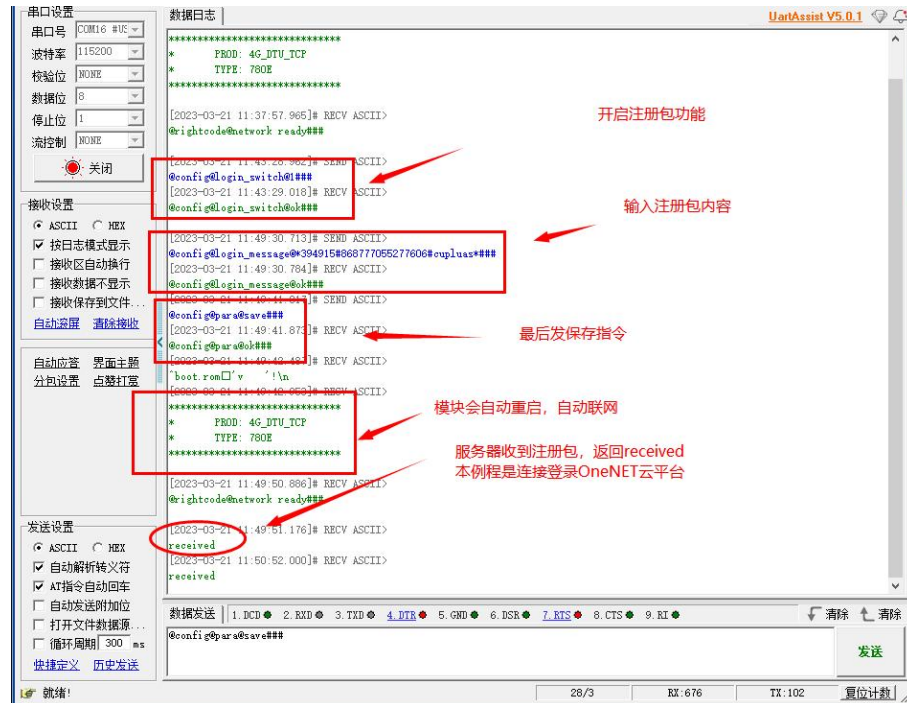
➤ 发送指令: @config@login_message@*394915#868777055277606#cupluas*###

➤ 返回值

成功: @config@login_message@ok###

失败: @config@login_message@error###

使用示例:



状态码列表

状态码	上报状态	描 述
错误码	@errorcode@sim card error###	SIM 卡异常
	@errorcode@open protocol error###	打开协议栈失败
	@errorcode@access net not allow###	网络接入异常
	@errorcode@apn error###	APN 设置异常
	@errorcode@pdp error###	基站连接失败
正确码	@rightcode@network ready###	连接网络成功
	@rightcode@domain name decode ok###	域名解析成功