



# MOVE SOLUTIONS

## MAMWLE FW-AT COMMAND DOCUMENT

### SUMMARY

AT DOCUMENTATION	<b>3</b>
Introduction	3
Power consumption (release mode)	3
FW memory occupation	4
Required software	4
First use	5
Release and debug modes	8
Node configuration	8
AT COMMAND	<b>11</b>
ATZ	11
AT?	11
AT+TXP=Power Index	12
AT+DEUI=Device EUI	13
AT+APPEUI=Application Key	14
AT+NWKKEY=Network Key	16
AT+APPKEY=Application Key	17
AT+JOIN=Join type	18
AT+CHANNEL=Mask0:Mask1:Mask2:Mask3:Mask4:Mask5:Mask6:Mask7:Mask8:Mask9:Mask10:Mask11	18
AT+ADR=ADR type	21
AT+DR=Data Rate	22
AT+CW=Timeout:Frequency:Power level	23
AT+STOPCW	24
AT+CFGSEND=Port:Number of send:Ack:Send timer:Datarate:Power index	25
AT+SEND=Payload Length:Payload raw	26
AT+STOPSEND	27
AT+VL=verbosity level:timestamp flag	27
AT+VER=?	28
AT+DADDR=device address	29

AT+DCS=value	30
AT+RX2FQ=frequency	31
AT+RX2DR=DataRate	32
AT+RXIDL=delay	33
AT+RX2DL=delay	33
AT+JN1DL=delay	34
AT+JN2DL=delay	36
AT+NWKID=network ID	37
AT+CLASS=class value	37
AT+LTIME=?	38
AT+TREQ	38
AT+UID	39
AT+NWKSKEY=network session key	39
AT+APPSKEY=application session key	40
AT+LINKC	40
AT+CERTIF=Join type	41
AT+BAT=?	41
AT+TTH=Start frequency:Stop frequency:frequency resolution:number of packets to send	42
AT+TCNF=TX frequency:TX power:BW:SF:4/CR:LNA state:Boost PA state:Modulation type:Payload length:Low DR optimization:BT FSK product	44
AT+TTX=Send's number:Channel frequency:Spreading Factor:BW:Payload Length:Payload	46
AT+TRX=Packet's number to receive:Channel frequency:Spreading Factor:BW	47
<b>REFERENCE DOCUMENTS</b>	<b>49</b>

## AT DOCUMENTATION

### Introduction

This brief note explains how to interface with the LoRaWAN® to manage LoRa® wireless link via AT commands. This document lists the set of AT commands, and a summary of the configuration needed, on the STM32WLE5JB and STM32WLE5JC (the only difference between the two is that the former has 64Kbytes of SRAM and 256Kbytes of Flash memory, while the latter has 48Kbytes of SRAM and 128Kbytes of Flash memory). The FW explained in this note is the 1.0 version. The TX and RX pins of the MAMWLEXX board are, respectively, PA2 and PA3.

### Power consumption (release mode)

The power consumption of the node had been measured in EU868 band, with the digital multimeter "Siglent SPD3303C", and the result are shown below. The mean values are:

- 2µA in sleep mode;
- 24mA in TX with maximum power (16dBm);
- 13mA in TX with minimum power (2dBm);
- 37mA in continuous wave TX with maximum power (15dBm);
- 14mA in continuous wave TX with minimum power (0dBm);
- 7mA in RX (class C);

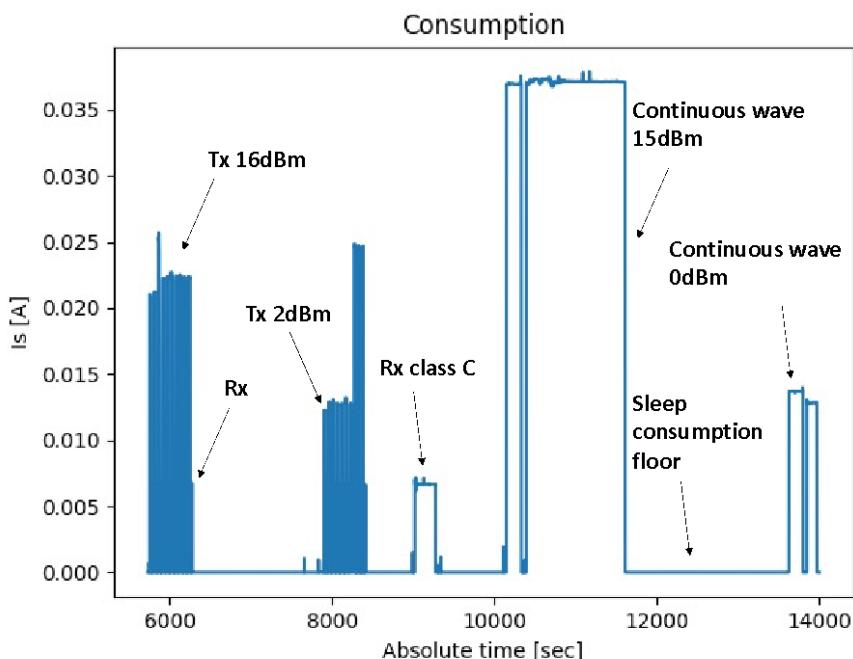


Figure 1, Power consumption in release mode

## FW memory occupation

The FW occupies 99.73 KBytes in Debug mode and 79.92 KBytes in Release mode.

Region	Start address	End address	Size	Free	Used	Usage (%)
RAM1	0x20000000	0x20006000	24 KB	13,98 KB	10,02 KB	41,73%
RAM2	0x20008000	0x2000e000	24 KB	24 KB	0 B	0,00%
FLASH	0x08000000	0x08020000	128 KB	27,95 KB	100,05 KB	78,16%

Figure 2 Debug mode

Region	Start address	End address	Size	Free	Used	Usage (%)
RAM1	0x20000000	0x20006000	24 KB	13,98 KB	10,02 KB	41,73%
RAM2	0x20008000	0x2000e000	24 KB	24 KB	0 B	0,00%
FLASH	0x08000000	0x08020000	128 KB	47,95 KB	80,05 KB	62,54%

Figure 3 Release mode

## Required software

- STM32CubeIDE
- Client for send/receive data on serial port (PuTTY for Windows, CoolTerm for Mac)

Configuration for serial communication:



Figure 4, Serial application settings

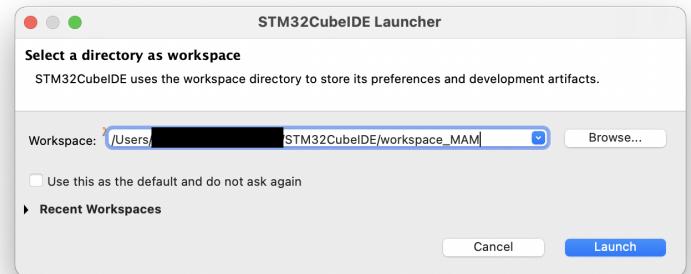


Figure 5, Creation of a new workspace

- Import the project

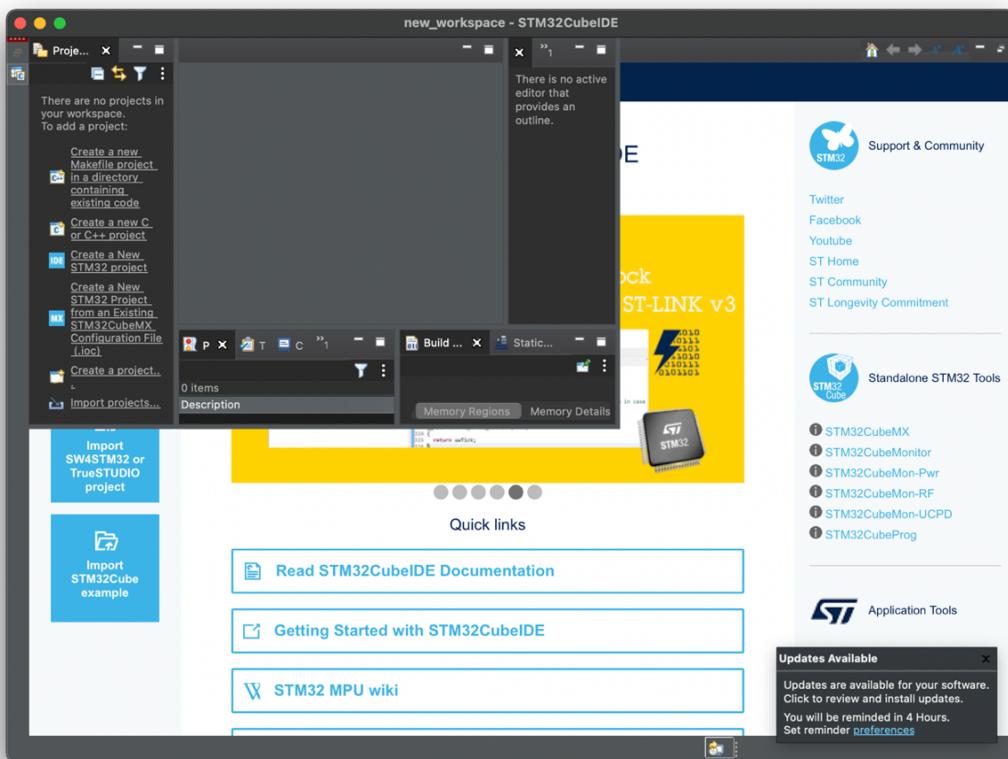


Figure 6, Import of a new project (1)

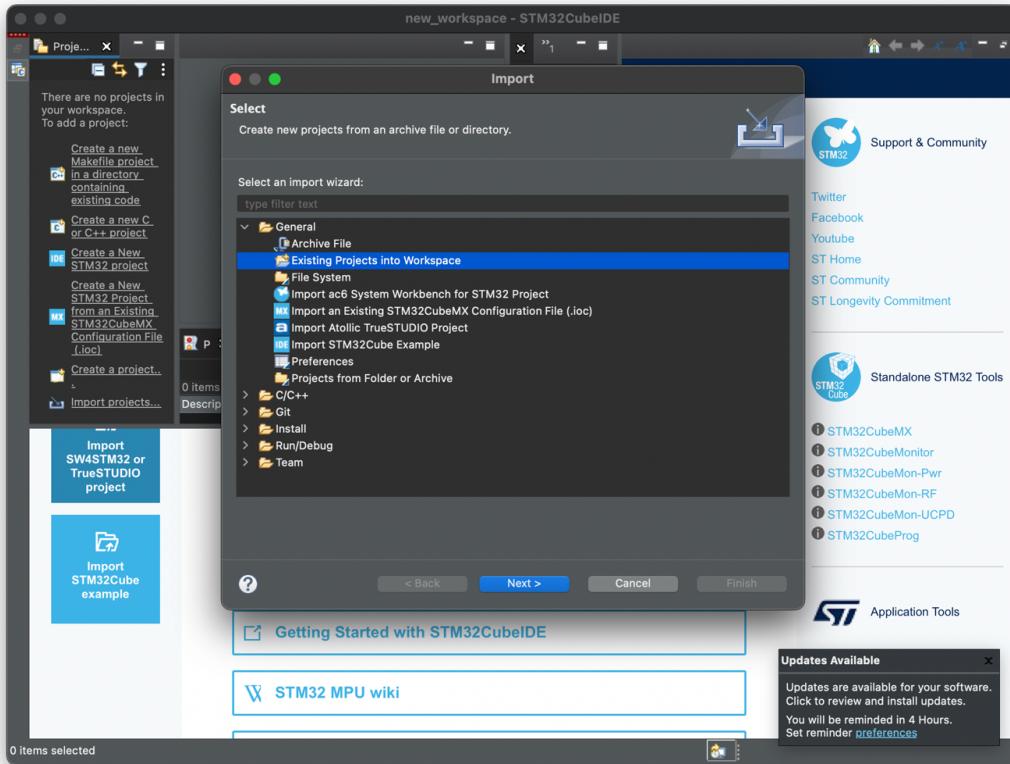


Figure 7, Import of a new project (2)

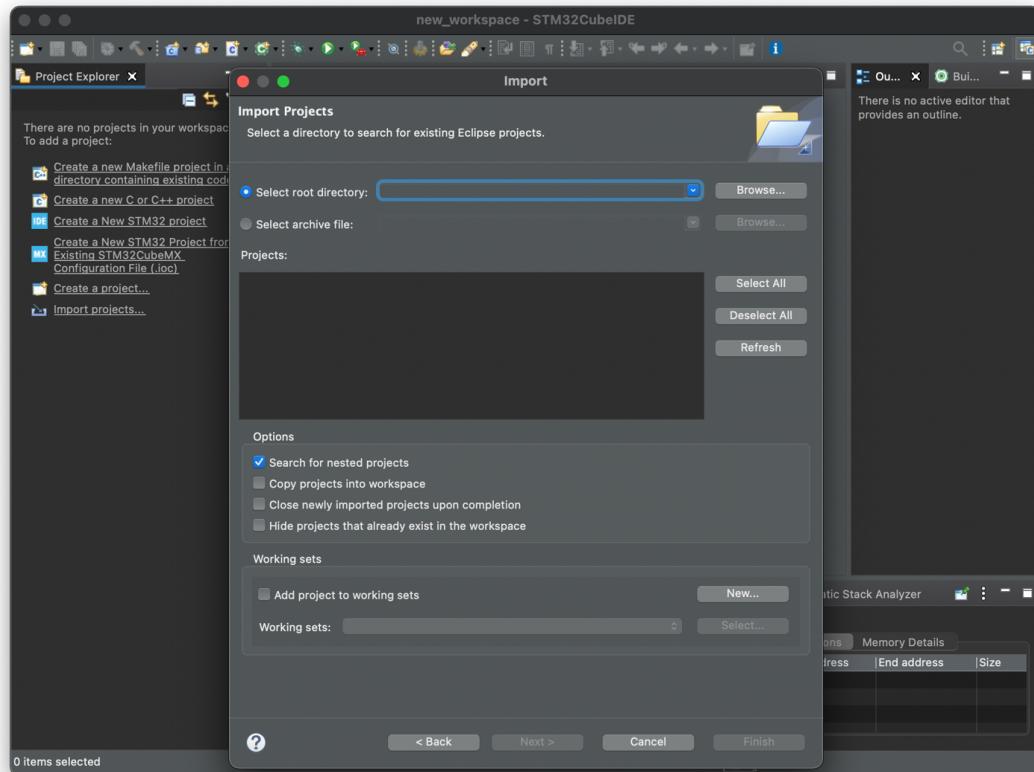


Figure 8, Import of a new project (3)

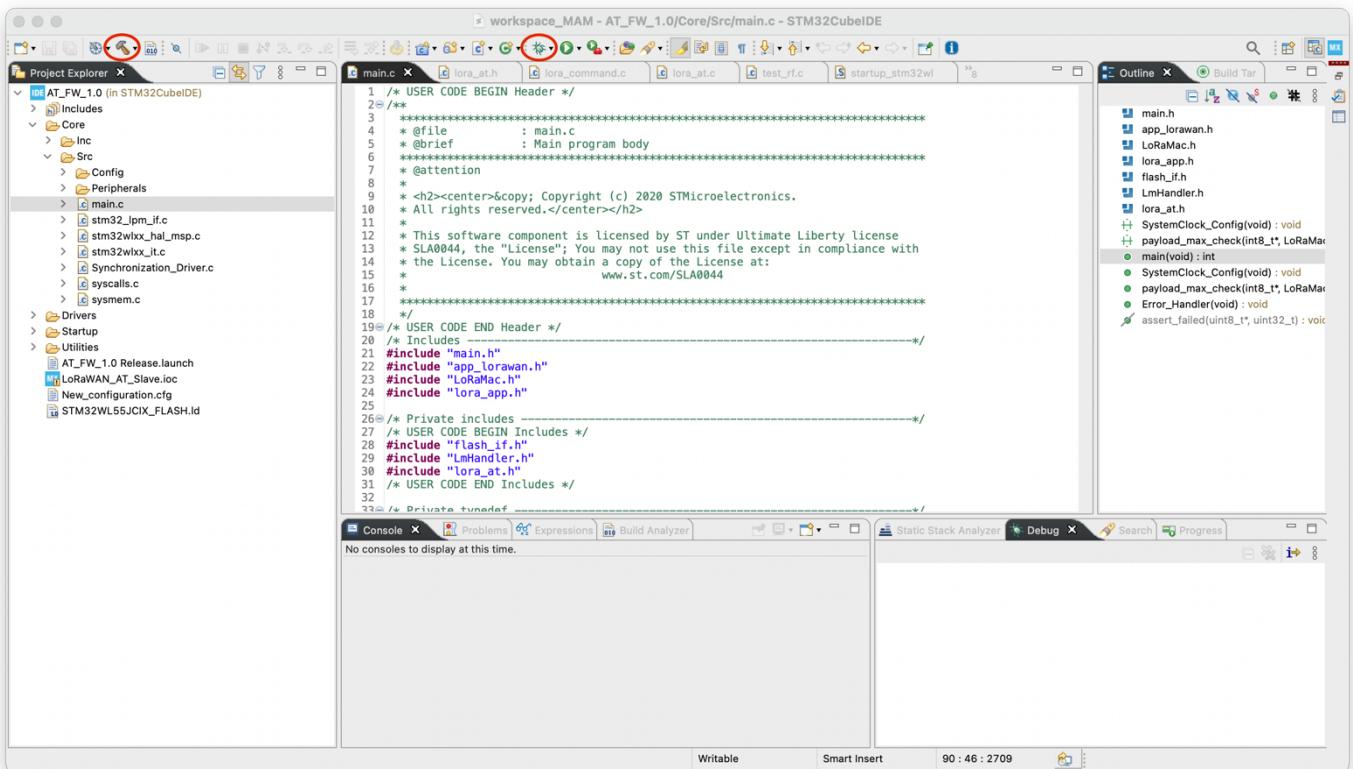


Figure 9, Import of a new project (4), with Debug and Flash button, respectively, highlighted

## Release and debug modes

The former is the commercial mode, with minimal and fixed output from the node, minimum dimension and power consumption, while the latter is the prototyping mode, with extensive message returned from the node. Going in release mode automatically **disable the debugger** and **activate the low power mode**.

## Node configuration

The node configuration, debug or release, is chosen in STM32CubeIDE: to set a new configuration go in "Debug configurations...", then create a new configuration by pressing the "New Configuration" button, and finally set the configuration parameters like the name, build configuration, and the serial number of the ST Link in the "Main" and "Debugger" windows, as showed in the pictures below:

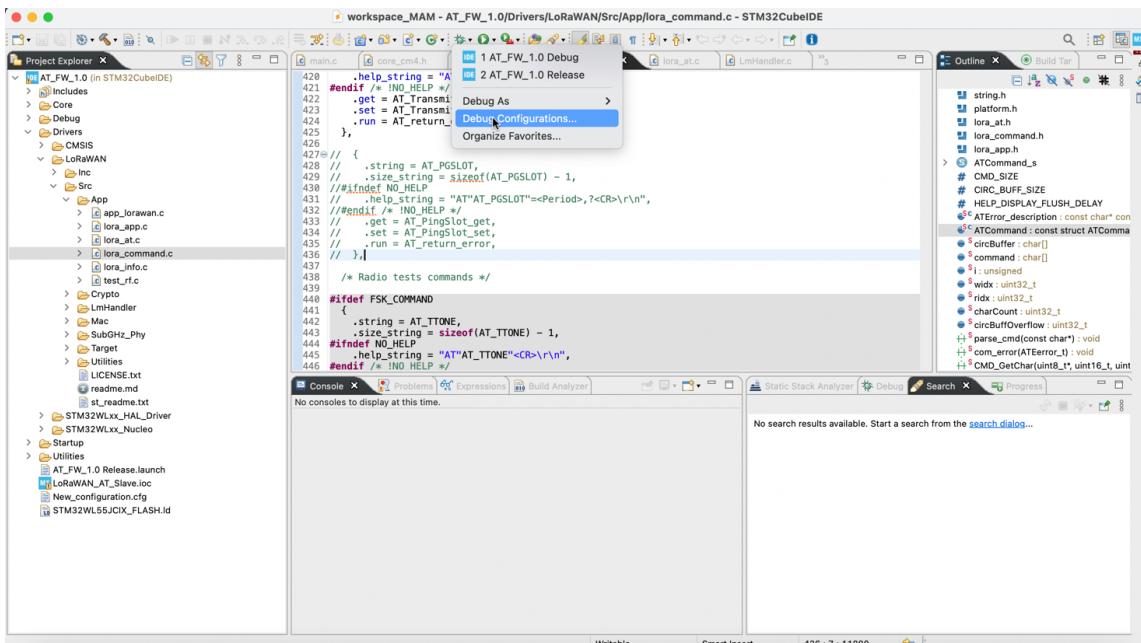


Figure 10, Debug configuration example (1)

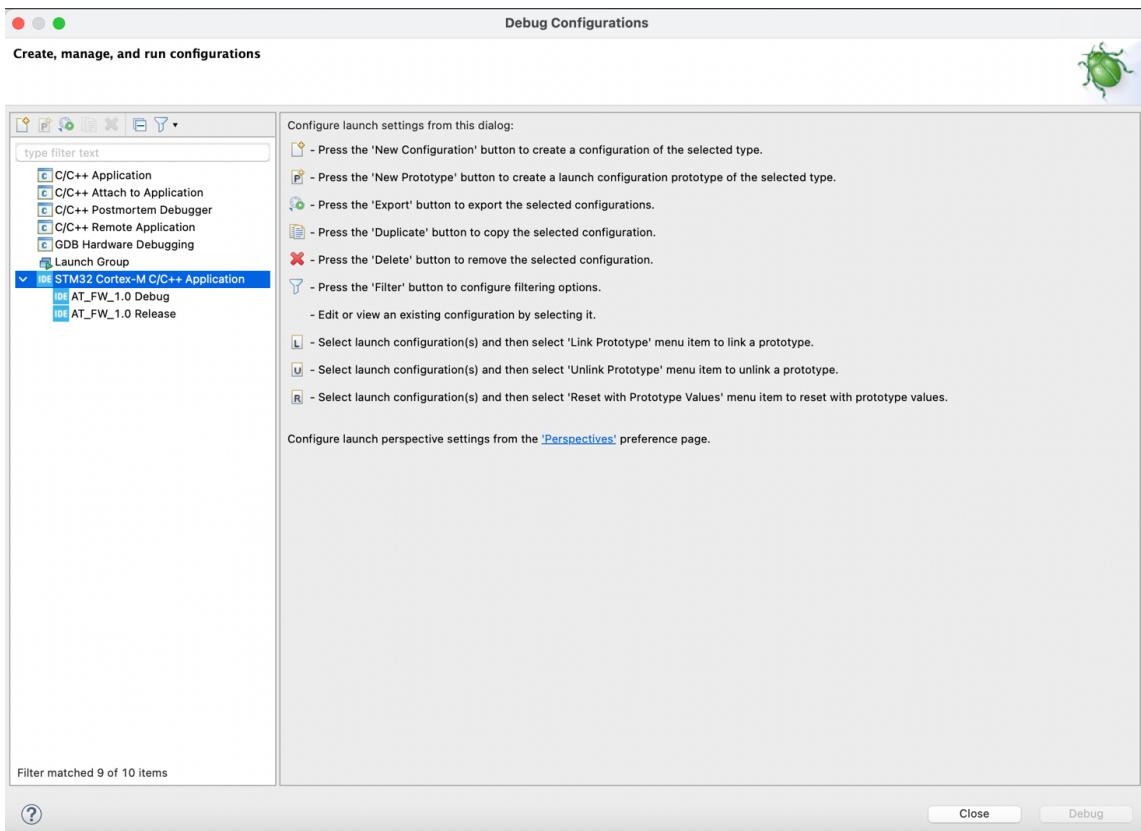


Figure 11, Debug configuration example (2)

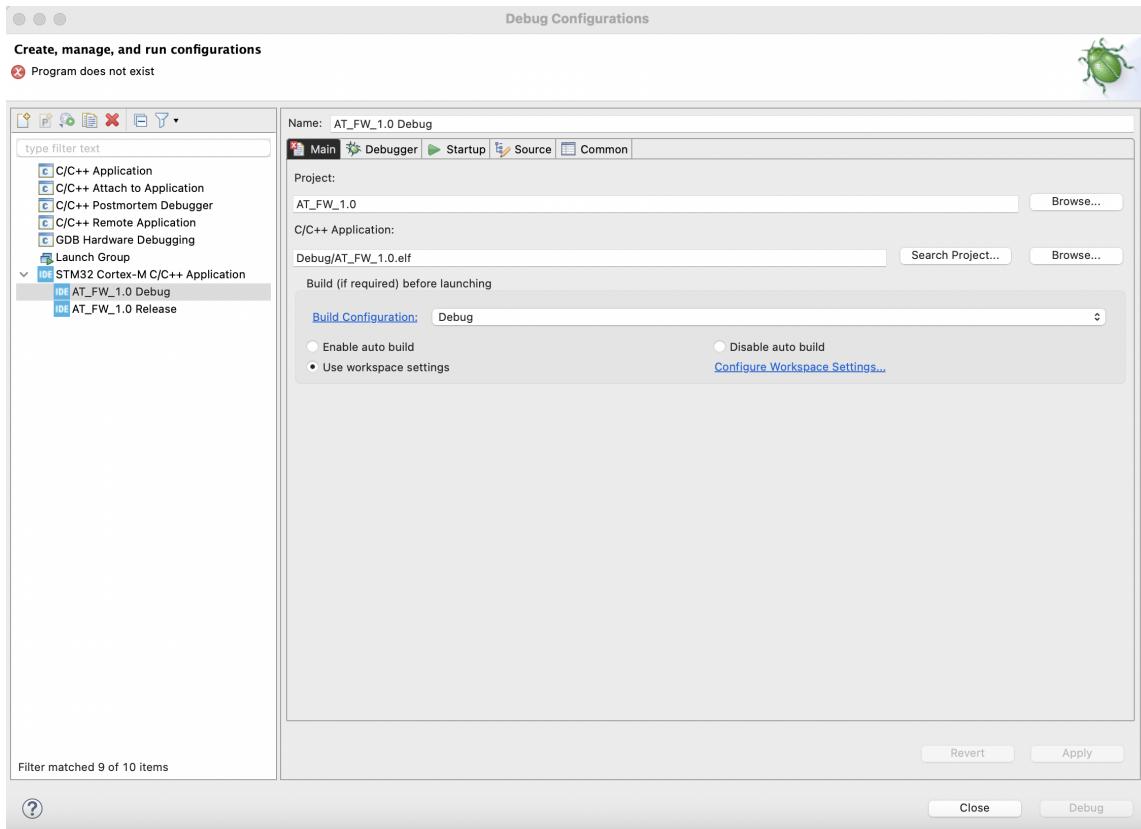


Figure 12, Debug configuration example (3)

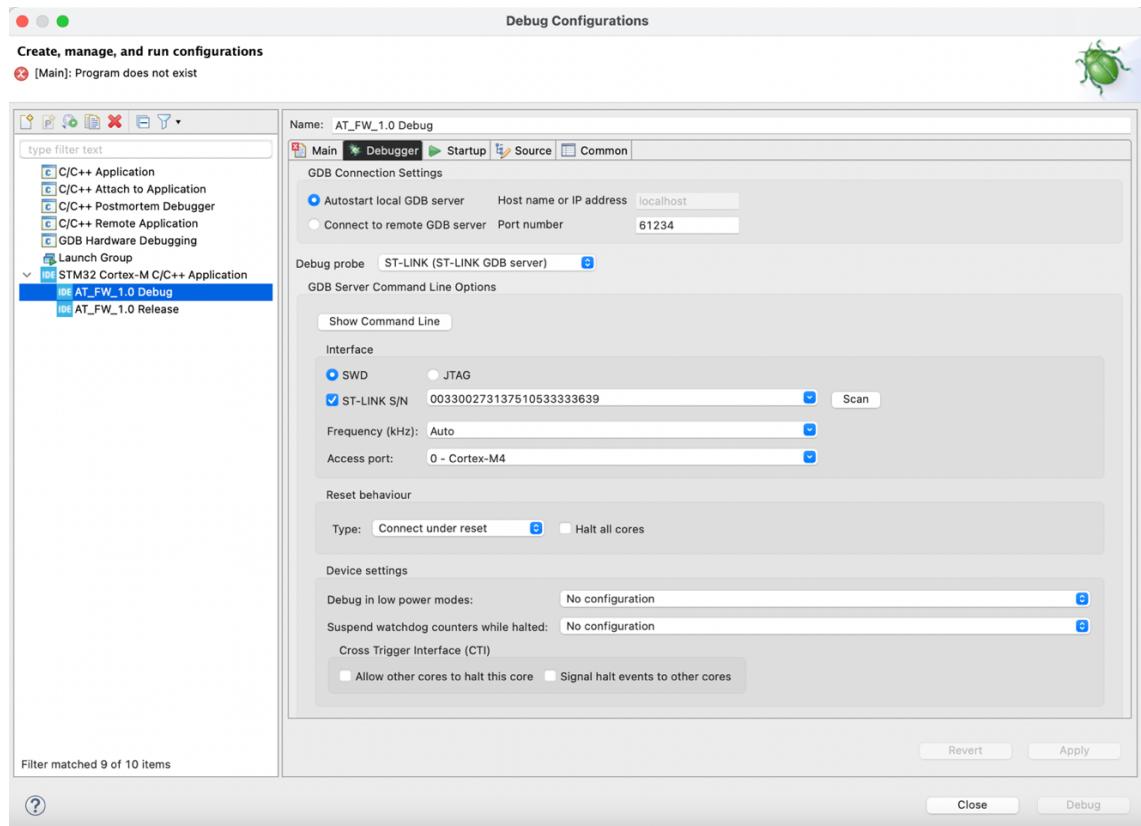


Figure 13, Debug configuration example (4)

## AT COMMAND

The values and the messages returned by the node, in the tables below any command, are examples. In almost every command there is an examples table, in which there are the messages returned by the node in release and debug mode.

### ATZ

Reset the whole system including radio and microprocessor. The node will return "start" message (in release mode, in debug mode it adds the principal key of the node, i.e. region, device eui, application keys, network key and application key).

Example:

ATZ	COMMAND RETURN VALUE
RELEASE MODE	Start
DEBUG MODE	Start ### REGION = US915 (value: 8) ### DEV_EUI = 00:00:00:00:00:00:01 ### APP_EUI = 00:00:00:00:00:00:0F ### APP_KEY = 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:0D ### NET_KEY = 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:0D

### AT?

If the flag "NO\_HELP" isn't set and the node is in debug mode, the node will return all commands with examples. If the user wish to set the above flag, or any flag, from STM32CubelDE, from "Project Explorer", right click on the active project, click on "Properties", from the drop-down menu on the left go to "C/C++ Build", from this menu select "Settings", it will open another window with a drop down menu, from which select "MCU GCC Compiler", so "Preprocessor" and, on the window that will open, add the flag "NO\_HELP" on "Define symbols".

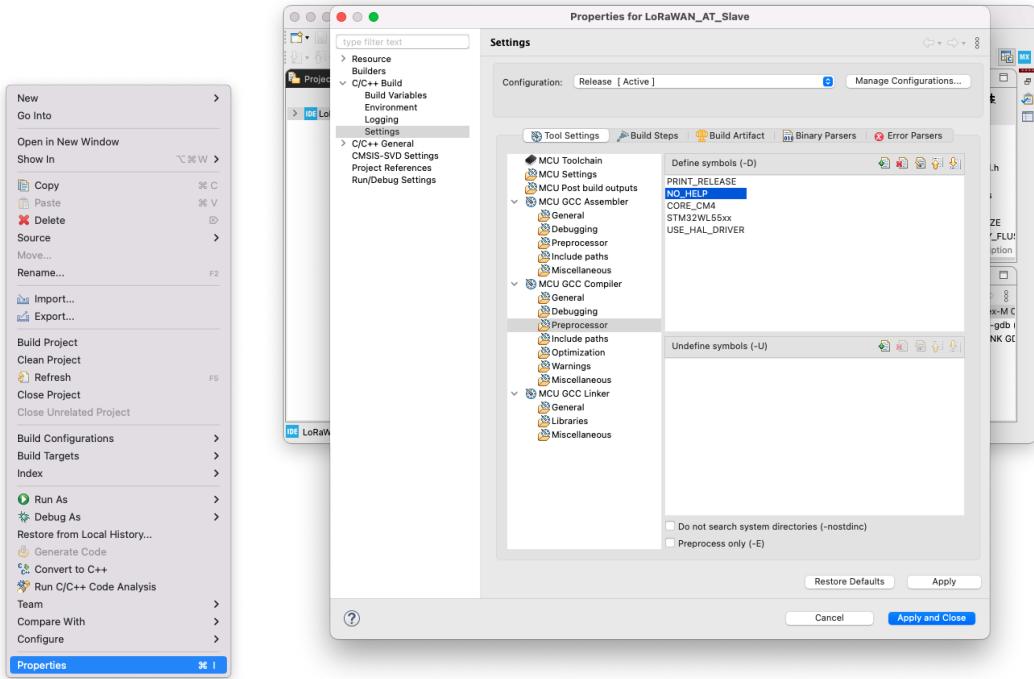


Figure 14, flag setting

## AT+TXP=Power Index

Sets the transmission power index. Admitted values: 0 **max** (16dBm), 7 **min** (2dBm) for **EU868 band**, 0 **max** (30dBm), 14 **min** (2dBm) for both **US915 and AU915 bands**. If the user sends a wrong power value, which depends on the chosen region, the node will return "c1" error ("AT\_ERROR" in debug mode), if the command is successful, the node will return "c0" ("AT\_OK" in debug mode). To check it digit "AT+TXP=?". It is also settable through "AT+CFGSEND" or "AT+TCONF"

Example:

AT+TXP=0	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	AT_OK
AT+TXP=?	COMMAND RETURN VALUE
RELEASE MODE	0
DEBUG MODE	0 AT_OK

## AT+BAND=LoRa Region

Sets the desired band region (**1** for **AU915**, **5** for **EU868**, **8** for **US915**), to check it digit "AT+BAND=?". When the band region on the device is changed it is entirely reset. If the user sends a wrong value the node will return "c1" error ("AT\_ERROR" in debug mode), if the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+BAND=5	COMMAND RETURN VALUE
RELEASE MODE	Start
DEBUG MODE	Start ### REGION = EU868 (value: 5) ### DEV_EUI = 00:00:00:00:00:00:00:01 ### APP_EUI = 00:00:00:00:00:00:00:0F ### APP_KEY = 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:0D ### NET_KEY = 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:0D
AT+BAND=?	COMMAND RETURN VALUE
RELEASE MODE	5 c0
DEBUG MODE	5 AT_OK

! **Warning:** It is not possible to modify the band region if the device **has already joined the network**, in that case it must be reset before. In the actual version of the FW it is possible to set only the US915, AU915 and EU868 bands. The band configuration holds even if the device is reset.

#### AT+DEUI=Device EUI

Sets the device EUI, to check it digit "AT+DEUI=?". If:

- the user enters more than 8 fields, or the format is wrong, the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+DEUI=00:00:00:00:00:00:00:01	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	AT_OK
AT+DEUI=?	COMMAND RETURN VALUE
RELEASE MODE	00:00:00:00:00:00:00:01 c0
DEBUG MODE	00:00:00:00:00:00:00:01 AT_OK

! **Warning:** This key holds even if the device is reset. It is not possible to modify this key if the device **has already joined the network**, in that case it must be reset before.

## AT+APPEUI=Application Key

Sets the application EUI for the join. APPEUI and JOIN\_EUI are used as synonyms in the FW and documentation. To check it digit "AT+APPEUI=?". If:

- the user enters more than 8 fields, or the format is wrong, the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+APPEUI=00:00:00:00:00:00 :00:0F	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	AT_OK
AT+APPEUI=?	COMMAND RETURN VALUE
RELEASE MODE	00:00:00:00:00:00:00:0F c0
DEBUG MODE	00:00:00:00:00:00:00:0F AT_OK



**Warning:** This key holds even if the device is reset. It is not possible to modify this key if the device **has already joined the network**, in that case it must be reset before.

## AT+NWKKEY=Network Key

Sets the device network key. To check it digit "AT+NWKKEY=?". If:

- the user enters more than 16 fields, or the format is wrong, the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

<b>AT+NWKKEY=00:00:00:00: 00:00:00:00:00:00:00:00: 0:0D</b>	<b>COMMAND RETURN VALUE</b>
RELEASE MODE	c0
DEBUG MODE	AT_OK
<b>AT+NWKKEY=?</b>	<b>COMMAND RETURN VALUE</b>
RELEASE MODE	00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:0D c0
DEBUG MODE	00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:0D AT_OK



**Warning:** This key holds even if the device is reset. It is not possible to modify this key if the device **has already joined the network**, in that case it must be reset before.

## AT+APPKEY=Application Key

Sets the device network key. To check it digit "AT+ APPKEY=?". If:

- the user enters more than 16 fields, or the format is wrong, the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

<b>AT+APPKEY=00:00:00:00:00:00: 0:00:00:00:00:00:00:00:00: 0D</b>	<b>COMMAND RETURN VALUE</b>
RELEASE MODE	c0
DEBUG MODE	AT_OK
<b>AT+APPKEY=?</b>	<b>COMMAND RETURN VALUE</b>
RELEASE MODE	00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:0D c0
DEBUG MODE	00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:0D AT_OK



**Warning:** This key holds even if the device is reset. It is not possible to modify this key if the device **has already joined the network**, in that case it must be reset before.

## AT+JOIN=Join type

Starts the joining process to the network, it can be of two types, ABP "Application By Personalization" (0) or OTAA "Over The Air Activation" (1). It must be performed before any data send/receiving command on the LoRaWAN standard. If:

- the user enters more than a field, or the field is wrong, the node will return "c1" error ("AT\_ERROR" in debug mode);
- a join was already performed the node will return "j2" message ("You have already joined" in debug mode);
- the join fail the node will return "j1" message ("+EVT:JOIN FAILED" in debug mode);
- the join goes well the node will return "j0" message ("+EVT:JOINED" in debug mode) only if the join type is OTAA, otherwise the network server doesn't respond to the join request so the node won't return any message if the join goes well;
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+JOIN=1 (OTAA join type)	COMMAND RETURN VALUE
RELEASE MODE	c0 j0
DEBUG MODE	TX on freq <FREQUENCY_TX> Hz at DR <DATARATE_TX> AT_OK RX_1 on freq <FREQUENCY_RX> Hz at DR <DATARATE_RX> +EVT:JOINED

## AT+CHANNEL=Mask0:Mask1:Mask2:Mask3:Mask4:Mask5:Mask6:Mask7:Mask8:Mask9:Mask10:Mask11

Sets the masks related to the random channel chosen during LoRaWAN transmission, only for AU and USA bands, showed below.

US915 uplink Channels									
902.3	902.5	902.7	902.9	903.1	903.3	903.5	903.7	Channels0 to 7	125kHz DR0 to DR3 4/5
903.9	904.1	904.3	904.5	904.7	904.9	905.1	905.3	Channels8 to 15	
905.5	905.7	905.9	906.1	906.3	906.5	906.7	906.9	Channels16 to 23	
907.1	907.3	907.5	907.7	907.9	908.1	908.3	908.5	Channels24 to 31	
908.7	908.9	909.1	909.3	909.5	909.7	909.9	910.1	Channels32 to 39	
910.3	910.5	910.7	910.9	911.1	911.3	911.5	911.7	Channels40 to 47	
911.9	912.1	912.3	912.5	912.7	912.9	913.1	913.3	Channels48 to 55	
913.5	913.7	913.9	914.1	914.3	914.5	914.7	914.9	Channels56 to 63	
903	904.6	906.2	907.8	909.4	911	912.6	914.2	Channels64 to 71	500kHz DR4

Figure 15, US link channels

AU915 uplink Channels									
915.2	915.4	915.6	915.8	916	916.2	916.4	916.6	Channels0 to 7	125kHz DR0 to DR5 4/5
916.8	917	917.2	917.4	917.6	917.8	918	918.2	Channels8 to 15	
918.4	918.6	918.8	919	919.2	919.4	919.6	919.8	Channels16 to 23	
920	920.2	920.4	920.6	920.8	921	921.2	921.4	Channels24 to 31	
921.6	921.8	922	922.2	922.4	922.6	922.8	923	Channels32 to 39	
923.2	923.4	923.6	923.8	924	924.2	924.4	924.6	Channels40 to 47	
924.8	925	925.2	925.4	925.6	925.8	926	926.2	Channels48 to 55	
926.4	926.6	926.8	927	927.2	927.4	927.6	927.8	Channels56 to 63	
915.9	917.5	919.1	920.7	922.3	923.9	925.5	927.1	Channels64 to 71	500kHz DR6

Figure 16, AU link channels

A join before this command is **mandatory**, if the join didn't take place the node will return "rj!", then "cl" error (in release mode, "You have to join before" and "AT\_ERROR" in debug mode).

The user **can't set less than three channels**, for the first 64 channels (8 bytes) available of AU and US bands, to place the command, due to dwell time restrictions. The default masks are, for US and AU bands, FF00:0000:0000:0000:8000:0000.

These masks, with the choosable channels, are visible with "AT+CHANNEL=?", with the following format:  
Channel id:DR max min:Band:Frequency.

**Channel id:** channel identifier;

**DR max min:** possible data rate assignable to the channel.

**Band:** subset of the channel.

**Frequency:** frequency of the channel.

The channels, in AU and USA bands, are 72 (9 bytes), so the mask that really choose the channels are from **mask0 to mask8**, see below.

ChMaskCntl	ChMask applies to
0	Channels 0 to 15
1	Channels 16 to 31
2	Channels 32 to 47
3	Channels 48 to 63
4	Channels 64 to 71
5	8LSBs controls Channel Blocks 0 to 7 (8MSBs are RFU)

Figure 17, bytes channel indexes

Example:

<b>AT+CHANNEL=00:FF:00:00:00: 00:00:00:00:03:00:00</b>	<b>COMMAND RETURN VALUE</b>
<b>RELEASE MODE</b>	c0
<b>DEBUG MODE</b>	AT_OK
<b>AT+CHANNEL=? (with EU868 band)</b>	<b>COMMAND RETURN VALUE</b>
<b>RELEASE MODE</b>	0:5_0:1:868100000 1:5_0:1:868300000 2:5_0:1:868500000 3:0_0:0 4:0_0:0 5:0_0:0 6:0_0:0 7:0_0:0 8:0_0:0 9:0_0:0 10:0_0:0 11:0_0:0 12:0_0:0 13:0_0:0 14:0_0:0 15:0_0:0 E000 c0 (see below)
<b>DEBUG MODE</b>	Channel id 0 Dr max-min 5-0 Band 1 Freq 868100000 Channel id 1 Dr max-min 5-0 Band 1 Freq 868300000 Channel id 2 Dr max-min 5-0 Band 1 Freq 868500000 Channel id 3 Dr max-min 0-0 Band 0 Freq 0 Channel id 4 Dr max-min 0-0 Band 0 Freq 0 Channel id 5 Dr max-min 0-0 Band 0 Freq 0 Channel id 6 Dr max-min 0-0 Band 0 Freq 0 Channel id 7 Dr max-min 0-0 Band 0 Freq 0 Channel id 8 Dr max-min 0-0 Band 0 Freq 0 Channel id 9 Dr max-min 0-0 Band 0 Freq 0 Channel id 10 Dr max-min 0-0 Band 0 Freq 0 Channel id 11 Dr max-min 0-0 Band 0 Freq 0 Channel id 12 Dr max-min 0-0 Band 0 Freq 0 Channel id 13 Dr max-min 0-0 Band 0 Freq 0 Channel id 14 Dr max-min 0-0 Band 0 Freq 0 Channel id 15 Dr max-min 0-0 Band 0 Freq 0 Mask E000 AT_OK (see below)

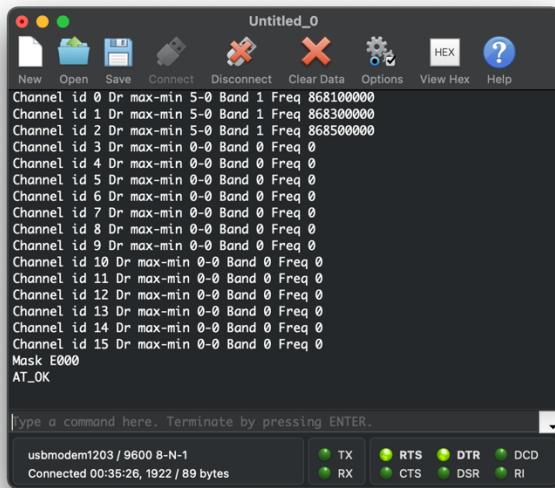


Figure 18 Debug mode

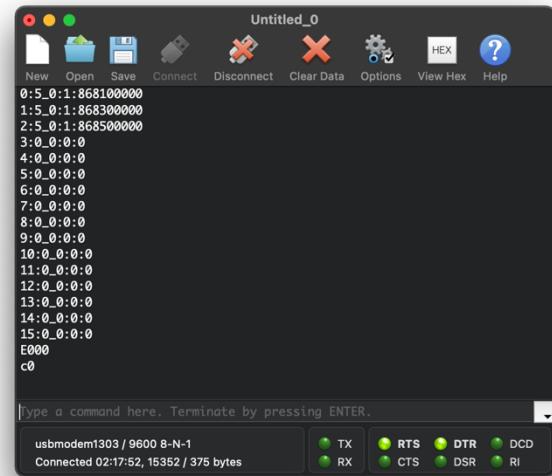


Figure 19 Release mode

### AT+ADR=ADR type

Activate or not the ADR. To check it digit "AT+ADR=?". 0 = OFF, 1 = ON, default = 1. If:

- the user enters more than a field, or the value is wrong, the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+ADR=1	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	AT_OK
AT+ADR=?	COMMAND RETURN VALUE
RELEASE MODE	1 c0
DEBUG MODE	1 AT_OK

**AT+DR=Data Rate**

Sets the data rate of the transmission. It is performable only if the ADR is **disabled**. To check it digit "AT+DR=?". It is also settable through "AT+CFGSEND", which automatically disables the ADR. Admitted values: 0-6 (EU868), 0-4 (US915), 0-6 (AU915). Lower data rates mean higher time on air, lower bit rate and higher reliability. If:

- the user enters more than a field, or the value is wrong, the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+DR=5	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	AT_OK
AT+DR=?	COMMAND RETURN VALUE
RELEASE MODE	5 c0
DEBUG MODE	5 AT_OK

## AT+CW=Timeout:Frequency:Power level

Sends a continuous tone, for “timeout” seconds”, at “frequency” frequency (100 Hz of resolution), in Hz or MHz, at “Power level” dBm power. The power ranges are **2-16 dBm** for **EU868 band** and **2-30 dBm** for **AU915** and **US915 bands**.

If:

- the user enters more than 2 fields, or the values are wrong, the node will return “c1” error (“AT\_ERROR” in debug mode);
- the command is successful, the node will return “c0” (“AT\_OK” in debug mode).

Example:

AT+CW=100:870:15 (with EU868 band)	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	Setting: Timeout = 100 sec   Frequency = 870 MHz   Power level = 15 dBm AT_OK

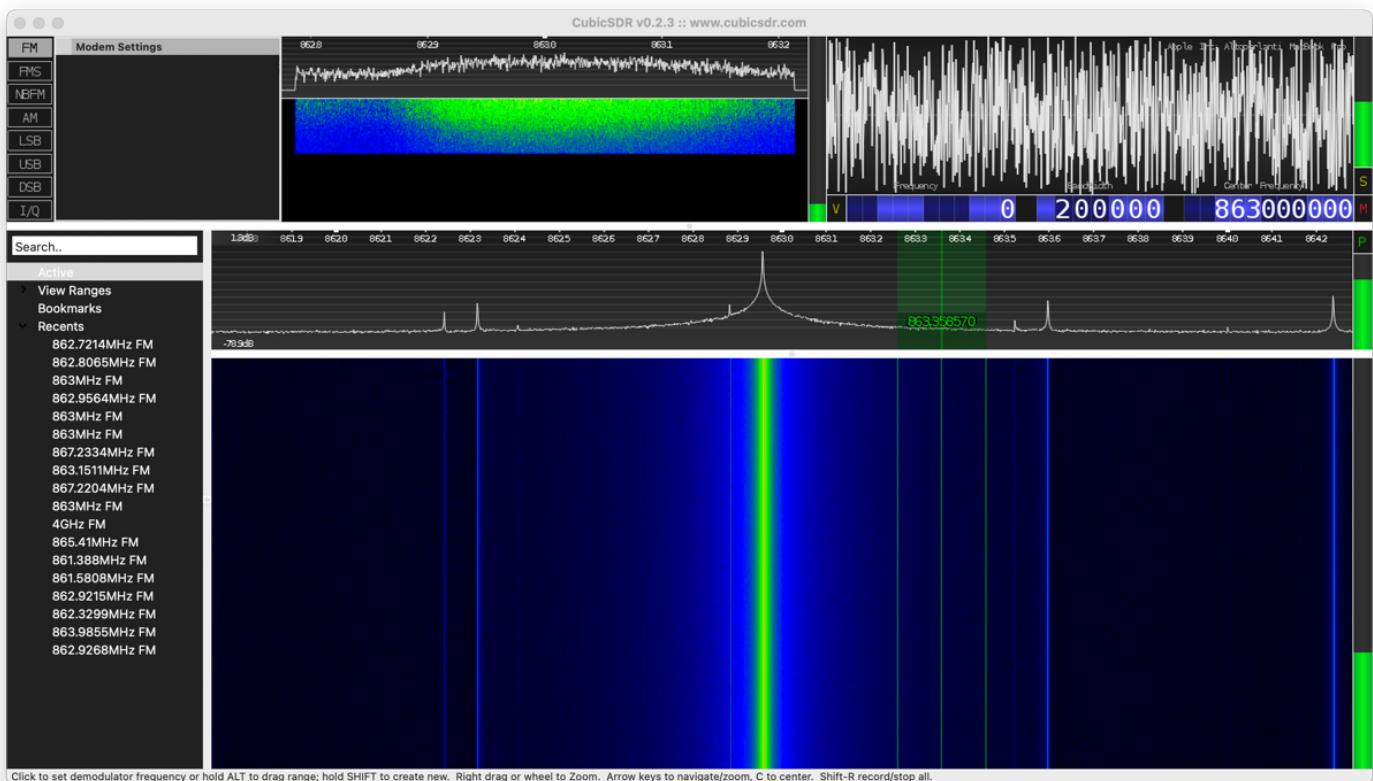


Figure 20, AT+CW example in CubicSDR



**Warning:** This mode can be used **only after the join** and is valid on the EU868 (863-870MHz), AU915 (915-928MHz) and US915 bands (902-928MHz).

**AT+STOPCW**

Interrupts instantaneously the CW mode. If the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+STOPCW	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	AT_OK

**AT+CFGSEND=Port:Number of send:Ack:Send timer:Datarate:Power index**

Sets "Port" port, "Number of send" number of packets to send, ACK or not (1 if yes, 0 otherwise), packet's delay "Send timer", in ms, datarate "datarate" and "Power index" power, in dBm (from 0-7 for EU868 band, 0-14 for AU915 and US915 bands). If:

- the user enters more than 6 fields, or the values are wrong, the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

To check it digit "AT+CFGSEND=?", the node will return Port:Number of send:Ack:Send timer:Datarate:Power index:Maximum size of packages, with the "Maximum size of packages" field automatically chosen by the node depending on the region and the datarate.

Example:

AT+CFGSEND=10:5:0:100 00:5:0	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	AT_OK
AT+CFGSEND=?	COMMAND RETURN VALUE
RELEASE MODE	10:5:0:10000:4:0:222 c0
DEBUG MODE	Settings: Port = 10   Number of packet send = 5   ACK = 0   Send timer = 10000   Data rate = 4   Power level = 0   Maximum size of packages = 222 Bytes AT_OK



**Warning:** The first packet send will be performed immediately after the command insertion, while the "Send timer" means the packet's interval transmission **after closing** the two receiving windows (RX1DL and RX2DL), which time can be set with AT+RX1DL and AT+RX2DL commands. By default, the sum of these two times is equal to approximately **four seconds**, so if the user set "send timer" equal 1000 the packet's interval transmission will be **five seconds**. If the datarate is **4**, with USA band, or **6**, with AU band, the user **can't set a "Send timer" lower than 10 seconds (10000)**, for Dwell time limitation. If the network can manage more than one channel with these tuple datarate-band, this limitation **is removed** only if the user adds **at least another 2 channels**, so the total channel **will be at least 3**. These checking and setting can be performed thanks to the command "AT+CHANNEL".

**AT+SEND=Payload Length:Payload raw**

Sends packets of data (raw in hex) of "Payload Length" bytes dimension, containing "Payload", in casual channels. If "Payload"s dimension is smaller than "Payload Length" the remaining bytes will be filled with zeros, otherwise if is greater the node will return "c1" ("AT\_ERROR" in debug mode). If:

- the user enters more than 2 fields, or the values are wrong, the node will return "c1" error ("AT\_ERROR" in debug mode);

- a join before this command is **mandatory**, if the join didn't take place the node will return "rj1", then "c1" error (in release mode, "You have to join before" and "AT\_ERROR" in debug mode);
- the user try to do a send while another one is already in place the node will return "c1" error (in release mode, "AT\_ERROR" in debug mode);
- the ack doesn't go well the node will return "ack1";
- the send doesn't go well (e.g. gateway not in radio coverage) the node will return "s1";
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode);
- the command is successful, but the send won't start the node will return "r1" and it will retry the send, otherwise use AT+STOPSEND.

Example:

AT+SEND=2:FF00(sen ds 0xFF00, with ACK)	COMMAND RETURN VALUE
<b>RELEASE MODE</b>	c0 ns=<packet number> 1:10:-56:9 ... s0 1:10:-56:9
<b>DEBUG MODE</b>	AT_OK SEND REQUEST:<number of send went well, total number of send> +EVT:RX_1, DR 10, RSSI -56, SNR 9 ... Finish or stop send
AT+SEND=2:FF00(sen ds 0xFF00, without ACK)	COMMAND RETURN VALUE
<b>RELEASE MODE</b>	c0 ns=#packet number ... s0
<b>DEBUG MODE</b>	AT_OK TX on freq <FREQUENCY_TX> Hz at DR <DATARATE_TX> SEND REQUEST: <INDEX OF ACTUAL SEND, TOTAL NUMBER OF SEND> RX_1 on freq <FREQUENCY_RX> Hz at DR <DATARATE_RX> RX_2 on freq <FREQUENCY_RX> Hz at DR <DATARATE_RX> ... Finish or stop send



**Warning:** Everytime the user performs a send, two receiving windows are opened so the user can receive a downlink packet. If the user receives one downlink then the node will return "rx:<Port>:<Payload length>:<Payload raw><CR><RX window index>:<Data rate>:<RSSI>:<SNR>".

**AT+STOPSEND**

Interrupts a flow of send (AT+SEND command), if a send is still performing the node will finish it. If the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+STOPSEND	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	AT_OK

**AT+VL=verbosity level:timestamp flag**

Sets the verbosity level of the application (0 OFF, 3 high level) and a flag for turning on timestamps into some messages returned from the node (0 OFF, 1 ON). The timestamp is in the format <Seconds>s<Thousandths>, while the time measured is from the last reset. The verbosity levels are:

- VLEVEL\_H (3), concerns application data print (i.e., they are present in lora\_app.c);
- VLEVEL\_M (2), turn on radio and middleware prints;
- VLEVEL\_L (1), turn on peripheral prints;
- VLEVEL\_ALWAYS, some print are always printed.

To check it digit "AT+VL=?". If:

- the user enters more than two field or the values are wrong the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+VL=3:1	COMMAND RETURN VALUE
RELEASE MODE	9s150:c0
DEBUG MODE	AT_OK
AT+VL=?	COMMAND RETURN VALUE
RELEASE MODE	9s150:3:1 c0
DEBUG MODE	9s150:3:1 AT_OK

**AT+VER=?**

Show the FW version.

Example:

AT+VER=?	COMMAND RETURN VALUE
RELEASE MODE	VER:1.0 c0
DEBUG MODE	FW_VERSION:1.0 AT_OK

**AT+DADDR=device address**

Sets the device address, **before a join**, in ABP join mode, otherwise (OTAA mode) is set randomly. To check it digit "AT+DADDR=?". If:

- the user enters more than 4 fields, or the values are wrong, the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+DADDR=01:02:0A:0B	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	AT_OK
AT+DADDR=?	COMMAND RETURN VALUE
RELEASE MODE	01:02:0A:0B c0
DEBUG MODE	01:02:0A:0B AT_OK

**AT+DCS=value**

Activate or not duty cycle (0 disabled, 1 enabled). To check it digit "AT+DCS=?". If:

- the user enters more than one field, or the value is wrong, the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+DCS=1	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	AT_OK
AT+DCS=?	COMMAND RETURN VALUE
RELEASE MODE	1 c0
DEBUG MODE	1 AT_OK

**AT+RX2FQ=frequency**

Sets the frequency of the second RX window in Hz. To check it digit "AT+RX2FQ=?". If:

- the user enters more than one field, or the value is wrong, the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

<b>AT+RX2FQ=869525000 (with EU868 band)</b>	<b>COMMAND RETURN VALUE</b>
<b>RELEASE MODE</b>	c0
<b>DEBUG MODE</b>	AT_OK
<b>AT+RX2FQ=? (with EU868 band)</b>	<b>COMMAND RETURN VALUE</b>
<b>RELEASE MODE</b>	869525000 c0
<b>DEBUG MODE</b>	869525000 AT_OK

**AT+RX2DR=Datarate**

Sets the DR of the second RX window (min=0, max=7 for **EU868**, min=8, max=13 for **AU915** and **US915**). To check it digit "AT+RX2DR=?". If:

- the user enters more than one field, or the value is wrong, the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+RX2DR=1	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	AT_OK
AT+RX2DR=?	COMMAND RETURN VALUE
RELEASE MODE	1 c0
DEBUG MODE	1 AT_OK

## AT+RX1DL=delay

Sets the delay of the first RX window in ms. Default value = 1000ms. To check it digit "AT+RX1DL=?". If:

- the user enters more than one field, or the value is wrong, the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+RX1DL=1000	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	AT_OK
AT+RX1DL=?	COMMAND RETURN VALUE
RELEASE MODE	1000 c0
DEBUG MODE	1000 AT_OK



**Warning:** This command must be carefully set because **it must be equal to the same one in the network server**. The network server sends this parameter to the node after the join, so we recommend to leave the default one.

## AT+RX2DL=delay

Sets the delay of the second RX window in ms. Default value = 2000ms. To check it digit "AT+RX2DL=?". If:

- the user enters more than one field, or the value is wrong, the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+RX2DL=2000	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	AT_OK
AT+RX2DL=?	COMMAND RETURN VALUE
RELEASE MODE	2000 c0
DEBUG MODE	2000 AT_OK



**Warning:** This command must be carefully set because **it must be equal to the same one in the network server**. The network server sends this parameter to the node after the join, so we recommend to leave the default one.

### AT+JNIDL=delay

Sets the join delay, in ms, of the first RX window. Default value = 5000ms. To check it digit "AT+JNIDL=?".

If:

- the user enters more than one field, or the value is wrong, the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+JNIDL=5000	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	AT_OK
AT+JNIDL=?	COMMAND RETURN VALUE
RELEASE MODE	5000 c0
DEBUG MODE	5000 AT_OK



**Warning:** This command must be carefully set because **it must be equal to the same one in the network server**, so we recommend to leave the default one.

**AT+JN2DL=delay**

Sets the join delay, in ms, of the second RX window. Default value = 6000ms. To check it digit "AT+JN2DL=?". If:

- the user enters more than one field, or the value is wrong, the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+JN2DL=6000	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	AT_OK
AT+JN2DL=?	COMMAND RETURN VALUE
RELEASE MODE	6000 c0
DEBUG MODE	6000 AT_OK



**Warning:** This command must be carefully set because **it must be equal to the same one in the network server**, so we recommend to leave the default one.

**AT+NWKID=network ID**

Sets the network ID (from 0 to 127), **before a join**, in ABP join mode, otherwise (OTAA mode) is set randomly. To check it digit "AT+NWKID=?". If:

- the user enters more than one field, or the value is wrong, the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+NWKID=0	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	AT_OK
AT+NWKID=?	COMMAND RETURN VALUE
RELEASE MODE	0 c0
DEBUG MODE	0 AT_OK

**AT+CLASS=class value**

Sets the LoRa class (A or C, B is unused) To check it digit "AT+CLASS=?". If:

- the user enters more than one field, or the value is wrong, the node will return "c1" error ("AT\_ERROR" in debug mode);
- a join before this command is **mandatory**, if the join didn't take place the node will return "rj1", then "c1" error (in release mode, "You have to join before" and "AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+CLASS=A	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	AT_OK
AT+CLASS=?	COMMAND RETURN VALUE
RELEASE MODE	A c0
DEBUG MODE	A AT_OK

**AT+LTIME=?**

Shows the local time in GPS POSIX-UNIX UTC format without any correction.

Example:

AT+LTIME=?	COMMAND RETURN VALUE
RELEASE MODE	01/01/1970 00:15:20 c0
DEBUG MODE	01/01/1970 00:15:20 AT_OK

### AT+TREQ

Receive the local time, in GPS POSIX-UNIX UTC format, from the gateway server, without applying any correction. A join before this command is **mandatory**, if the join didn't take place the node will return "rj", then "cl" error (in release mode, "You have to join before" and "AT\_ERROR" in debug mode).

Example:

AT+TREQ	COMMAND RETURN VALUE
RELEASE MODE	c0 1:10:-75:7
DEBUG MODE	AT_OK TX on freq <FREQUENCY_TX> Hz at DR <DATARATE_TX> RX_1 on freq <FREQUENCY_RX> Hz at DR <DATARATE_RX> +EVT:RX_1, DR 0, RSSI -66, SNR 7

**AT+UID**

Shows the unique ID of the device.

Example:

AT+UID=?	COMMAND RETURN VALUE
RELEASE MODE	00:80:E1:15:05:02:BF:DD c0
DEBUG MODE	00:80:E1:15:05:02:BF:DD AT_OK

**AT+NWKSKEY=network session key**

Sets the network session key, **before a join**, in ABP join mode, otherwise (OTAA mode) is set randomly. To check it digit "AT+NWKSKEY=?". If:

- the user enters more than 16 fields or the format is wrong the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+NWKSKEY=2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	AT_OK
AT+NWKSKEY=?	COMMAND RETURN VALUE
RELEASE MODE	2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C c0
DEBUG MODE	2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C AT_OK

## AT+APPSKEY=application session key

Sets the application session key, **before a join**, in ABP join mode, otherwise (OTAA mode) is set randomly. To check it digit "AT+APPSKEY=?". If:

- the user enters more than 16 fields or the format is wrong the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

<b>AT+APPSKEY=2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C</b>	<b>COMMAND RETURN VALUE</b>
<b>RELEASE MODE</b>	c0
<b>DEBUG MODE</b>	AT_OK
<b>AT+APPSKEY=?</b>	<b>COMMAND RETURN VALUE</b>
<b>RELEASE MODE</b>	2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C c0
<b>DEBUG MODE</b>	2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C AT_OK

## AT+LINKC

Transport link check MAC command request to the next uplink (piggybacking). A join before this command is **mandatory**, if the join didn't take place the node will return "rjl", then "c1" error (in release mode, "You have to join before" and "AT\_ERROR" in debug mode).

Example:

<b>AT+LINKC</b>	<b>COMMAND RETURN VALUE</b>
<b>RELEASE MODE</b>	c0
<b>DEBUG MODE</b>	AT_OK

**AT+CERTF=Join type**

Sets the module in LoRaWAN certification with join mode (0 = ABP, 1 = OTAA). The sequence of actions made by this command are:

- a join (ABP or OTAA);
- a TX, after 8 seconds, only to open the RX windows: if in that period of RX time the node receive the message "0x01010101" on the port 224 the node will enter in certification mode;
- in this mode the node, every 5 seconds, sends a packet to get command to perform the certification (for instance see reference [6]).

If the inserted join type is wrong the node will return "c1" error ("AT\_ERROR" in debug mode).

Example:

AT+CERTF=1	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	AT_OK

**AT+BAT=?**

Shows the device battery level in mV.

Example:

AT+BAT=?	COMMAND RETURN VALUE
RELEASE MODE	3269 c0
DEBUG MODE	3269 AT_OK

## AT+TTH=Start frequency:Stop frequency:frequency resolution:number of packets to send

Permits to do hopping between "start frequency" and "stop frequency" (in Hz or MHz), with "frequency resolution" (in Hz) resolution, sending "number of packets to send" packets. The maximum band of the frequency sweep is determined from the difference between "start frequency" and "stop frequency". The TX parameters can be previously set by AT+TCONF. If the number of packets to send is greater than the number of hop the remaining packets are sent restarting from "start frequency". If:

- the user enters more than 4 fields, the values are wrong, or a transmission is already in place the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+TTH=863000000:870000000:1000000:1 0	COMMAND RETURN VALUE
RELEASE MODE	c0
DEBUG MODE	AT_OK

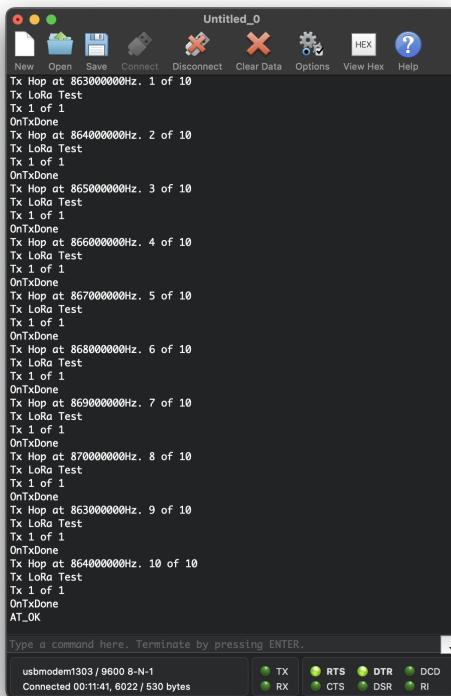


Figure 21, AT+TTH debug example



Figure 22 AT+TTH release example

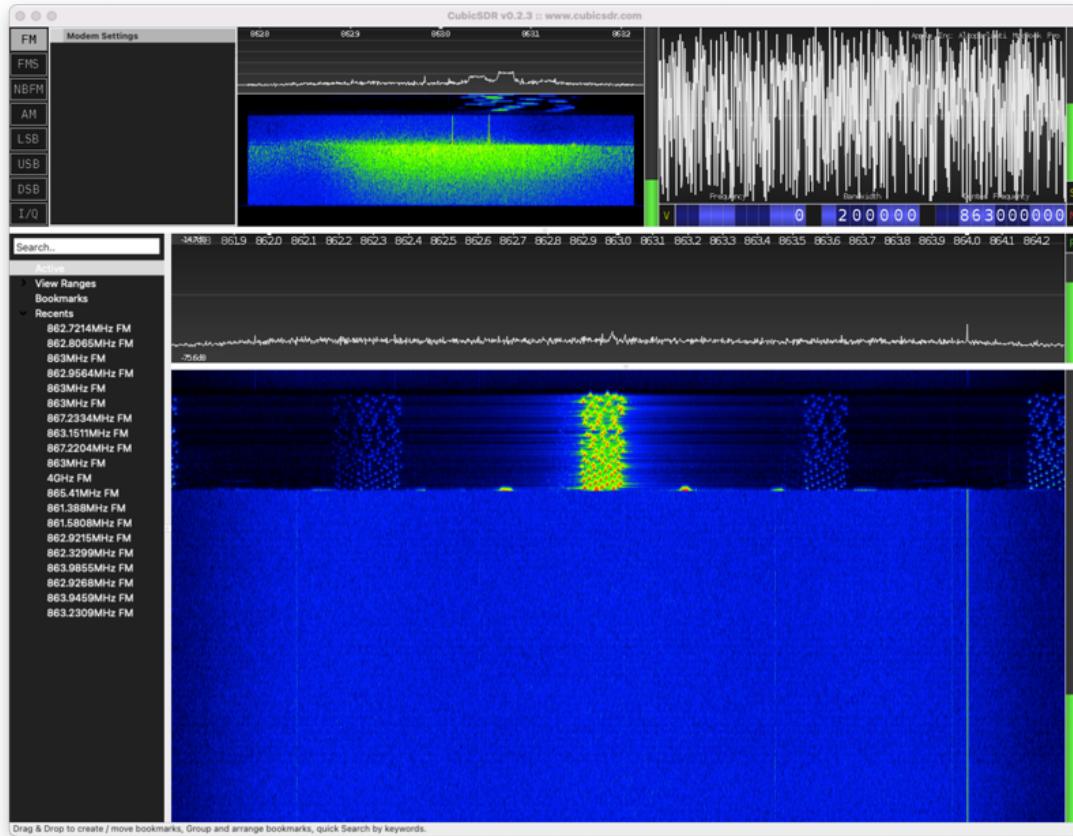


Figure 23, AT+TTH packets (showed thanks to CubicSDR)



**Warning:** After this command if we want to use AT+SEND or AT+CW commands **the device must be reset (ATZ).**

**AT+TCONF=TX frequency:TX power:BW:SF:4/CR:LNA state:Boost PA state:Modulation type:Payload length:Low DR optimization:BT FSK product**

Sets parameters **before** frequency hopping. To check it digit "AT+TCONF=?". If:

- the user enters more than 13 fields, or the values are wrong, the node will return "c1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

**TX frequency:** TX frequency expressed in Hz or MHz.

**TX power:** TX power in dBm, from -9 to 22.

**BW:** BW of the TX signal [6=500KHz, 5=250KHz, 4=125KHz].

**SF:** Spreading Factor, between 5 and 12.

**4/CR:** TX Code Rate, sets the redundancy of the message [5=4/5, 6=4/6, 7=4/7, 8=4/8].

**LNA state:** If =1 sets the RX LNA.

**PA Boost state:** If =1 sets the PA Boost.

**Modulation type:** TX modulation, in our case it must be set to 1 [0:FSK, 1:LoRa, 2:BPSK].

**Payload length:** Payload bytes length, between 1 and 222.

**Low DR optimization:** Sets the optimization for low DR in LoRa, must be set for SF11 and SF12 [0=OFF, 1=ON, 2: AUTO (i.e. 1 for SF11/SF12 and 0 otherwise)]. We recommend to set this to AUTO.

Example:

<b>AT+TCONF=868000000:14:4:12:4 /5:0:0:1:16:2</b>		<b>COMMAND RETURN VALUE</b>
<b>RELEASE MODE</b>		c0
<b>DEBUG MODE</b>		AT_OK
<b>AT+TCONF=?</b>		<b>COMMAND RETURN VALUE</b>
<b>RELEASE MODE</b>		868000000:14:4:12:4/5:0:0:1:16:2 c0
<b>DEBUG MODE</b>		1: Freq= 86800000 Hz 2: Power= 14 dBm 3: Bandwidth= 4 (=125000 Hz) 4: SF= 12 5: CR= 1 (=4/5) 6: LNA State= 0 7: PA Boost State= 0 8: modulation LORA 9: Payload len= 16 Bytes 10: LowDRopt[0 to 2]= 2

	<p>can be copy/paste in set cmd: AT+TCONF=868000000:14:4:12:4/5:0:0:1:16:2 AT_OK</p>
--	--

**AT+TTX=Send's number:Channel frequency:Spreading Factor:BW:Payload Length:Payload**

Sends "send's number" packets to "channel frequency" channel (in Hz or MHz) with "Spreading Factor" SF and "BW" band. The packets payload is specified by "Payload", and have "Payload Length" dimensions. If:

- the user enters more than 6 fields, the values are wrong, or a transmission is already in place the node will return "c1" error ("AT\_ERROR" in debug mode);
- the radio module is already busy in a send the node will return "b1" error ("AT\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+TTX=10:863:7:125:3:FABC11	COMMAND RETURN VALUE
<b>RELEASE MODE</b>	1_<send's number> ... <send's number>_<send's number> c0
<b>DEBUG MODE</b>	Tx at -- <Tx frequency> Mz -- of <send's number> packets Tx LoRa Test Tx 1 of <send's number> OnTxDone Tx LoRa Test Tx 2 of <send's number> OnTxDone ... Tx LoRa Test Tx <send's number> of <send's number> OnTxDone AT_OK



**Warning:** The TX parameters can be previously set by AT+TCONF. After this command if we want to use AT+SEND or AT+CW commands **the device must be reset (ATZ)**.

## **AT+TRX=Packet's number to receive:Channel frequency:Spreading Factor:BW**

Node waits on "channel frequency" channel (in Hz or MHz) "packet's number to receive" packets with "Spreading Factor" SF and "BW" band. The command will plot the payload received. If:

- the user enters more than 4 fields, the values are wrong, or a transmission is already in place the node will return "c1" error ("AT\_ERROR" in debug mode);
- the radio module is already busy in a receive the node will return "b1" error ("AT\_BUSY\_ERROR" in debug mode);
- the command is successful, the node will return "c0" ("AT\_OK" in debug mode).

Example:

AT+TRX=10:863:7:1 25	COMMAND RETURN VALUE
<b>RELEASE MODE</b>	<Payload> <Rssi value>:<send's number> 1_<send's number>:<PER percentage> ... payload = <Payload> <Rssi value>:<send's number> <send's number>_<send's number>:<PER percentage> c0
<b>DEBUG MODE</b>	Rx at -- <Rx frequency> Mz -- of <send's number> packets payload = <Payload> OnRxDone RssiValue=<Rssi value> dBm, SnrValue=<SNR value>dB Rx 1 of <send's number> >>> PER = <PER percentage>% payload = <Payload> OnRxDone RssiValue=<Rssi value> dBm, SnrValue=<SNR value>dB Rx 2 of <send's number> >>> PER = <PER percentage>% payload = <Payload> OnRxDone ... RssiValue=<Rssi value> dBm, SnrValue=<SNR value>dB Rx <send's number> of <send's number> >>> PER = <PER percentage>% payload = <Payload> OnRxDone AT_OK



**Warning:** The TX parameters can be previously set by AT+TCONF. After this command if we want to use AT+SEND or AT+CW, and therefore AT+JOIN, commands **the device must be reset** (ATZ).

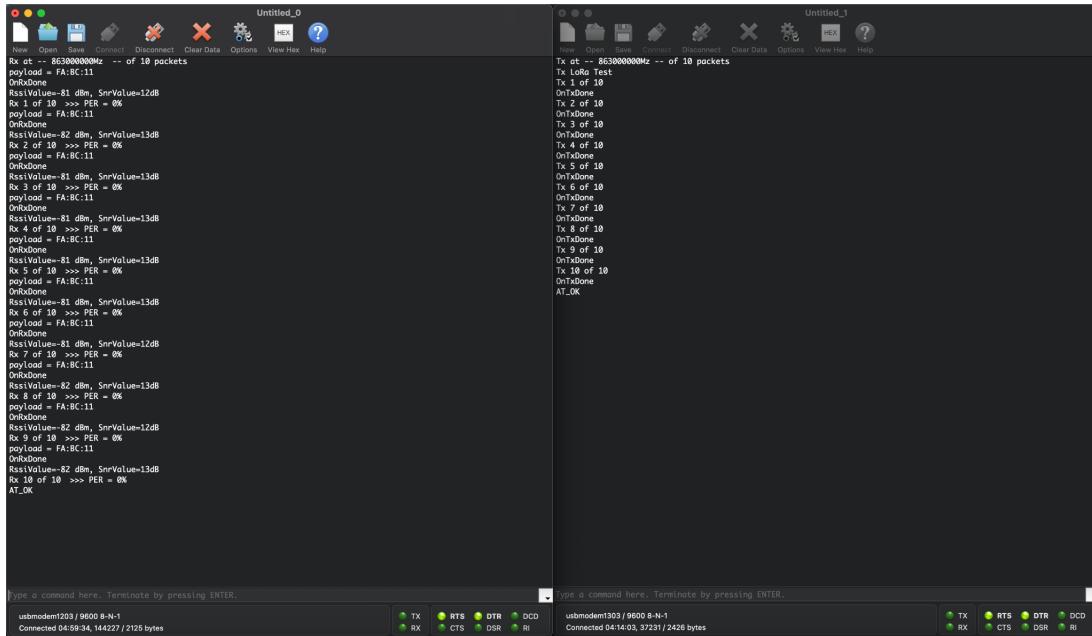


Figure 24 Example of AT+TTX and AT+TRX in debug mode

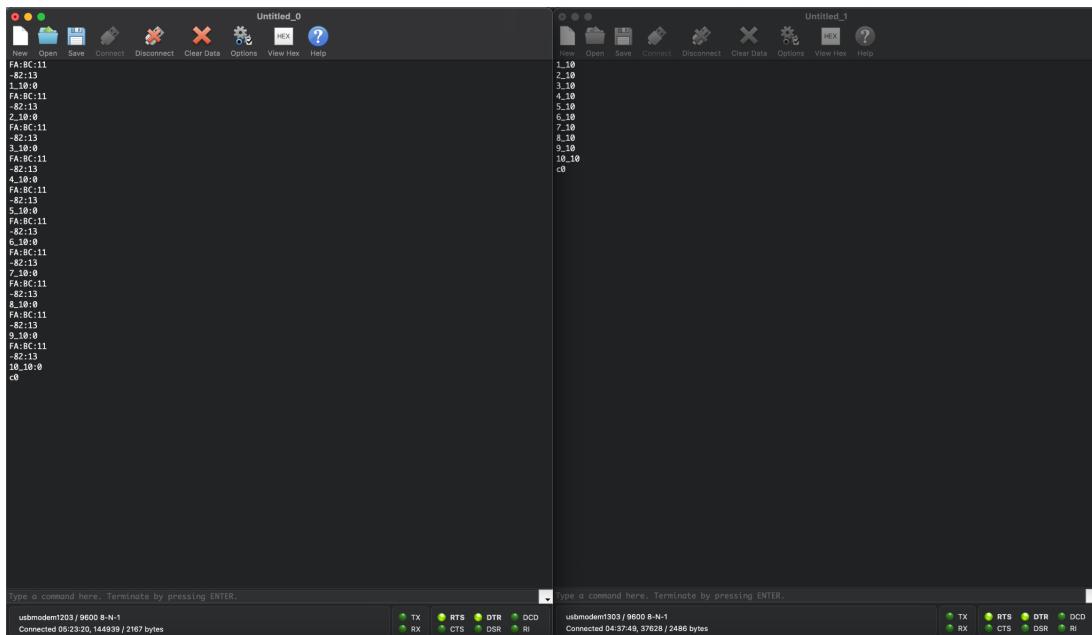


Figure 25 Example of AT+TTX and AT+TRX in release mode

## REFERENCE DOCUMENTS

1. **LoRaWAN 1.0.3 Specification by LoRa Alliance® Specification Protocol – 2018, January**  
([https://lora-alliance.org/resource\\_hub/lorawan-specification-v1-0-3/](https://lora-alliance.org/resource_hub/lorawan-specification-v1-0-3/))
2. **Application note How to build a LoRa application with STM32CubeWL (AN5406)**  
([https://www.st.com/resource/en/application\\_note/an5406-how-to-build-a-lora-application-with-stm32cubewl-stmicroelectronics.pdf](https://www.st.com/resource/en/application_note/an5406-how-to-build-a-lora-application-with-stm32cubewl-stmicroelectronics.pdf))
3. **User manual Description of STM32WL HAL and low-layer drivers (UM2642)**  
([https://www.st.com/resource/en/user\\_manual/dm00660673-description-of-stm32wl-hal-and-low-layer-drivers-stmicroelectronics.pdf](https://www.st.com/resource/en/user_manual/dm00660673-description-of-stm32wl-hal-and-low-layer-drivers-stmicroelectronics.pdf))
4. **STM32WLE5JB documentation (datasheet, manuals, application examples etc.)**  
(<https://www.st.com/en/microcontrollers-microprocessors/stm32wle5jb.html#documentation>)
5. **STM32WLE5JC documentation (datasheet, manuals, application examples etc.)**  
(<https://www.st.com/en/microcontrollers-microprocessors/stm32wle5jc.html>)
6. **LoRa alliance certification deepening**  
([https://lora-alliance.org/wp-content/uploads/2020/11/lora\\_alliance\\_certification\\_deep\\_dive.pdf](https://lora-alliance.org/wp-content/uploads/2020/11/lora_alliance_certification_deep_dive.pdf))