

# ST50

## Commands Set Reference

### for RF test

Document Name	ST50 Commands Set Reference for RF test
Version	V1.2.1
Doc No	
Date	Jul , 20, 2022

## Document History

Date	Revised Contents	Revised by	FW Version
Nov, 23, 2020	Initial version.	JC	V0.7.2
Aug ,31, 2021	Release version for Firmware v1.1.1	Yunlin	V1.1.1
Sep ,07, 2021	Add Appendix Region parameters	Yunlin	V1.1.1
Jan ,18, 2022	Update command result and reclassifications	Yunlin	V1.1.1
Mar,09, 2022	Release version for Firmware v1.1.2 and command update	Yunlin	V1.1.2
Mar,14, 2022	Correct AU915 US915 parameter table Frequency bound Rx to Tx	Yunlin	V1.1.2
Mar,25, 2022	Release version for Firmware v1.1.3 and command update	Yunlin	V1.1.3
Apr ,08, 2022	Release version for Firmware v1.1.4 and command update	Yunlin	V1.1.4
May ,12,2022	Release version for Firmware v1.2.0 and command update	Yunlin	V1.2.0
Jul,20,2022	Release version for Firmware v1.2.1 and enable Region KR920 IN865	Yunlin	V1.2.1

# Index

INDEX .....	2
RELEASE NOTE .....	5
1. CONFIGURATION.....	7
1.1 Software Configuration .....	7
2. COMMANDS SET REFERENCE .....	9
2.1 General commands.....	9
2.1.1 AT.....	9
2.1.2 AT?.....	10
2.1.3 ATZ.....	11
2.1.4 AT+VER .....	11
2.1.5 AT+FATRS .....	12
2.1.6 AT+VL.....	13
2.1.7 AT+BAT.....	14
2.1.8 AT+BR .....	15
2.2 MAC commands.....	16
2.2.1 AT+SAVE.....	16
2.2.2 AT+UID.....	17
2.2.3 AT+IEUID.....	17
2.2.4 AT+DEUI.....	18
2.2.5 AT+APPEUI.....	19
2.2.6 AT+DADDR.....	20
2.2.7 AT+APPKEY .....	21
2.2.8 AT+APPSKEY .....	22
2.2.9 AT+NWKSKEY.....	23
2.2.10 AT+JOIN .....	24
2.2.11 AT+CLASS.....	25
2.2.12 AT+BAND .....	26
2.2.13 AT+ UPDWELL.....	27
2.2.14 AT+ DWDWELL.....	28
2.2.15 AT+TXP.....	29
2.2.16 AT+SEND.....	30
2.2.17 AT+RPSEND.....	32
2.2.18 AT+ADR.....	33

2.2.19	AT+DR.....	34
2.2.20	AT+DCS.....	35
2.2.21	AT+RX1FQ.....	36
2.2.22	AT+RX1DL.....	37
2.2.23	AT+RX2DL.....	38
2.2.24	AT+RX2FQ.....	39
2.2.25	AT+RX2DR.....	40
2.2.26	AT+JN1DL.....	41
2.2.27	AT+JN2DL.....	42
2.2.28	AT+NWKID.....	43
2.2.29	AT+PGSLOT.....	44
2.2.30	AT+LINKC.....	45
2.2.31	AT+LTIME.....	45
2.2.32	AT+ULCNT.....	46
2.2.33	AT+DLCNT.....	47
2.2.34	AT+ PUBNWK.....	48
2.2.35	AT+TXRETRY.....	49
2.2.36	AT+DCBAND.....	50
<b>2.3</b>	<b>RF commands.....</b>	<b>51</b>
2.3.1	AT+RFMDE.....	51
2.3.2	AT+RFRLO.....	52
2.3.3	AT+RTCOUT.....	53
2.3.4	AT+MCO.....	54
2.3.5	AT+GPIOM.....	55
2.3.6	AT+GPIOS.....	56
2.3.7	AT+TOFF.....	57
2.3.8	AT+TCONF.....	57
2.3.9	AT+PWR.....	60
2.3.10	AT+TTONE.....	61
2.3.11	AT+TTLRA.....	62
2.3.12	AT+TRLRA.....	63
2.3.13	AT+SUPCH.....	65
2.3.14	AT+CHADD.....	66
2.3.15	AT+CHRE.....	67
2.3.16	AT+TRSSI.....	68
2.3.17	AT+TTH.....	69
2.3.18	AT+CERTIF.....	70



2.4      **Appendix**..... 71

2.4.1    *Region parameters*..... 71



# Release Note

FW Version	Revised Contents
V1.1.1	Initial version.
V1.1.2	<p><b>Normal</b></p> <p>1.Modify verbosity level 0 to LoRa Rx message only</p> <p><b>EU868 :</b></p> <p>1.When add channel to channel list, band will fill in after calculation automatically.</p> <p><b>AU915 :</b></p> <p>1.Enable region AU915.</p> <p>2.Add add/remove channel function.</p> <p>3.Add set/get Rx1 downlink channel command.</p> <p><b>US915 :</b></p> <p>1.Add set/get Rx1 downlink channel command.</p> <p><b>AS923 、 CN779 、 EU433 、 EU868 、 IN865 、 KR920 、 RU864 :</b></p> <p>1.When applied CFList channel list will not be cleared.</p>
V1.1.3	<p><b>Normal</b></p> <p>1.Print downlink mac command from server(gateway).</p> <p>2.Check EEPROM at boot, if is empty write default value, else read to buffer.</p> <p>3.Print string "Please disable the ADR function first.\r\n" in command AT+DR.</p> <p>4.Remove command "AT+READ".</p>
V1.1.4	<p><b>AU915 :</b></p> <p>1.Change Channel mask storage format in eeprom to fix the channel list display issue.</p> <p><b>US915 :</b></p> <p>1.Change Channel mask storage format in eeprom to fix the channel list display issue.</p> <p>2. Add dwell time switch command(AT+UPDWELL 、 AT+DWDWELL).</p> <p><b>Normal :</b></p> <p>1.Fix an issue where command AT+BR could not be used.</p> <p>2. Store baud rate in eeprom by using AT+SAVE.</p>

FW Version	Revised Contents
<b>V1.2.0</b>	<p>This version is base on STM official SDK V1.2.0 modified to apply to AcSip ST50 series.</p> <p><b>EU868 :</b></p> <ol style="list-style-type: none"><li>1. Fix Band duty cycle store format in eeprom.</li></ol> <p><b>US915 :</b></p> <ol style="list-style-type: none"><li>1. Don't check the number of active channels in mask set function.</li></ol> <p><b>Normal :</b></p> <ol style="list-style-type: none"><li>1. Update the AT commands help string.</li><li>2. Disable region : CN779 、EU433 、KR920 、IN865 、RU864.</li><li>3. Modify erase page 124~127 to 125~127 in command AT+FATRS.</li></ol>
<b>V1.2.1</b>	<p><b>Normal :</b></p> <ol style="list-style-type: none"><li>1. Enable region : KR920 、IN865.</li></ol>

# 1. Configuration

## 1.1 Software Configuration

The default baud rate of ST50 LPUART is set at **9600**. And the rest of LPUART setting, please follow these below settings:

Baud rate: **9600**

Data bits: **8**

Stop bits: **1**

Parity: **none**

Flow Control: **none**

Forward: **none**

To quickly start using ST50, the 1<sup>st</sup> step is using USB cable to connect EVB to PC/NB via micro USB port. The next step is checking whether the UART-To-USB bridge IC driver can be properly installed on PC/NB. By using win7/win10, the UART-To-USB bridge IC driver could be installed automatically and shows a USB serial com port after connecting well between EVB and PC/NB via USB cable.

After successful installation of USB driver, you can use any terminal program (suggesting free terminal software: [termite](#)) to connect to EVB. The commands set can be used through the terminal program.

By using [termite](#) or other terminal software, be aware of not being appended nothing in the end of a UART string (Figure 1.1).

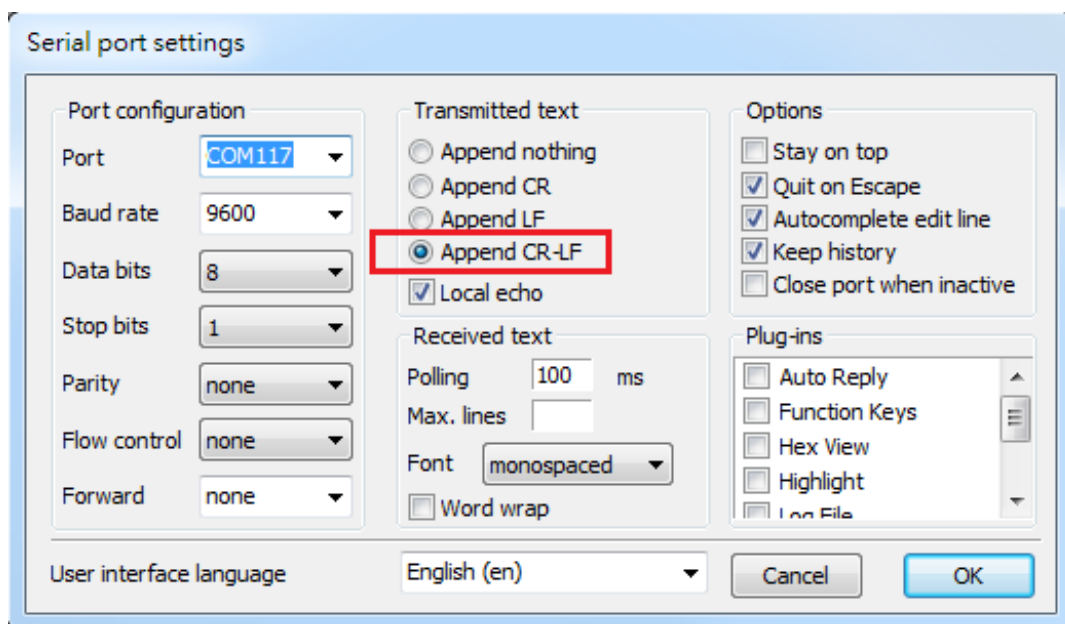


Figure 1.1



The AT commands have the standard format “AT+XXX”, with XXX denoting the command.

There are four available command behaviors:

- AT+XXX? provides a short help of the given command, for example AT+DEUI?
- AT+XXX is used to run a command, such as AT+JOIN
- AT+XXX=? is used to get the value of a given command, for example AT+CFS=?
- AT+XXX=<value> is used to provide a value to a command, for example  
AT+SEND=2:Hello

The output of the commands is provided on the UART. The output format is as below:

**<value><CR><LF>**

**<CR><LF><Status><CR><LF>**

<CR> stands for “carriage return” and <LF> stands for “line feed”

The <value><CR><LF> output is returned whenever the “help AT+XXX?” or the “get AT+XXX=?” commands are run.

When no value is returned, the <value><CR><LF> output is not returned at all.

Every command (except for ATZ used for MCU reset) returns a status string, which is preceded and followed by <CR><LF> in a “<CR><LF><Status><CR><LF>” format. The possible status are:

- OK: command run correctly without error.
- AT\_ERROR: generic error.
- AT\_PARAM\_ERROR: a parameter of the command is wrong.
- AT\_BUSY\_ERROR: the LoRa® network is busy, so the command has not been completed.
- AT\_TEST\_PARAM\_OVERFLOW: the parameter is too long.
- AT\_NO\_CLASSB\_ENABLE: End-node has not yet switched in Class B.
- AT\_NO\_NETWORK\_JOINED: the LoRa® network has not been joined yet.
- AT\_RX\_ERROR: error detection during the reception of the command.

More details on each command description and examples are given in the remainder of this section. Note that each command preceded by # is provided by the host to the module. Then the return of the module is printed.

## 2.Commands Set Reference

### 2.1 General commands

#### 2.1.1 AT

Purpose: This command is used to check that the link is working properly.

Response: OK.

Command	Input parameter	Return value	Return code	Command behavior
AT	-	-	OK	Run the command.

Example:

AT

OK

## 2.1.2 AT?

Purpose: This command provides short help for all the supported commands

Response: OK.

Command	Input parameter	Return value	Return code	Command behavior
AT?	-	All Command Help	OK	Provide help.

Example:

```

AT?
AT+<CMD>?      : Help on <CMD>
AT+<CMD>       : Run <CMD>
AT+<CMD>=<value> : Set the value
AT+<CMD>=?      : Get the value
ATZ.           : Trg a MCU reset.
AT+VL=<Level><CR>. Set the Verbose Level=[0:OFF/RF_RX .. 3:High].
AT+LTIME.      : Get the local time in UTC format.
AT+BR=<Baud Rate><CR>. Get or Set the baud rate of LPUART, default 9600, Baud_Rate=[1200, 2400, 4800, 9600].
AT+FAIRS.     : All LoRaWAN and radio configuration parameters will be set to default value and trig a MCU reset.
AT+APPEUI=<XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX><CR>. Get or Set the App Eui.
AT+APPKKEY=<XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX><CR>. Get or Set the Application Key.
AT+NWKSKEY=<XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX><CR>. Get or Set the Network Session Key.
AT+APPSKEY=<XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX><CR>. Get or Set the Application Session Key.
AT+DADDR=<XX:XX:XX:XX><CR>. Get or Set the Device address.
AT+DEUI=<XX:XX:XX:XX:XX:XX:XX:XX><CR>. Get or Set the Device EUI.
AT+NWKID=<NwkID><CR>. Get or Set the Network ID=[0..127].
AT+JOIN=<Mode><CR>. Join network with Mode=[0:ABF, 1:OTAA].
AT+LINKC.     : Piggyback a Link Check Request to the next uplink.
AT+SEND=<Port>:<Ack>:<Payload><CR>. Send binary data with the application Port=[1..199] and Ack=[0:unconfirmed, 1:confirmed].
AT+ULCNT.     : Get or Set the Uplink Counter.
AT+DLCNT.     : Get the Downlink Counter.
AT+TXRETRY.   : Get or Set the TX re-tries to get an acknowledge, Maximum number is 8.
AT+PUBNWK.    : Get or Set the Public Network setting(0: private network, 1: public network).
AT+UPDWELL.   : Get or Set the Uplink DWELL(0: OFF, 1: ON).
AT+DWDWELL.   : Get or Set the Downlink DWELL(0: OFF, 1: ON).
AT+DCBAND=<Band>:<DutyCycle><CR>. Get or Set the duty cycle of specified band.
AT+VER.       : Get the FW version.
AT+ADR=<ADR><CR>. Get or Set the Adaptive Data Rate setting ADR=[0:off, 1:on].
AT+DR=<DataRate><CR>. Get or Set the Tx DataRate=[0..7].
AT+BAND=<BandID><CR>. Get or Set the Active Region BandID=[0:AS923, 1:AUS915, 5:EU868, 6:KR920, 7:IN865, 8:US915].
AT+CLASS=<Class><CR>. Get or Set the Device Class=[A, B, C].
AT+DCS=<DutyCycle><CR>. Get or Set the ETSI DutyCycle=[0:disable, 1:enable] - Only for testing.
AT+UNIDL=<Delay><CR>. Get or Set the Join Accept Delay between the end of the Tx and the Join Rx Window 1 in ms.
AT+UN2DL=<Delay><CR>. Get or Set the Join Accept Delay between the end of the Tx and the Join Rx Window 2 in ms.
AT+RX1DL=<Delay><CR>. Get or Set the delay between the end of the Tx and the Rx Window 1 in ms.
AT+RX2DL=<Delay><CR>. Get or Set the delay between the end of the Tx and the Rx Window 2 in ms.
AT+RX2DR=<DataRate><CR>. Get or Set the Rx2 window DataRate=[0..7].
AT+RX2FQ=<Freq><CR>. Get or Set the Rx2 window Freq in Hz.
AT+TXP=<Power><CR>. Get or Set the Transmit Power=[0..15](valid range according to region).
AT+PGSLOT=<Period><CR>. Set or Get the unicast ping slot Period=[0:1s .. 7:128s](=2*Period).
AT+TTONE.     : Starts RF Tone test.
AT+TTLRA=<PacketNb>:<Interval>:<Payload><CR>. Sent payload with RF Tx test, PacketNb=[0-100000(0:Unlimit)], Interval=[0-300000(ms)], Payload=[size:1-254].
AT+TRLRA=<Payload><CR>. Received payload with RF Rx test, Payload=[size:1-254(optional)].
AT+TRSSI.     : Starts RF RSSI tone test.
AT+TCONF.     : Config RF test parameters.
AT+CERTIF=<Mode><CR>. Set the module in LoRaWAN Certification with Join Mode=[0:ABF, 1:OTAA].
AT+SAVE.      : Save the LoRa and LoRaWAN parameters.
AT+GPIOM.     : Set the GPIO mode.
AT+GPIOS.     : Set the GPIO output status or Get the input status.
AT+UID.       : Get the unique device ID(96-bit).
AT+IEUID.     : Get the IEEE 64-bit unique device ID.
AT+PWR.       : Let the ST50H enter low power mode (STOP2).
AT+RFMD=<Mode>. Get or Set the RF into standby or sleep mode, Mode=[SLEEP, STDBY_RC, STDBY_XOSC].
AT+RFRLO=<Mode>. Get or Set the RF regulator, Mode=[DCDC, LDO].
AT+RCOUT=<Switch><CR>. Set the GPIO pin PC13(RTC_OUT1) output the 4Hz signal, Switch=[4Hz, DISABLE].
AT+MCO=<Switch><CR>. Set the GPIO pin PA8(MCO) output the LSE or SYSClk signal, Switch=[LSE, SYSClk, DISABLE].
AT+CHRE.      : Remove the channel by Channel ID.
AT+SUPCH.     : Show the all uplink channels.
AT+CHADD.     : Add or set up the channel.
AT+RPSSEND.   : Repeat send the data for RF Certification(ex. CE, FCC, etc.).
AT+TTH=<Fstart>,<Fstop>,<Fdelta>,<PacketNb><CR>. Starts RF Tx hopping test from Fstart to Fstop in Hz or MHz, Fdelta in Hz.
AT+TOFF.      : Stops on-going RF test.
AT+RX1FQ=<Freq><Step><Count><CR>. Get or Set the Rx1 first channel Freq in Hz, Rx1 stepwidth and Rx1 Freq count (Only support AU915 and US915).
AT+BAT        : Get the battery Level in mV.
OK

```

### 2.1.3 ATZ

Purpose: This command resets the module and start FW over again.

Response: The beginning information since FW starts.

Command	Input parameter	Return value	Return code	Command behavior
ATZ?	-	ATZ. Trig a MCU reset	OK	Provide a short help.
ATZ	-	No return value and return code. The MCU is reset.	-	Run the command.

Example:

```

ATZ
-----
Tech Co., LTD
LoRaWAN v1.0.4 Ready
(Class A, B & C)
>> ST50H - V1.2.1 - AS923 - Jul 20 2022 - 09:13:51
  
```

### 2.1.4 AT+VER

Purpose: Get current firmware version.

Response: A string representing firmware version.

Command	Input parameter	Return value	Return code	Command behavior
AT+VER?	-	AT+VER. Get the FW version	OK	Provide a short help.
AT+VER=?	-	V.x.y.z	OK	Get the value.

Example:

```

AT+VER=?
AcSip_FW_VERSION:    V1.2.1
MW_LORAWAN_VERSION:  V2.4.0
MW_RADIO_VERSION:    V1.2.0
L2_SPEC_VERSION:     V1.0.4
RP_SPEC_VERSION:     V2-1.0.1
OK
  
```

### 2.1.5 AT+FATRS

**Purpose:** This command resets all LoRaWAN and radio configuration.

Reset verbose level, battery level, band, TX power, data rate, RX delay, RX frequency, Join delay, etc.

Response: The beginning information since FW starts.

Command	Input parameter	Return value	Return code	Command behavior
AT+FATRS?	-	AT+FATRS. All LoRaWAN and radio configuration parameters will be set to default value and trig a MCU reset	OK	Provide a short help.
AT+FATRS	-	No return value and return code. The MCU is reset.	-	Run the command.

Example:

AT+FATRS

```
>> ST50H - V1.2.1 - AS923 - Jul 20 2022 - 09:13:51
```

## 2.1.6 AT+VL=&lt;LEVEL&gt;

Purpose: Sets/gets the verbose level of the application.

<LEVEL> : 0, 1, 2, or 3.

- 0: VLEVEL\_OFF (Only show LoRa RX related message **block other message include command response**)
- 1: VLEVEL\_L
- 2: VLEVEL\_M
- 3: VLEVEL\_H

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+VL?	-	AT+VL=<Level><CR>. Set the Verbose Level=[0:Off .. 3:High]	OK	Provide a short help.
AT+VL=?	-	0, 1, 2 or 3	OK	Get the value.
AT+VL=<LEVEL>	0, 1, 2 or 3	-	OK AT_PARAM_ERROR	Set the value.

Example:

```
AT+VL=?
```

```
2
```

```
OK
```

```
AT+VL=2
```

```
OK
```

### 2.1.7 AT+BAT

Purpose: Get the battery Level.

Response: A integer representing the battery level.

Command	Input parameter	Return value	Return code	Command behavior
AT+BAT?	-	AT+BAT. Get the battery Level=[0..254]	OK	Provide a short help.
AT+BAT=?	-	<void>	OK	Get the value.

Example:

- Get the battery Level.

AT+BAT=?

3295

OK

### 2.1.8 AT+BR=<Baud Rate>

**Purpose:** Get or Set the baud rate of LPUART.

Note: Change baud rate will see unrecognizable message is normal, because baud rate is change.


Response: A integer representing the baud rate of LPUART.

Command	Input parameter	Return value	Return code	Command behavior
AT+BR?	-	AT+BR. Get or Set the baud rate of LPUART, default 9600, baud rate=[1200, 2400, 4800, 9600]	OK	Provide a short help.
AT+BR=?	-	<void>	OK	Get the value.
AT+BR=<Param>	1200, 2400, 4800, 9600	<void>	OK	Set the value.

Example:

- Get the baud rate of LPUART.

ATZ

 Tech Co., LTD  
LoRaWAN v1.0.4 Ready  
(Class A, B & C)

```
>> ST50H - V1.2.1 - AS923 - Jul 20 2022 - 09:13:51
```

$$AT + BR = ?$$

9600

OK

$$AT + BR = 4800$$

??????

AT+SAVE

OK

ATZ



Tech Co., LTD  
LoRaWAN v1.0.4 Ready  
(Class A, B & C)

>> ST50H - V1.2.1 - AS923 - Jul 20 2022 - 09:13:51

$$AT + BR = ?$$

4800

OK



## 2.2 MAC commands

### 2.2.1 AT+SAVE

**Purpose:** Save the LoRa and LoRaWAN parameters.

Response: **Ok**, if input arguments are valid.

**AT\_BUSY\_ERROR:** the LoRa® network is busy, so the command has not been completed.

Command	Input parameter	Return value	Return code	Command behavior
AT+SAVE?	-	AT+SAVE. Save the LoRa and LoRaWAN parameters	OK	Provide a short help.
AT+SAVE	-	-	OK AT_BUSY_ERROR	Run the command.

Example:

- Save LoRa and LoRaWAN parameters.

```
ATZ
      Tech Co., LTD
      LoRaWAN v1.0.4 Ready
      (Class A, B & C)

>> ST50H - V1.2.1 - AS923 - Jul 20 2022 - 09:13:51

AT+BR=?
9600

OK
AT+BR=4800
??????
```

AT+SAVE

OK  
ATZ

```
>> ST50H - V1.2.1 - AS923 - Jul 20 2022 - 09:13:51
AT+BR=?
4800
OK
```

### 2.2.2 AT+UID

Purpose: Each STM32 MCU device has its own unique ID, use this command to read it out.

Response: A string representing hardware STM32 MCU UID 96-bit value.

Command	Input parameter	Return value	Return code	Command behavior
AT+UID?	-	AT+UID: Get the unique device ID(96-bit)	OK	Provide a short help.
AT+UID=?	-	Unique device ID=xxxxxxxxxxxxxxxxxxxxxxxxxx	OK	Get the value.

Example:

```
AT+UID=?  
Unique device ID=2036395835565011002a004f  
OK
```

### 2.2.3 AT+IEUID

Purpose: Each STM32 MCU device has its own IEEE 64-bit unique device ID, use this command to read it out.

Response: A string representing hardware STM32 MCU IEEE 64-bit unique device ID value.

Command	Input parameter	Return value	Return code	Command behavior
AT+IEUID?	-	AT+IEUID: Get the IEEE 64-bit unique device ID	OK	Provide a short help.
AT+IEUID=?	-	IEEE 64-bit unique device ID=xxxxxxxxxxxxxxxx	OK	Get the value.

Example:

```
AT+IEUID=?  
IEEE 64-bit unique device ID=0080e1150006bfee  
OK
```

### 2.2.4 AT+DEUI=<DEUI>

Purpose: Allows the user to access the global end-device EUI.

<DEUI> an 8-byte hexadecimal string representing Device EUI used for LoRaWAN, 8 hexa separated by “:”.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+DEUI?	-	AT+DEUI: Get or Set the Device EUI.	OK	Provide a short help.
AT+DEUI=?	-	<8 hexa separated by:>	OK	Get the value.
AT+DEUI=<Param>	<8 hexa separated by:>	-	OK AT_PARAM_ERROR	Set the value.

Example:

- Get the Device EUI. The Device EUI is 0750363256375020.

```
AT+DEUI=?
00:80:e1:15:00:06:bf:ee
OK
```

- Set the Device EUI is 1122334455667788.

```
AT+DEUI=?
00:80:e1:15:00:06:bf:ee

OK
AT+DEUI=11:22:33:44:55:66:77:88

OK
AT+DEUI=?
11:22:33:44:55:66:77:88

OK
```

### 2.2.5 AT+APPEUI=<AEUI>

Purpose: Allows the user to access the global application identifier EUI.

<AEUI> an 8-byte hexadecimal string representing application identifier EUI used for LoRaWAN, 8 hexa separated by “:”.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+APPEUI?	-	AT+APPEUI: Get or Set the App EUI.	OK	Provide a short help.
AT+APPEUI=?	-	<8 hexa separated by:>	OK	Get the value.
AT+APPEUI=<Param>	<8 hexa separated by:>	-	OK AT_PARAM_ERROR	Set the value.

Example:

- Get the application identifier EUI. The application identifier EUI is 0101010101010101.

```
AT+APPEUI=?
01:01:01:01:01:01:01:01
OK
```

- Set the application identifier EUI is 1122334455667788.

```
AT+APPEUI=?
01:01:01:01:01:01:01:01
OK
AT+APPEUI=11:22:33:44:55:66:77:88
OK
AT+APPEUI=?
11:22:33:44:55:66:77:88
OK
```

### 2.2.6 AT+DADDR=<ADDR>

Purpose: Allows the user to access the device address.

<ADDR> a 4-byte hexadecimal string representing device address used for LoRaWAN, 4 hexa separated by “:”.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+DADDR?	-	AT+DADDR: Get or Set the Device address.	OK	Provide a short help.
AT+DADDR=?	-	<4 hexa separated by:>	OK	Get the value.
AT+DADDR=<Param>	<4 hexa separated by:>	-	OK AT_PARAM_ERROR	Set the value.

Example:

- Get the device address. The device address is 0100000a.

```

AT+DADDR=?
00:06:bf:ee
OK

```

- Set the device address is 11223344.

```

AT+DADDR=?
00:06:bf:ee

OK
AT+DADDR=11:22:33:44

OK
AT+DADDR=?
11:22:33:44

OK

```

### 2.2.7 AT+APPKEY=<KEY>

Purpose: Allows the user to access the application key.

<KEY> a 16-byte hexadecimal string representing application key used for LoRaWAN, 16 hexa separated by “:”.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+APPKEY?	-	AT+APPKEY: Get or Set the Application Key.	OK	Provide a short help.
AT+APPKEY=?	-	<16 hexa separated by:>	OK	Get the value.
AT+APPKEY=<Param>	<16 hexa separated by:>	-	OK AT_PARAM_ERROR	Set the value.

Example:

- Get the application key. The application key is 2b7e151628aed2a6abf7158809cf4f3c.

```
AT+APPKEY=?
2b:7e:15:16:28:ae:d2:a6:ab:f7:15:88:09:cf:4f:3c
OK
```

- Set the application key is 112233445566778899AABBCCDDEEFF11.

```
AT+APPKEY=?
2b:7e:15:16:28:ae:d2:a6:ab:f7:15:88:09:cf:4f:3c
OK
AT+APPKEY=11:22:33:44:55:66:77:88:99:AA:BB:CC:DD:EE:FF:11
OK
AT+APPKEY=?
11:22:33:44:55:66:77:88:99:aa:bb:cc:dd:ee:ff:11
OK
```

### 2.2.8 AT+APPSKEY=<KEY>

Purpose: Allows the user to set the application session key.

<KEY> a 16-byte hexadecimal string representing application session key used for LoRaWAN, 16 hexa separated by “:”.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+APPSKEY?	-	AT+APPSKEY: Set the Application Session Key.	OK	Provide a short help.
AT+APPSKEY=?	-	<16 hexa separated by:>	OK	Get the value.
AT+APPSKEY=<Param>	<16 hexa separated by:>	-	OK AT_PARAM_ERROR	Set the value.

Example:

- Get the application session key. The application key is 2b7e151628aed2a6abf7158809cf4f3c.

```
AT+APPSKEY=?
2b:7e:15:16:28:ae:d2:a6:ab:f7:15:88:09:cf:4f:3c
OK
```

- Set the application session key is 112233445566778899AABBCCDDEEFF11.

```
AT+APPSKEY=?
2b:7e:15:16:28:ae:d2:a6:ab:f7:15:88:09:cf:4f:3c
OK
AT+APPSKEY=11:22:33:44:55:66:77:88:99:AA:BB:CC:DD:EE:FF:11
OK
AT+APPSKEY=?
11:22:33:44:55:66:77:88:99:aa:bb:cc:dd:ee:ff:11
OK
```

### 2.2.9 AT+NWKSKEY=<KEY>

Purpose: Allows the user to set the network session key.

<KEY> a 16-byte hexadecimal string representing network session key used for LoRaWAN, 16 hexa separated by “:”.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+NWKSKEY?	-	AT+NWKSKEY: Set the Network Session Key.	OK	Provide a short help.
AT+ NWKSKEY=?	-	<16 hexa separated by:>	OK	Get the value.
AT+NWKSKEY=<Param>	<16 hexa separated by:>	-	OK AT_PARAM_ERROR	Set the value.

Example:

- Get the network session key. The application key is 2b7e151628aed2a6abf7158809cf4f3c.

AT+NWKSKEY=?

2b:7e:15:16:28:ae:d2:a6:ab:f7:15:88:09:cf:4f:3c

OK

- Set the network session key is 112233445566778899AABBCCDDEEFF11.

AT+NWKSKEY=?

2b:7e:15:16:28:ae:d2:a6:ab:f7:15:88:09:cf:4f:3c

OK

AT+NWKSKEY=11:22:33:44:55:66:77:88:99:AA:BB:CC:DD:EE:FF:11

OK

AT+NWKSKEY=?

11:22:33:44:55:66:77:88:99:aa:bb:cc:dd:ee:ff:11

OK



## 2.2.10 AT+JOIN=&lt;MODE&gt;

Purpose: This command does a join request to the network.

**<MODE>** : A decimal string representing join mode of LoRaWAN, can be 1 (otaa, over-the-air activation) or 0 (abp, activation by personalization).

Response: **Ok**, if input arguments are valid.

**AT\_BUSY\_ERROR**, the LoRa® network is busy, so the command has not been completed.

Command	Input parameter	Return value	Return code	Command behavior
AT+JOIN?	-	AT+JOIN: join network.	OK	Provide a short help.
AT+JOIN=<MODE>	0 or 1	-	OK AT_BUSY_ERROR	Set the value.

Example:

- Join LoRaWAN by OTAA.

```
AT+JOIN=1
6s424:TX on freq 923200000 Hz at DR 2
OK
--> OnRadioTxDone
6s806:MAC txDone
11s791:RX_1 on freq 923200000 Hz at DR 2
--> OnRadioRxDone rssi(-83) snr(8)
12s140:MAC rxDone
+EVT:JOINED
```

- Join LoRaWAN by ABP.

```
AT+JOIN=0
+EVT:JOINED
OK
```

### 2.2.11 AT+CLASS=<CLASS>

Purpose: Allow the user to access the LoRaWAN class.

**Note:** Use **AT+CLASS** must **JOIN(2.2.10)** network before.

<CLASS> : A, B or C.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+CLASS?	-	AT+CLASS: get or set the device class.	OK	Provide a short help.
AT+CLASS=?	-	A, B or C	OK	Get the value.
AT+CLASS=<CLASS>	A, B or C	-	OK AT_PARAM_ERROR AT_NO_CLASS_B_ENABLE AT_NO_NET_JOINED	Set the value.

Example:

- Get the LoRaWAN class.

```

AT+CLASS=?
A
OK

```

- Set the LoRaWAN class.

```

AT+CLASS=C
647s268:RX_C on freq 923200000 Hz at DR 2
Switch to Class C done
OK

```

### 2.2.12 AT+BAND=<BAND>

Purpose: Allows the user to access the Active Region.

**<BAND>** : A decimal string representing the band used for LoRaWAN.

- 0 : Asia band on 923MHz(AS923)
- 1 : Australia band on 915MHz(AU915)
- 5 : European band on 868MHz(EU868)
- 6 : South Korean band on 920MHz(KR920)
- 7 : India band on 865MHz(IN865)
- 8 : North American band on 915MHz(US915)

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

**AT\_BUSY\_ERROR**, the LoRa® network is busy, so the command has not been completed.

Command	Input parameter	Return value	Return code	Command behavior
AT+BAND?	-	AT+BAND: Get or Set the Active Region.	OK	Provide a short help.
AT+BAND=?	-	<integer>	OK AT_BUSY_ERROR	Get the value.
AT+BAND=<BAND>	<integer>	-	OK AT_PARAM_ERROR AT_BUSY_ERROR	Set the value.

Example:

- Get the active region.

```

AT+BAND=?
0:AS923
OK

```

### 2.2.13 AT+ UPDWELL=<OFF/ON >

Purpose: Get or Set the Uplink Dwell.

Note: Only use on AS923 、US915 and AU915.

<OFF/ON> 0 for OFF, 1 for ON.

Response: **Ok**, if input arguments are valid.

**AT\_ERROR**, generic error.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+ UPDWELL?	-	AT+UPDWELL. Get or Set the Uplink DWell. (0: OFF, 1: ON)	OK	Provide a short help.
AT+ UPDWELL=?	-	ON or OFF	OK	Get the value.
AT+ UPDWELL=<Param>	0 or 1	-	OK AT_ERROR AT_PARAM,_ERROR	Set the value.

Example:

- Set the Uplink Dwell.

```
AT+UPDWELL=1
```

```
OK
```

- Get the Uplink Dwell.

```
AT+UPDWELL=?
```

```
OFF
```

```
OK
```

### 2.2.14 AT+ DWDWELL=<OFF/ON >

Purpose: Get or Set the Downlink DWell.

Note: Only use on AS923 、US915 and AU915.

<OFF/ON> 0 for OFF, 1 for ON.

Response: **Ok**, if input arguments are valid.

**AT\_ERROR**, generic error.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+ DWDWELL?	-	AT+DWDWELL. Get or Set the Downlink DWell. (0: OFF, 1: ON)	OK	Provide a short help.
AT+ DWDWELL=?	-	ON or OFF	OK	Get the value.
AT+ DWDWELL=<Param>	0 or 1	-	OK AT_ERROR AT_PARAM,_ERROR	Set the value.

Example:

- Set the Downlink Dwell.

```
AT+DWDWELL=1
```

```
OK
```

- Get the Downlink Dwell.

```
AT+DWDWELL=?
```

```
ON
```

```
OK
```

### 2.2.15 AT+TXP=<POWER>

Purpose: Allows the user to access the transmit power.

<POWER> : A decimal string representing transmitting power in level.

**Note :** Value will vary depending on the region, please see 2.4.1 for detail.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+TXP?	-	AT+TXP: get or set the transmit power (0-5).	OK	Provide a short help.
AT+TXP=?	-	Value	OK	Get the value.
AT+TXP=<POWER>	2.4.1	-	OK AT_PARAM_ERROR	Set the value.

Example:

- Get the transmit power.

```
AT+TXP=?
```

```
0
```

```
OK
```

- Set the transmit power.

```
AT+TXP=5
```

```
OK
```

### 2.2.16 AT+SEND=<PORT>:<ACK>:<PAYLOAD>

Purpose: Allows the user to send binary data with the application port and confirmation mode.

**<PORT>** : A decimal string representing port number used for transmission, it can be from 1 to 223.

**<ACK>** : A decimal string representing type of transmitting message, can be 1 (confirmed) or 0 (unconfirmed).

**<PAYLOAD>** : A hexadecimal string representing data to be transmitted.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

**AT\_BUSY\_ERROR**, the LoRa® network is busy, so the command has not been completed.

**AT\_NO\_NETWORK\_JOINED**, the LoRa® network has not been joined yet

Command	Input parameter	Return value	Return code	Command behavior
AT+SEND?	-	AT+SEND: Send binary data with the application port and confirmation mode.	OK	Provide a short help.
AT+SEND=<INPUT>	<PORT>:<ACK>: <PAYLOAD>	-	OK AT_PARAM_ERROR AT_BUSY_ERROR AT_NO_NET_JOINED AT_DUTYCYCLE_RESTRICTED AT_CRYPTO_ERROR AT_ERROR	Set the value.

Example:

- Send the data with app port 20 and confirmation mode.

```
AT+SEND=20:1:8a9a1a2a3a
22s113:TX on freq 923200000 Hz at DR 2

OK
--> OnRadioTxDone

22s453:MAC txDone
23s438:RX_1 on freq 923200000 Hz at DR 2
--> OnRadioRxDone rssi(-75) snr(7)

23s747:MAC rxDone
+EVT:SEND_CONFIRMED
```

- Send the data with app port 20 and un-confirmation mode.

```
AT+SEND=20:0:8a9a1a2a3a
296s957:TX on freq 923200000 Hz at DR 2

OK
--> OnRadioTxDone

297s296:MAC txDone
298s282:RX_1 on freq 923200000 Hz at DR 2
--> OnRadioRxTimeout

298s358:MAC rxTimeout
299s282:RX_2 on freq 923200000 Hz at DR 2
--> OnRadioRxTimeout

299s358:MAC rxTimeout
```



### 2.2.17 AT+RPSEND=<Port>:<Ack>:<Payload>

Purpose: Repeat send the data for RF Certification(ex. CE, FCC, etc.).

**<PORT>** : A decimal string representing port number used for transmission, it can be from 1 to 223.

**<ACK>** : A decimal string representing type of transmitting message, can be 1 (confirmed) or 0 (unconfirmed).

**<PAYLOAD>** : A hexadecimal string representing data to be transmitted.

Response: **Ok**, if input arguments are valid.

**AT\_ERROR**, generic error.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+RPSEND?	-	AT+RPSEND. Repeat send the data for RF Certification(ex. CE, FCC, etc.)	OK	Provide a short help.
AT+RPSEND =<Param>	<Param> or OFF	-	OK AT_ERROR AT_PARAM,_ERROR	Set the value.

Example:

- Send the data with app port 20 and confirmation mode.

```
AT+RPSEND=80:0:11
```

```
The data payload is 1 bytes
```

```
511s949:TX on freq 923400000 Hz at DR 2, power 14 dBm
```

```
OK
```

```
512s249:MAC txDone
```

```
513s236:RX_1 on freq 923400000 Hz at DR 2
```

```
513s312:IRQ_RX_TX_TIMEOUT
```

```
513s312:MAC rxTimeOut
```

- Stop send the data

```
AT+RPSEND=OFF
```

```
OK
```

### 2.2.18 AT+ADR=<ON/OFF>

Purpose: Allows the user to access the adaptive data rate.

<ON/OFF> : A decimal string representing whether ADR is enable(1) or disable(0).

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+ADR?	-	AT+ADR: get or set the adaptive data rate setting (0 = off, 1 = on).	OK	Provide a short help.
AT+ADR=?	-	0 or 1	OK	Get the value.
AT+ADR=<ON/OFF>	0 or 1	-	OK AT_PARAM_ERROR	Set the value.

Example:

- Get the adaptive data rate setting.

```

AT+ADR=?
1
OK

```

- Set the (turn off) adaptive data rate.

```

AT+ADR=?
1
OK
AT+ADR=0
OK
AT+ADR=?
0
OK

```

### 2.2.19 AT+DR=<DR>

Purpose: Allow the user to access the data rate.

<DR> : A decimal string representing data rate used for LoRaWAN, it can be from 0 to 7.

**Note : Value will vary depending on the region, please see 2.4.1 for detail.**

**If ADR status is ON then AT+DR Command is not allow to use.**

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+DR?	-	AT+DR: Get or Set the Data Rate. (0-7 corresponding to DR_X).	OK	Provide a short help.
AT+DR=?	-	value	OK	Get the value.
AT+DR=<DR>	2.4.1	-	OK AT_PARAM_ERROR	Set the value.

Example:

- Get the data rate.

```

AT+DR=?
2
OK

```

- Set the data rate.

```

AT+DR=2
OK

```

### 2.2.20 AT+DCS=<ON/OFF>

Purpose: Allows the user to access the duty cycle parameter.

<ON/OFF> : A decimal string representing whether duty cycle is enable(1) or disable(0).

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+DCS?	-	AT+DCS: get or set the ETSI duty cycle setting: 0 = disable 1 = enable	OK	Provide a short help.
AT+DCS=?	-	0 or 1	OK	Get the value.
AT+DCS=<ON/OFF>	0 or 1	-	OK AT_PARAM_ERROR	Set the value.

Example:

- Get the duty cycle setting.

```

AT+DCS=?
0
OK

```

- Set the (turn on) duty cycle.

```

AT+DCS=?
0
OK
AT+DCS=1
OK
AT+DCS=?
1
OK

```

### 2.2.21 AT+RX1FQ=<Freq>:<Step>:<Count>

Purpose: Set Rx Frequency.

**Note: Only use on US915 and AU915.**

<Freq> : Get or Set the Rx1 first channel Freq in Hz.

<Step> : Rx1 step width.

<Count> : Rx1 Frequency count.

$\text{RX Frequency} = \text{Freq} + (\text{channel mod Count}) \times \text{Step}$

**Note: Channel is chosen by system it refer to AT+SUPCH list number.**

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+RX1FQ?	-	AT+RX1FQ=<Freq><Step><Count><CR>. Get or Set the Rx1 first channel Freq in Hz, Rx1 stepwidth and Rx1 Freq count.	OK	Provide a short help.
AT+RX1FQ=?	-	-	OK	Get the value.
AT+RX1FQ=<Input>		<Freq><Step><Count>-	OK AT_PARAM_ERROR	Set the value.

Example:

- Set Rx Frequency

```
AT+RX1FQ=923300000:600000:8
```

```
OK
```

### 2.2.22 AT+RX1DL=<TIME>

Purpose: Allows the user to access the delay of the received window 1.

**<TIME>** : A decimal string representing delay interval in milliseconds used for receive window 1.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

**AT\_BUSY\_ERROR**, the LoRa® network is busy, so the command has not been completed.

Command	Input parameter	Return value	Return code	Command behavior
AT+RX1DL?	-	AT+RX1DL: get or set the delay between the end of the Tx and the Rx window 1 in ms.	OK	Provide a short help.
AT+RX1DL=?	-	<integer>	OK AT_BUSY_ERROR	Get the value.
AT+RX1DL=<TIME>	<integer>	-	OK AT_BUSY_ERROR AT_PARAM_ERROR	Set the value.

Example:

- Get the delay of the received window 1.

```
AT+RX1DL=?
1000
```

OK

- Set the delay of the received window 1.

```
AT+RX1DL=1500
```

OK

### 2.2.23 AT+RX2DL=<TIME>

Purpose: Allows the user to access the delay of the received window 2.

**<TIME>** : A decimal string representing delay interval in milliseconds used for receive window 2.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

**AT\_BUSY\_ERROR**, the LoRa® network is busy, so the command has not been completed.

Command	Input parameter	Return value	Return code	Command behavior
AT+RX2DL?	-	AT+RX2DL: get or set the delay between the end of the Tx and the Rx window 2 in ms.	OK	Provide a short help.
AT+RX2DL=?	-	<integer>	OK AT_BUSY_ERROR	Get the value.
AT+RX2DL=<TIME>	<integer>	-	OK AT_BUSY_ERROR AT_PARAM_ERROR	Set the value.

Example:

- Get the delay of the received window 2.

```

AT+RX2DL=?
2000
OK

```

- Set the delay of the received window 2.

```

AT+RX2DL=3000
OK

```

### 2.2.24 AT+RX2FQ=<FREQ>

Purpose: Allows the user to access the frequency of the received window 2.

<FREQ>: A decimal string representing operation frequency of specified channel in Hz.

**Note :** Value will vary depending on the region, please see 2.4.1 for detail.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

**AT\_BUSY\_ERROR**, the LoRa® network is busy, so the command has not been completed.

Command	Input parameter	Return value	Return code	Command behavior
AT+RX2FQ?	-	AT+RX2FQ: get or set the Rx2 window frequency.	OK	Provide a short help.
AT+RX2FQ=?	-	<Frequency in Hz>	OK AT_BUSY_ERROR	Get the value.
AT+RX2FQ=<FREQ>	2.4.1	-	OK AT_BUSY_ERROR AT_PARAM_ERROR	Set the value.

Example:

- Get the frequency of the received window 2.

```
AT+RX2FQ=?
923200000
```

OK

- Set the frequency of the received window 2.

```
AT+RX2FQ=?
923000000

OK
AT+RX2FQ=922000000
```

```
OK
AT+RX2FQ=?
922000000
```

OK



### 2.2.25 AT+RX2DR=<DR>

Purpose: Allows the user to access the data rate of received window 2.

<DR> : A decimal string representing data rate used for LoRaWAN, it can be from 0 to 7.

**Note : Value will vary depending on the region, please see 2.4.1 for detail.**

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

**AT\_BUSY\_ERROR**, the LoRa® network is busy, so the command has not been completed.

Command	Input parameter	Return value	Return code	Command behavior
AT+RX2DR?	-	AT+RX2DR: get or set the Rx2 window data rate (0-7) corresponding to DR_X.	OK	Provide a short help.
AT+RX2DR=?	-	value	OK AT_BUSY_ERROR	Get the value.
AT+RX2DR=<DR>	2.4.1	-	OK AT_PARAM_ERROR AT_BUSY_ERROR	Set the value.

Example:

- Get the data rate of received window 2.

```
AT+RX2DR=?
2
```

OK

- Set the data rate of received window 2.

```
AT+RX2DR=?
2
```

OK

```
AT+RX2DR=3
```

OK

```
AT+RX2DR=?
3
```

OK

### 2.2.26 AT+JN1DL=<TIME>

Purpose: Allows the user to access the join delay on RX window 1.

**<TIME>** : A decimal string representing join delay interval in milliseconds used for receive window 1.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

**AT\_BUSY\_ERROR**, the LoRa® network is busy, so the command has not been completed.

Command	Input parameter	Return value	Return code	Command behavior
AT+JN1DL?	-	AT+JN1DL: get or set the joint accept delay between the end of the Tx and the join Rx window 1 in ms.	OK	Provide a short help.
AT+JN1DL=?	-	<integer>	OK AT_BUSY_ERROR	Get the value.
AT+JN1DL=<TIME>	<integer>	-	OK AT_PARAM_ERROR AT_BUSY_ERROR	Set the value.

Example:

- Get the delay of the join received window 1.

```
AT+JN1DL=?
5000
```

OK

- Set the delay of the join received window 1.

```
AT+JN1DL=?
5000
OK
AT+JN1DL=7500
```

```
OK
AT+JN1DL=?
7500
```

OK

### 2.2.27 AT+JN2DL=<TIME>

Purpose: Allows the user to access the join delay on RX window 2.

**<TIME>** : A decimal string representing join delay interval in milliseconds used for receive window 2.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

**AT\_BUSY\_ERROR**, the LoRa® network is busy, so the command has not been completed.

Command	Input parameter	Return value	Return code	Command behavior
AT+JN2DL?	-	AT+JN2DL: get or set the joint accept delay between the end of the Tx and the join Rx window 2 in ms.	OK	Provide a short help.
AT+JN2DL=?	-	<integer>	OK AT_BUSY_ERROR	Get the value.
AT+JN2DL=<TIME>	<integer>	-	OK AT_PARAM_ERROR AT_BUSY_ERROR	Set the value.

Example:

- Get the delay of the join received window 2.

AT+JN2DL=?

6000

OK

- Set the delay of the join received window 2.

AT+JN2DL=?

6000

OK

AT+JN2DL=8500

OK

AT+JN2DL=?

8500

OK

### 2.2.28 AT+NWKID=<ID>

Purpose: Allows the user to access the Network ID.

<ID> : [0..127].

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+NWKID?	-	AT+NWKID: Get or Set the Network ID.	OK	Provide a short help.
AT+NWKID=?	-	[0..127]	OK	Get the value.
AT+NWKID=<ID>	[0..127]	-	OK AT_PARAM_ERROR	Set the value.

Example:

- Get the Network ID. The Network ID is 0.

```
AT+NWKID=?
```

```
0
```

```
OK
```

- Set the Network ID is 11223344.

```
AT+NWKID=?
```

```
0
```

```
OK
```

```
AT+NWKID=100
```

```
OK
```

```
AT+NWKID=?
```

```
100
```

```
OK
```

### 2.2.29 AT+PGSLOT=<PERIOD>

Purpose: Allows the user to access the unicast ping slot periodicity.

**<PERIOD>** : A decimal string representing the unicast ping slot used for LoRaWAN Class B.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

**AT\_BUSY\_ERROR**, the LoRa® network is busy, so the command has not been completed.

**AT\_NO\_CLASS\_B\_ENABLE**, End-node has not yet switched in Class B.

Command	Input parameter	Return value	Return code	Command behavior
AT+PGSLOT?	-	AT+PGSLOT: Set or Get the unicast ping slot periodicity.	OK	Provide a short help.
AT+PGSLOT=?	-	0 ~ 7	OK AT_BUSY_ERROR AT_NO_CLASS_B_ENABLE	Get the value.
AT+PGSLOT=<PERIOD>	0 ~ 7	-	OK AT_PARAM_ERROR AT_BUSY_ERROR AT_NO_CLASS_B_ENABLE	Set the value.

Example:

- Get the unicast ping slot used for LoRaWAN Class B.

```
AT+PGSLOT=?
```

```
4
```

```
OK
```

- Set the unicast ping slot used for LoRaWAN Class B.

```
AT+PGSLOT=?
```

```
4
```

```
OK
```

```
AT+PGSLOT=3
```

```
OK
```

```
AT+PGSLOT=?
```

```
3
```

```
OK
```

### 2.2.30 AT+LINKC

Purpose: Piggyback a Link Check Request to the next uplink.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+LINKC?	-	AT+LINKC. Piggyback a Link Check Request to the next uplink	OK	Provide a short help.
AT+LINKC	-	-	OK	Run the command.

Example:

```
AT+JOIN=0
+EVT:JOINED

OK
AT+LINKC

OK
```

### 2.2.31 AT+LTIME

Purpose: Allows the user to get the local time in a UTC format.

Response: A string representing the local time in a UTC format.

Command	Input parameter	Return value	Return code	Command behavior
AT+LTIME?	-	AT+LTIME: Get the local time in UTC format.	OK	Provide a short help.
AT+LTIME=?	-	LTIME:xxhxxmxxs on DD/MM/YYYY	OK	Get the value.

Example:

```
AT+LTIME=?
LTIME:02h07m48s on 01/01/1970

OK
```

### 2.2.32 AT+ULCNT=<Counter>

Purpose: Get or Set the Uplink Counter

<Counter> A decimal string representing counter.

Response: **Ok**, if input arguments are valid.

**AT\_ERROR**, generic error.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+ ULCNT?	-	AT+ULCNT. Get or Set the Uplink Counter	OK	Provide a short help.
AT+ULCNT=?	-	Unjoin or Counter	OK AT_ERROR	Get the value.
AT+ULCNT=<Param>	-	-	OK AT_ERROR AT_PARAM,_ERROR	Set the value.

Example:

- Get the Uplink Counter

AT+ULCNT=?

0

OK

- Set the Uplink Counter.

AT+ULCNT=2

OK

AT+ULCNT=?

2

OK

### 2.2.33 AT+DLCNT

Purpose: Get the Downlink Counter

Response: **Ok**, if input arguments are valid.

**AT\_ERROR**, generic error.

Command	Input parameter	Return value	Return code	Command behavior
AT+ DLCNT?	-	AT+DLCNT. Get the Downlink Counter	OK	Provide a short help.
AT+ DLCNT=?	-	Unjoin or counter	OK AT_ERROR	Get the value.

Example:

- Get the Downlink Data.

AT+DLCNT=?

Not any received Downlink data from joined until now.

OK



## 2.2.34 AT+ PUBNWK=&lt;Private/ Public&gt;

Purpose: Get or Set the Public Network setting.

<ON/OFF> 0 for Private, 1 for Public.

Response: **Ok**, if input arguments are valid.

**AT\_ERROR**, generic error.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+ PUBNWK?	-	AT+PUBNWK. Get or Set the Public Network setting. (0: private network, 1: public network)	OK	Provide a short help.
AT+ PUBNWK=?	-	-	OK	Get the value.
AT+ PUBNWK=<Param>	0 or 1	-	OK AT_ERROR AT_PARAM,_ERROR	Set the value.

Example:

- Set the Public Network setting.  
AT+PUBNWK=0  
OK
- Get the Public Network setting.  
AT+PUBNWK=?  
Disable:Private Network  
OK

### 2.2.35 AT+TXRETRY=<times>

Purpose: Get or Set the TX re-tries to get an acknowledge.

**<times>** A decimal string representing retry times. Maximum number is 8

Response: **Ok**, if input arguments are valid.

**AT\_ERROR**, generic error.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+TXRETRY?	-	AT+TXRETRY. Get or Set the TX re-tries to get an acknowledge, Maximum number is 8	OK	Provide a short help.
AT+TXRETRY=?	-	-	OK	Get the value.
AT+TXRETRY=<Param>	0 to 8	-	OK AT_ERROR AT_PARAM,_ERROR	Set the value.

Example:

- Set the TX re-try times.

```
AT+TXRETRY=6
```

```
OK
```

- Get the TX re-try times.

```
AT+TXRETRY=?
```

```
6
```

```
OK
```

### 2.2.36 AT+DCBAND=<Band>:<Cycle>

Purpose: Get or Set the duty cycle of specified band.

**<BAND>** : A decimal string representing the band used for LoRaWAN.

- 0 : Asia band on 923MHz(AS923)
- 1 : Australia band on 915MHz(AU915)
- 5 : European band on 868MHz(EU868)
- 6 : South Korean band on 920MHz(KR920)
- 7 : India band on 865MHz(IN865)
- 8 : North American band on 915MHz(US915)

**<Cycle>** A decimal string representing Cycle from 1 to 65534.

Response: **Ok**, if input arguments are valid.

**AT\_ERROR**, generic error.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+DCBAND?	-	AT+DCBAND. Get or Set the duty cycle of specified band.	OK	Provide a short help.
AT+DCBAND=?	-	void	OK AT_ERROR	Get the value.
AT+DCBAND=<Param>	-	-	OK AT_ERROR AT_PARAM,_ERROR	Set the value.

Example:

- Set the duty cycle of specified.

```
AT+DCBAND=0:200
```

```
OK
```

- Get the duty cycle of specified.

```
AT+DCBAND=?
```

```
Band(0):Duty Cycle(1/200)
```

```
OK
```

## 2.3 RF commands

### 2.3.1 AT+RFMDE=<MODE>

Purpose: Allow the user to access the RF running mode.

<MODE> RF running mode, it can be strings **SLEEP**, **STDBY\_RC**, **STDBY\_XOSC**.

Note: Let the RF part enter sleep mode with warm start by keyword **SLEEP**. Let the RF part enter standby mode and the mode where only RC13M is used is called **STDBY\_RC** and the one with XOSC ON is called **STDBY\_XOSC**.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+RFMDE?	-	AT+RFMDE: Get or Set the RF into standby or sleep mode.	OK	Provide a short help.
AT+RFMDE=?	-	<Void>	OK	Get the mode.
AT+RFMDE=<MODE>	SLEEP, STDBY_RC or STDBY_XOSC	-	OK AT_PARAM_ERROR	Set the mode.

Example:

- Set the RF part enter sleep mode.

```
AT+RFMDE=STDBY_XOSC
```

```
OK
```

- Get the RF part running mode.

```
AT+RFMDE=?
```

```
RF in STDBY_XOSC mode
```

```
OK
```

### 2.3.2 AT+RFRLO=<REGULATOR>

Purpose: Allow the user to access the RF part regulator setting.

<REGULATOR> RF part regulator setting, it can be strings **DCDC**, **LDO**.

Note: DC-DC buck converter by string **DCDC** or linear LDO regulator by string **LDO**.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+RFRLO?	-	AT+RFRLO: Get or Set the RF regulator.	OK	Provide a short help.
AT+RFRLO=?	-	<Void>	OK	Get the setting.
AT+RFRLO=<MODE>	DCDC or LDO	-	OK AT_PARAM_ERROR	Set the setting.

Example:

- Set the RF part regulator setting in DCDC(DC-DC buck converter).

```
AT+RFRLO=DCDC
```

```
OK
```

- Get the RF part regulator setting.

```
AT+RFRLO=?
```

```
RF Regulator in DCDC mode
```

```
OK
```

### 2.3.3 AT+RTCOUT=<OUT>

Purpose: Allow the user to output the RTC\_OUT1 (4Hz square wave) signal on GPIO pin PC13.

<OUT> Output the RTC\_OUT1 (4Hz square wave) signal on GPIO pin PC13, it can be strings **4Hz**, or disable the output by string **DISABLE**.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+RTCOUT?	-	AT+RTCOUT: Set the GPIO pin PC13(RTC_OUT1) output the 4Hz signal. (4Hz or DISABLE).	OK	Provide a short help.
AT+RTCOUT=<OUT>	4Hz or DISABLE	-	OK AT_PARAM_ERROR	Set the setting.

Example:

- Output the RTC\_OUT1 (4Hz square wave) signal on GPIO pin PC13.

```
AT+RTCOUT=4Hz
```

```
OK
```

- Disable the RTC\_OUT1 output..

```
AT+RTCOUT=DISABLE
```

```
OK
```

### 2.3.4 AT+MCO=<OUT>

Purpose: Allow the user to output the LSE or SYSClk signals on GPIO pin PA8.

<OUT> Output the LSE (32768Hz) or SYSClk signals on GPIO pin PA8, it can be strings **LSE**, **SYSClk** or disable the output by string **DISABLE**.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+MCO?	-	AT+MCO: Set the GPIO pin PA8(MCO) output the LSE or SYSClk signal. (LSE, SYSClk or DISABLE).	OK	Provide a short help.
AT+MCO=<OUT>	LSE, SYSClk or DISABLE	-	OK AT_PARAM_ERROR	Set the setting.

Example:

- Output the LSE (32768Hz) signal on GPIO pin PA8.

AT+MCO=LSE

OK

- Disable the GPIO pin PA8 output.

AT+MCO=DISABLE

OK

### 2.3.5 AT+GPIOM=<Gpio\_Group>:<Gpio\_Pin\_Number>:<Gpio\_Mode>:<Gpio\_Pull>

Purpose: Assign STM32 GPIO pin mode as input or output.

**<Gpio\_Group>** A string representing STM32 GPIO pin groups, it can be these characters **PA**, **PB**, **PC** and **PH**.

**<Gpio\_Pin\_Number>** A decimal string representing STM32 GPIO pin number, it can be set from **0** to **15**.

Note: PA4, PA5, PA6, PA7, PA8, PA9, PA10, PA11, PA12, PA13, PA14, PA15, PB1, PB2, PB3, PB4, PB5, PB6, PB7, PB8, PB9, PB10, PB11, PB12, PB13, PB14, PB15, PC0, PC1, PC2, PC3, PC4, PC5, PC6, PC13, PH3, as mentioned above, all can used.

**<Gpio\_Mode>** A decimal string representing STM32 GPIO pin mode, it can be assigned as **INPUT**, **OUTPUT\_PP** or **OUTPUT\_OD**.

Note: **INPUT** meaning input mode, **OUTPUT\_PP** meaning output push-pull mode, **OUTPUT\_OD** meaning output open-drain mode.

**<Gpio\_Pull>** A decimal string representing STM32 GPIO pin pull activation, it can be assigned as **NOPULL**, **PULLUP** or **PULLDOWN**.

Note: **NOPULL** meaning GPIO no pull activation, **PULLUP** meaning GPIO pull-up activation, **PULLDOWN** meaning GPIO pull-down activation.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+GPIOM?	-	AT+GPIOM: Set the GPIO mode	OK	Provide a short help.
AT+GPIOM=<Param>	Void	Void	OK AT_PARAM_ERROR	Set the value.

Example:

- Set PA0 as output push-pull mode with pull-up activation.

```
AT+GPIOM=PA:0:OUTPUT_PP:PULLUP
```

```
OK
```



### 2.3.6 AT+GPIO=<Gpio\_Group>:<Gpio\_Pin\_Number>:<Gpio\_Value>

Purpose: Assign STM32 GPIO pin output state as high or low, or get the GPIO pin input state.

**<Gpio\_Group>** A string representing STM32 GPIO pin groups, it can be these characters **PA**, **PB**, **PC** and **PH**.

**<Gpio\_Pin\_Number>** A decimal string representing STM32 GPIO pin number, it can be set from **0** to **15**.

Note: PA4, PA5, PA6, PA7, PA8, PA9, PA10, PA11, PA12, PA13, PA14, PA15, PB1, PB2, PB3, PB4, PB5, PB6, PB7, PB8, PB9, PB10, PB11, PB12, PB13, PB14, PB15, PC0, PC1, PC2, PC3, PC4, PC5, PC6, PC13, PH3, as mentioned above, all can used.

**<Gpio\_Value >** A decimal string representing STM32 GPIO pin value, it can be assigned as high or low, set **1** would let pin be high state and **0** let it as low state. If assigned as **?**, meaning to get the GPIO pin input state.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+GPIO?	-	AT+GPIO: Set the GPIO output status or Get the input status	OK	Provide a short help.
AT+GPIO=<Param>	Void	Void (when set) PX=y (when get)	OK AT_PARAM_ERROR	Set or Get the value.

Example:

- Set PA0 as output high state.

```
AT+GPIO=PA:0:1
OK
```

- Get PH3 input state.

```
AT+GPIO=PH:3:?
PH3=0
OK
```

### 2.3.7 AT+TOFF

Purpose: Stop ongoing radio frequency test.

Response: **Ok**, if input arguments are valid.

Command	Input parameter	Return value	Return code	Command behavior
AT+TOFF?	-	AT+TOFF: Stops on-going RF test	OK	Provide a short help.
AT+TOFF	-	Test Stop.	OK	Run the command.

Example:

Stop RF test.

```
AT+TOFF
Test Stop
OK
```

### 2.3.8 AT+TCONF=<Freq>:<Power>:<Bandwidth>:<SF\_Datarate>:<crNum>:<CodingRate>:<LNA>:<PaBoost>:<Modulation>:<fskDeviation>:<lowDrOpt>:<BTproduct>:<Preamble>:<CRC>:<IQ Inversion>

Purpose: Allows the user to access the LoRa® configuration test.

**<Freq>** a decimal string representing communication frequency in Hz, it can be values from **150000000** to **960000000**.

**<Power>** a decimal string representing transmitting power in dBm, it can be from **-9** to **22**.

**<Bandwidth>** a decimal string representing signal bandwidth in kHz, if for LoRa it can be: **125, 250, 500**. If for FSK it can be **4800 to 467000**.

**<SF\_Datarate>** a decimal string representing spreading factor used for communication, if for LoRa modulation it can be: **5, 6, 7, 8, 9, 10, 11 and 12**. If for FSK modulation it can be **600 to 300000**.

**<crNum>** a decimal string representing CR number, always be number **4**.

**<CodingRate>** a decimal string representing coding rate, can be: **5, 6, 7, 8**.

**<LNA>** a decimal string representing Low Noise Amplifier, can be **0(off)** or **1(on)**.

**<PaBoost>** a decimal string representing Power Amplifier, can be **0(off)** or **1(on)**. **(Not use)**

**<Modulation>** a decimal string representing modulation, can be **0(FSK)**, **1(LoRa)**.

**<fskDeviation>** a decimal string representing Frequency deviation(Fdev), can be **600 to 200000**. **FSK only**.

**<lowDrOpt>** a decimal string representing Low Data Rate Optimization, can be **0**(off), **1**(on), **2**(Auto). This parameter is usually set when the LoRa® symbol time is equal or above 16.38 ms (typically for SF11 with BW125 and SF12 with BW125 and BW250). **LoRa only.**

**<BTproduct>** a decimal string representing Bandwidth-Time bit period product, can be **0**(no Gaussian Filter Applied), **1**(Gaussian BT=0.3), **2**(Gaussian BT=0.5), **3**(Gaussian BT=0.7), **4**(Gaussian BT=1). **FSK only.**

**<Preamble>** a decimal string representing preamble length, it can be from **8** to **65535**.

**<CRC>** a decimal string representing the CRC header is **1**(on) or **0**(off).

**<IQ Inversion>** a decimal string representing the Invert IQ functionality is **1**(on) or **0**(off).

Response: **Ok**, if input arguments are valid.

**AT\_ERROR**, generic error.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+TCNF?	-	AT+TCNF: Config RF parameters.	OK	Provide a short help.
AT+TCNF=?	Void	Void	OK AT_ERROR	Get the value.
AT+TCNF=<Param>	Void	Void	OK AT_PARAM_ERROR	Set the value.

Example:

- Get the RF parameters.

```
AT+TCNF=?
1: Freq= 868000000 Hz
2: Power= 14 dBm
3: Bandwidth= 125000 Hz
4: SF= 12
5: CR= 4/5
6: LNA State= 0
7: PA Boost State= 0
8: modulation LORA
9: Frequency deviation not applicable
10: LowDrOpt[0 to 2]= 2
11: BT product not applicable
12: Preamble= 12 bits
13: CRC State= 1
14: IQ_Inversion State= 0
can be copy/paste in set cmd: AT+TCNF=868000000:14:125000:12:4/5:0:1:25000:2:3:12:1:0
OK
```

- Set the RF parameters.
  - 1: Freq= 923000000 Hz
  - 2: Power= 14 dBm
  - 3: Bandwidth= 125000 Hz
  - 4: SF= 12
  - 5: CR= 4/5
  - 6: LNA State= 0
  - 7: PA Boost State= 0
  - 8: modulation LORA
  - 9: Frequency deviation not applicable
  - 10: LowDRopt[0 to 2]= 2
  - 11: BT product not applicable
  - 12: Preamble= 12 bits
  - 13: CRC State= 1
  - 14: IQ\_Inversion State= 0

```
AT+TCONF=923:14:125:12:4/5:0:0:1:25000:2:3:13:1:0
```

```
OK
```

```
AT+TCONF=?
```

```
1: Freq= 923000000 Hz
```

```
2: Power= 14 dBm
```

```
3: Bandwidth= 125000 Hz
```

```
4: SF= 12
```

```
5: CR= 4/5
```

```
6: LNA State= 0
```

```
7: PA Boost State= 0
```

```
8: modulation LORA
```

```
9: Frequency deviation not applicable
```

```
10: LowDRopt[0 to 2]= 2
```

```
11: BT product not applicable
```

```
12: Preamble= 13 bits
```

```
13: CRC State= 1
```

```
14: IQ_Inversion State= 0
```

```
can be copy/paste in set cmd: AT+TCONF=923000000:14:125000:12:4/5:0:0:1:25000:2:3:13:1:0
```

```
OK
```

### 2.3.9 AT+PWR=<MODE>

Purpose: Let ST50H enter low power (STOP2) mode.

<MODE> Low power mode, it can be string **STOP2**.

Note1: Just supported the "STOP2" low power mode. Wake up by any key input from AT cmd.

Note2: MCU enter "STOP2" low power mode, and RF part enter sleep mode with warm start.

Response: **Ok**, if input arguments are valid, and wake up after.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+PWR?	-	AT+PWR: Let the ST50H enter low power mode	OK	Provide a short help.
AT+PWR=<MODE>	STOP2	-	OK AT_PARAM_ERROR	Enter low power mode.

Example:

- ST50H enter "STOP2" low power mode.

AT+PWR=STOP2

OK

### 2.3.10 AT+TTONE

Purpose: Allows the user to start TX the RF tone test.

Response: **Ok**, if input arguments are valid.

**AT\_BUSY\_ERROR**: the LoRa® network is busy, so the command has not been completed.

Command	Input parameter	Return value	Return code	Command behavior
AT+TTONE?	-	AT+TTONE: Starts RF Tone test	OK	Provide a short help.
AT+TTONE	-	Tx TONE Test.	OK	Run the command.

Example:

- Start TX tone test.

```
AT+TTONE
Tx TONE Test
OK
```

- Stop TX tone test.

```
AT+TTONE
Tx TONE Test

OK
AT+TOFF
Test Stop
OK
```

### 2.3.11 AT+TTLRA=<Times>:<Interval>:<Data>

Purpose: Allows the user to start the RF Tx LoRa® test and to choose as input the number of packets to be sent.

**<Times>** a decimal string representing how many time of TX counts, it can be values from **0** to **100000**, "0" means TX would not stop until "AT+TOFF" send.

**<Interval>** a decimal string representing LoRa® TX interval in **ms**, it can be values from **0** to **300000**.

**<Data>** a hexadecimal string representing data to be transmitted. The maximum transfer length: **254** bytes

Response: **Ok**, if input arguments are valid.

**AT\_BUSY\_ERROR**: the LoRa® network is busy, so the command has not been completed.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+TTLRA?	-	AT+TTLRA: Set Nb of packets sent with RF Tx test.	OK	Provide a short help.
AT+TTLRA=<Param>	Void	Void	OK AT_BUSY_ERROR AT_PARAM_ERROR	Set the value and run the TX test.

Example:

- TX times is 100, TX interval is 200ms, TX data is 0xaa, 0xaa, 0x55, 0x55 that is 4 bytes.

```
AT+TTLRA=100:200:aaaa5555
>> AT+TTLRA(1)
```

```
OK
OnTxDone
```

```
OnTxDone
```

```
OnTxDone
```

```
OnTxDone
```

Note: (AT+TTLRA(n)) would be only shown when n is 1, 10, 20, 30, ...)

- Stop TX test.

```
AT+TTLRA=100:200:aaaa5555
>> AT+TTLRA(1)
```

```
OK
OnTxDone
```

```
OnTxDone
```

```
OnTxDone
```

```
OnTxDone
```

```
AT+TOFF
```

```
>> AT+TTLRA(5)
Test Stop
```

```
OK
```

Note: When stop, AT+TTLRA(n) would be shown the n meaning the TX total times.

### 2.3.12 AT+TRLRA=<Data>

Purpose: Allows the user to start the RF Rx LoRa® test and to choose as input the data of packets to be received.

<Data> a hexadecimal string representing that demands to be matched. Max length limitation is **254** bytes.

Response: **Ok**, if input arguments are valid.

**AT\_BUSY\_ERROR**: the LoRa® network is busy, so the command has not been completed.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+TRLRA?	-	AT+TRLRA: Set Nb of packets to be received with RF Rx test.	OK	Provide a short help.
AT+TRLRA	-	Void	OK AT_BUSY_ERROR	Run the RX test. RX the any packet.
AT+TRLRA=<Data>	Data string	Void	OK AT_BUSY_ERROR AT_PARAM_ERROR	Set the value and run the RX test. RX the input packet.



Example:

- Start the RX test with input packet data aaaa5555.

```
AT+TRLRA=aaaa5555
Freq=923000000, SForDataRate=12, Bandwidth=4, CodingRate=1
PREAMBLE_LENGTH=13, SYMBOL_TIMEOUT=0(Symbols), CRC=1
OK
```

Note: (AT+TRLRA(n) would be only shown when n is 1, 10, 20, 30, ...)

- Start the RX test with any packet.

```
AT+TRLRA
Freq=923000000, SForDataRate=12, Bandwidth=4, CodingRate=1
PREAMBLE_LENGTH=13, SYMBOL_TIMEOUT=0(Symbols), CRC=1
OK
```

- Stop the RX test.

```
AT+TRLRA=aaaa5555
Freq=923000000, SForDataRate=12, Bandwidth=4, CodingRate=1
PREAMBLE_LENGTH=13, SYMBOL_TIMEOUT=0(Symbols), CRC=1
OK
AT+TOFF
>> AT+TRLRA(0)
Test Stop
OK
```

Note: When stop, AT+TRLRA(n) would be shown the n meaning the RX total times.

### 2.3.13 AT+SUPCH

Purpose: Show the all uplink channels.

Response: **Ok**, if input arguments are valid.

Command	Input parameter	Return value	Return code	Command behavior
AT+SUPCH?	-	AT+SUPCH. Show the all uplink channels	OK	Provide a short help.
AT+SUPCH=?	-	viod	OK	Get the value..

Example:

- Show the all uplink channels.

AT+SUPCH=?

Channel(0):Freq(923200000), Rx1Freq(0), DRmax(5), DRmin(0), Band(0)

Channel(1):Freq(923400000), Rx1Freq(0), DRmax(5), DRmin(0), Band(0)

OK

**2.3.14 AT+CHADD=<ID>:<TX\_Freq>:<RX\_Freq>:<DX\_Max>:<DR\_Min>:<Band>**

Purpose: Add or set up the channel.

<ID> a decimal string representing Channel ID.

<TX\_Freq> a decimal string representing communication frequency in Hz, it can be values from **150000000** to **960000000**.

<RX\_Freq> a decimal string representing communication frequency in Hz, it can be values below **960000000**.

<DX\_Max> a decimal string representing DX Max, it can be values below **16**.

<DR\_Min> a decimal string representing DR Min, it can be values below **16**.

<BAND> : A decimal string representing the band used for LoRaWAN.

**Note : Value will vary depending on the region, please see 2.4.1 for detail.**

Response: **Ok**, if input arguments are valid.

**AT\_ERROR**, generic error.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+CHADD?	-	AT+CHADD. Add or set up the channel	OK	Provide a short help.
AT+CHADD=<Param>	-	-	OK AT_ERROR AT_PARAM_ERROR	Set the value.

Example:

AT+SUPCH=?

Channel(0):Freq(923200000), Rx1Freq(0), DRmax(5), DRmin(0), Band(0)

Channel(1):Freq(923400000), Rx1Freq(0), DRmax(5), DRmin(0), Band(0)

OK

AT+CHADD=2:923500000:0:5:0:0

OK

AT+SUPCH=?

Channel(0):Freq(923200000), Rx1Freq(0), DRmax(5), DRmin(0), Band(0)

Channel(1):Freq(923400000), Rx1Freq(0), DRmax(5), DRmin(0), Band(0)

Channel(2):Freq(923500000), Rx1Freq(0), DRmax(5), DRmin(0), Band(0)

OK

### 2.3.15 AT+CHRE=<ID> or AT+CHRE=<ID\_Head>:<ID\_Tail>

Purpose: Remove the channel by Channel ID.

<ID>,<ID\_Head>,<ID\_Tail> Channel ID and Tail cant smaller than Head.

Response: **Ok**, if input arguments are valid.

**AT\_ERROR**, generic error.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+CHRE?	-	AT+CHRE. Remove the channel by Channel ID	OK	Provide a short help.
AT+CHRE=<Param>	-	viod	OK AT_ERROR AT_PARAM_ERROR	Set the value.

Example:

- Remove the channel by Channel ID.

```

AT+SUPCH=?
Channel(0):Freq(868100000), Rx1Freq(0), DRmax(5), DRmin(0), Band(1)
Channel(1):Freq(868300000), Rx1Freq(0), DRmax(5), DRmin(0), Band(1)
Channel(2):Freq(868500000), Rx1Freq(0), DRmax(5), DRmin(0), Band(1)

OK
AT+CHRE=2

OK
AT+SUPCH=?
Channel(0):Freq(868100000), Rx1Freq(0), DRmax(5), DRmin(0), Band(1)
Channel(1):Freq(868300000), Rx1Freq(0), DRmax(5), DRmin(0), Band(1)

OK

```

### 2.3.16 AT+TRSSI

Purpose: Starts RF RSSI tone test.

Response: **Ok**, if input arguments are valid.

**AT\_ERROR**, generic error.

Command	Input parameter	Return value	Return code	Command behavior
AT+TRSSI?	-	AT+TRSSI. Starts RF RSSI tone test	OK	Provide a short help.
AT+TRSS	-	-	OK AT_ERROR	Run the command.

Example:

```
AT+TRSSI
628s547:Px FSK Test
628s585:>>> RSSI Value= -125 dBm

OK
```

### 2.3.17 AT+TTH=<Fstart>:<Fstop>:<Fdelta>:<PacketNb>:<Data>

Purpose: Starts RF Tx hopping test from Fstart to Fstop in Hz or MHz, Fdelta in Hz.

<Fstart> a decimal string representing communication frequency Start in MHz, it can be values below 1000.

<Fstop> a decimal string representing communication frequency End in MHz, it can be values below 1000.

<Fdelta> a decimal string representing communication hop frequency in Hz.

<PacketNb> a decimal string representing packet number.

<Data> a hexadecimal string representing Data, Maximum is 255.

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

Command	Input parameter	Return value	Return code	Command behavior
AT+ TTH?	-	AT+TTH=<Fstart>:<Fstop>:<Fdelta>:<PacketNb>:<Data><CR>. Starts RF Tx hopping test from Fstart to Fstop in Hz or MHz, Fdelta in Hz	OK	Provide a short help.
AT+ TTH=<Param>	-	Void	OK AT_PARAM_ERROR	Set the value.

Example:

- Start RF Tx hopping test.  

```

AT+TTH=823:824:100000:4:11
3796s736:Tx Hop at 823000000Hz. 1 of 4
OnTxDone

3797s706:Tx Hop at 823100000Hz. 2 of 4
OnTxDone

3798s674:Tx Hop at 823200000Hz. 3 of 4
OnTxDone

3799s643:Tx Hop at 823300000Hz. 4 of 4
OnTxDone

OK

```

### 2.3.18 AT+CERTIF=<MODE>

Purpose: Set the module in LoRaWAN Certification mode.

**<MODE>** : A decimal string representing join mode of LoRaWAN, can be 1 (otaa, over-the-air activation) or 0 (abp, activation by personalization).

Response: **Ok**, if input arguments are valid.

**AT\_PARAM\_ERROR**, if input argument are not valid or out of range.

**AT\_BUSY\_ERROR**, the LoRa® network is busy, so the command has not been completed.

Command	Input parameter	Return value	Return code	Command behavior
AT+CERTIF?	-	AT+CERTIF: Set the module in LoraWan Certification with join Mode (0: ABP, 1: OTAA).	OK	Provide a short help.
AT+CERTIF=<MODE>	0 or 1	-	OK AT_BUSY_ERROR AT_PARAM_ERROR	Set the value.

Example:

- Set the module in LoRaWAN Certification with ABP join Mode

```

AT+CERTIF=0
##### DevAddr: 12266751
##### NwkSKey: 2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C
##### AppSKey: 2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C
+EVT:JOINED
99s712:TX on freq 923400000 Hz at DR 2

OK
--> OnRadioTxDone

100s094:MAC txDone
105s080:RX_1 on freq 923400000 Hz at DR 2
--> OnRadioRxDone rssi(-80) snr(7)

105s430:MAC rxDone
+EVT:JOINED

```

## 2.4 Appendix

### 2.4.1 Region parameters

0: Asia band on 923MHz(AS923)

	MIN	MAX
Frequency(MHz)	915	928
PHY TX DR	0	7
PHY TX DR(UPDWELL)	2	7
RX DR	0	7
RX DR(UPDWELL)	2	7
TX POWER	7	0

1: Australia band on 915MHz(AU915)

	MIN	MAX
Frequency(MHz)	915.2	927.8
PHY TX DR	0	6
PHY TX DR(UPDWELL)	2	6
RX DR	8	13
RX DR(UPDWELL)	2	13
TX POWER	0	14

5: European band on 868MHz(EU868)

	MIN	MAX
Frequency(MHz)		
Band 0	865	868
Band 1	868	868.6
Band 2	863	865
Band 3	869.4	869.65
Band 4	869.7	870
Band 5	868.7	869.2
PHY TX DR	0	7
RX DR	0	7
TX POWER	7	0



## 6: South Korean band on 920MHz(KR920)

	MIN	MAX
Frequency(MHz)	920.9	923.3
PHY TX DR	0	7
RX DR	0	7
TX POWER	7	0

## 7: India band on 865MHz(IN865)

	MIN	MAX
Frequency(MHz)	865	867
PHY TX DR(except 6)	0	7
RX DR(except 6)	0	7
TX POWER	10	0

## 8: North American band on 915MHz(US915)

	MIN	MAX
Frequency(MHz) must be multiple of 600000	902.3	927.5
PHY TX DR	0	4
RX DR	8	13
TX POWER	14	0