ST50H

Commands Set Reference

For SDK

Document Name	**ST50H Commands Set Reference For SDK**
Version	**V1.1.0**
Doc No	
Date	**Aug, 12, 2021**

Document History

Date	Revised Contents	Revised by	Version
Dec, 4, 2020	Initial version.	JC	V0.7.0_rc1
Oct, 30, 2020	Release version.	JC	V0.7.2
Dec, 22, 2020	Release version for SDK v1.0.0	JC	V1.0.0
Aug, 12, 2021	Release version for SDK v1.1.0	Chiachang	V1.1.0

Index

1. Configuration

1.1 Software Configuration

The default baud rate of ST50H 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**

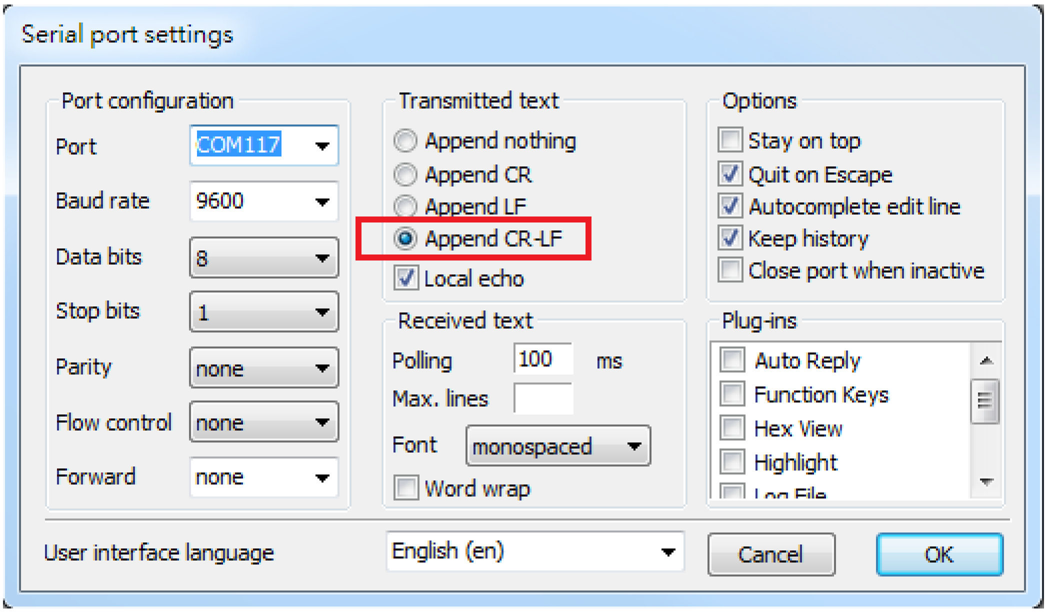
Flow Control: **none**

Forward: **none**

To quickly start using ST50H, the 1st 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).



Port configuration

Port: COM117

Baud rate: 9600

Data bits: 8

Stop bits: 1

Parity: none

Flow control: none

Forward: none

Transmitted text

○ Append nothing

○ Append CR

○ Append LF

● Append CR-LF

☑ Local echo

Received text

Polling: 100 ms

Max. lines:

Font: monospaced

□ Word wrap

Options

□ Stay on top

☑ Quit on Escape

☑ Autocomplete edit line

☑ Keep history

□ Close port when inactive

Plug-ins

□ Auto Reply

□ Function Keys

□ Hex View

□ Highlight

□ Log File

User interface language: English (en)

Figure 1.1

1.2 Command Structure

The syntax rules followed the Hayes AT commands. A command line is made up of three elements: the prefix, the body and the termination character.

- The command line prefix consists of the characters “AT”.
- The body is the basic command and sub-parameter.
- The termination character is default being <CR> with <LF>.

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

Example: AT+CMD1<CR><LF> where AT is the command line prefix, CMD1 is the body of a basic command and <CR><LF> is the command line terminator character.

AT+<CMD>?	: Help on <CMD>
AT+<CMD>	: Run <CMD>
AT+<CMD>=<value>	: Set the value
AT+<CMD>=?	: Get the value

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_CLASS_B_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

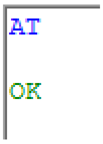
2.1.1 AT

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
```

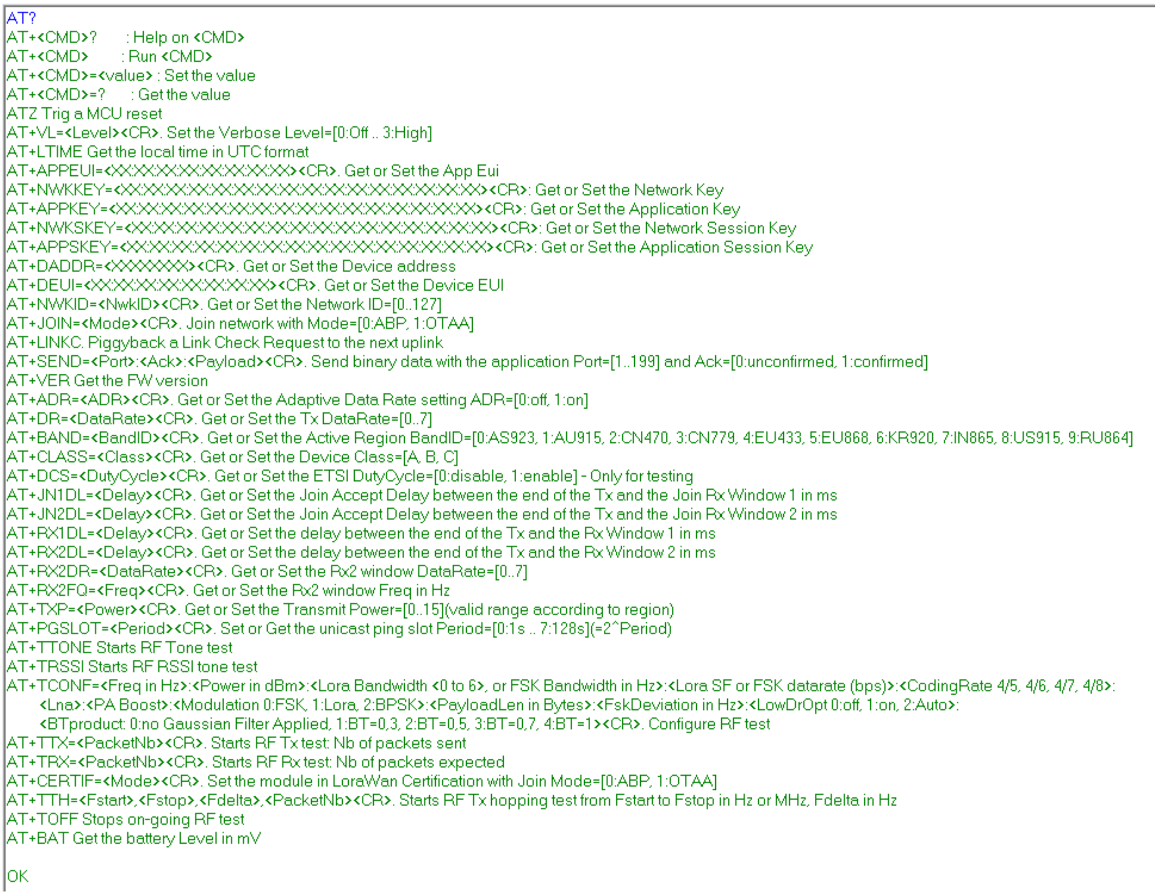
2.1.2 AT?

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

Response: OK.

Command	Input parameter	Return value	Return code	Command behavior
AT?	-	AT+<CMD>?: help on <CMD> AT+<CMD>: run <CMD> AT+<CMD>=<value>: set the value AT+<CMD>=?: get the value <followed by the help of all commands>	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 Trig a MCU reset
AT+VL=<Level><CR>. Set the Verbose Level=[0:Off .. 3:High]
AT+LTIME Get the local time in UTC format
AT+APPEUI=<XX:XX:XX:XX:XX:XX:XX:XX><CR>. Get or Set the App Eui
AT+NWKKEY=<XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX><CR>: Get or Set the Network Key
AT+APPKEY=<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=<XXXXXXXX><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:ABP, 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+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:AU915, 2:CN470, 3:CN779, 4:EU433, 5:EU868, 6:KR920, 7:IN865, 8:US915, 9:RU864]
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+JN1DL=<Delay><CR>. Get or Set the Join Accept Delay between the end of the Tx and the Join Rx Window 1 in ms
AT+JN2DL=<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+TRSSI Starts RF RSSI tone test
AT+TCONF=<Freq in Hz>:<Power in dBm>:<Lora Bandwidth <0 to 6>, or FSK Bandwidth in Hz>:<Lora SF or FSK datarate (bps)>:<CodingRate 4/5, 4/6, 4/7, 4/8>:
     <Lna>:<PA Boost>:<Modulation 0:FSK, 1:Lora, 2:BPSK>:<PayloadLen in Bytes>:<FskDeviation in Hz>:<LowDrOpt 0:off, 1:on, 2:Auto>:
     <BTproduct: 0:no Gaussian Filter Applied, 1:BT=0,3, 2:BT=0,5, 3:BT=0,7, 4:BT=1><CR>. Configure RF test
AT+TTX=<PacketNb><CR>. Starts RF Tx test: Nb of packets sent
AT+TRX=<PacketNb><CR>. Starts RF Rx test: Nb of packets expected
AT+CERTIF=<Mode><CR>. Set the module in LoraWan Certification with Join Mode=[0:ABP, 1:OTAA]
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+BAT Get the battery Level in mV

OK
```

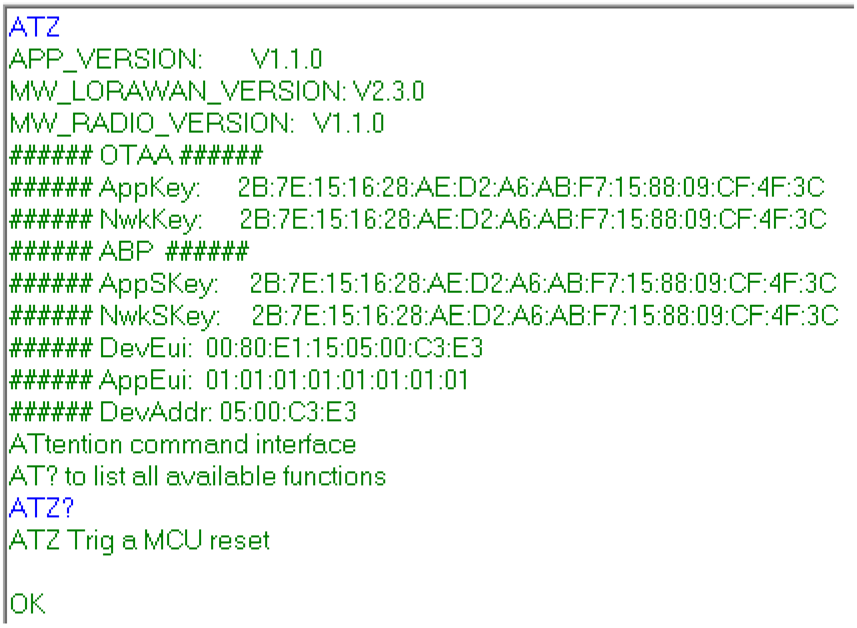
2.1.3 ATZ

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

Response: The beginning information since FW starts.

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

Example:



```
ATZ
APP_VERSION:      V1.1.0
MW_LORAWAN_VERSION: V2.3.0
MW_RADIO_VERSION:   V1.1.0
###### OTAA ######
###### AppKey:     2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
###### NwkKey:     2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
###### ABP  ######
###### AppSKey:    2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
###### NwkSKey:    2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
###### DevEui:  00:80:E1:15:05:00:C3:E3
###### AppEui:  01:01:01:01:01:01:01:01
###### DevAddr: 05:00:C3:E3
ATtention command interface
AT? to list all available functions
ATZ?
ATZ Trig a MCU reset

OK
```

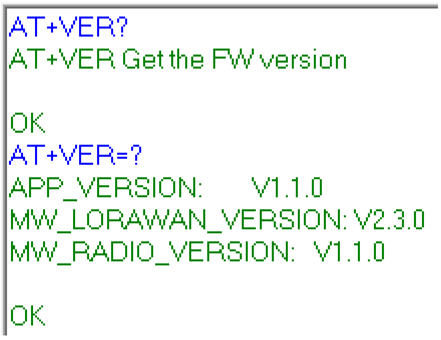
2.1.4 AT+VER

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 version of the FW	OK	Provide a short help.
AT+VER=?	-	V.x.y.z	OK	Get the value.

Example:



```
AT+VER?
AT+VER Get the FW version

OK
AT+VER=?
APP_VERSION:      V1.1.0
MW_LORAWAN_VERSION: V2.3.0
MW_RADIO_VERSION:   V1.1.0

OK
```

2.1.5 AT+VL=<LEVEL>

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

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

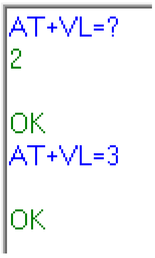
- 0: VLEVEL_OFF
- 1: VLEVEL_L
- 2: VLEVEL_M
- 3: VLEVEL_H

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=3

OK
```

2.1.6 AT+LTIME

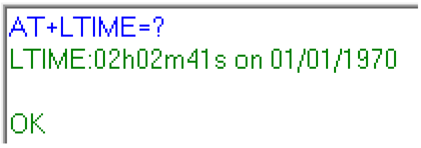
Purpose: Allows the user to get the local time in a UTC format.

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	OK	Provide a short help.

		UTC format.		
AT+LTIME=?	-	LTIME:xxhxxmxxs on DD/MM/YYYY	OK	Get the value.

Example:



```
AT+LTIME=?
LTIME:02h02m41s on 01/01/1970

OK
```

2.2 MAC commands

2.2.1 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^{TM}, 8 hexa separated by ":".

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

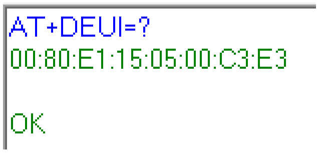
AT_PARAM_ERROR, if input argument are not valid or out of range.

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:

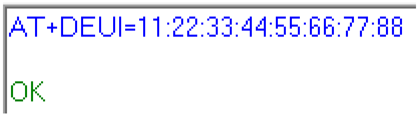
- Get the Device EUI. The Device EUI is 0750363256375020.



```
AT+DEUI=?
00:80:E1:15:05:00:C3:E3

OK
```

- Set the Device EUI is 1122334455667788.



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

OK
```

2.2.2 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^{TM}, 8 hexa separated by “:”.

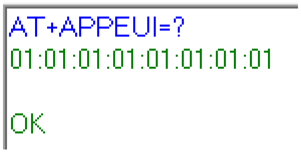
Response: **Ok**, if input arguments are valid.

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:

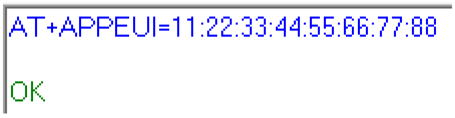
- Get the application identifier EUI. The application identifier EUI is 0101010101010101.



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

OK
```

- Set the application identifier EUI is 1122334455667788.



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

OK
```

2.2.3 AT+DADDR=<ADDR>

Purpose: Allows the user to access the device address.

<ADDR> : A 4-byte hexadecimal string representing device address used for LoRaWAN^{TM}, 4 hexa separated by “:”.

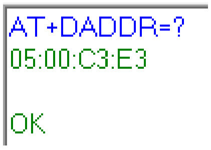
Response: **Ok**, if input arguments are valid.

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:

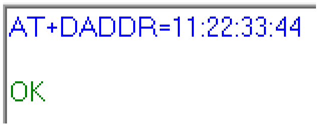
- Get the device address.



```
AT+DADDR=?
05:00:C3:E3

OK
```

- Set the device address is 11223344.



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

OK
```

2.2.4 AT+CLASS=<CLASS>

Purpose: Allow the user to access the LoRaWAN® class.

<CLASS> : A, B or C.

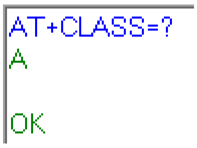
Response: **Ok**, if input arguments are valid.

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	Set the value.

Example:

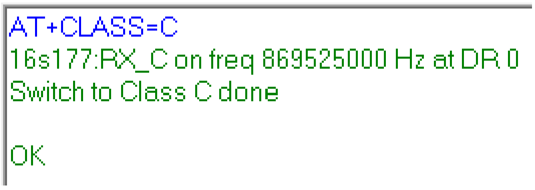
- Get the LoRaWAN class.



```
AT+CLASS=?
A

OK
```

- Set the LoRaWAN class.



```
AT+CLASS=C
16s177:RX_C on freq 869525000 Hz at DR 0
Switch to Class C done

OK
```

2.2.5 AT+JOIN=<MODE>

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).

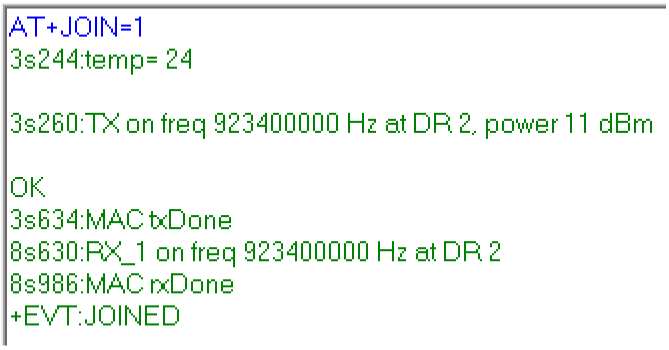
Response: **Ok**, if input arguments are valid.

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.

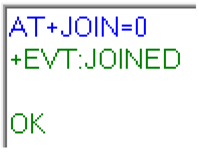


```
AT+JOIN=1
3s244:temp= 24

3s260:TX on freq 923400000 Hz at DR 2, power 11 dBm

OK
3s634:MAC txDone
8s630:RX_1 on freq 923400000 Hz at DR 2
8s986:MAC rxDone
+EVT:JOINED
```

- Join LoRaWAN by ABP.



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

OK
```

2.2.6 AT+TXP=<POWER>

Purpose: Allows the user to access the transmit power.

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

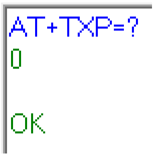
Response: **Ok**, if input arguments are valid.

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=?	-	0, 1, 2, 3, 4 or 5	OK	Get the value.
AT+TXP=<POWER>	0, 1, 2, 3, 4 or 5	-	OK AT_PARAM_ERROR	Set the value.

Example:

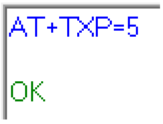
- Get the transmit power.



```
AT+TXP=?
0

OK
```

- Set the transmit power.



```
AT+TXP=5

OK
```

2.2.7 AT+SEND=<PORT>:<ACK>:<PAYLOAD>

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

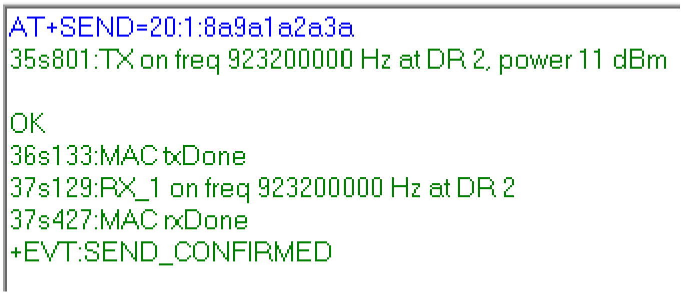
<PORT> : A decimal string representing port number used for transmission, it can be from 1 to 223.
<ACK> : A decimal string representing type of transmitting message, can be 1 (confirmed) or 0 (unconfirmed).
<PAYLOAD> : A hexadecimal string representing data to be transmitted.

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_NETWORK_JOINED	Set the value.

Example:

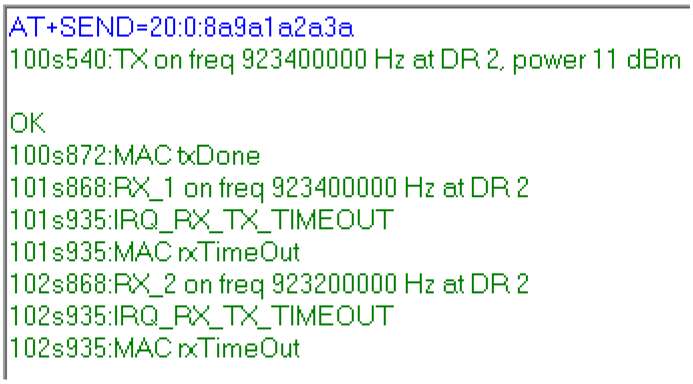
- Send the data with app port 20 and confirmation mode.



```
AT+SEND=20:1:8a9a1a2a3a
35s801:TX on freq 923200000 Hz at DR 2, power 11 dBm

OK
36s133:MAC txDone
37s129:RX_1 on freq 923200000 Hz at DR 2
37s427:MAC rxDone
+EVT:SEND_CONFIRMED
```

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



```
AT+SEND=20:0:8a9a1a2a3a
100s540:TX on freq 923400000 Hz at DR 2, power 11 dBm

OK
100s872:MAC txDone
101s868:RX_1 on freq 923400000 Hz at DR 2
101s935:IRQ_RX_TX_TIMEOUT
101s935:MAC rxTimeOut
102s868:RX_2 on freq 923200000 Hz at DR 2
102s935:IRQ_RX_TX_TIMEOUT
102s935:MAC rxTimeOut
```

2.2.8 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.

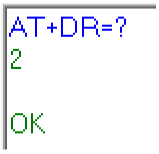
Response: **Ok**, if input arguments are valid.

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=?	-	0 ~ 7	OK	Get the value.
AT+DR=<DR>	0 ~ 7	-	OK AT_PARAM_ERROR	Set the value.

Example:

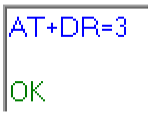
- Get the data rate.



```
AT+DR=?
2

OK
```

- Set the data rate.



```
AT+DR=3

OK
```

2.2.9 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).

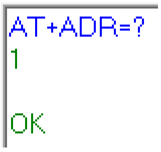
Response: **Ok**, if input arguments are valid.

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:

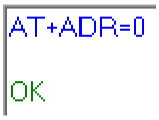
- Get the adaptive data rate setting.



```
AT+ADR=?
1

OK
```

- Set the (turn off) adaptive data rate.



```
AT+ADR=0

OK
```

2.2.10AT+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).

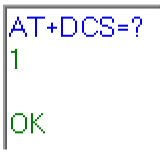
Response: **Ok**, if input arguments are valid.

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:

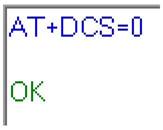
- Get the duty cycle setting.



```
AT+DCS=?
1

OK
```

- Set the (turn off) duty cycle.



```
AT+DCS=0

OK
```

2.2.11AT+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.

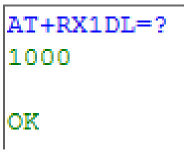
AT_PARAM_ERROR, if input argument are not valid or out of range.

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:

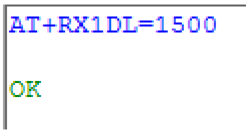
- Get the delay of the received window 1.



```
AT+RX1DL=?
1000

OK
```

- Set the delay of the received window 1.



```
AT+RX1DL=1500

OK
```

2.2.12AT+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.

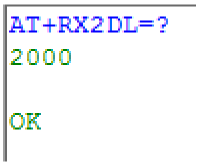
AT_PARAM_ERROR, if input argument are not valid or out of range.

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:

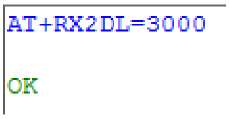
- Get the delay of the received window 2.



```
AT+RX2DL=?
2000

OK
```

- Set the delay of the received window 2.



```
AT+RX2DL=3000

OK
```

2.2.13AT+RX2FQ=<FREQ>

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

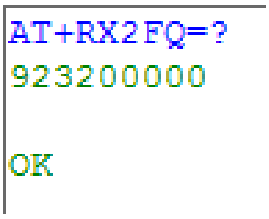
<FREQ>: A decimal string representing operation frequency of specified channel in Hz.

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>	<Frequency in Hz>	-	OK AT_BUSY_ERROR AT_PARAM_ERROR	Set the value.

Example:

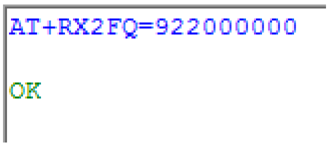
- Get the frequency of the received window 2.



```
AT+RX2FQ=?
923200000

OK
```

- Set the frequency of the received window 2.



```
AT+RX2FQ=922000000

OK
```

2.2.14AT+RX2DR=<DR>

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

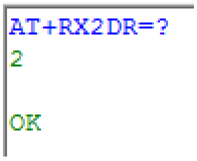
<DR> : A decimal string representing data rate used for LoRaWAN, it can be from 0 to 7.

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=?	-	0 ~ 7	OK AT_BUSY_ERROR	Get the value.
AT+RX2DR=<DR>	0 ~ 7	-	OK AT_PARAM_ERROR AT_BUSY_ERROR	Set the value.

Example:

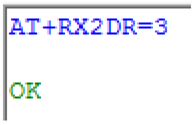
- Get the data rate of received window 2.



```
AT+RX2DR=?
2

OK
```

- Set the data rate of received window 2.



```
AT+RX2DR=3

OK
```

2.2.15AT+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.

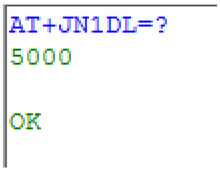
AT_PARAM_ERROR, if input argument are not valid or out of range.

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:

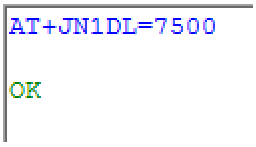
- Get the delay of the join received window 1.



```
AT+JN1DL=?
5000

OK
```

- Set the delay of the join received window 1.



```
AT+JN1DL=7500

OK
```

2.2.16AT+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.

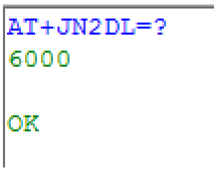
AT_PARAM_ERROR, if input argument are not valid or out of range.

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:

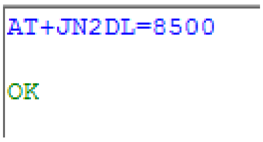
- Get the delay of the join received window 2.



```
AT+JN2DL=?
6000

OK
```

- Set the delay of the join received window 2.



```
AT+JN2DL=8500

OK
```

2.2.17AT+NWKID=<ID>

Purpose: Allows the user to access the Network ID.

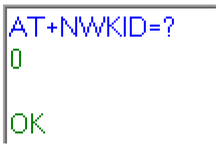
<ID> : A decimal string representing device address used for Network ID.

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=?	-	<integer>	OK	Get the value.
AT+NWKID=<ID>	<integer>	-	OK AT_PARAM_ERROR	Set the value.

Example:

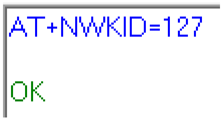
- Get the Network ID. The Network ID is 0.



```
AT+NWKID=?
0

OK
```

- Set the Network ID is 127.



```
AT+NWKID=127

OK
```

2.2.18AT+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：Australian band on 915MHz(AU915)
- 2：Chinese band on 470MHz(CN470)
- 3：Chinese band on 779MHz(CN779)

- 4：European band on 433MHz(EU433)
- 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)
- 9：Russia band on 864MHz(RU864)

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

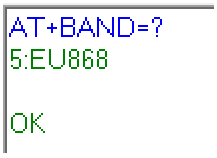
AT_PARAM_ERROR, if input argument are not valid or out of range.

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:

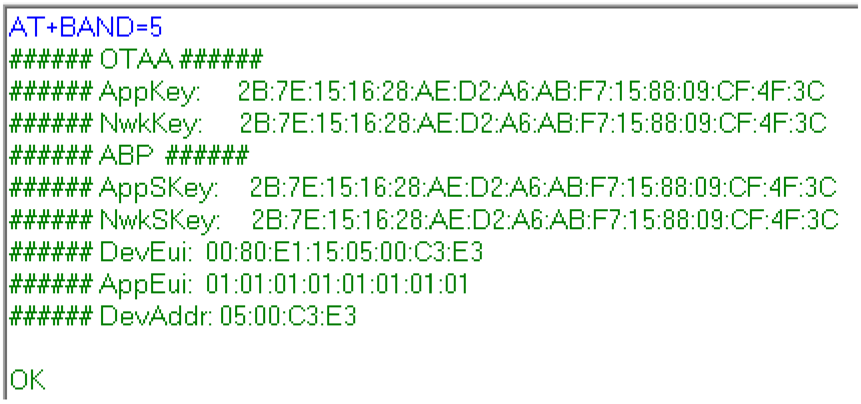
- Get the active region.



```
AT+BAND=?
5:EU868

OK
```

- Set the active region in EU868(European band on 868MHz).



```
AT+BAND=5
###### OTAA ######
###### AppKey:    2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
###### NwkKey:    2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
###### ABP  ######
###### AppSKey:   2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
###### NwkSKey:   2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
###### DevEui:  00:80:E1:15:05:00:C3:E3
###### AppEui:  01:01:01:01:01:01:01:01
###### DevAddr: 05:00:C3:E3

OK
```

2.2.19AT+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.

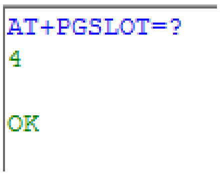
AT_BUSY_ERROR, the LoRa® network is busy, so the command has not been completed.

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:

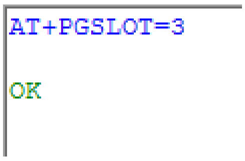
- Get the unicast ping slot used for LoRaWAN Class B.



```
AT+PGSLOT=?
4

OK
```

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



```
AT+PGSLOT=3

OK
```

2.2.20AT+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.

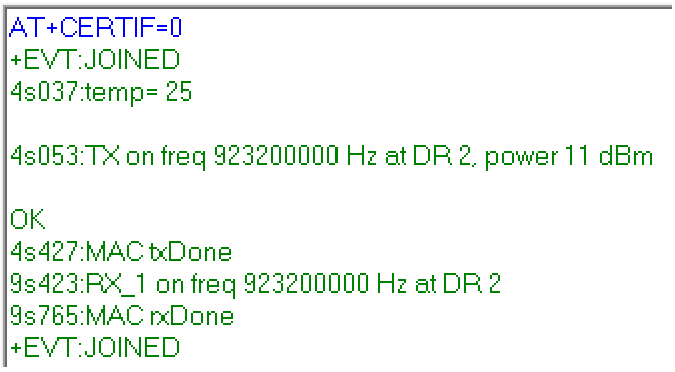
AT_PARAM_ERROR, if input argument are not valid or out of range.

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
+EVT:JOINED
4s037:temp= 25

4s053:TX on freq 923200000 Hz at DR 2, power 11 dBm

OK
4s427:MAC txDone
9s423:RX_1 on freq 923200000 Hz at DR 2
9s765:MAC rxDone
+EVT:JOINED
```