# LD-11
# Operation Manual


**Version: 1.0**

**GlobalSat WorldCom Corporation**

16F., No. 186, Jian 1st Rd, Zhonghe Dist.,

New Taipei City 23553, Taiwan

Tel: 886.2.8226.3799/ Fax: 886.2.8226.3899
service@globalsat.com.tw
www.globalsat.com.tw

**USGlobalSat Incorporated**

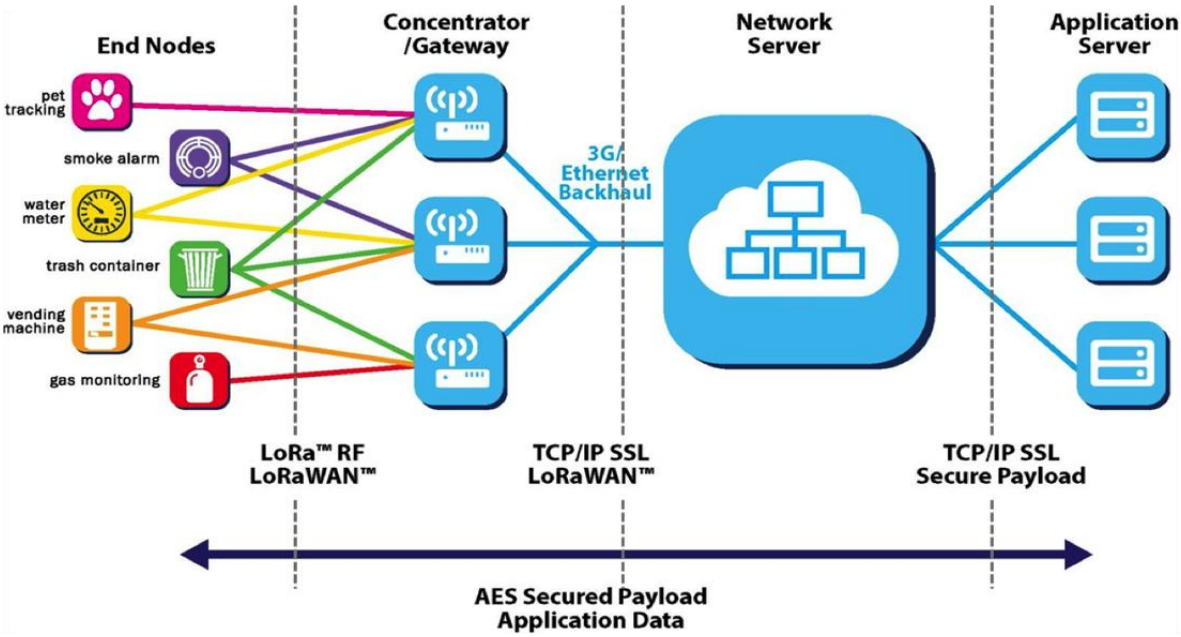
14740 Yorba Court Chino, CA 91710

Tel: 888.323.8720 / Fax: 909.597.8532
sales@usglobalsat.com
www.usglobalsat.com

# CONTENT

# 1. Introduction

**LD-11 is a LoRaWAN™ compliant wireless communication module with MiniPCIe connector interface. It is a low power-consuming and half-duplex module which can wirelessly transmit data to long-distance. It has built-in high-speed, low power-consuming MCU and SX1276 modulation chipset. This chipset is applied with forward error correction technique which greatly improves interference immunity and advances sensitivity. The coding can detect errors and automatically filter out errors and false data. LD-11 fully complies with LoRaWAN™ specification and can work with any LoRaWAN™ compliant gateway. It is suitable for M2M world especially for long-distance transmission or use in harsh environments.**

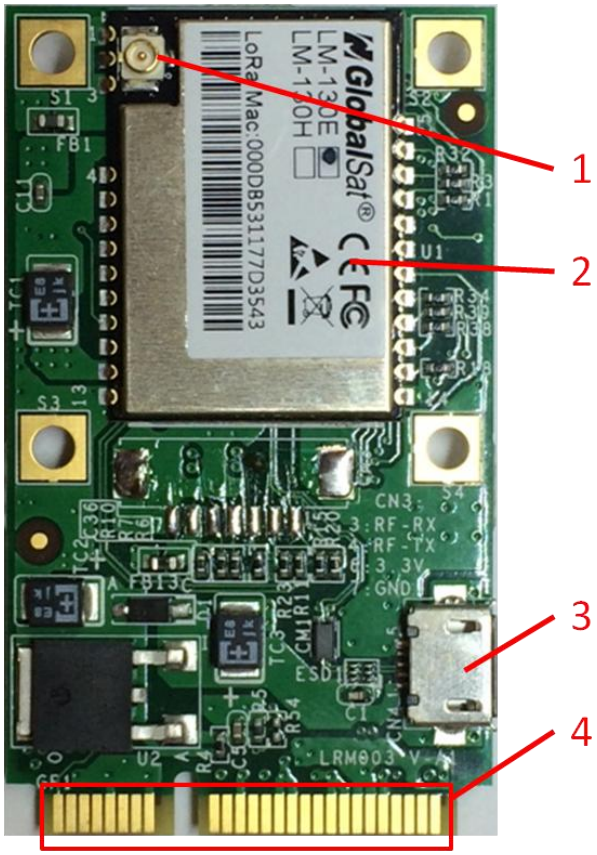



## Features:

- **Mini PCI Express form factor**
- **LoRaWAN™ compliant**
- **Long range transmission (1km to 10km)**
- **2-way duplex communication**
- **High penetration & strong anti-interference capability**

# 2. Setup LD-11

LD-11’s component introduction is as below:



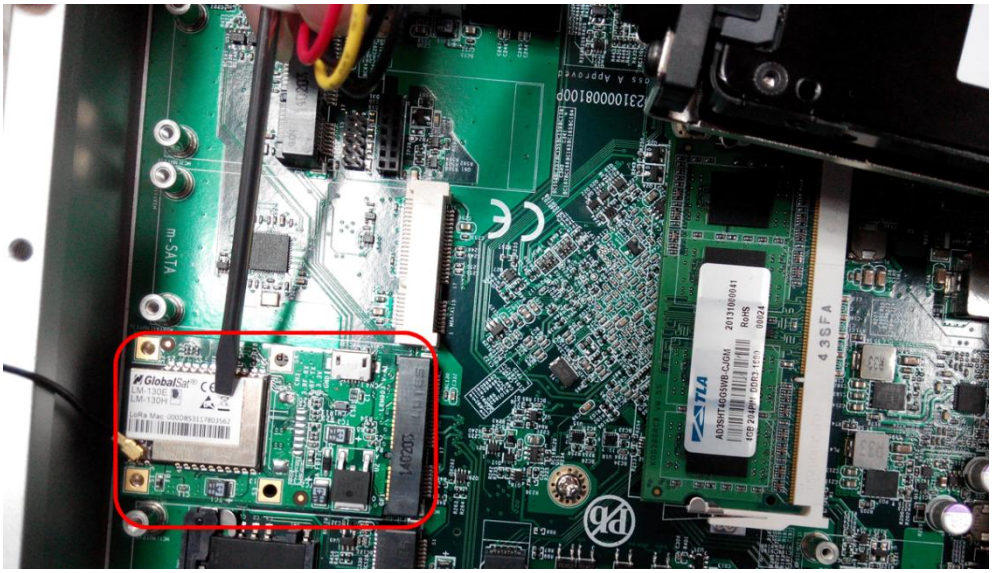

| | | | |
|---|---|---|---|
| 1 | **IPEX antenna connector** | 3 | **Micro USB** |
| 2 | **LM-130 Module** | 4 | **Mini PCIe** |

**To check LD-11's condition in Windows**

1) Please install FT323 driver for micro USB to work in Windows.
2) Please connect LD-11 with PC through micro USB cable.
3) Please install the proper IPEX antenna which supports to your RF frequency.
4) Please refer to "3. Access into network server" to access into network server.

**To Install LD-11 into IPC in Linux**

5) Please insert LD-11 and set it with your mother board properly as below picture.
   Note: If your mPCIe slot has the screws, please remove them first.
   The LD-11 fits in one orientation only. If it doesn't fit, try reversing it.




6) In Linux, please send command (**ls /dev/ttyUSB***) to check whether virtual Com port is. You need to write the program to send AT command through Com port.
7) Please refer to "3. Access into network server" to access into network server.

**Note: Micro USB is used for FW updating and sending AT commands in Windows. The mPCIe is used for sending AT command only in Linux.**

# 3. Access into network server

End-device can be activated by “Over-The-Air Activation” (OTAA) or by “Activation By Personalization” (ABP). You might need to send AT commands to activate OTAA or ABP by serial terminal tools. We recommend “RealTerm”. You could download it at the following link http://realterm.sourceforge.net/.

**Please follow the steps below for activating LD-11 to access into the network server.**

1. **For ABP Mode**
   1) Make sure the gateway already connect to the internet.
   2) Make sure LD-11’s information (DevAddr, AppSKey, NwkSKey) has been successfully registered to the network server.
   3) Use AT command (AAT2 JoinMode=0) to switch LD-11 from OTAA to ABP mode. You would see “ok” when LD-11 is switched to ABP mode. Use AT command (AAT1 Save) to save the setting to the flash. You would see “ok” when the settings are saved. Use AT command (AAT1 Reset) to run the new settings.
   4) Use AT command (AAT2 Tx=2,cnf,aabbccdd) to perform the interop testing. The payload here is “aabbccdd”. You could set other payload uplink to server.
   5) Check if the payload on the network server is correct.
2. **For OTAA Mode**
   1) Make sure the gateway already connect to the internet.
   2) Make sure LD-11’s information (DevEUI, AppEUI, AppKey) has been successfully registered to the network server.
   3) Use AT command (AAT2 JoinMode=1) to switch LD-11 from ABP to OTAA mode. You would see “ok” when LD-11 is switched to ABP mode. Use AT command (AAT1 Save) to save the setting to the flash. You would see “ok” when the settings are saved. Use AT command (AAT1 Reset) to run the new settings.
   4) Use AT command (AAT2 Tx=2,cnf,aabbccdd) to perform the interop testing. The payload here is “aabbccdd”. You could set other payload uplink to server.
   5) Check if the payload on the network server is correct.

**Note:**

The example of our DevEUI, DevAddr and Default key is as following table. **DevEUI and**

**DevAddr would be unique for each device.** DevEUI and DevAddr can be retrieved by following AT commands. (AAT2 DevEui=?, AAT2 DevAddr=?)

| | **US ISM Band** | **EU ISM Band** |
|---|---|---|
| **DevEUI** | 000DB533188A3572 | 000DB5331760356B |
| **DevAddr** | 188A3572 | 1760356B |
| **Default Key** | **1. ABP:**<br>NwkSKey=28AED22B7E1516A609CFABF715884F3C<br>(Hexadecimal number)<br>AppSKey=1628AE2B7E15D2A6ABF7CF4F3C158809<br>(Hexadecimal number)<br><br>**2. OTAA**<br>AppEUI=0000000000010203<br>(Hexadecimal number)<br>AppKey=0123456789ABCDEFEFCDAB8967452301<br>(Hexadecimal number) | |

# 4. Specifications

| | |
|---|---|
| LoRa Module | GlobalSat LM-130 LoRaWAN™ Module |
| RF Band | 868/ 915 MHz |
| USB to UART IC | FTD® 232HL |
| RF Output Power | Max. 20dBm |
| Receiver Sensitivity | -130dBm |
| Current Consumption | Receiving: 89mA (typical)<br>Transmitting: 211mA (typical)<br>Sleeping: 65mA (typical) |
| Dimension | 50 x 29 x 9 mm |
| Interface | Mini PCIe / USB |
| Operation Temperature | -40°C~85°C |
| Storage Temperature | -40°C~85°C |
| Operation Humidity | 5%~95% (non-condensing) |
| System Requirements | Windows® 7/ 8/ 10 and Linux® Ubuntu 12.04 |

# 5. AT Commands

All of the LM-130 module's settings and commands are transmitted over UART using the ASCII interface. All commands need to be terminated with <CR><LF> and any replies they generate will also be terminated by the same sequence.

**After setting LM-130 module, you must use AT command (AAT1 Save) to save the settings to the flash. You would see "ok" when the settings are saved. Then use AT command (AAT1 Reset) to run the new settings.**

The settings for the UART interface are 57600 bps, 8 bits, no parity, 1 stop bit, no flow control.

**Note: All AT commands are case sensitive.**

| Command | Description |
|---|---|
| AAT1 UpdateFW | Upgrade the LM-130 module firmware.<br><br>Response ***ok*** after entering the command. |
| AAT1 Save | All parameters are saved.<br><br>Response ***ok*** after parameters are saved. |
| AAT1 FwVersion | Show up firmware version. |
| AAT1 Reset | Resets and restarts the LM-130 module.<br><br>Response ***ok*** after entering the command. |
| AAT1 SLEEP | Put LM-130 into sleep mode.<br>To leave sleep mode, just Input 0xFF by UART to wake up LM-130.<br><br>Response ***ok*** after entering the command. |
| AAT1 Restore | Restore the defaults of FW.<br><br>Response ***ok*** after entering the command. |
| AAT1 TestMode=[parameter1] | [parameter1]:<br>0: Disable **(Active Report Mode: Off)**<br>1: Enable **(Active Report Mode: On)**<br><br>Response:<br>***ok*** if value is valid<br>***invalid_param*** if parameter1 is not valid |

| Command | Description |
|---|---|
| | **This command sets the state of the active report mode for the module.** |
| AAT1 TestMode=? | Response:<br>0: disable (Active Report Mode: Off)<br>1: enable (Active Report Mode: On)<br><br>**This command will return the state of the active report mode.** |
| AAT2 DevAddr=[parameter1] | [parameter1]: 4-byte hexadecimal number representing the device address, from 00000001 – FFFFFFFF.<br><br>Response:<br>***ok*** if address is valid<br>***invalid_param*** if parameter1 is not valid<br><br>This command configures the module with a 4-byte unique network device address [parameter1]. The [parameter1] must be unique to the current network. This must be directly set solely for activation by personalization devices. This parameter must not be set before attempting to join using over-the-air activation because it will be overwritten once the join process is over. |
| AAT2 DevAddr=? | Response: 4-byte hexadecimal number representing the device address, from 00000001 to FFFFFFFF.<br><br>This command will return present end-device address of the module. |
| AAT2 DevEui=[parameter1] | [parameter1]: 8-byte hexadecimal number representing the device EUI.<br><br>Response:<br>***ok*** if address is valid<br>***invalid_param*** if parameter1 is not valid<br><br>This command sets the globally unique device identifier for the module. The identifier must be set by the host MCU. The module contains a pre-programmed unique EUI and |

| Command | Description |
|---|---|
| | can be retrieved using user provided EUI can be configured using the AAT2 DevEui command. |
| AAT2 DevEui=? | Response: 8-byte hexadecimal number representing the device EUI. This command returns the globally unique end-device identifier, as set in the module. |
| AAT2 AppEui=[parameter1] | [parameter1]: 8-byte hexadecimal number representing the application EUI.<br><br>Response:<br>***ok*** if address is valid<br>***invalid_param*** if parameter1 is not valid<br><br>This command sets the application identifier for the module. |
| AAT2 AppEui=? | Response: 8-byte hexadecimal number representing the application EUI. This command will return the application identifier for the module. The application identifier is a value given to the device by the network. |
| AAT2 NwkSKey=[parameter1] | [parameter1]: 16-byte hexadecimal number representing the network session key.<br><br>Response:<br>***ok*** if address is valid<br>***invalid_param*** if parameter1 is not valid<br><br>This command sets the network session key for the module. This key is 16 bytes in length, and should be modified with each session between the module and network. The key should remain the same until the communication session between devices is terminated. |
| AAT2 NwkSKey=? | Response: [parameter1]: 16-byte hexadecimal number representing the network session key.<br><br>This command sets the network session key for the module. |
| AAT2 AppSKey=[parameter1] | [parameter1]: 16-byte hexadecimal number representing the application session key.<br><br>Response: |

| Command | Description |
|---|---|
| | ***ok*** if address is valid<br>***invalid_param*** if parameter1 is not valid<br><br>This command sets the application session key for the module. This key is unique, created for each occurrence of communication, when the network requests an action taken by the application. |
| AAT2 AppSKey=? | Response: [parameter1]: 16-byte hexadecimal number representing the application session key.<br><br>This command sets the application session key for the module. |
| AAT2 AppKey=[parameter1] | [parameter1]: 16-byte hexadecimal number representing the application key.<br><br>Response:<br>***ok*** if address is valid<br>***invalid_param*** if parameter1 is not valid<br><br>This command sets the application key for the module. The application key is used to identify a grouping over module units which perform the same or similar task. |
| AAT2 AppKey=? | Response: [parameter1]: 16-byte hexadecimal number representing the application key.<br><br>This command sets the application key for the module. |
| AAT2 ADR=[parameter1] | [parameter1]:<br>0: disable<br>1: enable<br><br>Response:<br>***ok*** if address is valid<br>***invalid_param*** if parameter1 is not valid<br><br>This command sets if the adaptive data rate (ADR) is to be enabled, or disabled. The server is informed about the status of the module's ADR in every uplink frame it receives from the ADR field in uplink data packet. If ADR is |

| Command | Description |
|---|---|
| | enabled, the server will optimize the data rate and the transmission power of the module based on the information collected from the network. |
| AAT2 ADR=? | Response:<br>0: disable<br>1: enable<br><br>This command will return the state of the adaptive data rate mechanism. |
| AAT1 EVK_TxCycle=[parameter1] | [parameter1]: decimal number representing the report interval in seconds, from 1 to 254.<br>This command will only take effect when “TestMode”=1.<br><br>Response:<br>***ok*** if parameter1 is valid<br>***invalid_param*** if parameter1 is not valid<br><br>This command sets the report interval for the module. |
| AAT1 EVK_TxCycle=? | Response: decimal number representing the interval, in seconds, for EVK_TxCycle, from 1 to 254.<br><br>This command will return the interval, in seconds, for EVK_TxCycle. |
| AAT2 JoinMode=[parameter1] | [parameter1]:<br>0: ABP mode<br>1: OTAA mode<br><br>Response:<br>***ok*** if address is valid<br>***invalid_param*** if parameter1 is not valid<br><br>This command informs the ***module activation type.*** |
| AAT2 JoinMode=? | Response:<br>0: ABP mode<br>1: OTAA mode<br><br>This command will return the ***activation type*** of module. |

| Command | Description |
|---|---|
| AAT2 reTx=[parameter1] | [parameter1]: decimal number representing the number of retransmissions for an uplink confirmed packet, from 0 to 8.<br><br>Response:<br>***ok*** if address is valid<br>***invalid_param*** if parameter1 is not valid<br><br>This command sets the number of retransmissions to be used for an uplink confirmed packet, if no downlink acknowledgment is received from the server. |
| AAT2 reTx=? | Response: decimal number representing the number of retransmissions, from 0 to 8.<br><br>This command will return the currently configured number of retransmissions which are attempted for a confirmed uplink communication when no downlink response has been received. |
| AAT2 RxDelay1=[parameter1] | [parameter1]: decimal number representing the delay between the transmission and the first reception window in microseconds, from 100000 to 10000000.<br><br>Response:<br>***ok*** if address is valid<br>***invalid_param*** if parameter1 is not valid<br><br>This command will set the delay between the transmission and the first reception window to the [parameter1] in microseconds. The delay between the transmission and the second Reception window is calculated in software as the delay between the transmission and the first Reception window + 1000000 (µs). |
| AAT2 RxDelay1=? | Response: decimal number representing the interval, in microseconds, for RxDelay1.<br><br>This command will return the interval, in microseconds, for RxDelay1. |

| Command | Description |
|---|---|
| AAT2 Tx=[parameter1],<br>[parameter2],<br>[parameter3] | [parameter1]: decimal number representing the port number, from 1 to 223.<br>[parameter2]: string representing the uplink payload type, either ***"cnf"*** or ***"uncnf".*** (cnf = confirmed, uncnf = unconfirmed)<br>[parameter3]: hexadecimal number representing the payload value.<br>The length of [parameter3] bytes capable of being transmitted are dependent upon the set data rate. (Please refer to the LoRaWAN™ Specification for further details)<br><br>Response: This command may reply with two responses. The first response will be received immediately is valid (ok reply received), the second reply will be received after the end of the uplink transmission. (Please refer to the the LoRaWAN™ Specification for further details.)<br><br>Response after entering the command:<br>● ***ok*** - if parameters and configurations are valid.<br>● ***Invalid_param*** – if parameters ( [parameter1],[parameter2],[parameter3]) are not valid.<br>● ***Tx_ok*** - if "cnf" radio Tx return with ACK<br>● ***Tx_ok*** - if "uncnf" radio Tx return<br>● ***Tx_noACK*** – if "cnf" radio Tx return without ACK<br>● ***Rx < parameter1> < parameter2>*** – if transmission was successful, [parameter1] port number, from 1 to 223; [parameter2] hexadecimal number that was received from the server. |
| AAT2 DutyCycle=[parameter1] | [parameter1]:<br>0: disable<br>1: enable<br><br>Response: |

| Command | Description |
|---|---|
| | ***ok*** if address is valid<br>***invalid_param*** if parameter1 is not valid<br><br>This command sets the state of the duty cycle. You could set the ratio of duty cycle by **AAT2 Tx_Band=[parameter1],[parameter2],[parameter3]**. For the details, please refer to the command above. |
| AAT2 DutyCycle=? | Response:<br>0: disable<br>1: enable<br><br>This command will return the state of the duty cycle. |
| AAT2 PLCheck=[parameter1] | [parameter1]:<br>0: disable<br>1: enable<br><br>Response:<br>***ok*** if address is valid<br>***invalid_param*** if parameter1 is not valid<br><br>This command sets the payload length's verification. |
| AAT2 PLCheck=? | Response:<br>0: disable<br>1: enable<br><br>This command will return the state of payload length's verification. |
| AAT2 Rx2_Freq_DR= [parameter1],[parameter2] | [parameter1]: decimal number representing Rx2 frequency, from 000000001 to 999999999 in Hz.<br><br>[parameter2]: decimal number representing Rx2 Data Rate, from 0 to 15.<br><br>Response:<br>***ok*** if address is valid<br>***invalid_param*** if parameter1 is not valid |

| Command | Description |
|---|---|
| | This command sets the frequency and Data Rate of Rx2. |
| AAT2 Rx2_Freq_DR=? | Response: decimal number representing the frequency and Data Rate of RX2.<br><br>This command will return the frequency and Data Rate of RX2. When RX2 frequency is 915MHz and Data Rate is 3, the response message is as “Freq.915000000, DR3”. |
| AAT2 ClassMode=[parameter1] | [parameter1]:<br>0: Class A<br>2: Class C<br><br>This command sets the operation mode of the module. |
| AAT2 ClassMode=? | Response:<br>0: Class A<br>2: Class C<br><br>This command will return the operation mode of module. |
| AAT2 Rx1DrOffset=[parameter1] | [parameter1]: decimal number representing the index of Rx1 offset, from 0 to 3.<br><br>This command sets the index of Rx1 offset. The Rx1DrOffset sets the offset between the uplink data rate and the downlink data rate used to communicate with the end-device on the first reception slot (Rx1). As a default this offset is 0. The offset is used to take into account maximum power density constraints for base stations in some regions and to balance the uplink and downlink radio link margins. |
| AAT2 Rx1DrOffset=? | Response: decimal number representing the index of Rx1 offset.<br><br>This command will return the index of Rx1 offset. |
| AAT2 Tx_Channel= [parameter1],[parameter2] [parameter3],[parameter4] [parameter5] | [parameter1]: decimal number representing the channel number. The range for US is from 0 to 71. The range for EU is from 0 to 15.<br>[parameter2]: decimal number representing the frequency of TX channel, from 000000001 to 999999999 in Hz. |

| Command | Description |
|---|---|
| | [parameter3]: 2 digit decimal numbers representing the operating range of Data Rate. (The left one is DR's Max, the right one is DR's Min.) The range of DR is from 0 to 15. Note: According to LoRaWAN_Regional_Parameter.pdf, Data Rate in some regions will be limited in a particular range. For example, upstream 64 channels numbered 0 to 63 utilizing LoRa 125 kHz BW varying from DR0 to DR3 for US.<br><br>[parameter4]: 0/1 representing the channel is close/open.<br>[parameter5]: decimal number representing the number of band grouping. The range for US is 0. The range for EU is from 0 to 3.<br>Please refer to **AAT2 Tx_Band=[parameter1], [parameter2],[parameter3] for further understanding.**<br><br>Response:<br>***ok*** if address is valid<br>***invalid_param*** if one of parameters is not valid.<br><br>This command sets the frequency, Data Rate, status and the number of band grouping for assigned Tx channel.<br>For example:<br>AAT2 Tx_Channel=3,973300000,40,1,0<br>Description:<br>Modify Channel 3 to have the settings below.<br>973300000 = Frequency<br>40 = DR Range; (4=DR Max, 0=DR Min)<br>1=Status ;(1=open)<br>0= the number of band grouping |
| AAT2 Tx_Channel***x***=? | Response: the assigned Tx channel will be shown. Variable ***x*** needs to be assigned as channel number.<br><br>This command will show the assigned Tx channel.<br>For example:<br>AAT2 Tx_Channel15=?<br>Channel15 message will be shown as below: |

| Command | Description |
|---|---|
| | channel_15,Freq.905300000,DrRange.0-3,Status0, Band0 |
| AAT2 Tx_Band=<br>[parameter1],<br>[parameter2],<br>[parameter3] | [parameter1]: decimal number representing the number of band grouping. The range for US is 0. The range for EU is from 0 to 3.<br>[parameter2]: decimal number representing the value of duty cycle, from 1 to 9999. The real duty cycle could be calculated as (100% / duty cycle value).<br>[parameter3]: decimal number representing the index of Tx power, from 0 to 15.<br><br>Response:<br>***ok*** if address is valid<br>***invalid_param*** if one of parameters is not valid.<br><br>This command sets duty cycle and Tx power for the assigned number of band grouping.<br>For example:<br>AAT2 Tx_Band=0,50,5 (for US)<br>Description:<br>The real duty cycle = 100% / 50 = 2%<br>Modify band grouping 0 to set duty cycle as 2% and Tx power index as 5 (20dBm). |
| AAT2 Tx_Band=? | Response: the list of all Tx bands will be shown.<br><br>This command will show the list of all Tx bands. |
| AAT2 Tx_Band***x***=? | Response: the assigned Tx band will be shown. Variable ***x*** needs to be assigned as band grouping number.<br><br>This command will show the assigned Tx band.<br>For example:<br>AAT2 Tx_Band0=?<br>The message will be shown as below:<br>Band_0, DutyCycle.1, TxPower.5 |
| AAT2 Uplink_Count=? | Response: decimal number representing the number of uplink frame counter.<br><br>This command will return the number of uplink |

| Command | Description |
|---|---|
| | frame counter. |
| AAT2 Downlink_Count=? | Response: decimal number representing the number of downlink frame counter.<br><br>This command will return the number of downlink frame counter. |
| AAT2 Tx_Power=<br>[parameter1],[parameter2] | [parameter1]: decimal number representing the index of Tx power from 0 to 15.<br>[parameter2]: decimal number representing the real TxPower. The range for US is 0 to 30. The range for EU is from 0 to 20.<br><br>Response:<br>***ok*** if address is valid<br>***invalid_param*** if one of parameters is not valid<br><br>This command sets the index of Tx power and the real Tx power for Tx power table. |
| AAT2 Tx_Power=? | Response: the Tx power table will be shown.<br><br>This command will show Tx power table. |
| AAT2 Tx_Power***x***=? | Response: the real power of assigned Tx power index will be shown. Variable ***x*** needs to be assigned as Tx power index.<br><br>This command will show the real power of the assigned Tx power index.<br>For example:<br>AAT2 Tx_Power2=?<br>The message will be shown as below:<br>TxPower_2, 26 dBm. |
| AAT2 Pl_Max_Length=<br>[parameter1],[parameter2] | [parameter1]: decimal number representing Data Rate from 0 to 15.<br>[parameter2]: decimal number representing maximum application payload length (N) from 0 to 255.<br><br>Response:<br>***ok*** if address is valid<br>***invalid_param*** if one of parameters is not valid |

| Command | Description |
|---|---|
| | This command sets maximum application payload length (N) **(without repeater)** according to the assigned Data Rate. |
| AAT2 Pl_Max_Length=? | Response: the list of all Pl_Max_Length settings will be shown.<br><br>This command will show the list of all Pl_Max_Length settings. |
| AAT2 Pl_Max_Length***x***=? | Response: the list of assigned Pl_Max_Length setting will be shown. Variable ***x*** needs to be assigned as the level of Data Rate.<br><br>This command will show the assigned Pl_Max_Length setting. |
| AAT2 Plre_Max_Length= [parameter1],[parameter2] | [parameter1]: decimal number representing DataRate from 0 to 15.<br>[parameter2]: decimal number representing maximum application payload length (N) from 0 to 255.<br><br>Response:<br>***ok*** if address is valid<br>***invalid_param*** if one of parameters is not valid<br><br>This command sets maximum application payload length (N) **(with repeater)** according to the assigned Data Rate. |
| AAT2 Plre_Max_Length=? | Response: the list of all Plre_Max_Length settings will be shown.<br><br>This command will show the list of all Plre_Max_Length settings. |
| AAT2 Plre_Max_Length***x***=? | Response: the list of assigned Plre_Max_Length setting will be shown. Variable ***x*** needs to be assigned as the level of Data Rate.<br><br>This command will show the assigned Plre_Max_Length setting. |