

AT Commands

We recommend “RealTerm” tool to send AT command. You could download it at the following link <http://realterm.sourceforge.net/>.

The sensor node’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.

When you set the parameters, please use AT command (DEV+DTX_STOP) to stop the transmission for the sensor node. Otherwise, it may cause the command failed.

- For AT commands with header “DEV”, they would take effect immediately after setting.
- For AT commands with header “AAT”, you must use AT command (AAT1 Save) to save the settings to the flash after setting. 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
DEV+DTX_START	Data transmission starts. It always starts after reboot. Response <i>data_tx_started</i> after entering the command.
DEV+DTX_STOP	Data transmission stops. Response <i>data_tx_stopped</i> after entering the command.
DEV+DTX_STATE?	This command will return the state of the data transmission. Response <i>data_tx_started</i> or <i>data_tx_stopped</i> .
DEV+DTX_RATE=[parameter1]	[parameter1]: decimal number representing report interval in second, from 5 to 65535. The default value is 60. Response: <i>tx_transmission_rate = [parameter1] seconds</i> if parameter1 is valid <i>invalid_parameters</i> if parameter1 is not valid

Command	Description
	This command sets the report interval for the sensor node.
DEV+DTX_RATE?	<p>Response: decimal number representing the interval, in seconds, for DEV+DTX_RATE, from 5 to 65535.</p> <p>This command will return the report interval, in seconds, for the sensor node.</p>
DEV+LORA_PORT=[parameter1]	<p>[parameter1]: decimal number representing the port number, from 1 to 223.</p> <p>Response: lorawan_port=[parameter1] if parameter1 is valid invalid_parameters if parameter1 is not valid</p> <p>This command will set the port number for the sensor node.</p>
DEV+LORA_PORT?	<p>Response: decimal number representing the port number, , for DEV+LORA_PORT, from 1 to 223.</p> <p>This command will return the port number for the sensor node.</p>
DEV+LORA_CNF=[parameter1]	<p>[parameter1]: 0: un-confirmed transmission 1: confirmed transmission</p> <p>Response: cnf_setting=[parameter1] if parameter1 is valid unknown_command if parameter1 is not valid</p> <p>This command sets if the data transmission is to be confirmed, or un-confirmed. If the parameter is out of range, it will response unknown_command.</p>
DEV+LORA_CNF?	<p>Response: 0: disable 1: enable</p> <p>This command will return the current transmission mode.</p>
AAT1 UpdateFW	<p>Upgrade the LM-130 module firmware.</p> <p>Response ok after entering the command.</p>

Command	Description
AAT1 Save	All parameters are saved. Response ok after parameters are saved.
AAT1 FwVersion	Show up firmware version.
AAT1 Reset	Resets and restarts the LM-130 module. Response ok after entering the command.
AAT1 SLEEP	Put LM-130 into sleep mode. To leave sleep mode, just Input 0xFF by UART to wake up LM-130. Response ok after entering the command.
AAT1 Restore	Restore the defaults of FW. Response ok after entering the command.
AAT2 DevAddr=[parameter1]	[parameter1]: 4-byte hexadecimal number representing the device address, from 00000000 – FFFFFFFF. Response: ok if address is valid invalid_param if parameter1 is not valid 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 00000000 to FFFFFFFF. This command will return present end-device address of the module.
AAT2 DevEui=[parameter1]	[parameter1]: 8-byte hexadecimal number representing the device EUI. Response: ok if address is valid invalid_param if parameter1 is not valid

Command	Description
	<p>This command sets the globally unique device identifier for the sensor node. The identifier must be set by the host MCU. The module contains a pre-programmed unique EUI and can be retrieved using user provided EUI can be configured using the AAT2 DevEui command.</p>
AAT2 DevEui=?	<p>Response: 8-byte hexadecimal number representing the device EUI. This command returns the globally unique end-device identifier, as set in the module.</p>
AAT2 AppEui=[parameter1]	<p>[parameter1]: 8-byte hexadecimal number representing the application EUI.</p> <p>Response: ok if address is valid invalid_param if parameter1 is not valid</p> <p>This command sets the application identifier for the sensor node.</p>
AAT2 AppEui=?	<p>Response: 8-byte hexadecimal number representing the application EUI. This command will return the application identifier for the sensor node. The application identifier is a value given to the device by the network.</p>
AAT2 NwkSKey=[parameter1]	<p>[parameter1]: 16-byte hexadecimal number representing the network session key.</p> <p>Response: ok if address is valid invalid_param if parameter1 is not valid</p> <p>This command sets the network session key for the sensor node. 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.</p>
AAT2 NwkSKey=?	<p>Response: [parameter1]: 16-byte hexadecimal number representing the network session key.</p> <p>This command sets the network session key for the sensor node.</p>

Command	Description
AAT2 AppSKey=[parameter1]	<p>[parameter1]: 16-byte hexadecimal number representing the application session key.</p> <p>Response: ok if address is valid invalid_param if parameter1 is not valid</p> <p>This command sets the application session key for the sensor node. This key is unique, created for each occurrence of communication, when the network requests an action taken by the application.</p>
AAT2 AppSKey=?	<p>Response: [parameter1]: 16-byte hexadecimal number representing the application session key.</p> <p>This command sets the application session key for the sensor node.</p>
AAT2 AppKey=[parameter1]	<p>[parameter1]: 16-byte hexadecimal number representing the application key.</p> <p>Response: ok if address is valid invalid_param if parameter1 is not valid</p> <p>This command sets the application key for the sensor node. The application key is used to identify a grouping over module units which perform the same or similar task.</p>
AAT2 AppKey=?	<p>Response: [parameter1]: 16-byte hexadecimal number representing the application key.</p> <p>This command sets the application key for the sensor node.</p>
AAT2 ADR=[parameter1]	<p>[parameter1]: 0: disable 1: enable</p> <p>Response: ok if address is valid invalid_param if parameter1 is not valid</p>

Command	Description
	<p>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 enabled, the server will optimize the data rate and the transmission power of the module based on the information collected from the network.</p>
AAT2 ADR=?	<p>Response: 0: disable 1: enable</p> <p>This command will return the state of the adaptive data rate mechanism.</p>
AAT2 JoinMode=[parameter1]	<p>[parameter1]: 0: ABP mode 1: OTAA mode</p> <p>Response: ok if address is valid invalid_param if parameter1 is not valid</p> <p>This command informs the module activation type.</p>
AAT2 JoinMode=?	<p>Response: 0: ABP mode 1: OTAA mode</p> <p>This command will return the activation type of module.</p>
AAT2 reTx=[parameter1]	<p>[parameter1]: decimal number representing the number of retransmissions for an uplink confirmed packet, from 0 to 10.</p> <p>Response: ok if address is valid invalid_param if parameter1 is not valid</p> <p>This command sets the number of retransmissions to be</p>

Command	Description
	used for an uplink confirmed packet, if no downlink acknowledgment is received from the server.
AAT2 reTx=?	<p>Response: decimal number representing the number of retransmissions, from 0 to 10.</p> <p>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.</p>
AAT2 RxDelay1=[parameter1]	<p>[parameter1]: decimal number representing the delay between the transmission and the first reception window in microseconds, from 100000 to 10000000.</p> <p>Response: ok if address is valid invalid_param if parameter1 is not valid</p> <p>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).</p>
AAT2 RxDelay1=?	<p>Response: decimal number representing the interval, in microseconds, for RxDelay1.</p> <p>This command will return the interval, in microseconds, for RxDelay1.</p>

Contact Information

GlobalSat WorldCom Corporation

16F. No. 186, Jian 1st Rd, Zhonghe Dist., New Taipei City 23553, Taiwan (R.O.C)

Tel: 886.2.8226.3799/ Fax: 886.2.8226.3899

Email: lora@globalsat.com.tw

www.globalsat.com.tw

USGlobalSat Incorporated

14740 Yorba Court Chino, CA 91710

Tel: 888.323.8720 / Fax: 909.597.8532

Email: sales@usglobalsat.com

www.usglobalsat.com