18-SEP-2023 56312E33

RYLR993 AT COMMAND SET

Contents

List of tables

1. General information

The document applies to the REYAX RYLR993 module that are LoRa SOC core-based devices.

It compatible with LoRaWAN® 1.0.3 Specification.

Table 1. Acronyms

Acronym	Definition
ABP	Activation by personalization
ETSI	European telecommunications standards institute
LoRa	Long range radio technology
LoRaWAN	LoRa wide-area network
OTAA	Over-the-air activation
RF	Radio frequency
RSSI	Received signal strength indicator
SNR	Signal-to-noise ratio

2. Overview

The following sections contain the interface description, the AT commands definition, and the description of some use cases and of the embedded software.

3. AT commands

The AT command set is a standard developed by REYAX to control module. The command set consists of a series of short text strings for performing operations such as joining, dataexchange and parameters setting.

The AT commands are used to drive the LoRa module and to send data. The ATcommands are sent through the UART.

- Baud rate: 9600
- Data: 8 bits
- Parity: none
- Stop: 1 bit
- Flow control: none

All commands are of the form AT+XXX, with XXX denoting the command. The following command behaviors are available:

- AT+XXX? provides a short help of the given command (such as `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 (such as `AT+CFS=?`).
- AT+XXX=<value> is used to provide a value to a command (such as `AT+SEND=2:Hello`).

Output of the commands is provided on the UART. The output format is typically:

```
<value><CR><LF>
<CR><LF><Status<CR><LF>
```

Considering:

- <value><CR><LF> is returned when help AT+XXX? and get AT+XXX=? commands are run.
- <CR> and <LF> stands for the carriage return and line feed.
- When no value is returned, then <value><CR><LF> is not returned at all.
- Every command, except ATZ (MCU reset), returns a status string, that is preceded and followed by <CR><LF>. Possible status are:
  - OK: command run correctly without error.
  - AT_ERROR: generic error
  - AT_PARAM_ERROR: parameter of the command is wrong.
  - AT_BUSY_ERROR: LoRa network is busy, so the command could not complete.
  - AT_TEST_PARAM_OVERFLOW: parameter is too long.
  - AT_NO_NETWORK_JOINED: LoRa network is not joined.
  - AT_RX_ERROR: error detection during the reception of the command

More details on each command description and examples are given in the next sections. Each command preceded by # is provided by the host to the module, then the return of the module is printed.

3.1 AT_RX_ERROR

In case of AT_RX_ERROR, the command is corrupted when received in AT_Slave. Hence the command is not run.

However, in case of long commands, some spurious characters can still be in the queue, ready to be processed as a command. So, in case the user receives an AT_RX_ERROR, the user must first send <CR><LF> to purge the queue, and then send back the same command so that it is processed.

Example

```
# AT+APPKEY=2b:7e:15:16:28:ae:d2:a6:ab:f7:15:88:09:cf:4f:3c<CR><LF>
<CR><LF>AT_RX_ERROR<CR><LF>  /* a RX error has been encountered */
<CR><LF>AT_ERROR<CR><LF> /* after the command, AT_Slave have processed "something" which is
not a command - that could result in an error */
# <CR><LF>  /* newline to purge */
<CR><LF>AT_ERROR<CR><LF> /* purge could result in an error */
/* now it is ok to resend the command */
# AT+APPKEY=2b:7e:15:16:28:ae:d2:a6:ab:f7:15:88:09:cf:4f:3c<CR><LF>
```

3.2 AT command overview

Table 2. AT commands

Command	Parameters	Description
General Commands		
AT	None	Check if the interface is available.
AT	[?]	Help of all supported commands.
ATZ	None	Reset
AT+VL	[=verb_lvl], where verb_lvl = [0:3]	Sets/gets the verbose level.
AT+LTIME	[=?]	Gets the local time in UTC format.
Keys, IDs and EUIs management commands		
AT+APPEUI	[=01:02:03:04:05:06:07:08]	Sets/gets the application EUI.
AT+NWKKEY	[=2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C]	Sets/gets the network root key
AT+APPKEY	[=2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C]	Sets/gets the application root key.

Command	Parameters	Description
AT+APPSKEY	[=2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C]	Sets/gets the application session key.
AT+NWKSKEY	[=2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C]	Sets/gets the network session key.
AT+DADDR	[=01:02:0A:0B]	Sets/gets the device address.
AT+DEUI	[=01:23:45:67:89:AB:CD:EF]	Sets/gets the module unique ID.
AT+NWKID	[=127]	Sets/gets the network ID.
LoRa join and send data commands		
AT+JOIN	[=mode] where mode = 0 (ABP) or mode = 1 (OTAA)	Joins the network.
AT+LINKC	-	Piggyback link check MAC command request to the next uplink
AT+SEND	[=port_nb:confirmedmode:data] where confirmedmode = 0 or 1.	Sends packets to the network.
LoRa network management commands		
AT+VER	[=?]	Gets the LoRaWAN version.
AT+ADR	[=adr_enable] where adr_enable = 0 or 1	Sets/gets the adaptive data rate functionality.
AT+DR	[=datarate] where datarate = [0:7]	Sets/gets the data rate.
AT+BAND	[=region] where region = [0:9]	Sets/gets the active region
AT+CLASS	[=class] where class = [A, B or C]	Sets/gets the LoRa class.
AT+DCS	[=dutycycle] where dutycycle = 0 or 1	Sets/gets duty cycle settings.
AT+JN1DL	[=delay] where delay in ms	Sets/gets the join delay on Rx window 1.
AT+JN2DL		Sets/gets the join delay on Rx window 2.
AT+RX1DL		Sets/gets the delay of the Rx window 1.
AT+RX2DL		Sets/gets the delay of the Rx window 2.
AT+RX2DR	[=datarate] where X = [0:7]	Sets/gets data rate of the Rx window 2.
AT+RX2FQ	[=freq] where freq in Hz	Sets/gets the frequency of the Rx window 2.
AT+TXP	[=txpow] where txpow = [0:7]	Sets/gets the transmit power.
AT+PGSLOT	[=periodicity]	Sets/gets the ping slot.
Radio tests commands		
AT+TTONE	None	Sets the RF tone test.
AT+TRSSI		Sets the RF RSSI tone test.
AT+TCONF	[=freq:pow:bw:sf:cr:lna:pa:mod:paylen:freqdev:lowdropt:BT] [=868000000:14:125:12:4/5:0:0:1:255:0:0:0 for example	Sets/gets the config LoRa RF test.
AT+TTX	[=nb_packets_sent]	Sets the number of packets to be sent for PER RF Tx test.
AT+TRX	[=nb_packets_received]	Sets the number of packets to be received for PER RF Rx test.

Command	Parameters	Description
AT+CERTIF	[=mode] where mode = 0 (ABP) or mode = 1 (OTAA)	Sets the module in LoRaWAN certification with join mode.
AT+TTH	[=<Fstart>, <Fstop>, <FDelta>,<PacketNb>]	Starts RF Tx hopping test from Fstart to Fstop (in Hz or MHz), Fdelta in Hz
AT+TOFF	None	Stops RF tests.
Information command		
AT+BAT	None	Gets the battery level.

3.3 Event table

The table below details the events that the AT_Slave application sends as a notification to the host module.

Table 3. Event table

Event	Return value	Description
+EVT:JOINED	None	Notifies an host module has been join on the gateway by OTAA.
+EVT:JOIN FAILED	None	Notifies the host module has not completed the join transaction (ID/Keys error, Tx not received by the gateway, Rx not received or not decrypted). In this case, the AT+JOIN must be recalled.
+EVT:	:<port>:<size>:<payload>	Notifies the host module that an asynchronous frame has been received on a RX window with downlink frame.
+EVT:	RX_<slot>:<DR>:<RSSI>:<SNR>	Notifies the host module that an asynchronous frame has been received on a RX window with downlink parameters.
+EVT:	RX_<slot>:<DR>:<RSSI>:<SNR> :<DMODM>:<GWN>	Notifies the host module that an asynchronous frame has been received on a RX window with extended downlink parameters. This event replaces the previous event when at least one link check request (AT+LINKC) has been executed.
+EVT:SEND_CONFIRMED	None	Notifies the host module that a Tx frame has been acknowledge by the gateway.

3.4 General commands

3.4.1 AT

Description	Attention is used to check if the link is working properly.
Syntax	AT<CR>
Arguments	None
Response	None
Result code	<CR><LF>OK<CR><LF>

Example:

```
/* Example: check the AT link is working properly*/
# AT<CR>
<CR>
OK<CR>
```

3.4.2 AT?

Description	Provides the short help of all supported commands.
Syntax	AT?<CR>
Arguments	None
Response	None
Result code	<CR><LF>OK<CR><LF>

Example:

```
/* Example: Get the short help of ALL AT commands*/
# AT?<CR>
AT+<CMD>?
AT+<CMD>          : Run <CMD>
AT+<CMD>=<value> : Set the value
AT+<CMD>=?       : Get the value
<List of all commands help>
<CR>
OK<CR>
```

3.4.3 ATZ - MCU reset

Description	The command generates a NVIC reset: resets the whole system including radio and microprocessor.
Syntax	ATZ<CR>
Arguments	None
Response	None
Result code	None (NVIC_Reset action)

Example:

```
/* Example: set NVIC system reset */
# ATZ<CR>
APP_VERSION:          V1.1.0<CR>
MW_LORAWAN_VERSION:   V2.3.0<CR>
MW_RADIO_VERSION:  V1.1.0<CR>
######  DevEui:  AA:BB:CC:DD:EE:FF:00:11<CR>
######  AppEui:  01:02:03:04:05:06:07:08<CR>
###### DevAddr: 12:34:56:78<CR>
ATtention command interface<CR>
AT? to list all available functions<CR>
```

Note: The displayed keys by command above after `######` (`DevEUI`, `AppEui`, and `DevAddr`) are just informative and not a command response.

3.4.4 AT+VL - Verbose level

Description	Sets/gets the verbose level of the application.
Syntax	AT+VL=<verbose_leve><CR> AT+VL=?<CR>
Arguments	<verbose_level>, the default is 1 (VLEVEL_L) 0: VLEVEL_OFF 1: VLEVEL_L 2: VLEVEL_M 3: VLEVEL_H
Response	<verbose_level><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Examples:

```
/* Example1: set verbose level */
# AT+VL=3<CR>
<CR>
OK<CR>

/* Example2: get verbose level */
# AT+VL =?<CR>
3<CR>
<CR>
OK<CR>
```

3.4.5 AT+LTIME - Local time in UTC format

Description	Gets the local time in UTC format.
Syntax	AT+LTIME=?<CR>
Arguments	None
Response	<local time><CR><LF>
Result code	<CR><LF>OK<CR><LF>

Example:

```
/* Example: Get the local time in UTC format */
#AT+ LTIME =?<CR>
LTIME:02h14m52s on 01/01/1970<CR>
<CR>
OK<CR> /* module returns the command error code */
```

3.4.6 AT+OPMODE – Set operating mode

Description	Select LoRaWAN mode or REYAX RYLR998 proprietary mode
Syntax	AT+OPMODE=<mode><CR> AT+OPMODE=?<CR>
Arguments	<mode>, the default is 0 (LoRaWAN mode) 0: LoRaWAN mode 1: REYAX RYLR998 proprietary mode ***RYLR993 can't set up the NETWORKID, only accept the default NETWORKID=18 of the RYLR998. When entry this mode please refer to the AT command of RYLR998.**
Response	<mode><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Example:

```
/* Example 1: set operating mode*/
#AT+OPMODE=1<CR>
<CR>
OK<CR>

/* Example 2: get operating mode*/
#AT+VL=?<CR>
1<CR>
<CR>
OK<CR>
```

3.5 Keys, IDs and EUIs management

3.5.1 AT+APPEUI - Application identifier

Description	Sets/gets the application EUI.
Syntax	AT+APPEUI=<id><CR> AT+APPEUI=?<CR>
Arguments	<id>, 8-byte value separated by ":" (hexadecimal format string)
Response	<id><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_ERROR<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Examples:

```
/* Example1: set APP EUI */
# AT+APPEUI=01:02:03:04:05:06:07:08<CR>
<CR>
OK<CR>

/* Example2: get APP EUI */
# AT+APPEUI=?<CR>
01:02:03:04:05:06:07:08<CR>
<CR>
OK<CR>
```

3.5.2 AT+NWKKEY - Network root key

Description	Sets/gets the network root key. This key is used only in OTAA mode.
Syntax	AT+NWKKEY=<key><CR> AT+NWKKEY=?<CR>
Arguments	<id>, 4-byte value separated by ":" (hexadecimal format string)
Response	<key><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_ERROR<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Examples:

```
/* Example1: set NWK Key */
# AT+NWKKEY=2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C<CR>
<CR>
OK<CR>

/* Example2: get NWK Key when #define KEY_EXTRACTABLE 1 */
# AT+NWKKEY=?<CR>
2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C<CR>
<CR>
OK<CR>

/* Example3: get NWK Key when #define KEY_EXTRACTABLE 0 */
# AT+NWKKEY=?<CR>
<CR>
AT_ERROR<CR>
```

3.5.3 AT+APPKEY - Application root key

Description	Sets/gets the application root key. This key is used only in OTAA mode.
Syntax	AT+APPKEY=<key><CR> AT+APPKEY=?<CR>
Arguments	<key>, 16-byte value separated by ":" (hexadecimal format string)
Response	<key><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_ERROR<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Examples:

```
/* Example1: set APP Key */
# AT+APPKEY=2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C<CR>
<CR>
OK<CR>

/* Example2: get APP Key when #define KEY_EXTRACTABLE 1 */
# AT+APPKEY=?<CR>
2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C<CR>
<CR>
OK<CR>

/* Example3: get APP Key when #define KEY_EXTRACTABLE 0 */
# AT+APPKEY=?<CR>
<CR>
AT_ERROR<CR>
```

3.5.4 AT+APPSKEY - Application session key

Description	Sets/gets the application session key. This key is used only in OTAA and APB modes. In OTAA mode, this key is replaced during the derivation process with the application root key and `JoinAccept` response information.
Syntax	`AT+APPSKEY=<key><CR>` `AT+APPSKEY=?<CR>`
Arguments	`<key>`, 16-byte value separated by ":" (hexadecimal format string)
Response	`<key><CR><LF>`

Result code	`<CR><LF>OK<CR><LF>`
	`<CR><LF>AT_ERROR<CR><LF>`
	`<CR><LF>AT_PARAM_ERROR<CR><LF>`

Example:

```
/* Example1: set APP Session Key */
# AT+APPSKEY=2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C<CR>
<CR>
OK<CR>

/* Example2: get APP Session Key when #define KEY_EXTRACTABLE 1 */
# AT+APPSKEY=?<CR>
2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C<CR>
<CR>
OK<CR>

/* Example3: get APP Session Key when #define KEY_EXTRACTABLE 0 */
# AT+APPSKEY=?<CR>
<CR>
AT_ERROR<CR>
```

3.5.5 AT+NWKSKEY - Network session key

Description	Sets/gets the network session key. This key is used in OTAA and ABP modes. In OTAA mode, this key is replaced during the derivation process with the network's root key and `JoinAccept` response information.
Syntax	`AT+NWKSKEY=<key><CR>` `AT+NWKSEY=?<CR>`
Arguments	`<key>`, 16-byte value separated by ":" (hexadecimal format string)
Response	`<key><CR><LF>`

Result code	`<CR><LF>OK<CR><LF>`
	`<CR><LF>AT_ERROR<CR><LF>`
	`<CR><LF>AT_PARAM_ERROR<CR><LF>`

Example:

```
/* Example1: set NWK Session Key */
# AT+NWKSKEY=2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C<CR>
<CR>
OK<CR>

/* Example2: get NWK Session Key when #define KEY_EXTRACTABLE 1 */
# AT+NWKSKEY=?<CR>
2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C<CR>
<CR>
OK<CR>

/* Example3: get NWK Session Key when #define KEY_EXTRACTABLE 0 */
# AT+NWKSKEY=?<CR>
<CR>
AT_ERROR<CR>
```

3.5.6 AT+DADDR - Device address

Description	Sets/gets the device address.
Syntax	`AT+DADDR=<address><CR>` `AT+DADDR=?<CR>`
Arguments	`<address>`, 4-byte value separated by ":" (hexadecimal format string)
Response	`<address><CR><LF>`

Result code	`<CR><LF>OK<CR><LF>`
	`<CR><LF>AT_ERROR<CR><LF>`
	`<CR><LF>AT_PARAM_ERROR<CR><LF>`

Examples:

```
/* Example1: set device address*/
# AT+DADDR=01:02:0A:0B<CR>
<CR>
OK<CR>

/* Example2: get device address*/
# AT+DADDR=?<CR>
01:02:0A:0B<CR>
<CR>
OK<CR>
```

3.5.7 AT+DEUI - Device EUI

Description	Sets/gets the device EUI.
Syntax	AT+DEUI=<EUI><CR> AT+DEUI=?<CR>
Arguments	<EUI>, 8-byte value separated by ":" (hexadecimal format string)
Response	<EUI><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_ERROR<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Examples:

```
/* Example1: set device EUI*/
# AT+DEUI=01:02:03:04:05:06:07:08<CR>
<CR>
OK<CR>

/* Example2: get device EUI */
# AT+DEUI=?<CR>
01:02:03:04:05:06:07:08<CR>
<CR>
OK<CR>
```

3.5.8 AT+NWKID - Network ID

Description	Sets/gets the network ID.
Syntax	AT+NWKID=<id><CR> AT+NWKID=?<CR>
Arguments	<id>, 1-byte decimal value from 0 to 127
Response	<id><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_ERROR<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Examples:

```
/* Example1: set the network ID */
# AT+NWKID=127<CR>
<CR>
OK<CR>

/* Example2: get the network ID */
# AT+NWKID=?<CR>
127<CR>
<CR>
OK<CR>
```

3.6 Join and send data on LoRa network

3.6.1 AT+JOIN - Join LoRa network

Description	Join the LoRa network.
Syntax	AT+JOIN=<mode><CR>
Arguments	<mode> 0: join to a network by ABP 1: join to a network by OTAA
Response	+EVT:JOINED or +EVT:JOIN_FAILED
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Examples:

```
/* Example1: Join a network by ABP */
#AT+JOIN=0<CR>
+EVT:JOINED<CR>   /* event: ABP configuration done. Ready to start Tx */
<CR>
OK<CR>

/* Example2: Join a network by OTAA (Success result) */
#AT+JOIN=1<CR>
<CR>
OK<CR>

+EVT:JOINED<CR>   /* Event : OTAA join successful event */

/* Example3: Join a network by OTAA (Fail result) */
#AT+JOIN=1<CR>
<CR>
OK<CR>

+EVT:JOIN FAILED<CR> /* Event : OTAA join failed event. LoRaWAN network offline or keys not
                        aligned with the network configuration */
```

3.6.2 AT+LINKC - Link check request

Description	Piggyback link check MAC command request to the next uplink. The DemodMargin and NbGateways output information is provided into the extended Rx events +EVT:RX.
Syntax	AT+LINKC<CR>
Arguments	None
Response	None
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Examples:

```
/* Example: Piggyback Link Check Request to the next uplink */
#AT+LINKC<CR>
<CR>
OK<CR>
```

3.6.3 AT+SEND - Send data to LoRa network

Description	Sends application packets with specified and AppPort and payload to LoRaWAN network.
Syntax	AT+SEND=<port>:<ack>:<payload><CR>
Arguments	• <port>: application port to be transmitted • <ack> – 0: unconfirmed message – 1: confirmed message • <payload>: payload in hexadecimal format strings (maximum length is 242 bytes)
Response	+EVT:SEND_CONFIRMED
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF> <CR><LF>AT_DUTYCYLE_RESTRICTED<CR><LF> <CR><LF>AT_NO_NET_JOINED<CR><LF> <CR><LF>AT_BUSY_ERROR<CR><LF> <CR><LF>AT_CRYPTO_ERROR<CR><LF> <CR><LF>AT_ERROR<CR><LF>

Examples:

```
/* Example1: Send a packet to the gateway in unconfirmed mode */
#AT+SEND=2:0:ABCD<CR>/* send a packet : "ABCD", with APP port is 2, unconfirmed message */
<CR>
OK<CR>

/* Example2: Send a packet to the gateway in confirmed mode */
# AT+SEND=10:1:7FFF<CR>/* send a packet : "7FFF", with APP port is 10, confirmed message */
<CR>
OK<CR>

+EVT:SEND_CONFIRMED
```

3.7 LoRa network management

3.7.1 AT+VER - Firmware version

Description	Gets the version of the AT_Slave firmware.
Syntax	APP_VERSION: Vx.y.z<CR><LF> MW_LORAWAN_VERSION: Va.b.c<CR><LF> MW_RADION_VERSION: Vd.e.f<CR><LF>
Arguments	None
Response	<version><CR><LF>
Result code	<CR><LF>OK<CR><LF>

Example:

```
/* Example: Get the Application and Middleware versions */
#AT+VER=?
RYLR993_FW_VERSION:   V0.1.0<CR>
MW_LORAWAN_VERSION:   V2.3.0<CR>
MW_RADION_VERSION:  V1.1.0<CR>
<CR>
OK<CR>
```

3.7.2 AT+ADR - Adaptive data rate functionality

Description	Sets/gets the adaptive data rate functionality.
Syntax	AT+ADR=<enabled><CR> AT+ADR=?<CR>
Arguments	<enabled> • 0: ADR disabled • 1: ADR enabled (default)
Response	<enabled><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Examples:

```
/* Example1: Disable ADR */
#AT+ADR=0<CR>/* Disable ADR*/
<CR>
OK<CR> /* module returns the command error code */

/* Example2: Check ADR status */
# AT+ADR=?<CR>
0<CR>            /* module returns ADR status */
<CR>
OK<CR>/* module returns the command error code */
```

3.7.3 AT+DR - Data rate

Description	Sets/gets the Tx data rate.
Syntax	AT+DR=<data rate><CR> AT+DR=?<CR>
Arguments	<data rate> in the range [0,1,2,3,4,5,6,7]
Response	<data rate><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_ERROR<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Note: To be able to set data rate, the ADR must be disabled.

Examples:

```
/* Example1: Set TX Data Rate */
#AT+DR=2<CR>/* Set TX Data Rate */
<CR>
OK<CR> /* module returns the command error code */

/* Example2: Get Data rate with Adaptive DataRate disabled */
#AT+ADR=?<CR>
0<CR>
<CR>
OK<CR>
# AT+DR=?<CR>
2<CR> /* module returns TX data rate */
<CR>
OK<CR>

/* Example3: Get Data rate with Adaptive DataRate enabled */
#AT+ADR=?<CR>
1<CR>
<CR>
OK<CR>
# AT+DR=?<CR>
<CR>
AT_ERROR<CR>
```

3.7.4 AT+BAND - Active region

Description	Sets/gets the active region.
Syntax	AT+BAND=<band><CR> AT+BAND=?<CR>
Arguments	<band>: number corresponding to active regions 0: AS923 1: AU915 5: EU868 6: KR920 7: IN865 8: US915<default> 9: RU864
Response	<band><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Examples:

```
/* Example1: Set Active region */
#AT+BAND=0<CR>/* Set AS923 as active region*/
<CR>
OK<CR> /* module returns the command error code */

/* Example2: Get Active region */
# AT+BAND=?<CR>
5:EU868<CR>             /* module returns Active region */
<CR>
OK<CR> /* module returns the command error code */
```

3.7.5 AT+CLASS - LoRa class

Description	Sets/gets the LoRa class.
Syntax	AT+CLASS=<class><CR> AT+CLASS=?<CR>
Arguments	<class>: must be A, B or C.
Response	<class><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_ERROR<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF> <CR><LF>AT_NO_CLASS_B_ENABLE<CR><LF> <CR><LF>AT_NO_NET_JOINED<CR><LF>

Examples:

```
/* Example1: Set the LoRa Class */
#AT+CLASS=C<CR>/* Set Class C on device */
<CR>
OK<CR> /* module returns the command error code */

/* Example2: Get the LoRa Class */
# AT+CLASS=?<CR>
C<CR>            /* module returns Active Class */
<CR>
OK<CR> /* module returns the command error code */
```

3.7.6 AT+DCS - Duty cycle settings

Description	Sets/gets the duty cycle settings.
Syntax	AT+DCS=<dutyCycleEnable><CR> AT+DCS=?<CR>
Arguments	<dutyCycleEnable> 0: duty cycle disabled 1: duty cycle enabled
Response	<dutyCycleEnable><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Examples:

```
/* Example1: Enable Duty cycle */
#AT+DCS=1<CR>
<CR>
OK<CR> /* module returns the command error code */

/* Example2: Get Duty cycle */
# AT+DCS=?<CR>
1<CR>             /* module returns Duty cycle */
<CR>
OK<CR> /* module returns the command error code */
```

3.7.7 AT+JN1DL - Join delay on Rx window 1

Description	Sets/gets the join accept delay between the end of the Tx and the join Rx window 1 (in ms).
Syntax	AT+JN1DL=<delay><CR> AT+JN1DL=?<CR>
Arguments	<delay>: value in ms
Response	<delay><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Examples:

```
/* Example1: Set Join Delay on RX window 1*/
#AT+JN1DL=5000<CR>
<CR>
OK<CR> /* module returns the command error code */

/* Example2: Get Join Delay on RX window 1*/
# AT+JN1DL=?<CR>
5000<CR>        /* module returns Join Delay on RX window 1 in ms*/
<CR>
OK<CR> /* module returns the command error code */
```

3.7.8 AT+JN2DL - Join delay on Rx window 2

Description	Sets/gets the join accept delay between the end of the Tx and the join Rx window 2 (in ms).
Syntax	AT+JN2DL=<delay><CR> AT+JN2DL=?<CR>
Arguments	<delay>: value in ms
Response	<delay><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Examples:

```
/* Example1: Set Join Delay on RX window 2*/
#AT+JN2DL=8000<CR>
<CR>
OK<CR> /* module returns the command error code */

/* Example2: Get Join Delay on RX window 2*/
# AT+JN2DL=?<CR>
8000<CR>        /* module returns Join Delay on RX window 2 in ms*/
<CR>
OK<CR> /* module returns the command error code */
```

3.7.9 AT+RX1DL - Delay of the Rx window 1

Description	Sets/gets the delay between the end of the Tx and the Rx window 1 (in ms).
Syntax	AT+RX1DL=<delay><CR> AT+RX1DL=?<CR>
Arguments	<delay>: value in ms
Response	<delay><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Examples:

```
/* Example1: Set Delay on RX window 1*/
#AT+RX1DL=1500<CR>
<CR>
OK<CR> /* module returns the command error code */

/* Example2: Get Delay on RX window 1*/
# AT+RX1DL=?<CR>
1500<CR>        /* module returns Delay on RX window 1 in ms*/
<CR>
OK<CR> /* module returns the command error code */
```

3.7.10 AT+RX2DL - Delay of the Rx window 2

Description	Sets/gets the delay between the end of the Tx and the Rx window 2 (in ms).
Syntax	AT+RX2DL=<delay><CR> AT+RX2DL=?<CR>
Arguments	<delay>: value in ms
Response	<delay><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Examples:

```
/* Example1: Set Delay on RX window 2*/
#AT+RX2DL=2500<CR>
<CR>
OK<CR> /* module returns the command error code */

/* Example2: Get delay on RX window 2*/
# AT+RX2DL=?<CR>
2500<CR>        /* module returns Delay on RX window 2 in ms*/
<CR>
OK<CR> /* module returns the command error code */
```

3.7.11 AT+RX2DR - Data rate of the Rx window 2

Description	Sets/gets the Rx window 2 data rate (0-7 corresponding to DR_X).
Syntax	AT+RX2DR=<datarate><CR> AT+RX2DR=?<CR>
Arguments	<datarate>: value in range [0:15]
Response	<datarate><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Examples:

```
/* Example1: Set RX window 2 Data rate*/
#AT+RX2DR=5<CR>
<CR>
OK<CR> /* module returns the command error code */

/* Example2: Get RX window 2 Data rate */
# AT+RX2DR=?<CR>
5<CR>        /* module returns RX window 2 Data rate */
<CR>
OK<CR> /* module returns the command error code */
```

3.7.12 AT+RX2FQ - Frequency of the Rx window 2

Description	Sets/gets the Rx window 2 frequency.
Syntax	AT+RX2FQ=<freq><CR> AT+RX2FQ=?<CR>
Arguments	<freq>: value in Hz
Response	<freq><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Examples:

```
/* Example1: Set RX window 2 Frequency */
#AT+RX2FQ=869535000<CR>
<CR>
OK<CR> /* module returns the command error code */

/* Example2: Get RX window 2 Frequency */
# AT+RX2FQ=?<CR>
869535000<CR>        /* module returns RX window 2 Frequency */
<CR>
OK<CR> /* module returns the command error code */
```

3.7.13 AT+TXP - Transmit power

Description	Sets/gets the transmit power.
Syntax	AT+TXP=<TxPow><CR> AT+TXP=?<CR>
Arguments	<TxPow>: must be in the range of the region activated in the range [0:15].
Response	<TxPow><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Examples:

```
/* Example1: Set Transmit power */
#AT+TXP=3<CR>
<CR>
OK<CR> /* module returns the command error code */

/* Example2: Get Transmit power */
# AT+TXP=?<CR>
3<CR>             /* module returns Transmit power */
<CR>
OK<CR> /* module returns the command error code */
```

3.7.14 AT+PGSLOT - Ping slot

Description	Sets/gets the unicast ping slot periodicity.
Syntax	AT+PGSLOT=<periodicity><CR> AT+PGSLOT=?<CR>
Arguments	<periodicity>: periodicity to be transmitted, must be in the range [0:7] Ping slot periodicity is $2^{<\text{periodicity}>}$, in seconds.
Response	<periodicity><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Example:

```
/* Example1: Set Ping Slot */
#AT+PGSLOT=4<CR>/* Set Ping Slot periodicity to 2^4= 16 seconds*/
<CR>
OK<CR> /* module returns the command error code */

/* Example2: Set Ping Slot */
#AT+PGSLOT=?<CR>
4<CR>
<CR>
OK<CR>/* module returns the command error code */
```

3.8 Radio test commands

3.8.1 AT+TTONE - RF tone test

Description	Starts a RF tone test.
Syntax	AT+TTONE<CR>
Arguments	None
Response	None
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_BUSY_ERROR<CR><LF>

Example:

```
/* Example: starts a RF Tone test */
# AT+TTONE<CR>
[TimeDisplay]: Tx FSK Test<CR>
<CR>
OK<CR>
```

3.8.2 AT+TRSSI - RF RSSI tone test

Description	Starts a RF RSSI tone test.
Syntax	AT+TRSSI<CR>
Arguments	None
Response	<rssi_lvl><CR><LF>: value in dBm
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_BUSY_ERROR<CR><LF>

Example:

```
/* Example: starts a RSSI tone test */
# AT+TRSSI<CR>
[TimeDisplay]: Rx FSK Test<CR>
[TimeDisplay]:>>> RSSI Value= -7 dBm<CR>
<CR>
OK<CR>
```

3.8.3 AT+TCONF - LoRa RF test configuration

Description	Sets/gets the LoRa RF test configuration.
Syntax	AT+TCONF=<freq>:<pow>:<bw>:<sf>:<cr>:<lna>:<pa>:<mod>:<paylen>:<freq dev>:<lowdropt>:<BT><CR> AT+TCONF=?<CR>

Arguments	• <freq>: frequency in Hz • <pow>: Tx power in range [-9:22] dBm • <bw>: – LoRa (in kHz) ◦ 0: 7.8125 ◦ 1: 15.625 ◦ 2: 31.25 ◦ 3: 62.5 ◦ 4: 125 ◦ 5: 250 ◦ 6: 500 – FSK: 4800 to 467000 Hz • <sf>: – LoRa: SF5 to SF12 bit/s – FSK: 600 to 300000 bit/s • <cr>: LoRa only – 1: 4/5 – 2: 4/6 – 3: 4/7 – 4: 4/8 • <lna>: low-noise amplifier – 0: Off – 1: On • <pa>: PA boost – 0: Off – 1: On • <mod>: modulation – [0: FSK – 1: LoRa – 2: BPSK(Tx) • <paylen>: payload length 1 to 256 • <freqdev>: FSK only 4800 to 467000 • <lowdropt>: low DR optimization, LoRa only – 0: Off – 1: On – 2: Auto (1 when SF11 or SF12, 0 otherwise) • <BT>: FSK only – 0: no Gaussian filter applied – 1: BT = 0,3 – 2: BT = 0,5 – 3: BT = 0,7 – 4: BT = 1

Response	• Freq= <freq> Hz<CR> • Power= <pow> dBm<CR> • Bandwidth= <bw> (=125000 Hz)<CR> • SF= <sf><CR> • CR= <cr> (=4/5)<CR> • LNA State= <lna><CR> • PA Boost State= <pa><CR> • Modulation <mod><CR> • Payload len= <paylen> Bytes<CR> • <freqdev><CR> • LowDRopt[0 to 2]= <lowdropt><CR> • <BT><CR>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Note: *<pa>, <freqdev>, <lowdropt> and <BT> arguments are required by the command syntax but are not used on the B-L072Z-LRWAN1 platform.*

Examples:

```
/* Example1: Set LoRa RF test configuration */
#AT+TCONF=868000000:14:4:12:4/5:0:0:1:16:25000:2:3<CR>
<CR>
OK<CR> /* module returns the command error code */

/* Example2: Get LoRa RF test configuration */
# AT+TCONF=?<CR>
1: Freq= 868000000 Hz<CR>
2: Power= 14 dBm<CR>
3: Bandwidth= 4 (=125000 Hz)<CR>
4: SF= 12<CR>
5: CR= 1 (=4/5)<CR>
6: LNA State= 0<CR>
7: modulation LORA<CR>
8: Payload len= 16 Bytes<CR>
can be copy/paste in set cmd: AT+TCONF=868000000:14:4:12:4/5:0:0:1:16:25000:2:3<CR>
<CR>
OK<CR>
```

3.8.4 AT+TTX - Packets to be sent for PER RF TX test

Description	Starts a PER RF TX test with the number of packets to be sent.
Syntax	AT+TTX=<nb_packets><CR>
Arguments	<nb_packets>
Response	None
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF> <CR><LF>AT_BUSY_ERROR<CR><LF>

Example:

```
/* Example: Starts a PER RF TX test with the number of packets to be sent. */
# AT+TTX=4<CR>
[TimeDisplay]:Tx Test<CR>
[TimeDisplay]:Tx Test: Packet 1 of 4<CR>
[TimeDisplay]:OnTxDone<CR>
[TimeDisplay]:Tx Test: Packet 2 of 4<CR>
[TimeDisplay]:OnTxDone<CR>
[TimeDisplay]:Tx Test: Packet 3 of 4<CR>
[TimeDisplay]:OnTxDone<CR>
[TimeDisplay]:Tx Test: Packet 4 of 4<CR>
[TimeDisplay]:OnTxDone<CR>
<CR>
OK<CR>
```

3.8.5 AT+TRX - Packets to be received for PER RF RX test

Description	Starts a PER RF RX test with the number of packets to be received.
Syntax	AT+TRX=<nb_packets><CR>
Arguments	<nb_packets>
Response	None
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF> <CR><LF>AT_BUSY_ERROR<CR><LF>

Example:

```
/* Example: Starts a PER RF RX test with the number of packets to be received. */
# AT+TRLRA=4<CR>
[TimeDisplay]:PRE OK<CR>
[TimeDisplay]:HDR OK<CR>
[TimeDisplay]:OnRxDone<CR>
[TimeDisplay]:RssiValue=-7 dBm, SnrValue=7<CR>
[TimeDisplay]:Rx: 1 of 4 >>> PER= 0 %<CR> /* PER percentage is updated/displayed after each
reception*/
[TimeDisplay]:PRE OK<CR>
[TimeDisplay]:HDR OK<CR>
[TimeDisplay]:OnRxDone<CR>
[TimeDisplay]:RssiValue=-7 dBm, SnrValue=6<CR>
[TimeDisplay]:Rx: 2 of 4 >>> PER= 0 %<CR> /* PER percentage is updated/displayed after each
reception*/
[TimeDisplay]:PRE OK<CR>
[TimeDisplay]:HDR OK<CR>
[TimeDisplay]:OnRxDone<CR>
[TimeDisplay]:RssiValue=-7 dBm, SnrValue=5<CR>
[TimeDisplay]:Rx: 3 of 4 >>> PER= 0 %<CR> /* PER percentage is updated/displayed after each
reception*/
[TimeDisplay]:PRE OK<CR>
[TimeDisplay]:HDR OK<CR>
[TimeDisplay]:OnRxDone<CR>
[TimeDisplay]:RssiValue=-7 dBm, SnrValue=6<CR>
[TimeDisplay]:Rx: 4 of 4 >>> PER= 0 %<CR> /* PER percentage is updated/displayed after each
reception*/
<CR>
OK<CR>
```

3.8.6 AT+TTH - RF Tx hopping test

Description	Starts RF Tx hopping test from Fstart to Fstop, with Fdelta steps.
Syntax	AT+TTH=<Fstart>,<Fstop>,<FDelta>,<nb_packets><CR>
Arguments	• <Fstart>: frequency start (in Hz or MHz) • <Fstop>: frequency stop (in Hz or MHz) • <FDelta>: frequency bandwidth (in Hz) • <nb_packets>: number of packets to be sent
Response	None
Result code	<CR><LF> OK <CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF> <CR><LF>AT_BUSY_ERROR<CR><LF>

Example:

```
 /* Example: set TX hopping test  from 868 to 868,5 MHz  with 6 steps of 100 kHz */

# AT+TTH=868000000,868500000,100000,6<CR>
[TimeDisplay]: Tx Hop at 868000000Hz. 0 of 6<CR>
[TimeDisplay]:Tx LoRa Test<CR>
[TimeDisplay]:Tx 1 of 1<CR>
[TimeDisplay]:OnTxDone<CR>
[TimeDisplay]:Tx Hop at 868100000Hz. 1 of 6<CR>
[TimeDisplay]:Tx LoRa Test<CR>
[TimeDisplay]:Tx 1 of 1<CR>
[TimeDisplay]:OnTxDone<CR>
[TimeDisplay]:Tx Hop at 868200000Hz. 2 of 6<CR>
[TimeDisplay]:Tx LoRa Test<CR>
[TimeDisplay]:Tx 1 of 1<CR>
[TimeDisplay]:OnTxDone<CR>
[TimeDisplay]:Tx Hop at 868300000Hz. 3 of 6<CR>
[TimeDisplay]:Tx LoRa Test<CR>
[TimeDisplay]:Tx 1 of 1<CR>
[TimeDisplay]:OnTxDone<CR>
[TimeDisplay]:Tx Hop at 868400000Hz. 4 of 6<CR>
[TimeDisplay]:Tx LoRa Test<CR>
[TimeDisplay]:Tx 1 of 1<CR>
[TimeDisplay]:OnTxDone<CR>
[TimeDisplay]:Tx Hop at 868500000Hz. 5 of 6<CR>
[TimeDisplay]:Tx LoRa Test<CR>
[TimeDisplay]:Tx 1 of 1<CR>
[TimeDisplay]:OnTxDone<CR>
<CR>
OK<CR>
```

3.8.7 AT+CERTIF - Module in LoRaWAN certification with join mode

Description	Starts the module in LoRaWAN certification and with the choice of join mode.
Syntax	AT+CERTIF=<mode><CR>
Arguments	<mode> 0: join to a network by ABP 1: join to a network by OTAA
Response	+EVT:JOINED +EVT:JOIN_FAILED
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Examples:

```
/* Example1: Set the module in LoRaWAN certification and Join network by ABP */
#AT+CERTIF=0<CR>
+EVT:JOINED<CR> /* event: ABP configuration done. Ready to start Tx */
<CR>
OK<CR>

/* Example2: Set the module in LoRaWAN certification and Join network by OTAA */
#AT+CERTIF=1<CR>
<CR>
OK<CR>

+EVT:JOINED<CR> /* Event : OTAA join successful event */
```

3.8.8 AT+TOFF - RF test

Description	Stops the RF test.
Syntax	AT+TOFF<CR>
Arguments	None
Response	None
Result code	<CR><LF>OK<CR><LF>

Example:

```
/* Example: stops RF test */
# AT+TOFF<CR>
Test Stop<CR>
<CR>
OK<CR> /* module returns the command error code */
```

3.9 Information

3.9.1 AT+BAT - Battery level

Description	Gets the battery level (in mV).
Syntax	AT+BAT=?<CR>
Arguments	None
Response	<level><CR><LF>: value is in mV
Result code	<CR><LF>OK<CR><LF>

Example:

```
/* Example: Get the battery level in mV */
#AT+ BAT=?<CR>
3300<CR> /* battery level in mV */
<CR>
OK<CR> /* module returns the command error code */
```

3.9.2 AT+TEMP – Temperature

Description	Gets the temperature of module (in Celsius degree °C).
Syntax	AT+TEMP=?<CR>
Arguments	None
Response	<Celsius degree><CR><LF>: value is in Celsius degree.
Result code	<CR><LF>OK<CR><LF>

Example:

```
/* Example: Get the temperature */
#AT+TEMP=?<CR>
25<CR> /* value is in Celsius degree. */
<CR>
OK<CR> /* module returns the command error code */
```

4. Examples

Here are some basic examples that describe how to use the AT commands. In the following sections, commands provided by the host are preceded by `#`, and comments are embraced with `/* */`.

4.1 Join and send in unconfirmed mode

```
/* Check AT Link is OK */
#AT<CR>
<CR>
OK<CR>
/* Join in OTAA mode */
#AT+JOIN=1<CR>
 +EVT:JOINED<CR> /* Event: OTAA join successful event */
<CR>
OK<CR>
/* Network is joined, now data can be sent */
#AT+SEND=50:0:01234ABCD<CR>/* Send hexadecimal values in unconfirmed mode to port 50 */
<CR>
OK<CR>
```

4.2 Join and send in confirmed mode

```
/* Check AT Link is OK */
#AT<CR>
<CR>
OK<CR>
/* Join in OTAA mode */
#AT+JOIN=1<CR>
 +EVT:JOINED<CR> /* Event: OTAA join successful event */
<CR>
OK<CR>
/* Network is joined, now data can be sent */
#AT+SEND=50:1:01234ABCD<CR>/* Send hexadecimal values in confirmed mode to port 50 */
+EVT:SEND_CONFIRMED<CR>
<CR>
OK<CR>
```

4.3 Rx received data

It is possible to retrieve data sent from a specified port, when `+EVT:RX` is received.

```
/* Check AT Link is OK */
#AT<CR>
<CR>
OK<CR>
/* Join in OTAA mode */
#AT+JOIN=1<CR>
JOINED<CR> /* Event: OTAA join successful event */
<CR>
OK<CR>
/* Network is joined, now data can be sent */
#AT+SEND=50:0:01234ABCD<CR>/* Send hexadecimal values in unconfirmed mode to port 50 */
<CR>
OK<CR>
+EVT:50:4:ABCD<CR> /*Receive downlink frame */
+EVT:RX_1, DR 0, RSSI -49, SNR 5 <CR> /*Receive downlink parameters */
```

4.4 Class B enable request

The example below shows how to do a Class B request through an AT command sequence.

```
/* Join request in OTAA mode */
#AT+JOIN=1<CR>
<CR>
OK<CR>
/* wait for few seconds to wait for join to complete */
+EVT:JOINED<CR> /* end-device has joined the network */

/* now the network is joined, a request to enter into a Class B mode can be made */
#AT+CLASS=B<CR> /* Request to switch to Class B "enable" */
OK<CR>

/* A built-in MAC message is sent to the network to acquire the system time "Device Time
Req" */
#AT+SEND=50:0:0123<CR> /* Send data will allow piggybacking the MAC Device Time Req - could
be a dummy message */
OK<CR>

/* --> MAC Ping Device Time ANS is received by end-node in hidden way */
#AT+CLASS=?<CR>
B, S0<CR> /* Beacon Acquisition on-going */
OK<CR>
/* Loop on AT+CLASS=? until Beacon Acquisition on-going */

#AT+CLASS=?<CR>
B, S1<CR> /* Beacon Acquisition locked */
OK<CR>

#AT+PGSLOT=4<CR>/* Set Ping Slot periodicity to 2^4= 16 seconds and Send PingSlotInfoReq */
OK<CR>
/* --> MAC Ping Slot Info ANS is received by end-node in hidden way */

/* now the end-node is Class B "enable" */
#AT+CLASS=?<CR>
B<CR> /*Class B "enable"*/
OK<CR>

/* example: Local Time can be requested */
#AT+LTIME=?<CR>
LTIME:01h01m01s on 01/01/2021<CR>
<CR>
OK<CR>
```

5. Proprietary mode

The REYAX RYLR993 can support REYAX RYLR998 proprietary mode by sending the command “AT+OPMODE” to select operation mode.

Description	Select LoRaWAN mode or REYAX RYLR998 proprietary mode
Syntax	AT+OPMODE=<mode><CR> AT+OPMODE=?<CR>
Arguments	<mode>, the default is 0 (LoRaWAN mode) 0: LoRaWAN mode 1: REYAX RYLR998 proprietary mode ***RYLR993 can’t set up the NETWORKID, only accept the default NETWORKID=18 of the RYLR998. When entry this mode please refer to the AT command of RYLR998.**
Response	<mode><CR><LF>
Result code	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

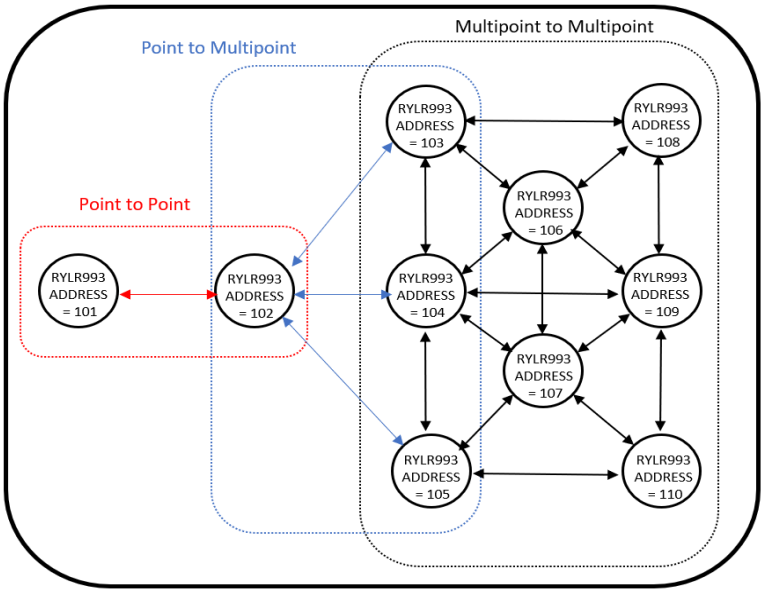
Example:

```
/* Example 1: set operating mode*/
#AT+OPMODE=1<CR>
<CR>
OK<CR>

/* Example 2: get operating mode*/
#AT+VL=?<CR>
1<CR>
<CR>
OK<CR>
```

5.1 Proprietary mode network structure

With the own LoRa® wireless transceiver function and the application program designed by customers, the RYLR993 can achieve different network architectures such as "Point to Point", "Point to Multipoint" or " Multipoint to Multipoint ". The figure below shows that the modules can communicate with each other in same network ID. If the ADDRESS of specified receiver belongs to different group, it is not able to communicate with each other.



5.2 THE SEQUENCE OF USING AT COMMAND

1. Use "**AT+ADDRESS**" to set ADDRESS. The ADDRESS is regard as the identification of transmitter or specified receiver.

2. Use" **AT+BAND**" to set the center frequency of wireless band. The transmitter and the receiver are required to use the same frequency to communicate with each other.

3. Use" **AT+PARAMETER**" to set the RF wireless parameters. The transmitter and the receiver are required to set the same parameters to communicate with each other. The parameters of which as follows:

[1] <Spreading Factor>: The larger the SF is, the better the sensitivity is. But the transmission time will take longer.

[2] <Bandwidth>: The smaller the bandwidth is, the better the sensitivity is. But the transmission time will take longer.

[3] <Coding Rate>: The coding rate will be the fastest if setting it as 1.

[4] <Programmed Preamble>: Preamble code. If the preamble code is bigger, it will result in the less opportunity of losing data. Generally, preamble code can be set above 10 if under the permission of the transmission time. Recommend to set "**AT + PARAMETER = 9,7,1,12**"

[5] When the Payload length is greater than 100Bytes, Recommend to set "AT + PARAMETER = 8,7,1,12"

4. Use "**AT+SEND**" to send data to the specified ADDRESS. Please use "LoRa® Modem Calculator Tool" to calculate the transmission time. Due to the program used by the module, the payload part will increase more 8 bytes than the actual data length.

5.3 Proprietary mode AT Commands

It is required to key in "**enter**" or "**\r\n**" in the end of all AT Commands.

Add " **=?** " in the end of the commands to ask the current setting value.

It is required to wait until the module replies **"OK"** so that you can execute the next AT command.

5.3.1 AT

Description	Test if the module can respond to Commands.
Syntax	AT<CR>
Arguments	None
Response	None
Result code	<CR><LF>OK<CR><LF>

5.3.2 Software RESET

Description	Software Reset
Syntax	ATZ
Arguments	None
Response	None
Result code	REYAX RYLR993<CR><LF> FREQ:915000000<CR><LF> +READY<CR><LF>

5.3.3 AT+BAND– Set the RF Frequency

Description	Set the RF Frequency
Syntax	AT+BAND=<parameter> AT+BAND=?
Arguments	<parameter> is the RF Frequency, unit is “Hz”. 915000000:915000000Hz (default)
Response	None
Result code	<CR><LF>OK<CR><LF> +BAND=<parameter><CR><LF> OK<CR><LF>

Examples:

```
/* Example1: Set the frequency as 868500000Hz
# AT+BAND=868500000
<CR><LF>OK<CR><LF>

/* Example2: Check the current Frequency band
# AT+BAND=?
+BAND=<868500000><CR><LF>
OK<CR><LF>
```

5.3.4 AT+PARAMETER – Set the RF parameter

Description	Set the RF parameter
Syntax	AT+PARAMETER=<Spreading Factor>,<Bandwidth>,<Coding Rate>,<Programmed Preamble> AT+PARAMETER=?
Arguments	<Spreading Factor>: 7~11(default:9) *SF7to SF9 at 125kHz, SF7 to SF10 at 250kHz, and SF7 to SF11 at 500kHz <Bandwidth>: 7~9, list as below 7: 125 KHz (default) 8: 250 KHz 9: 500 KHz <Coding Rate>: 1~4,(default:1) <Programmed Preamble>: (default 12) When NETWORKID=18, The value can be configured to 4~24. Other NETWORKID can only be configured to 12.
Response	None
Result code	<CR><LF>OK<CR><LF> +PARAMETER=<Spreading Factor>,<Bandwidth>,<Coding Rate>, <Programmed Preamble><CR><LF> OK<CR><LF>

Examples:

```
/* Example1: Set the parameters as:<Spreading Factor>7,<Bandwidth>500KHz,<Coding Rate>4, <Programmed Preamble>12
 # AT+PARAMETER=7,9,4,12
<CR><LF>OK<CR><LF>

/* Example2: Check the current RF parameter
# AT+PARAMETER=?
+PARAMETER=7,9,4,12<CR><LF>
OK<CR><LF>
```

5.3.5 AT+ADDRESS – Set the ADDRESS ID of module LoRa®

Description	Set the ADDRESS ID of module LoRa®
Syntax	AT+ADDRESS=<Address> AT+ADDRESS=?
Arguments	<Address>:0~65535(default 0)
Response	None
Result code	<CR><LF>OK<CR><LF> +ADDRESS=<Address><CR><LF> OK<CR><LF>

Examples:

```
/* Example1: Set the address of module as 120.
# AT+ADDRESS=120
<CR><LF>OK<CR><LF>

/* Example2: Check the current address
# AT+ADDRESS=?
+ADDRESS=120<CR><LF>
OK<CR><LF>
```

The settings will be memorized in Flash.

5.3.6 AT+CPIN – Set the domain password

Description	Set the domain password
Syntax	AT+CPIN=<password> AT+CPIN=?
Arguments	<Password>An 8 character long password from 00000001 to FFFFFFFF, Only by using same password can the data be recognized. After resetting, the previously password will disappear. After resetting, the previously password will disappear.
Response	None
Result code	<CR><LF>OK<CR><LF> +CPIN=00000000<CR><LF> OK<CR><LF> +CPIN=EEDCAA90<CR><LF> OK<CR><LF>

Examples:

```
/* Example1: Set the password to EEDCAA90
# AT+CPIN=EEDCAA90
<CR><LF>OK<CR><LF>

/* Example2: Check the current password (default)
# AT+CPIN=?
+CPIN=00000000<CR><LF>
OK<CR><LF>

/* Example3: Check the current password (after setting the password)
# AT+CPIN=?
+CPIN= EEDCAA90<CR><LF>
OK<CR><LF>
```

5.3.7 AT+CRFOP – Set the RF output power

Description	Set the RF output power
Syntax	AT+CRFOP=<power><CR> AT+CRFOP=?
Arguments	<power>: 0~22 dBm 22: 22dBm(default) 21: 21dBm 20: 20dBm …… 01: 1dBm 00: 0dBm
Response	None
Result code	<CR><LF>OK<CR><LF> +CRFOP=<power><CR><LF> OK<CR><LF>

Examples:

```
/* Example1: Set the output power as 10dBm
# AT+CRFOP=10
<CR><LF>OK<CR><LF>

/* Example2: Check the current output power
# AT+CRFOP=?
+CRFOP=10<CR><LF>
OK<CR><LF>
```

**** RF Output Power must be set to less than AT+CRFOP=14 to comply CE certification***

5.3.8 AT+SEND – Send data to the appointed address by Command Mode.

Description	Send data to the appointed address by Command Mode.
Syntax	AT+SEND=<address>,<Payload Length>,<Data> AT+SEND=?<CR>
Arguments	<Address>:0~65535, When the <Address> is 0, it will send data to all address (From 0 to 65535.) <Payload Length>:Maximum 240bytes <Data>:ASCII Format
Response	None
Result code	OK +SEND=<address>,<Payload Length>,<Data><CR><LF> OK<CR><LF>

Examples:

```
/* Example1: Send HELLO string to the Address 50
# AT+SEND=50,5,HELLO
OK

/* Example2: Search last transmit data,
# AT+SEND=?
+SEND=50,5,HELLO<CR><LF>
OK<CR><LF>
```

5.3.9 +RCV – Show the received data actively

Description	Show the received data actively
Syntax	+RCV=<Address>,<Length>,<Data>,<RSSI>,<SNR><CR>
Arguments	<Address>: Transmitter Address ID <Length>: Data Length <Data>: ASCll Format Data <RSSI>: Received Signal Strength Indicator <SNR>: Signal-to-noise ratio
Response	None
Result code	None

Examples:

```
/* Example1: Module received the ID Address 50 send 5 bytes data, Content is HELLO string, RSSI is -99dBm, SNR is 40, It will show as below.
#+RCV=50,5,HELLO,-99,-40OK
```

5.3.10 AT+UID – To inquire module ID. 12BYTES

Description	To inquire module ID. 12BYTES
Syntax	AT+UID=?
Arguments	None
Response	None
Result code	+UID=00:00:00:00:00:00:00:00<CR><LF> OK<CR><LF>

5.3.11 AT+VER – To inquire the firmware version

Description	To inquire the firmware version
Syntax	AT+VER=?
Arguments	None
Response	None
Result code	RYLR993_FW_VERSION: V0.2.6<CR><LF> MW_LORAWAN_VERSION: V2.3.0<CR><LF> MW_RADIO_VERSION: V1.1.0<CR><LF> OK<CR><LF>

5.3.12 AT+FACTORY – Set all current parameters to manufacturer defaults

Description	Set all current parameters to manufacturer defaults
Syntax	AT+FACTORY
Arguments	Manufacturer defaults: BAND：915MHz UART：115200 Spreading Factor：9 Bandwidth：125kHz Coding Rate：1 Preamble Length：12 Address：0 Network ID：18 CRFOP：22
Response	None
Result code	<CR><LF>OK<CR><LF>

5.3.13 Other Message

Narrative	After Reset
Arguments	None
Response	None
Result code	REYAX RYLR993<CR><LF> FREQ:915000000<CR><LF> +READY<CR><LF>

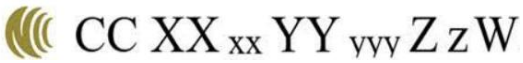
5.3.14 Error result code

1. If the head of AT command is not “**AT string**” or if it’s a “**unknow command**”, the device will show “**AT_ERROR**”
2. If the parameters are set incorrectly, the device will show “**AT_PARAM_ERROR**”

CERTIFICATION INFORMATION

National Communications Commission Statement

取得審驗證明之低功率射頻器材，非經核准，公司、商號或使用者均不得擅自變更頻率、加大功率或變更原設計之特性及功能。低功率射頻器材之使用不得影響飛航安全及干擾合法通信;經發現有干擾現象時，應立即停用，並改善至無干擾時方得繼續使用。前述合法通信，指依電信管理法規定作業之無線電通信。低功率射頻器材須忍受合法通信或工業、科學及醫療用電波輻射性電機設備之干擾



若此認證標籤隨同RYLR993安裝在主體裝置或設備內而無法看見，則安裝RYLR993之主體裝置或設備上必須標示【內含發射器模組: CC XXxx YYyyy ZzW】。

Federal Communication Commission Statement

The RYLR993 module has received Federal Communications Commission (FCC) CFR47 Telecommunications, Part 15 Subpart C modular approval in accordance with Part 15.212 Modular Transmitter Statement about FCC

Any changes or modifications not expressly approved by the party responsible for compliance could void your authority to operate the equipment.

This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Trace antenna designs

The RF_PORT (Module Pin No. 43) signal may be directly connected to an on-board chip antenna or terminated in an RF connector of any form factor for enabling the use of external antennas.
The RF traces on RF_PORT (Module Pin No. 43) should have a characteristic impedance of 50 Ohms. Any standard 50 Ohms RF trace (Microstrip or Coplanar wave guide) may be used. The width of the 50 Ohms line depends on the PCB stack, e.g., the dielectric of the PCB, thickness of the copper, thickness of the dielectric and other factors. Consult the PCB fabrication unit to get these factors right.

RF exposure considerations

1. This Transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.
2. This equipment complies with FCC RF radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with a minimum distance of 20 centimeters between the radiator and your body or nearby persons.
3. The modular transmitter is only FCC authorized for the specific rule parts (i.e., FCC transmitter rules) listed on the grant. OEM integrators are responsible for host-product evaluation or any additional compliance requirements with this modular transmitter installed.

Antennas

Brand	Antenna Model	Antenna Type	Gain	Qualified Region
REYAX	RYAI915	Helical Antenna	2 dBi	FCC/NCC
REYAX	RYBF915	Dipole Antenna	5.7 dBi	FCC/NCC

The final end product must be labeled in a visible area with the following " Contains TX FCC ID : QLYRYLR993 ". If the size of the end product is larger than 8x10cm, then the following FCC part 15.19 statement has to also be available on the label: This device complies with Part 15 of FCC rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference and (2) this device must accept any interference received, including interference that may cause undesired operation.
If the FCC ID is not visible when the RYRL993 is installed inside another device, then the outside of the device into which the RYLR993 is installed must also display a label such as the following Transceiver Module FCC ID : QLYRYLR993.

Additional testing, Part 15B disclaimer :
This module only complied with FCC 15.247 requirements as listed on the grant, and therefore, the host product manufacturer is responsible for FCC Part 15B compliance testing with the modular transmitter installed.

REYAX
TECHNOLOGY CORPORATION, LTD

E-mail: sales@reyax.com
Website:https://reyax.com