# SMART Modular Technologies

## LoRa® Smart Module: AT COMMAND USER MANUAL

**August 09, 2021**

**Rev 2.14**



This manual provides a complete AT Command list, description and syntaxes of the SMART Modular Technologies LoRa® Module family.
The specification and information contained in this manual are subject to changes without prior notice. Any change will be integrated into the latest release of this document.
Refer to **LoRa® Smart Module Datasheet** for device specification, operation and pinouts.

# Contents





This manual provides a complete AT Command list, description and syntaxes of the SMART Modular Technologies LoRa® Module family.
The specification and information contained in this manual are subject to changes without prior notice. Any change will be integrated into the latest release of this document.
Refer to **LoRa® Smart Module Datasheet** for device specification, operation and pinouts.

This manual provides a complete AT Command list, description and syntaxes of the SMART Modular Technologies LoRa® Module family.
The specification and information contained in this manual are subject to changes without prior notice. Any change will be integrated into the latest release of this document.
Refer to **LoRa® Smart Module Datasheet** for device specification, operation and pinouts.

This manual provides a complete AT Command list, description and syntaxes of the SMART Modular Technologies LoRa® Module family.
The specification and information contained in this manual are subject to changes without prior notice. Any change will be integrated into the latest release of this document.
Refer to **LoRa® Smart Module Datasheet** for device specification, operation and pinouts.

# List of tables





This manual provides a complete AT Command list, description and syntaxes of the SMART Modular Technologies LoRa® Module family.
The specification and information contained in this manual are subject to changes without prior notice. Any change will be integrated into the latest release of this document.
Refer to **LoRa® Smart Module Datasheet** for device specification, operation and pinouts.

# List of figures




This manual provides a complete AT Command list, description and syntaxes of the SMART Modular Technologies LoRa® Module family.
The specification and information contained in this manual are subject to changes without prior notice. Any change will be integrated into the latest release of this document.
Refer to **LoRa® Smart Module Datasheet** for device specification, operation and pinouts.

# 1 Acronyms

Table 1. List of acronyms

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


This manual provides a complete AT Command list, description and syntaxes of the SMART Modular Technologies LoRa® Module family.
The specification and information contained in this manual are subject to changes without prior notice. Any change will be integrated into the latest release of this document.
Refer to **LoRa® Smart Module Datasheet** for device specification, operation and pinouts.

# 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 “Hayes” to control modems. AT stands for attention.

The command set consists of a series of short text strings providing operations such as joining, data exchange and parameters setting.

In a context of LoRa® modem, the Hayes command set is a variation of the standard AT Hayes commands.

The AT commands are used to drive the LoRa® module and to send data (refer to document

*1*). The AT commands are sent through the UART.

As described in document *2*, the LoRa® modem can be controlled either through a terminal emulation like Tera Term or PuTTY (see *Figure 1*), or through an embedded AT master module (see *Figure 2*).

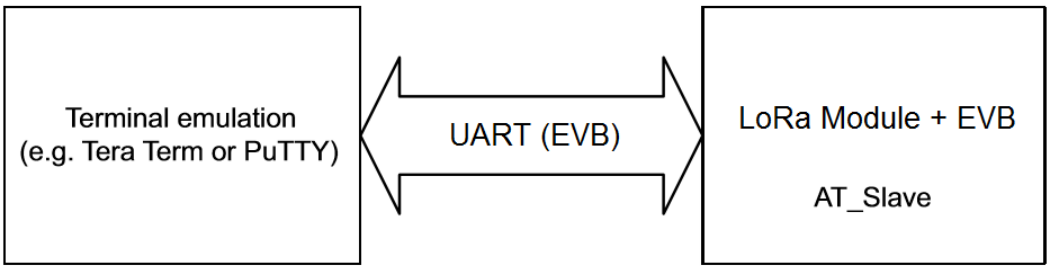


Figure 1. Terminal emulation mode

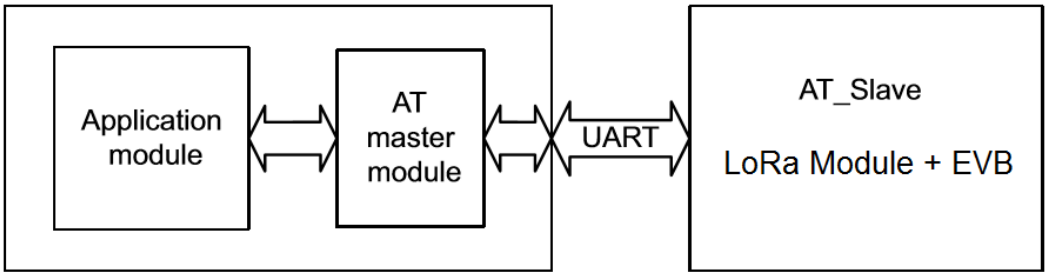


Figure 2. AT master mode

For illustration purposes, the rest of the document is based on the relation “terminal emulation” with the SMART Modular Technologies - LoRa Module + EVB.

An UART over SMART Modular Technologies - LoRa EVB (SMW-SX1262M0EB) can then be used with

standard Windows® software such as Tera Term or PuTTY. The chosen software has to be configured with the following parameters:

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

Figure 3 and Figure 4 show the standard configuration for Tera Term to use the UART over the SMW-SX1262M0EB

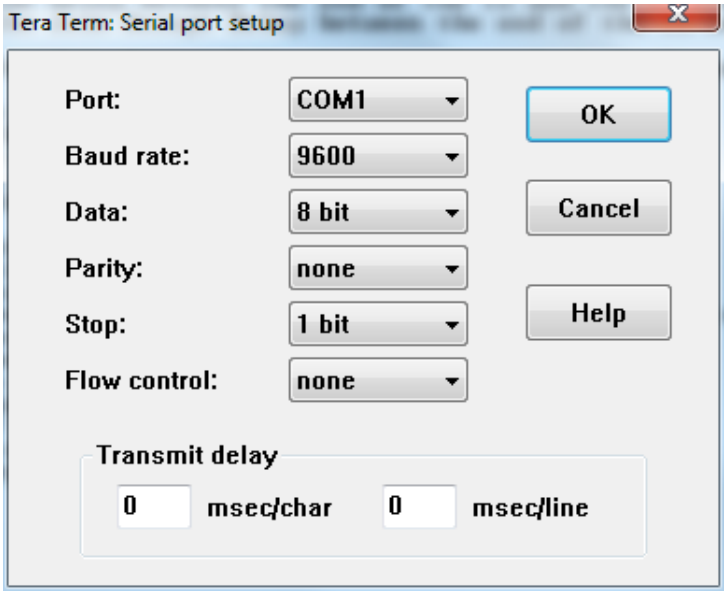


Figure 3. Tera Term port setup example

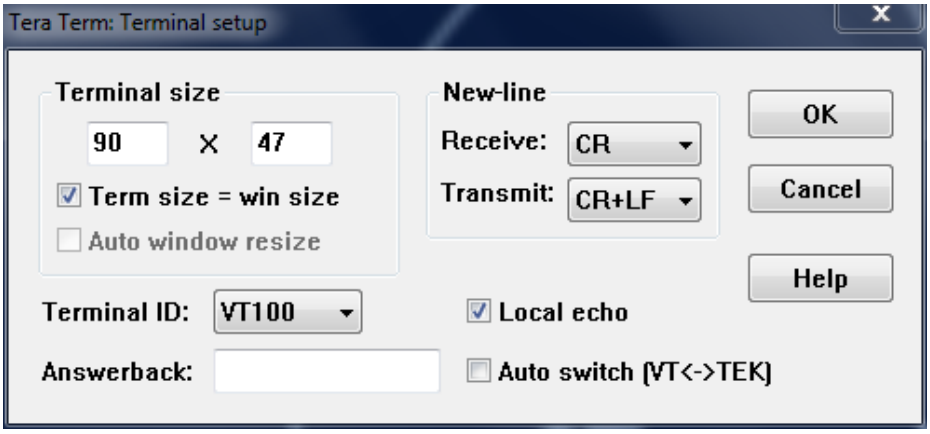


Figure 4. Tera Term terminal setup example

The AT commands have the standard format “AT+XXX”, with XXX denoting the command. There are four available command behaviors:

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

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

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

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

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

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

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

- OK: command run correctly without error.
- AT_ERROR: generic error
- AT_PARAM_ERROR: a parameter of the command is wrong
- AT_BUSY_ERROR: the LoRa® network is busy, so the command has not been completed
- AT_TEST_PARAM_OVERFLOW: the parameter is too long
- AT_NO_CLASSB_ENABLE: End-node has not yet switched in Class B
- AT_NO_NETWORK_JOINED: the LoRa® network has not been joined yet
- AT_RX_ERROR: error detection during the reception of the command

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

Then the return of the module is printed.

## 3.1 General commands

This section describes the commands related to “attention” help list, link control and CPU AT_Slave reset.

### 3.1.1 AT: attention

This command is used to check that the link is working properly (refer to *Table 2* for details).

Table 2. Link check command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT | - | - | OK |


This manual provides a complete AT Command list, description and syntaxes of the SMART Modular Technologies LoRa® Module family.
The specification and information contained in this manual are subject to changes without prior notice. Any change will be integrated into the latest release of this document.
Refer to **LoRa® Smart Module Datasheet** for device specification, operation and pinouts.

### 3.1.2 AT?: short help

This command provides short help for all the supported commands (refer to *Table 3* for details).

Table 3. Short help command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT? | - | AT+<CMD>?: help on <CMD<br>AT+<CMD>: run <CMD><br>AT+<CMD>=<value>: set the value<br>AT+<CMD>=?: get the value<br><followed by the help of all commands> | OK |

### 3.1.3 ATZ: MCU reset

This command is used to trig a CPU reset of the SMW-SX1262M0 board (refer to *Table 4* for details).

Table 4. MCU reset command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| ATZ? | - | ATZ: triggers a reset of the MCU | OK |
| ATZ | - | No return value and return code. The MCU is reset. | Void |

## 3.2 Keys, IDs and EUIs management

This section describes the commands related to the activation of the end device.

### 3.2.1 AT+APPEUI: application identifier

This command allows the user to access the global application identifier (refer to *Table 5* for details).

Table 5. Application identifier command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+APPEUI? | - | AT+APPEUI: get or set the application EUI | OK |
| AT+APPEUI=? | - | <8 hexa separated by:> | OK |
| AT+APPEUI= <Param> | <8 hexa separated by:> | - | OK / AT_PARAM_ERROR(1) |
| Example AT+APPEUI= | 01:2:a:FB:A1:CD:4D:20 | - | OK |
| Example AT+APPEUI= | 01:2:a:FB:A1:CD:4D | - | AT_PARAM_ERROR(1) |
| Example AT+APPEUI=? | - | 01:02:03:04:05:06:07:08 | OK |

1. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

### 3.2.2 AT+APPKEY: application key

This command allows the user to access the application session key (refer to *Table 6* for details).

Table 6. Application key command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+APPKEY? | - | AT+APPKEY: get or set the application key | OK |
| AT+APPKEY=? | - | <16 hexa separated by:> | OK |
| AT+APPKEY= <Param> | <16 hexa separated by:> | void | OK / AT_PARAM_ERROR(1) |
| Example AT+APPKEY= | 01:2:a:FB:A1:CD:4D:20: 01:02:30:40:5a:6b:7f:88 | - | OK |
| Example AT+APPKEY= | 01:2:a:FB:A1:CD:4D:20: 01:02:30:40:5a:6b:7f | - | AT_PARAM_ERROR(1) |
| Example AT+APPKEY=? | - | 2b:7e:15:16:28:ae:d2:a6: ab:f7:15:88:09:cf:4f:3c | OK |

1. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

### 3.2.3 AT+APPSKEY: application session key

This command allows the user to set the application session key (refer to *Table 7* for details).

Table 7. Application session key command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+APPSKEY? | - | AT+APPSKEY:<br>set the application session key | OK |
| AT+APPSKEY=? | - | <16 hexa separated by:> | OK / AT_ ERROR(1) |
| AT+APPSKEY=<Param> | <16 hexa separated by:> | void | OK / AT_PARAM_ ERROR(2) |
| Example AT+APPSKEY= | 01:2:a:FB:A1:CD:4D:20:01:02:30:40:5a:6b:7f:88 | - | OK |
| Example AT+APPSKEY= | 01:2:a:FB:A1:CD:4D:20:01:02:30:40:5a:6b:7f | - | AT_PARAM_ ERROR(2) |

1. AT_ ERROR is returned when the Activation Mode is set to OTAA.
2. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

### 3.2.4 AT+DADDR: device address

This command allows the user to access the device address (refer to *Table 8* for details).

Table 8. Device address command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+DADDR? | - | AT+DADDR: get or set the device address | OK |
| AT+DADDR=? | - | <4 hexa separated by:> | OK |
| AT+DADDR=<Param> | <4 hexa separated by:> | - | OK / AT_PARAM_ ERROR(1) |
| Example AT+DADDR= | 01:2:a:FB | - | OK |
| Example AT+DADDR=? | - | 11:22:33:44 | OK |

1. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

### 3.2.5 AT+DEUI: device EUI

This command allows the user to access the global end-device ID (refer to *Table 9* for details).

Table 9. Device EUI command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+DEUI? | - | AT+DEUI: get the device EUI | OK |
| AT+DEUI=? | - | <8 hexa separated by:> | OK |
| Example AT+DEUI=? | - | 11:22:33:44:55:66:77:88 | OK |

1. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

### 3.2.6 AT+NWKID: network ID

This command allows the user to access the network identifier (refer to *Table 10* for details).

Table 10. Network ID command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+NWKID? | - | AT+NWKID: get or set the network ID | OK |
| AT+NWKID=? | - | <4 hexa separated by:> | OK |
| AT+NWKID= <Param> | <4 hexa separated by:> | - | OK / AT_PARAM_ ERROR(1) |
| Example AT+NWKID=? | - | 11:22:33:44 | OK |
| Example AT+NWKID= | 11:22:33:44 | - | OK |

1. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

### 3.2.7 AT+NWKSKEY: network session key

This command allows the user to set the network session key (refer to *Table 11* for details).

Table 11. Network session key command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+NWKSKEY? | - | AT+NWKSKEY:<br>set the network session key | OK |
| AT+NWKSKEY=? | - | <16 hexa separated by:> | OK / AT_ ERROR(1) |
| AT+NWKSKEY=<Param> | <16 hexa separated by:> | - | OK / AT_PARAM_ ERROR(2) |
| Example<br>AT+NWKSKEY= | 0:1:2:3:4:5:6:7:8:9:A:B:C:D:E:F | - | OK |

1. AT_ ERROR is returned when the Activation Mode is set to OTAA.
2. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

## 3.3 Joining and sending data on LoRa® network

This section gives description of the commands related to the join procedure and to the data path.

### 3.3.1 AT+CFM: confirm mode

This command allows the user to access the notification on received data coming from network (refer to *Table 12* for details).

Table 12. Confirm mode command

| Command | Input parameter or Parmeter | Return value | Return code |
|---|---|---|---|
| AT+CFM? | - | AT+CFM: get or set the confirm mode (0-1) | OK |
| AT+CFM=? | - | 0 or 1 | OK |
| AT+CFM=<Param> | 0 or 1 | - | OK / AT_PARAM_ERROR(1) |
| Example<br>AT+CFM= | 1 | - | OK |
| Example<br>AT+CFM=?(2) | - | 1 | OK |

1. AT_PARAM_ERROR is returned when setting a wrong or malformed value.
2. When the confirmation mode is 1, each sent message must be confirmed. AT+CFS=? allows the user to know whether the last sent message has been confirmed or not.

### 3.3.2 AT+CFS: confirm status

This command allows the user to access the status of the last “SEND” command (refer to *Table 13* for details).

Table 13. Confirm status command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+CFS? | - | AT+CFS: get the confirmation status of the last AT+SEND (0-1) | OK |
| AT+CFS=? | - | 0 or 1 | OK |
| Example<br>AT+CFS=? | - | 0 | OK |

### 3.3.3 AT+JOIN: join LoRa® network

This command does a join request to the network (refer to *Table 14* for details).

Table 14. Join LoRa® network command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+JOIN? | - | AT+JOIN: join network | OK |
| AT+JOIN | Void | Void | OK/<br>AT_BUSY_ERROR(1) |
| Example<br>AT+JOIN | - | - | OK |

1. AT_BUSY_ERROR is returned when a joining process is already running.

This is an asynchronous command. OK means that the join is being run. The completion of the JOIN must be verified with AT+NJS=?.

### 3.3.4 AT+NJM: LoRa® network join mode

This command allows the user to access the network join mode (refer to *Table 15* for details).

Table 15. LoRa® network join mode command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+NJM? | - | AT+NJM: get or set the network join mode (0 = ABP, 1 = OTAA) | OK |
| AT+NJM | - | | OK/ |
| AT+NJM=<Input> | 0 or 1 | - | OK/<br>AT_PARAM_ERROR(1) |
| Example AT+NJM=? | - | 0 | OK |
| Example AT+NJM= | 1 | - | OK |
| Example AT+NJM= | 2 | - | AT_PARAM_ERROR |

1. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

### 3.3.5 AT+NJS: LoRa® network join status

This command allows the user to access the current status of the LoRa® link (refer to *Table 16* for details).

Table 16. LoRa® network join status command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+NJS? | - | AT+NJS: get the join status | OK |
| AT+NJS=? | - | 0 or 1 | OK |
| Example AT+NJS=? | - | 0 (network not joined) | OK |
| Example AT+NJS=? | - | 1 (network joined) | OK |

### 3.3.6 AT+RECV: last received text data

This command allows the user to access the last received text data in raw format (refer to *Table 17* for details).

Table 17. Last received text data command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+RECV? | - | AT+RECV: print the last received data in raw format | OK |
| AT+RECV=? | - | Raw (string format) | OK |
| Example AT+RECV=? | - | 45: hello world | OK |

This command returns the last received data in a text form, along with the port on which it was received. The format of the output is:

```
<port>:<text data><CR><LF>
<CR><LF>OK<CR><LF>
```

When called twice, without new data received between the calls, the second AT+RECV=? returns an empty value as shown below:

```
45:<CR><LF>
<CR><LF>OK<CR><LF>
```

### 3.3.7 AT+RECVB: last received binary data

This command allows the user to access the last received text data in binary format (refer to *Table 18* for details). In Table 18 the binary data is received on port 45.

Table 18. Last received binary data command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+RECVB? | - | AT+RECVB: print the last received data in binary format (with hexadecimal values) | OK |
| AT+RECVB=? | - | <port>:<binary>, | OK |
| Example AT+RECVB=? | - | 45:48656c6c6f20576f726c64 | OK |

### 3.3.8 AT+SEND: send text data

This command provides the way to send text data on a dedicated port number (refer to *Table 19* for details). In Table 19 the text data is received on port 12.

Table 19. Send text data command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+SEND? | - | AT+SEND: send text data along with the application port | OK |
| AT+SEND=<input> | port text | - | OK/ AT_PARAM_ERROR(1)/ AT_BUSY_ERROR(2)/ AT_NO_NETWORK_JOINED(3) |
| Example AT+SEND= | 12: hello world | - | OK |

1. AT_PARAM_ERROR is returned when the setting does not have the correct format <port>:<text>, with <port> being a decimal value.
2. AT_BUSY_ERROR is returned when the previous send is not complete (send waiting for duty cycle, rx window not consumed…).
3. AT_NO_NETWORK_JOINED is returned when the network is not yet joined.

### 3.3.9 AT+SENB: send binary data

This command provides the way to send text data in binary format on a dedicated port number (refer to *Table 20* for details).
Each byte of the binary data is provided as two characters denoting the value in hexadecimal. Hence, the length of the binary data is always even.
In the example of Table 20, 8 bytes are sent on port 12: 0xAB, 0xCD, 0xEF, 0x01 (note that the example passes "01", passing only "1" would fail), 0x23, 0x45, 0x67 and 0x89.

Table 20. Send binary data command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+SENDB? | - | AT+SENDB: send hexadecimal data along with the application port | OK |
| AT+SENDB=<input> | <port>:<binary>, | - | OK/ AT_PARAM_ERROR(1)/ AT_BUSY_ERROR(2)/ AT_NO_NETWORK_JOINED(3) |
| Example AT+SENDB= | 12:abcdef0123456789 | - | OK |
| Example AT+SENDB= | abcdef0123456789 | - | AT_PARAM_ERROR |

1. AT_PARAM_ERROR is returned when the setting has not the correct format <port>:<binary>, with <port> being a decimal value, and <binary> following hexadecimal format using two characters as described above.
2. AT_BUSY_ERROR is returned when the previous send is not complete (send waiting for duty cycle, rx window not consumed…).
3. AT_NO_NETWORK_JOINED is returned when the network is not joined yet.

### 3.3.10 AT+CNTUP: Uplink Counter

This command provides the way to get or set the current uplink counter (refer to *Table* 21 for details).

Table 21. Uplink Counter command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+CNTUP? | - | AT+CNTUP: Get or Set Counter Up | OK |
| AT+CNTUP=? | - | <integer> | OK |
| AT+CNTUP=<input> | <integer> | - | OK / AT_PARAM_ERROR(1) |
| Example AT+CNTUP= | 12 | - | OK |
| Example AT+CNTUP= | abcdef | - | AT_PARAM_ERROR |

1. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

**(*) NOTE:** The CNTUP command is not available on all Hardware and Firmware versions. Currently, only accepted for the SMW-SX1262M0 board.

## 3.4 LoRa® network management

This section provides a set of commands for network management.

### 3.4.1 AT+ADR: adaptive rate

This command allows the user to access the adaptive data rate (refer to *Table 22* for details). The default value of the ADR is 1 (enabled).

Table 22. Adaptive rate command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+ADR? | - | AT+ADR: get or set the adaptive data rate setting (0 = off, 1 = on) | OK |
| AT+ADR=? | - | 0 or 1 | OK |
| AT+ADR=<Input> | 0 or 1 | - | OK/ AT_PARAM_ERROR(1) |
| Example AT+ADR= | 0 | - | OK |
| Example AT+ADR=? | - | 0 | OK |

1. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

### 3.4.2 AT+CLASS: LoRa® class

This command allows the user to access the LoRaWAN™ class (refer to *Table 23* for details).

Table 23. LoRa® class command

| Command | Input parameter | Return value(1) | Return code |
|---|---|---|---|
| AT+CLASS? | - | AT+CLASS: get or set the device class | OK |
| AT+CLASS=? | - | A, B or C | OK |
| AT+CLASS=<Input> | A, B(2) or C | - | OK/ AT_PARAM_ERROR(3) |
| Example AT+CLASS=? | - | A | - |

1. This release version supports the LoRaWAN V1.0.3 stack.
2. B, S0: Beacon searching

   B, S1: Beacon locked
   B, S2: Beacon failed
   B: Class B enabled
3. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

### 3.4.3 AT+DCS: duty cycle settings

This command allows the user to access the duty cycle parameter (refer to *Table 24* for details).

Table 24. Duty cycle settings command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+DCS? | - | AT+DCS: get or set the ETSI duty cycle setting:<br>– 0 = disable<br>– 1 = enable - only for testing (refer to document *2*) | OK |
| AT+DCS=? | - | 0 or 1 | OK |
| AT+DCS=<Input> | 0 or 1 | - | OK/ AT_PARAM_ERROR(1) |
| Example AT+DCS? | - | 1 | OK |
| Example AT+DCS= | 1 | - | OK |

1. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

### 3.4.4 AT+DR: data rate

This command allows the user to access the data rate (refer to *Table 25* for details).

Table 25. Data rate command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+DR? | - | AT+DR: get or set the data rate (0-6 corresponding to DR_X) | OK |
| AT+DR=? | - | [0,1,2,3,4,5,6](2) | OK |
| AT+DR=<Input> | [0,1,2,3,4,5,6] | - | OK/ AT_PARAM_ERROR(1) |
| Example AT+DR=? | - | 3 | OK |
| Example AT+DR= | 2 | - | OK |

1. AT_PARAM_ERROR is returned when setting a wrong or malformed value.
2. The data rate range is dependent on the frequency plan for each LoRaWAN regional parameters, check your region for more information.

### 3.4.5 AT+JN1DL: join delay on RX window 1

This command allows the user to access the join delay on RX window 1 (refer to *Table 26* for details).

Table 26. Join delay on RX window 1 command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| 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 |
| AT+JN1DL=? | - | <integer> | OK/ AT_BUSY_ERROR(1) |
| AT+JN1DL=<input> | <integer> | - | OK/ AT_PARAM_ERROR(2) AT_BUSY_ERROR(1) |
| Example AT+JN1DL=? | - | 5000 | OK |
| Example AT+JN1DL= | 10000 | - | OK |

1. AT_BUSY_ERROR is returned when a join or a send is being processed.
2. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

### 3.4.6 AT+JN2DL: join delay on RX window 2

This command allows the user to access the join delay on RX window 2 (refer to *Table 27* for details).

Table 27. Join delay on RX window 2 command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| 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 |
| AT+JN2DL=? | - | <integer> | OK/ AT_BUSY_ERROR(1) |
| AT+JN2DL=<input> | <integer> | - | OK/ AT_PARAM_ERROR(2) AT_BUSY_ERROR(1) |
| Example AT+JN2DL=? | - | 6000 | OK |
| Example AT+JN2DL= | 20000 | - | OK |

1. AT_BUSY_ERROR is returned when a join or a send is being processed.
2. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

### 3.4.7 AT+PNM: public network mode

This command allows the user to access the public network mode (refer to *Table 28* for details).

Table 28. Public network mode command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+PNM? | - | AT+PNM: get or set the public network mode (0 = off, 1 = on) | OK |
| AT+PNM=? | - | 0 or 1 | OK |
| AT+PNM=<input> | 0 or 1 | - | OK/ AT_PARAM_ERROR(1) |
| Example AT+PNM=? | - | 0 | OK |
| Example AT+PNM= | 1 | - | OK |
| Example AT+PNM= | 2 | - | AT_PARAM_ERROR(1) |

1. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

### 3.4.8 AT+RX1DL: delay of the received window 1

This command allows the user to access the delay of the received window 1 (refer to *Table 29* for details).

Table 29. Delay of the received window 1 command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+RX1DL? | - | AT+RX1DL: get or set the delay between the end of the Tx and the Rx window 1 in ms | OK |
| AT+RX1DL=? | - | <integer> | OK/ AT_BUSY_ERROR(1) |
| AT+RX1DL=<input> | <integer> | - | OK/ AT_PARAM_ERROR(2) AT_BUSY_ERROR(1) |
| Example AT+RX1DL=? | - | 1000 | OK |
| Example AT+RX1DL= | 1500 | - | OK |

1. AT_BUSY_ERROR is returned when a join or a send is being processed.
2. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

### 3.4.9 AT+RX2DL: delay of the received window 2

This command allows the user to access the delay of the received window 2 (refer to *Table 30* for details).

Table 30. Delay of the received window 2 command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+RX2DL? | - | AT+RX2DL: get or set the delay between the end of the Tx and the Rx window 2 in ms | OK |
| AT+RX2DL=? | - | <integer> | OK/ AT_BUSY_ERROR(1) |
| AT+RX2DL=<input> | <integer> | - | OK/ AT_PARAM_ERROR(2) AT_BUSY_ERROR(1) |
| Example AT+RX2DL=? | - | 2000 | OK |
| Example AT+RX2DL= | 2500 | - | OK |

1. AT_BUSY_ERROR is returned when a join or a send is being processed.
2. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

### 3.4.10 AT+RX2DR: data rate of the received window 2

This command allows the user to access the data rate of received window 2 (refer to *Table 31* for details).

Table 31. Data rate of the received window 2 command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+RX2DR? | - | AT+RX2DR: get or set the Rx2 window data rate (8-13) corresponding to DR_X | OK |
| AT+RX2DR=? | - | [8,9,10,11,12,13](3) | OK/ AT_BUSY_ERROR(1) |
| AT+RX2DR=<input> | [8,9,10,11,12,13] | - | OK/ AT_PARAM_ERROR(1) AT_BUSY_ERROR(2) |
| Example AT+RX2DR=? | - | 8 | OK |
| Example AT+RX2DR= | 10 | - | OK |

1. AT_BUSY_ERROR is returned when a join or a send is being processed.
2. AT_PARAM_ERROR is returned when setting a wrong or malformed value.
3. The data rate range is dependent on the frequency plan for each LoRaWAN regional parameters, check your region for more information.

### 3.4.11 AT+RX2FQ: frequency of the received window 2

This command allows the user to access the frequency of the received window 2 (refer to *Table 32* for details).

Table 32. Frequency of the received window 2 command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+RX2FQ? | - | AT+RX2FQ: get or set the Rx2 window frequency | OK |
| AT+RX2FQ=? | - | Frequency in Hz | OK/ AT_BUSY_ERROR(1) |
| AT+RX2FQ=869535000 | Frequency in Hz | - | OK/ AT_PARAM_ERROR(2) AT_BUSY_ERROR(1) |
| Example AT+RX2FQ=? | - | 869535000 | OK |
| Example AT+RX2FQ= | 869535000 | - | OK |

1. AT_BUSY_ERROR is returned when a join or a send is being processed.
2. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

### 3.4.12 AT+TXP: transmit power

This command allows the user to access the transmit power (refer to *Table 33* for details).

Table 33. Transmit power command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+TXP? | - | AT+TXP: get or set the transmit power (0-10) | OK |
| AT+TXP=? | - | [0,1,2,3,4,5,6,7,8,9,10] | OK AT_PARAM_ERROR(1) |
| AT+TXP=<input> | [0,1,2,3,4,5,6,7,8,9,10] | - | OK AT_PARAM_ERROR(1) |
| Example AT+TXP=? | - | 1 | OK |
| Example AT+TXP= | 4 | - | OK |

1. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

## 3.5 Class B mode

This section provides a set of commands for Class B mode management. Class B mode is only available depending on Firmware version.

### 3.5.1 AT+PGSLOT

This command allows the user to set or to get the unicast ping slot periodicity.

Table 34. Slot periodicity command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+PGSLOT? | - | PS: periodicity, DRx, psfreq(1) | OK |
| AT+PGSLOT=<input> | [0,1,2,3,4,5,6,7] | - | OK<br>AT_PARAM_ERROR(2) |
| Example AT+PGSLOT=? | - | PS: 2, 3, 869.525 | OK |
| Example AT+PGSLOT= | 1 | - | OK |

1. Default value for EU868 SF9/125 MHz. Periodicity (see *2*).
2. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

### 3.5.2 AT+BFREQ

This command allows the user to access the current beacon (default broadcast) frequency.

Table 35. Beacon frequency command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+BFREQ? | - | AT+BFREQ: “Get the Beacon frequency” | OK |
| AT+BFREQ=? | - | BCON: DRx, psfreq(1) | |
| Example AT+BFREQ=? | - | BCON: 3, 869.525 | OK |

1. Default value for EU868 SF9 / 125 MHz.

### 3.5.3 AT+BTIME

This command(1) allows the user to access the current beacon time(2).

Table 36. Beacon frequency command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+BTIME? | - | BTIME: “Get the Beacon Time (GPS Epoch time)” | OK |
| AT+BTIME=? | - | BTIME: “GPS epoch time” | OK |
| Example AT+BTIME=? | - | BTIME: 1226592311 | OK |

1. Only applicable when the end-node is in “beacon locked” state.
2. Time in seconds since January 6, 1980 00:00:00 UTC (start of the GPS epoch) modulo $2^{32}$.

### 3.5.4 AT+BGW

This command(2) allows the user to access the GW GPS coordinate, NetID and GwID.

Table 37. GW GPS coordinate command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+BGW? | - | AT+BGW: Get the Gateway GPS coordinate, NetID and GwID | OK |
| AT+BGW=? | - | BGW: INFO(1), netid, gwid, longitude, latitude | OK |
| Example AT+BGW=? | - | BGW: 3, “NetID”, “GwID”, “N/A”, “N/A” | OK |

1. For a single omnidirectional antenna gateway the INFO value is 0 with GPS coordinates. For a site featuring three sectored antennas, for example, the first antenna INFO equals 0 with GPS coordinates, the second antenna INFO equals 1 with GPS coordinate, etc. In this case netid and gwid are not relevant. When INFO = 3, the content of network NetID plus a freely allocated gateway gwid and longitude and latitude are not relevant.
2. Only applicable when the end-node is in “beacon locked” state.

### 3.5.5 AT+LTIME

This command(1) allows the user to access the local time in a UTC format.

Table 38. Local time command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+LTIME? | - | AT+LTIME: Get the local time in UTC format | OK |
| AT+LTIME=? | - | LTIME: YYYY-MM-DD | OK |
| Example AT+LTIME=? | - | LTIME: 2018-11-14 | OK |

1. Only relevant after the end-node has received from the network the answer of the “DeviceTimeReq” request.

## 3.6 Asynchronous events

*Table 39* lists the possible events sent from the Smart Module + EVB to host serial port actively. Host parser may need to handle this event at any time.

Table 39. Asynchronous events

| Event | | Parameter | Description |
|---|---|---|---|
| Beacon acquisition process | +BC: | <status> | FAILED: if beacon not found during Class B switching mode process. |
| | | | ACQ: the beacon acquisition process is ongoing. |
| | | | REACQ: missed a beacon, need to update the beacon time. Reacquisition is needed. |
| | | | LOST: beacon lost, modem does a Beacon Reacquisition. |
| | | | LOCKED: beacon found. The modem is synchronized. |
| Pingslot process | +PS: | <status> | DONE: at this time, ping slots will be opened periodically. The modem is now in Class B mode. |
| Class B downlink | +EVT: | <status> | UNICAST: lets host know that the Rx is in unicast Class B mode. |
| | | | PortNumber: “12345678”: received binary data on PortNumber. |
| | | | RX3, RSSI -110, SNR 5: indicates that data has been received on pingslot received window. |
| Class A/C downlink | +EVT: | <status> | PortNumber: “binary or string format”: received binary data or raw string format data on PortNumber. |

## 3.7 Information

This section provides a set of commands for battery level, RF signal quality and FW version.

### 3.7.1 AT+RSSI: RSSI on reception

This command allows the user to access the RSSI on reception (refer to *Table 40* for details).

Table 40. RSSI on reception command

| Command(1) | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+RSSI? | - | AT+RSSI: get the RSSI of the last received packet | OK |
| AT+RSSI=? | - | Integer | OK |
| Example AT+RSSI=? | - | -31 | OK |

1. AT+RSSI=? provides a value in dBm.

### 3.7.2 AT+SNR: signal noise ratio

This command allows the user to access the SNR of the last received packet (refer to *Table 41* for details).

Table 41. Signal noise ratio command

| Command(1) | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+SNR? | - | AT+SNR: get the SNR of the last received packet | OK |
| AT+SNR=? | - | Integer | OK |
| Example AT+SNR=? | - | 32 | OK |

1. AT+SNR=? provides a value in dBm.

### 3.7.3 AT+ID: Return the unique ID

This command allows the user to access unique ID of the module (refer to *Table 42* for details).

Table 42. Return the unique ID command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+ID? | - | AT+ID: Get the unique ID | OK |
| AT+ID=? | - | XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX | OK |
| Example AT+ID=? | - | 12:34:56:78:90:AB:CD:EF:12:34:56:78 | OK |

### 3.7.4 AT+VER: version of the firmware

This command allows the user to access the version of the module firmware (refer to *Table 43* for details).

Table 43. Version of the firmware command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+VER? | - | AT+VER: get the version of the AT_iSlave FW | OK |
| AT+VER=? | - | V.x.y | OK |
| Example AT+VER=? | - | 1.0.0 | OK |

## 3.8 RF tests

This section provides a set of commands for the RF test management.

### 3.8.1 AT+TXLRA: Start RF Tx LoRa® test

This command allows the user to start the RF Tx LoRa® test (refer to *Table 44* for details).

Table 44. Start RF Tx LoRa® test command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+TXLRA? | - | AT+TXLRA: set RF Tx LoRa® test | OK |
| AT+TXLRA | Void | Void | AT_ERROR<br>AT_BUSY_ERROR |
| AT+TXLRA =<param> | Freq (kHz):<br>Continuous Tx mode on/off (1/0):<br>Text to be Txed | Void | OK<br>AT_PARAM_ERROR<br>AT_BUSY_ERROR |
| Example AT+TXLRA = | 915200:1:Text Example | ..... | OK |
| Example AT+TXLRA = | 915200:1 | - | AT_PARAM_ERROR<br>(missing text) |

AT_PARAM_ERROR is returned when the setting does not have the correct format, or when it is outside the required set:

- Frequency = 902000-907400kHz, 915200-927800kHz;
- Continuous Mode = {0 (OFF), 1 (ON)};
- Text = from 1 to 64 characters;

### 3.8.2 AT+TXTONE: Start RF Tx Tone test

This command allows the user to start the RF Tx Tone test (refer to *Table 45* for details).
Table 45. Start RF Tx Tone test command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+TXTONE? | - | AT+ RXLRA: Set RF Tx Tone test | OK |
| AT+TXTONE | Void | Void | AT_ERROR<br>AT_BUSY_ERROR |
| AT+TXTONE=<param> | Freq (kHz) | Void | OK<br>AT_PARAM_ERROR<br>AT_BUSY_ERROR |
| Example AT+TXTONE= | 915200 | - | OK |
| Example AT+TXTONE= | 928000 | - | AT_PARAM_ERROR<br>(freq. out of range) |

AT_PARAM_ERROR is returned when the setting does not have the correct format, or when it is outside the required set:

- Frequency = 902000-907400kHz, 915200-927800kHz

### 3.8.3 AT+TIMER: Start Timer test

This command allows the user to start the Timer test (refer to *Table 46* for details).

When the configured timer expires, the module internally calculates and returns the elapsed time. The elapsed time should not exceed a pre-determined tolerance.
Table 46. Start Timer test command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+TIMER? | - | AT+TIMER: Set Timer Test | OK |
| AT+TIMER | Void | Void | AT_ERROR<br>AT_BUSY_ERROR |
| AT+TIMER=<param> | Time (seconds) | Void | OK<br>AT_PARAM_ERROR<br>AT_BUSY_ERROR |
| Example AT+TIMER= | 2 | Elapsed time: 2001ms | OK |
| Example AT+TIMER= | **10** | - | AT_PARAM_ERROR<br>(time out of range) |

AT_PARAM_ERROR is returned when the setting does not have the correct format, or when it is outside the required set:

- Time = 1 – 4 seconds

### 3.8.4 AT+RXLRA: Start RF Rx LoRa® test

This command allows the user to start the RF Rx LoRa® test (refer to *Table 47* for details).

Table 47. Start RF Rx LoRa® test command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+RXLRA? | - | AT+ RXLRA: starts Rx LoRa® test | OK |
| AT+RXLRA | Void | Void | AT_ERROR<br>AT_BUSY_ERROR |
| AT+RXLRA =<param> | Freq (kHz):<br>Continuous Rx mode on/off (1/0) | Void | OK<br>AT_PARAM_ERROR<br>AT_BUSY_ERROR |
| Example AT+RXLRA = | 915200:0 | RSSI=-9 dBm SNR=6 dBm Rx Text-> Text Example | OK |
| Example AT+RXLRA = | **928000**:1 | - | AT_PARAM_ERROR<br>(freq. out of range) |

AT_PARAM_ERROR is returned when the setting does not have the correct format, or when it is outside the required set:

- Frequency = 902000-907400kHz, 915200-927800kHz;
- Continuous Mode = {0 (OFF), 1 (ON)};

### 3.8.5 AT+TCONF: Config LoRa® RF test

This command allows the user to access the LoRa® configuration test (refer to *Table 48* for details).

Table 48. Config LoRa® RF test command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+TCONF? | - | T+TCONF: configure LoRa® RF test | OK |
| AT+TCONF=? | Void | Void | OK AT_ERROR |
| AT+TCONF=<param> | Void | Void | OK AT_PARAM_ERROR |
| Example AT+TCONF=? | - | Freq = 915200 kHz Power = 22 dBm Bandwidth = 125 KHz SF = 9<br>CR = 4 / 8<br>LNA State = 0 PA boost state = 0 | OK |
| Example AT+TCONF= | 915200:12:125:12:4/8:0:0 | - | OK |
| Example AT+TCONF= | 915200:12:**300**:12:4/8:0:0 | - | AT_PARAM_ERROR<br>(error on bandwidth setting) |

AT_PARAM_ERROR is returned when the setting does not have the correct format (being a decimal value), or when it is outside the required set:

- Bandwidth = {125, 250, 500};
- SF = {7, 8, 9, 10, 11, 12};
- CR = {4/5, 4/6, 4/7, 4/8}.

### 3.8.6 AT+TOFF: Stop ongoing radio frequency test

This command allows the user to stop the ongoing RF test or Timer test (refer to *Table 49* for details).

Table 49. Stop radio frequency test command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+TOFF? | - | AT+TOFF: stop ongoing RF test | OK |
| AT+TOFF | Void | Void | OK |

### 3.8.7 AT+CERTIF: Set the module in LoRaWAN™ Certification mode

This command allows the user to start the RF Rx LoRa test (refer to *Table 50* for details).

Table 50. Set the module in LoRaWAN™ Certification mode command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+CERTIF? | - | AT+CERTIF: set the module in LoraWAN™ Certification mode | OK |
| AT+CERTIF | Void | Void | OK<br>AT_BUSY_ERROR |

AT+CERTIF puts the timer to handler data transmission equal to 5 s.

## 3.9 GPIO

This section provides a set of commands for the GPIO control.

### 3.9.1 AT+GPIO: Get or Set the GPIO

This command allows the user to control GPIOs (refer to *Table 51* for details).

Table 51. Get or Set the GPIO command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+GPIO? | - | AT+GPIO: Get or Set the GPIO # | OK |
| AT+GPIO | Void | Void | AT_PARAM_ERROR |
| AT+GPIO=? | Void | Void | OK |
| AT+GPIO=<param> | Port:Cmd | Void | OK<br>AT_PARAM_ERROR |
| AT+GPIO=? | Void | GPIO 3 = 1<br>GPIO 5 = 0<br>GPIO 6 = 0 | OK |
| Example AT+GPIO= | 1:U | ..... | OK |
| Example AT+GPIO= | 3:0 | - | OK |

AT+GPIO=? command only shows the state from GPIOs which were previously set up with Z or U Cmd value.

AT_PARAM_ERROR is returned when the setting does not have the correct format, or when it is outside the required set:

- Port = 1-6;
- Cmd = 0 Pin drives low
  1 Pin drives high
  Z Pin is high impedance input
  U Pin is input with pull−up
  O Pin is output (1) (*)
  S Pin is switch to transmit a pre-configured text (2)
  A Pin is analog input (3)
  T Pin is UART transmitter (TX) (4)
  R Pin is UART receiver (RX) (5)

**(*) NOTE:** The O, S, A, T and R Cmds are not available on all Hardware and Firmware versions. Currently, only accepted for the SMW-SX1262M0 board.

1. Every data received from the Network Server in the corresponding Port (121 to 126) will be sent to the output;
2. When the switch is activated, the module transmits a pre-configured text through the corresponding Port (121 to 126);
3. This command is accepted only for GPIO 3. The acquired analog value will be transmitted to the Network Server through Port 123. If the GTIME command was executed before this command, GPIO3 will be activated and period will be maintained; otherwise, GPIO3 will be activated and period will be set to the default value (1 minute);
4. This command is accepted only for GPIO 5. Every data received from the Network Server at Port 125 will be transmitted through this pin;
5. This command is accepted only for GPIO 6. Every data received by this pin will be transmitted to the Network Server at Port 126. Data will be transmitted up to 32 bytes at a time; there isn't an expected terminating character, after receiving the first character, the receive channel will be opened for 200ms. This command can also be used in conjunction with the GTIME command.

Refer to Appendix A for further details on remote GPIO operations.

| Port Number | Function Associated |
|---|---|
| 120 | TIME |
| 121 | GPIO1 |
| 122 | GPIO2 |
| 123 | GPIO3 |
| 124 | GPIO4 |
| 125 | GPIO5 |
| 126 | GPIO6 |
| 127 | TEXT |
| 128 | BATTERY LEVEL |
| 129 | TEMPERATURE |

### 3.9.2 AT+TEXT: Get or Set the Text to be transmitted

This command allows the user to save a 16 text characters that will be used by Switch command and GTIME command (refer to *Table 52* for details). (*)

Table 52. Send text data command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+TEXT? | - | AT+TEXT: Get or Set the text that will be transmitted when the configured switch is pressed | OK |
| AT+TEXT=<input> | text | - | OK |
| AT+TEXT=? | Void | text | OK |
| Example AT+TEXT= | hello world | - | OK |
| Example AT+TEXT=? | - | hello world | OK |

**(*) NOTE:** The TEXT command is not available on all Hardware and Firmware versions. Currently, only accepted for the SMW-SX1262M0 board.

### 3.9.3 AT+GTIME: Get or Set the input acquisition period

This command allows the user to define the acquisition/transmission period (in minutes) that will be used by the GPIOs configured as inputs, a pre-configured Text, the battery level and the temperature. The sequential values passed as input parameter determine the activation or deactivation of these values. Refer to *Table 53* for details. (*)

When the timer expires, each configured value will be read and transmitted sequentially to the Network Server through the corresponding Port (121 to 129).

Table 53. Get or Set the input acquisition period command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+GTIME? | - | AT+ GTIME: Get or Set the acquisition time in minutes | OK |
| AT+GTIME=<param> | Period:GPIO1:GPIO2:GPIO3:GPIO4:GPIO5:GPIO6:TEXT:BATT:TEMP | - | OK<br>AT_PARAM_ERROR(3) |
| AT+GTIME=? | Void | Period | OK |
| Example AT+GTIME= | 5:1:0:1:0:1:0:1:1:0(2) | - | OK |
| Example AT+GTIME=? | - | 5:1:0:1:0:1:0:1:1:0 | OK |
| Example AT+GTIME= | 0(3) | - | OK |

**(*) NOTE:** The GTIME command is not available on all Hardware and Firmware versions. Currently, only accepted for the SMW-SX1262M0 board.

1. In this example, the GPIOs that will be acquired and transmitted are GPIO1, GPIO3 and GPIO5 (Ports 121, 123 and 125). Also, the pre-configured Text will be transmitted through Port 127 and Battery Level through Port 128.

| Command | TIME | GPIO1 | GPIO2 | GPIO3 | GPIO4 | GPIO5 | GPIO6 | TEXT | BATTERY LEVEL | TEMPERATURE |
|---|---|---|---|---|---|---|---|---|---|---|
| AT+GTIME | Period minutes | ON/ OFF | ON/ OFF | ON/ OFF | ON/ OFF | ON/ OFF | ON/ OFF | ON/ OFF | ON/ OFF | ON/ OFF |
| Example: AT+GTIME= | 5 | :1 | :0 | :1 | :0 | :1 | :0 | :1 | :1 | :0 |

The transmitted payload format is the following:

| Port Number | Payload Format | Configuration |
|---|---|---|
| 121 | "GP1=0;" | Digital Input (0 or 1) |
| | "TXT=SMART_MODULAR;" | Switch (pre-configured Text) |
| 122 | "GP2=1;" | Digital Input (0 or 1) |
| | "TXT=SMART_MODULAR;" | Switch (pre-configured Text) |
| 123 | "GP3=0;" | Digital Input (0 or 1) |
| | "TXT=SMART_MODULAR;" | Switch (pre-configured Text) |
| | "GP3=4095;" | Analog Input (From 1 to 4095) |
| 124 | "GP4=1;" | Digital Input (0 or 1) |
| | "TXT=SMART_MODULAR;" | Switch (pre-configured Text) |
| 125 | "GP5=0;" | Digital Input (0 or 1) |
| | "TXT=SMART_MODULAR;" | Switch (pre-configured Text) |
| 126 | "GP6=1;" | Digital Input (0 or 1) |
| | "TXT=SMART_MODULAR;" | Switch (pre-configured Text) |
| 127 | "TXT=SMART_MODULAR;" | Text (pre-configured Text) |
| 128 | "BAT=3300;" | Battery Level (value in mV) |
| 129 | "TMP=29.5;" | Temperature (value in ºC) |

1. When the input parameter is “0”, the timer will be deactivated and the acquisition and transmissions will stop.
2. AT_PARAM_ERROR is returned when the setting does not have the correct format, or when it is outside the required set:
    - Period = From 0 to 1440 minutes;
    - Port Activation = {0 (OFF), 1 (ON)}.

## 3.10 Non Volatile Configuration

This section provides a set of commands for saving and restoring all module configurations.

### 3.10.1 AT+SAVE: Save the current configuration

This command allows the user to save the current configuration of the module (refer to *Table 54* for details). **(*)**

Table 54. Save the current configuration command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+SAVE? | - | AT+SAVE: Save current configuration | OK |
| AT+SAVE(1) | Void | Void | OK |

1. Any new configuration made after the AT+SAVE command will be lost if the module is reset without another call to the AT+SAVE command.

Refer to Appendix B for the complete list of parameters that are saved with this command.

**(*) NOTE:** The SAVE command is not available on all Hardware and Firmware versions. Currently, only accepted for the SMW-SX1262M0 board.

### 3.10.2 AT+DEFLT: Restore the default manufacturer configuration

This command allows the user to restore the default manufacturer configuration (refer to *Table 55* for details). **(*)**

Table 55. Restore the default manufacturer configuration command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+DEFLT? | - | AT+DEFLT: Restore default configuration | OK |
| AT+DEFLT | Void | Void | OK |

**(*) NOTE:** The DEFLT command is not available on all Hardware and Firmware versions. Currently, only accepted for the SMW-SX1262M0 board.

### 3.10.3 AT+AJOIN: Get or Set the Auto-Join mode

This command allows the user to set the Auto-Join mode when the module is set to OTAA (refer to *Table 56* for details). (*)

Table 56. Get or Set the Auto-Join mode command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+AJOIN? | - | AT+AJOIN: Get or Set the Auto-Join mode<br>- 0 = disable<br>- 1 = enable | OK |
| AT+AJOIN=? | Void | 0 or 1 | OK |
| AT+AJOIN=<Input> | 0 or 1 | - | OK/ AT_PARAM_ERROR(1) |
| Example AT+AJOIN? | - | 1 | OK |
| Example AT+AJOIN= | 1 | - | OK |

1. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

**(*) NOTE:** The AJOIN command is not available on all Hardware and Firmware versions. Currently, only accepted for the SMW-SX1262M0 board.

### 3.10.4 AT+SETPMODE: Set Protect Mode

This command allows the user to set the Protect mode. In this mode of operation, all keys, IDs and EUIs (described in *section 3.2*) are not displayed over UART. In order to set the Protect mode, an alphanumeric password (containing from 6 to 32 characters) needs to be configured (refer to *Table 57* for details). (*)

Table 57. Set protect mode command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+SETPMODE? | - | AT+SETPMODE: Set Protect Mode | OK |
| AT+SETPMODE= <input> | text | - | OK / AT_PARAM_ERROR |
| Example AT+SETPMODE= | PASS123456 | - | OK |

1. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

**(*) NOTE:** The SETPMODE command is not available on all Hardware and Firmware versions. Currently, only accepted for the SMW-SX1262M0 board.

### 3.10.5 AT+RESETPMODE: Reset Protect Mode

This command allows the user to reset the Protect mode. All keys, IDs and EUIs (described in *section 3.2*) are displayed over UART. In order to reset the Protect mode, the configured alphanumeric password needs to be passed as the input parameter. (refer to *Table 58* for details). **(*)**

Table 58. Reset protect mode command

| Command | Input parameter | Return value | Return code |
|---|---|---|---|
| AT+RESETPMODE? | - | AT+RESETPMODE:<br>Reset Protect Mode | OK |
| AT+RESETPMODE=<br><input> | text | - | OK /<br>AT_PARAM_ERROR |
| Example<br>AT+RESETPMODE= | PASS123456 | - | OK |

1. AT_PARAM_ERROR is returned when setting a wrong or malformed value.

**(*) NOTE:** The RESETPMODE command is not available on all Hardware and Firmware versions. Currently, only accepted for the SMW-SX1262M0 board.

# 4 Examples

This section provides examples of join and send, receiving and confirmation of data.

## 4.1 Join and send

This example shows the complete join procedure and the way to send data on the LoRa® link.

```
# AT
<CR><LF>OK<CR><LF>
# AT+JOIN
<CR><LF>OK<CR><LF>
# AT+NJS=?
0<CR><LF>/* Network is not joined yet */
<CR><LF>OK<CR><LF>
/* wait for few seconds to wait for join to complete */
# AT+NJS=?
1<CR><LF>/* Network is now joined */
<CR><LF>OK<CR><LF>
/* now the network is joined, data can be sent */
# AT+SEND=50:Hello World/* Send text to port 50 */
<CR><LF>OK<CR><LF>
# AT+SENDB=60:0123 /* Send data (2 bytes: 0x01 and 0x23) on port 60 */
<CR><LF>OK<CR><LF>
/* Note that the result could be AT_BUSY_ERROR in case the previous send is
not completed, because of the duty cycle restriction, or because RX windows
are not completed */
```

## 4.2 Confirmation

This example shows how to transmit data on the LoRa® link. This example assumes that the network is already joined.

```
# AT+NJS=?
1<CR><LF>/* Network is already joined */
<CR><LF>OK<CR><LF>
# AT+CFM=?
0<CR><LF>/* Unconfirmed data */
<CR><LF>OK<CR><LF>
# AT+CFM=1
<CR><LF>OK<CR><LF>
# AT+CFM=?
0<CR><LF>/* Confirmed data */
<CR><LF>OK<CR><LF>
# AT+SEND=50:Hello World/* Send text to port 50 */
<CR><LF>OK<CR><LF>
# AT+CFS=?
0<CR><LF>/* Message is not confirmed yet */
<CR><LF>OK<CR><LF>
```

```
/* wait for few seconds to wait for the confirmation */
# AT+CFS=?
# AT+NJS=?
1<CR><LF>/* Network is already joined */
<CR><LF>OK<CR><LF>
# AT+CFM=?
0<CR><LF>/* Unconfirmed data */
<CR><LF>OK<CR><LF>
# AT+CFM=1
<CR><LF>OK<CR><LF>
# AT+CFM=?
0<CR><LF>/* Confirmed data */
<CR><LF>OK<CR><LF>
# AT+SEND=50:Hello World/* Send text to port 50 */
<CR><LF>OK<CR><LF>
# AT+CFS=?
0<CR><LF>/* Message is not confirmed yet */
<CR><LF>OK<CR><LF>
/* wait for few seconds to wait for the confirmation */
# AT+CFS=?
```

## 4.3 Receiving data

This example shows how to receive data on the LoRa® link, with the assumption that the network is already joined.

```
# AT+NJS=?
1<CR><LF>/* Network is already joined */
<CR><LF>OK<CR><LF>
# AT+RECV=?
0:<CR><LF>/* Nothing has been received */
<CR><LF>OK<CR><LF>
/* Server is sending Binary Data 0x01 0xA0 0x23 on port 20
   Need to send data to receive the one from the server */
# AT+SENDB=2:ab
<CR><LF>OK<CR><LF>
# AT+RECVB=?
0:<CR><LF>/* Nothing has been received yet */
<CR><LF>OK<CR><LF>
/* wait the received windows to complete */
# AT+RECVB=?
20:01a023<CR><LF> /* Message is now received */
<CR><LF>OK<CR><LF>
# AT+RECVB=?
20:<CR><LF>/* No new message received */
<CR><LF>OK<CR><LF>
```

## 4.4 Class B enable request

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

```
#AT+JOIN
<CR><LF>OK<CR><LF>
/* wait for few seconds to have a join complete */
# AT+NJS=?
1<CR><LF>                    /* Network is now joined */
<CR><LF>OK<CR><LF>
/* now the network is joined, data can be sent */
/* --> A build in MAC message is sent to the network  to acquire the system
time "Device Time req"*/
AT+SEND=2:hello             /* Send data will allow to piggybacking the MAC
Device Time Req -could be a dummy message*/
<CR><LF>OK<CR><LF>
/* --> MAC Ping Device Time ANS is received by end-node in hidden way*/
AT+CLASS=B                   /* Request to switch to Class B "enable"*/
<CR><LF>OK<CR><LF>
AT+CLASS=?
B, S0<CR><LF>                /*Beacon Acquisition on-going*/
+BC: LOCKED<CR><LF>          /*Asynchronous event : End-Node locked on
Beacon*/
/* now the End-node is locked, Beacon Time can be requested */
AT+BTIME=?
BTIME: 1538759296<CR><LF>
<CR><LF>OK<CR><LF>
/* --> A build in MAC messages are sent to the network "link check req" and
"ping slot info req"*/
AT+SEND=2:hello         /* will allow to piggybacking the MAC messages */
<CR><LF>OK<CR><LF>
/* --> MAC Ping Slot Info ANS is received by end-node in hidden way*/
+PS: DONE<CR><LF>            /*Asynchronous event : pingslot will be opened
periodically*/
/* now the end-node is Class B "enable"*/
AT+CLASS=?
B<CR><LF>                    /*Class B "enable"*/
+BC: LOST<CR><LF>        /* Asynchronous event: Beacon lost, modem does a
Beacon Reacquisition:*/
+BC: REACQ<CR><LF>       /* Asynchronous event : Beacon Reacquisition:*/
+BC: LOCKED<CR><LF>     /* Asynchronous event : Beacon found. The modem is
synchronized
/* Since the End-node is locked, Local Time can be requested */
AT+LTIME=?
LTIME: 17h08m16s on 05/10/2018<CR><LF>
<CR><LF>OK<CR><LF>
```

## 4.5 Set GTIME

This example shows how to configure GTIME.

```
# AT+JOIN
<CR><LF>OK<CR><LF>
/* wait for few seconds to wait for join to complete */
# AT+NJS=?
1<CR><LF>/* Network is now joined */
<CR><LF>OK<CR><LF>
/* now the network is joined, data can be sent */
# AT+GPIO=1:Z
<CR><LF>OK<CR><LF>
# AT+GPIO=2:U
<CR><LF>OK<CR><LF>
# AT+GPIO=3:A
<CR><LF>OK<CR><LF>
# AT+GPIO=4:O
<CR><LF>OK<CR><LF>
# AT+GPIO=5:T
<CR><LF>OK<CR><LF>
# AT+GPIO=6:R
<CR><LF>OK<CR><LF>
# AT+GTIME=5:1:1:1:1:1:1:1:1:1 /* Configure 5 minutes timer */
<CR><LF>OK<CR><LF>
/* After 5 minutes, the module will send GPIO1, GPIO2, GPIO3, GPIO6, TEXT,
Battery Level and Temperature values to the network server */
/* Although GTIME was configured for every GPIO, GPIO4 and GPIO5 values are
not sent, since they're not configured as input */
```

## 4.6 Save current configuration

This example shows how to save the parameters currently configured in the module.

```
AT+AJOIN=1
<CR><LF>OK<CR><LF>
AT+ADR=0
<CR><LF>OK<CR><LF>
AT+DR=2
<CR><LF>OK<CR><LF>
AT+CFM=1
<CR><LF>OK<CR><LF>
AT+TEXT=Warning!
<CR><LF>OK<CR><LF>
# AT+GPIO=1:U
<CR><LF>OK<CR><LF>
# AT+GPIO=4:U
<CR><LF>OK<CR><LF>
```

```
# AT+GTIME=1:1:0:0:1:0:0:1:1:0
<CR><LF>OK<CR><LF>
AT+SAVE
<CR><LF>OK<CR><LF>
ATZ
/* The module is reset and gets the configuration that was configured before
the SAVE command */
/* Since Auto-Join is enabled, it will Join the Network Server and start
sending the values configured with GTIME */
AT+AJOIN=?
1<CR><LF>
<CR><LF>OK<CR><LF>
AT+ADR=?
0<CR><LF>
<CR><LF>OK<CR><LF>
AT+DR=?
2<CR><LF>
<CR><LF>OK<CR><LF>
AT+CFM=?
1<CR><LF>
<CR><LF>OK<CR><LF>
AT+TEXT=?
Warning!<CR><LF>
<CR><LF>OK<CR><LF>
```

## 4.7 Restore default configuration

This example shows how to restore the default configuration of the module.

```
/* After changing and saving the configuration as in the last example */
AT+DEFLT /* Execute the DEFLT command to restore default configuration */
<CR><LF>OK<CR><LF>
ATZ
/* The module is reset and gets the default configuration */
AT+AJOIN=?
0<CR><LF>
<CR><LF>OK<CR><LF>
AT+ADR=?
1<CR><LF>
<CR><LF>OK<CR><LF>
AT+DR=?
0<CR><LF>
<CR><LF>OK<CR><LF>
AT+CFM=?
0<CR><LF>
<CR><LF>OK<CR><LF>
AT+TEXT=?
SMART_MODULAR<CR><LF>
<CR><LF>OK<CR><LF>
```

# 5 Appendix A

This section describes operations that can be performed from the Network Server side. In addition to the commands described in *section 3.9*, the following functionalities were created to allow controlling GPIOs remotely (refer to *Table 59* for details).

Table 59. Function associated to each Port

| Port Number | Function Associated |
|---|---|
| 120 | TIME |
| 121 | GPIO1 |
| 122 | GPIO2 |
| 123 | GPIO3 |
| 124 | GPIO4 |
| 125 | GPIO5 |
| 126 | GPIO6 |
| 127 | TEXT |
| 128 | BATTERY LEVEL |
| 129 | TEMPERATURE |

**(*) NOTE:** The operations described in this section are not available on all Hardware and Firmware versions. Currently, only accepted for the SMW-SX1262M0 board.

## 5.1 Update timer value

After the execution of the GTIME command or Analog command, the acquisition period can be reconfigured by Network Server through Port 120.

The new period will begin after the current timer expires. The value transmitted through Port 120 must be between 1 and 1440 (minutes).

If the new period is equal to zero, the timer will stop.

The Network Server is not able to start the timer and set the active ports that will be acquired and transmitted, the GTIME command or Analog command must be executed.

## 5.2 Update text

This functionality is identical to the TEXT command. The Network Server is able to update the text that will be used by Switch command and GTIME command through Port 127.

The module will store up to 16 text characters that are transmitted through Port 127.

## 5.3 Control Outputs

When a GPIO was configured as Output, Network Server is able to control this GPIO remotely through the corresponding Port (121 to 126), refers to Table 59.

If value 1 is transmitted through Ports 121 to 126, the corresponding GPIO (1 to 6) will be set.
If value 0 is transmitted through Ports 121 to 126, the corresponding GPIO (1 to 6) will be reset.

## 5.4 Serial Transmission and Reception

When GPIO5 was configured as an UART transmitter pin (TX), Network Server is able to transmit data through Port 125. Every data received by the module will be transmitted through GPIO5.

When GPIO6 was configured as an UART receiver pin (RX), Network Server will receive data through Port 126. Every data received by the module at GPIO6 will be transmitted through Port 126. Data will be transmitted up to 32 bytes at a time.

# 6 Appendix B

This section describes the configuration parameters that are saved when the *Non Volatile Configuration* commands are executed. Refer to *section 3.10* for details regarding the execution of these commands.

The following *Table 60* lists all configuration parameters that can be saved (Refer to each corresponding section for details):

> **(*) NOTE:** The SAVE, DEFLT and AJOIN commands are not available on all Hardware and Firmware versions. Currently, only accepted for the SMW-SX1262M0 board.

Table 60. Configuration Parameters that can be saved

| Configuration Parameter | Value | Observation |
|---|---|---|
| AT+DADDR | <4 hexa separated by:> | Saved only when in ABP Mode |
| AT+APPKEY | <16 hexa separated by:> | Saved only when in OTAA Mode |
| AT+NWKSKEY | <16 hexa separated by:> | Saved only when in ABP Mode |
| AT+APPSKEY | <16 hexa separated by:> | Saved only when in ABP Mode |
| AT+APPEUI | <8 hexa separated by:> | - |
| AT+ADR | (0: off, 1: on) | - |
| AT+TXP | (0-5) | - |
| AT+DR | (0-7 corresponding to DR_X) | - |
| AT+DCS | (0: disable, 1: enable) | - |
| AT+PNM | (0: off, 1: on) | - |
| AT+RX2FQ | Frequency in Hz | - |
| AT+RX2DR | (8-13 corresponding to DR_X) | - |
| AT+RX1DL | Delay in ms | - |
| AT+RX2DL | Delay in ms | - |
| AT+JN1DL | Delay in ms | - |
| AT+JN2DL | Delay in ms | - |
| AT+NJM | (0: ABP, 1: OTAA) | - |
| AT+NWKID | <4 hexa separated by:> | Saved only when in ABP Mode |
| AT+CLASS | (A, B or C) | - |
| AT+CFM | (0: off, 1: on) | - |
| AT+GPIO | Port and Cmd | If configured as Output, the current state is also saved |
| AT+TEXT | 16 characters text | - |
| AT+GTIME | Period and Active GPIOs | - |
| AT+AJOIN | (0: disable, 1: enable) | - |
| AT+SETPMODE | Alphanumeric password | - |