AN4967
Application note

Examples of AT commands on I-CUBE-LRWAN

Introduction

I-CUBE-LRWAN is a LoRaWAN® software Expansion Package for STM32Cube, consisting in a set of libraries and application examples for microcontrollers of the STM32L0, STM32L1 and STM32L4 Series acting as end devices.

The I-CUBE-LRWAN main features are:

- Easy add-on of the low-power LoRa® solution
- Extremely low CPU load
- No latency requirements
- Small STM32L0 Series memory footprint

This application note describes the set of AT commands for the B-L072Z-LRWAN1 Discovery board embedding the CMWX1ZZABZ-091 LoRa® module.

This document explains how to interface with the LoRaWAN® to manage the LoRa® wireless link using AT commands.

For more information on the LoRa® embedded Expansion Package (I-CUBE-LRWAN) implementation on the STM32Lx Series, refer to *STM32 LoRa® Expansion Package for STM32Cube* (UM2073), available at *www.st.com*.

Contents

List of tables

List of figures

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

2 Reference documents

1. LoRaWAN® Specification by LoRa Alliance™ (Version 1.0.3, March 2018, Final, Released), available at www.lora-alliance.org
2. *STM32 LoRa software expansion for STM32Cube* (UM2073), available at *www.st.com*

3 Overview

The B-L072Z-LRWAN1 Discovery board embeds the CMWX1ZZABZ-091 LoRa® firmware.

This firmware implements the AT_Slave module (see document *2*) that supports a set of AT commands to drive the LoRaWAN® communications and the LoRa® RF test.

It applies to microcontrollers of the STM32L0, STM32L1 and STM32L4 Series, all based on Arm®(a) cores.

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

a. Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

4 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 for performing 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

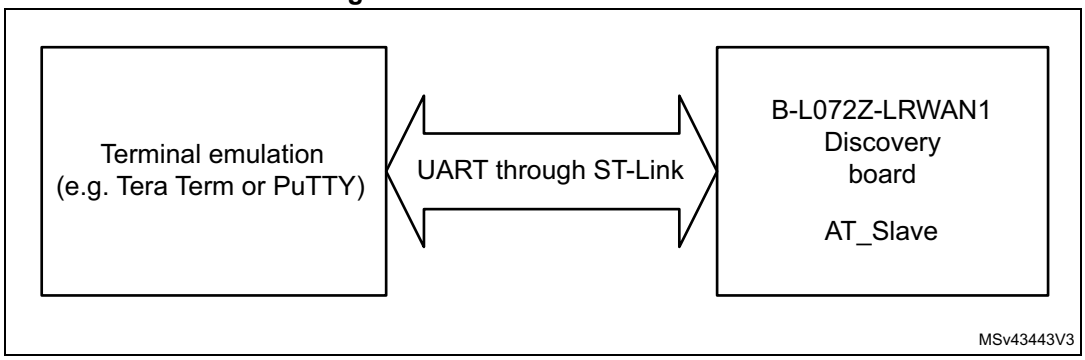
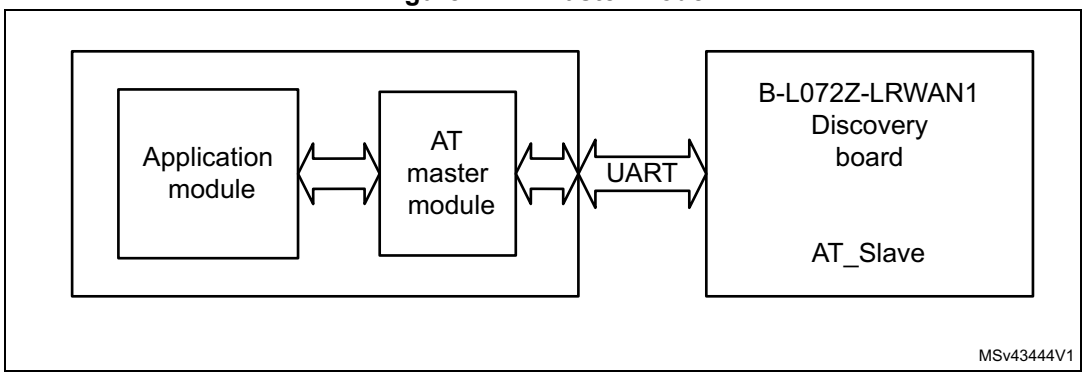


Figure 2. AT master mode



For illustration purposes, the rest of the document is based on the relation “terminal emulation” with the B-L072Z-LRWAN1 Discovery board.

An UART over ST-Link 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 ST-LINK.

Figure 3. Tera Term port setup example

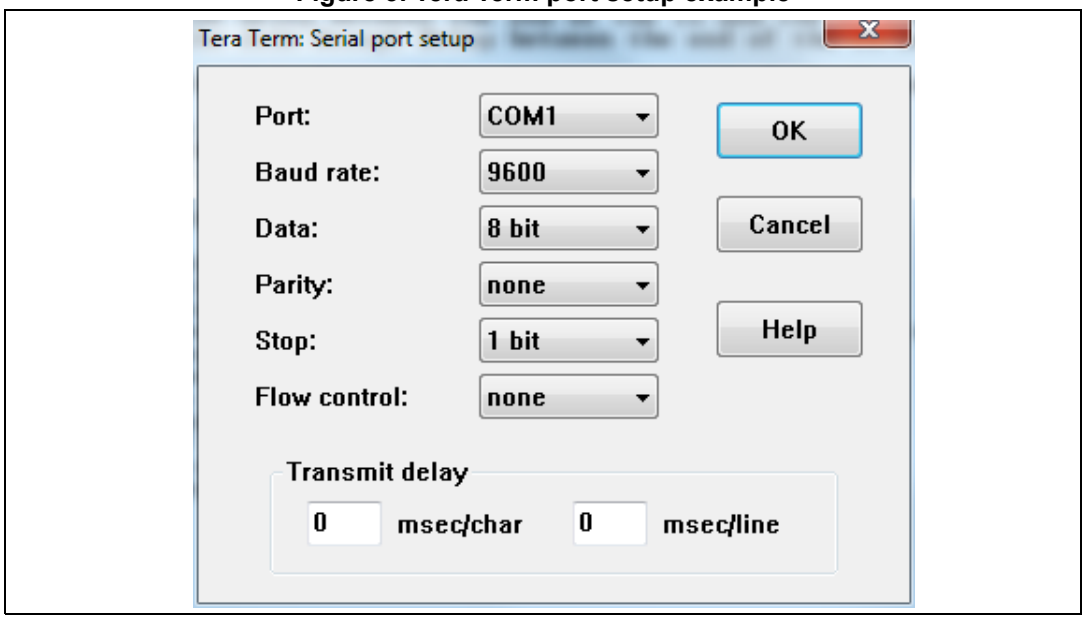
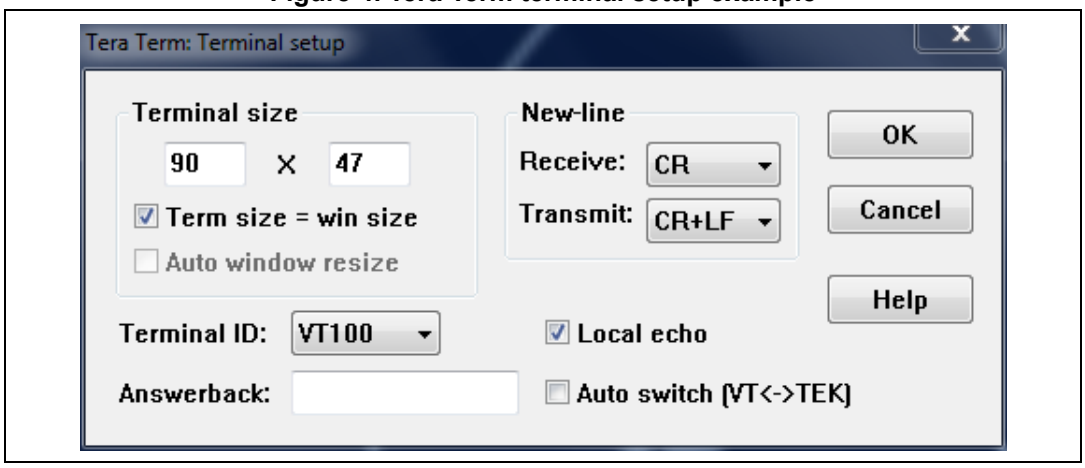


Figure 4. Tera Term terminal setup example



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+TOFF**
- **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=2:0:1234**

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. 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. Each command preceded by # is provided by the host to the module, then the return of the module is printed.

4.1 General commands

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

4.1.1 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

4.1.2 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 AT+<CMD>: run <CMD> AT+<CMD>=<value>: set the value AT+<CMD>=?: get the value <followed by the help of all commands>	OK

4.1.3 ATZ: MCU reset

This command is used to trigger a CPU reset of the B-L072Z-LRWAN1 Discovery 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

4.1.4 AT+VL: Verbose level

This command is used to set/get the verbose level of the application (refer to *Table 5* for details).

Table 5. Verbose level

Command	Input parameter	Return value	Return code
AT+VL?	-	AT+VL get or set the verbose level	OK
AT+VL=?	-	0 (VLEVEL_OFF) or 1 (VLEVEL_L) or 2 (VLEVEL_M or 3 (VLEVEL_H)	OK
AT+VL=<param>(1)	0 or 1 or 2 or 3	-	OK / AT_PARAM_ERROR(2)
Example AT+VL=	1	-	OK
Example AT+VL=?	-	1	OK

1. 0 = VLEVEL_OFF
   1 = VLEVEL_L
   2 = VLEVEL_M
   3 = VLEVEL_H.
2. Returned when setting a wrong or malformed value.

4.2 Keys, IDs and EUIs management

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

4.2.1 AT+APPEUI: Application identifier

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

Table 6. 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=	01:02:03:04:05:06:07:08	-	OK
	01:2:03:04:05:06:07:08	-	AT_PARAM_ERROR(1)
Example AT+APPEUI=?	-	01:02:03:04:05:06:07:08	OK

1. Returned when setting a wrong or malformed value.

4.2.2 AT+DADDR: Device address

This command allows the user to access the device address (refer to *Table 7* for details).

Table 7. 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=	11:22:33:44	-	OK
Example AT+DADDR=?	11:22:33:44	11:22:33:44	OK

1. Returned when setting a wrong or malformed value.

4.2.3 AT+DEUI: Device EUI

This command allows the user to access the global end-device ID (refer to *Table 8* for details).

Table 8. Device EUI command

Command	Input parameter	Return value	Return code
AT+DEUI?	-	AT+DEUI: get or set the device EUI	OK
AT+DEUI=?	-	<8 hexa separated by :>	OK
AT+DEUI= <Param>	<8 hexa separated by :>	-	OK / AT_PARAM_ ERROR(1)
Example AT+DEUI=?	-	11:22:33:44:55:66:77:88	OK
Example AT+DEUI=	11:22:33:44:55:66:77:88	-	OK

1. Returned when setting a wrong or malformed value.

4.2.4 AT+NWKID: Network ID

This command allows the user to access the network identifier (refer to *Table 9* for details).

Table 9. 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. Returned when setting a wrong or malformed value.

4.3 Joining and sending data on LoRa network

This section describes the commands related to the join procedure and to the data path.

4.3.1 AT+JOIN: Join LoRa network

This command does a join request to the network (refer to *Table 10* for details).

Table 10. Join LoRa network command

Command	Input parameter	Return value	Return code
AT+JOIN?	-	AT+JOIN: join network	OK
AT+JOIN=<param>	– 0: indicates the ABP join mode – 1: indicates the OTAA join mode	See *Section 4.6*	OK / AT_BUSY_ERROR(1)
Example(2) AT+JOIN=	1	+EVT: JOINED	OK
Example(3) AT+JOIN=	1	+EVT: JOIN FAILED	OK

1. Returned when a joining process is already running.
2. OTAA join request procedure successful.
3. OTAA join request procedure fail.

This is an asynchronous command. OK means that the join is running, its completion is notified by an asynchronous event (see *Section 4.6*).

4.3.2 AT+SEND: Send binary data

This command provides the way to send text a confirmed or unconfirmed data message in binary format on a dedicated port number (refer to *Table 11* for details).

Each data byte is provided as two characters denoting the value in hexadecimal. Hence, the length is always even.

In the example of *Table 11*, 8 bytes are sent on port 12:1: 0xAB, 0xCD, 0xEF, 0x01 (note that the example passes “01”, passing only “1” fails), 0x23, 0x45, 0x67 and 0x89.

Table 11. Send data command

Command	Input parameter	Return value	Return code
AT+SEND?	-	AT+SEND: send hexadecimal data along with the application port	OK
AT+SEND= <input>	<port>:<ack>:<payload> – 0: indicates an unconfirmed message – 1: indicates a confirmed message	-	OK/ AT_PARAM_ERROR(1)/ AT_BUSY_ERROR(2)/ AT_NO_NETWORK_JOINED(3)/ AT_DUTYCYCLE_RESTRICTED(4)/ AT_CRYPTO_ERROR(5)
Example AT+SEND=	12:1:abcdef0123456789	-	OK
	abcdef0123456789	-	AT_PARAM_ERROR

1. Returned when the setting has not the correct format <port>:<ack>:<payload>, where <port> and <ack> are a decimal value and <payload> follows hexadecimal format using two characters as described above.
2. Returned when the previous send is not complete (e.g. send waiting for duty cycle, rx window not consumed).
3. Returned when the network is not joined yet.
4. Returned due to the duty cycle constraint.
5. Returned when the modem has encountered a problem during the cryptographic process.

4.4 LoRa network management

This section provides a set of commands for network management.

4.4.1 AT+ADR: Adaptive rate

This command allows the user to access the adaptive data rate (refer to *Table 12* for details). The default value of the ADR is 1 (enabled).

Table 12. 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. Returned when setting a wrong or malformed value.

4.4.2 AT+CLASS: LoRa class

This command allows the user to access the LoRaWAN class (refer to *Table 13* for details).

Table 13. 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(2) or C	OK
AT+CLASS=<Input>	A, B 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. Returned when setting a wrong or malformed value.

4.4.3 AT+DCS: Duty cycle settings

This command allows the user to access the duty cycle parameter (refer to *Table 14* for details).

Table 14. Duty cycle settings command

Command	Input parameter	Return value	Return code
AT+DCS?	-	AT+DCS: get or set the ETSI duty cycle setting: – 0 = disable (testing only, refer to document 2) – 1 = enable (default setting)	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. Returned when setting a wrong or malformed value.

4.4.4 AT+DR: Data rate

This command allows the user to access the data rate (refer to *Table 15* for details).

Table 15. Data rate command

Command	Input parameter	Return value	Return code
AT+DR?	-	AT+DR: get or set the data rate (0-7 corresponding to DR_X)	OK
AT+DR=?	-	[0,1,2,3,4,5,6,7]	OK
AT+DR=<Input>	[0,1,2,3,4,5,6,7]	-	OK/ AT_PARAM_ERROR(1)
Example AT+DR=?	-	3	OK
Example AT+DR=	2	-	OK

1. Returned when setting a wrong or malformed value.

4.4.5 AT+BAND: Active region

This command is used to set the active region of the end-device (refer to *Table 16* for details).

Table 16. Active region

Command	Input parameter	Return value	Return code
AT+BAND?	-	AT+BAND: get or set the active region	OK
AT+BAND=?	-	[0,1,3,4,5,6,7,8,9]	OK
AT+BAND=<param>	[0,1,2,3,4,5,6,7,8,9]	-	OK/ AT_PARAM_ERROR(1)
Example AT+BAND=	5	-	OK
Example AT+BAND=?	-	5	OK

1. Returned when setting a wrong or malformed value, or when the region is not supported by the modem (not compiled in the binary).

4.4.6 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 17* for details).

Table 17. 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. Returned when a join or a send is being processed.
2. Returned when setting a wrong or malformed value.

4.4.7 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 18* for details).

Table 18. 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. Returned when a join or a send is being processed.
2. Returned when setting a wrong or malformed value.

4.4.8 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 19* for details).

Table 19. 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. Returned when a join or a send is being processed.
2. Returned when setting a wrong or malformed value.

4.4.9 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 20* for details).

Table 20. 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. Returned when a join or a send is being processed.
2. Returned when setting a wrong or malformed value.

4.4.10 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 21* for details).

Table 21. 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 (0-7) corresponding to DR_X	OK
AT+RX2DR=?	-	[0,1,2,3,4,5,6,7]	OK/ AT_BUSY_ERROR(1)
AT+RX2DR=<input>	[0,1,2,3,4,5,6,7]	-	OK/ AT_PARAM_ERROR(1) AT_BUSY_ERROR(2)
Example AT+RX2DR=?	-	6	OK
Example AT+RX2DR=	5	-	OK

1. Returned when a join or a send is being processed.
2. Returned when setting a wrong or malformed value.

4.4.11 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 22* for details).

Table 22. 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. Returned when a join or a send is being processed.
2. Returned when setting a wrong or malformed value.

4.4.12 AT+TXP: Transmit power

This command allows the user to access the transmit power (refer to *Table 23* for details).

Table 23. Transmit power command

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

1. Returned when setting a wrong or malformed value.

4.5 Class B mode

This section provides a set of commands for Class B mode management.

4.5.1 AT+PGSLOT

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

Table 24. 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 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. Returned when setting a wrong or malformed value.

4.5.2 AT+LTIME

This command[a] allows the user to access the local time in a UTC format.

Table 25. 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

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

4.6 Asynchronous events

Table 26 lists the possible events sent from the B-L072Z-LRWAN1 Discovery board to host serial port actively. Host parser may need to handle this event at any time.

Table 26. Asynchronous events

Event		Parameter	Description
Beacon acquisition process	+BC:	<status>	FAILED: 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 are opened periodically. The modem is now in Class B mode.
Class B downlink	+EVT:	<status>	UNICAST: lets host know that 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>:<size>: <payload>. A payload of size has been received on a Rx window on PortNumber.
Join request success	+EVT:	<status>	JOINED: end-device has joined the network. Join request successful.
Join request failed	+EVT:	<status>	JOIN FAILED: end-device has not joined the network. Join request failed.
Tx confirmed msg	+EVT:	<status>	SEND_CONFIRMED: Tx frame in confirmed mode has been acknowledged by the network.

4.7 Information

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

4.7.1 AT+BAT: Battery level

This command allows the user to access the battery level of the end-device (refer to *Table 27* for details).

Table 27. Battery level command

Command	Input parameter	Return value(1)	Return code
AT+BAT?	-	AT+BAT: get the battery level	OK
AT+BAT=?	-	[1 ... 254]	OK
Example AT+BAT=?	-	254	OK

1. Battery level ranges from 1 to 254, 254 meaning fully charged.

4.7.2 AT+VER: Version of the firmware

This command allows the user to access the version of the B-L072Z-LRWAN1 Discovery board firmware (refer to *Table 28* for details).

Table 28. Version of the firmware command

Command	Input parameter	Return value	Return code
AT+VER?	-	AT+VER: get the version of the AT_Slave FW	OK
AT+VER=?	-	– APP_VERSION= VX.Y.Z – MAC_VERSION= VA.B.C_rcD	OK
Example AT+VER=?	-	– APP_VERSION= V1.0.0 – MAC_VERSION= V1.0.0_rcA	OK

4.8 RF tests

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

4.8.1 AT+TRSSI: Start radio frequency RSSI tone test

This command allows the user to start the RF RSSI tone test (refer to *Table 29* for details).

Table 29. Start radio frequency RSSI tone command

Command	Input parameter	Return value	Return code
AT+TRSSI?	-	AT+TRSSI: start RF RSSI tone test	OK
AT+TRSSI	Void	Void	OK AT_BUSY_ERROR
Example AT+TRSSI	-	-	OK

4.8.2 AT+TTONE: Start radio frequency tone test

This command allows the user to start the RF tone test (refer to *Table 30* for details).

Table 30. Start radio frequency tone test command

Command	Input parameter	Return value	Return code
AT+TTONE?	-	AT+TTONE: start RF tone test	OK
AT+TTONEI	Void	Void	OK AT_BUSY_ERROR
Example AT+TTONE	-	-	OK

4.8.3 AT+TTX: Start RF Tx test

This command allows the user to start the RF Tx test and to choose as input the number of packets to be sent (refer to *Table 31* for details). Packet type/size is fixed (by default equal to 12-byte long).

Table 31. Start RF Tx test command(1)

Command	Input parameter	Return value	Return code
AT+TTX?	-	AT+TTX: set number of packets sent with RF LoRa® test	OK
AT+TTX = <Param>	0 < integer < 64	-	OK AT_PARAM_ERROR
Example AT+TTX 10	-	-	OK

1. The RF Tx test can be done on different modulations (FSK, LoRa®, BPSK). The setup modulation is done thanks to the AT+TCONF command. AT+TTX command is used in place of the AT+TTLRA command.

4.8.4 AT+TRX: Start RF Rx test

This command allows the user to start the RF Rx test and to choose as input the number of packets to be received (refer to *Table 32* for details). Packet error rate (PER) is displayed as return value.

Table 32. Start RF Rx test command(1)

Command	Input parameter	Return value	Return code
AT+TRX?	-	AT+TRX: set number of packets received with RF LoRa® test	OK
AT+TRX = <Param>	0 < integer < 64	Void	OK AT_PARAM_ERROR
Example AT+TRX 10	10	0 < PER < 100	OK

1. The RF Rx test can be done on different modulations (FSK, LoRa®, BPSK). The setup modulation is done thanks to the AT+TCONF command. AT+TRX command is used in place of the AT+TTLRA command.

4.8.5 AT+TCONF: Configure LoRa RF test

This command allows the user to access the LoRa configuration test (refer to *Table 33* for details).

Table 33. Configure LoRa RF test command

Command	Input parameter	Return value	Return code
AT+TCONF?	-	AT+TCONF: configure LoRa® RF test	OK
AT+TCONF=?	-	-	OK AT_ERROR
AT+TCONF=<param>	<freq>:<power>:<bandwith>:<SF>:<CR>:<LNA>:<PA>:<mod>:<payloadlen>:<fskDev>	-	OK AT_PARAM_ERROR
Example AT+TCONF?	-	Freq = 868000 Hz Power = 14 dbm Bandwidth = [0,1,2,3,4,5,6] SF = 12 CR = 4 / 8 LNA State = 0 PA boost state = 0 Modulation = [0,1,2] payloadlen = [0:256]	OK
Example AT+TCONF=	868000000:14:4:12:4/5:0:0:1:16:25000:2:3	-	OK
Example AT+TCONF=	868000000:14:**8**:12:4/5:0:0:1:16:25000:2:3	-	AT_PARAM_ERROR (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 = {7.8125,15,625,31.25,62.5,125,250,500};
- SF = {7, 8, 9, 10, 11, 12};
- CR = {4/5, 4/6, 4/7, 4/8}.

4.8.6 AT+TOFF: Stop ongoing radio frequency test

This command allows the user to stop the ongoing RF test (refer to *Table 34* for details).

Table 34. 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

4.8.7 AT+CERTIF: Set the module in LoRaWAN certification mode

This command allows the user to set the module in LoRaWAN certification mode with the choice of join mode (refer to *Table 35* for details).

Table 35. 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=<param>	0 or 1(1)	Return an asynchronous event +EVT:JOINED	OK AT_PARAM_ERROR(2)
Example AT+CERTIF=	1	+EVT:JOINED	-
	0	+EVT:JOINED	-

1. 0: indicates the join network in ABP mode
   1: indicates the join network in OTAA mode.
2. Returned when setting a wrong or malformed value.

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

4.8.8 AT+TTH: RF Tx hopping test

Table 36. RF Tx hopping

Command	Input parameter	Return value	Return code
AT+TTH?	-	AT+TTH start RF Tx hopping test from Fstart to Fstop with Fdelta step	OK
AT+TTH=<param>	Fstart, Fstop, Fdelta, PacketNb	-	OK AT_PARAM_ERROR(1)
Example(2) AT+TTH=	868000000, 868500000, 100000, 6		OK

1. Returned when setting a wrong or malformed value.
2. Set Tx hopping test from 868 MHz (Fstart) to 868.5 MHz (Fstop) with six steps (PacketNb) of 100 kHz (Fdelta).

5 Examples

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

5.1 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=1 /* join request in OTAA mode*/
<CR><LF>OK<CR><LF>
/* wait for few seconds to wait for join to complete */
+EVT:JOINED<CR><LF>   /*end-device has joined the network */
/* now the network is joined, data can be sent */
# AT+SEND=50:0:0123 /* Send data in unconfirm mode  on port 50 */
<CR><LF>OK<CR><LF>
# AT+SEND=60:1:0123  /* Send data in confirm mode (2 bytes: 0x01 and 0x23)
on port 60 */
<CR><LF>OK<CR><LF>
+EVT:SEND_CONFIRMED /* Tx frame in confirmed mode has been acknowledged by
the network*/
/* 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 */
```

5.2 Confirmation

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

```
# AT+SEND=50:0:0123 /* Send data in unconfirm mode  on port 50 */
<CR><LF>OK<CR><LF>
# AT+SEND=60:1:0123  /* Send data in confirm mode (2 bytes: 0x01 and 0x23)
on port 60 */
<CR><LF>OK<CR><LF>
+EVT:SEND_CONFIRMED /* Tx frame in confirmed mode has been acknowledged by
the network*/
```

5.3 Receiving data

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

```
/* Network is joined, data can be sent */
# AT+SEND=50:0:0123 /* Send data in unconfirm mode  on port 50 */
<CR><LF>OK<CR><LF>
```

```
+EVT:50:6:ABCDEF<CR><LF>   /*receive down frame*/
+EVT:RX_1, DR 0, RSSI -49, SNR 5<CR><LF>   /* receive down frame
parameters*/
```

5.4 Class B enable request

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

```
# AT+JOIN=1 /* join request in OTAA mode*/
<CR><LF>OK<CR><LF>
/* wait for few seconds to wait for join to complete */
+EVT:JOINED<CR><LF>   /*end-device has joined the network */
/* 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=50:0:0123 /* 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=50:0:0123 /* 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>
```

6 Embedded software description

This section gives an overview of the firmware architecture of the B-L072Z-LRWAN1 Discovery board. To see the complete description of the software implementation in I-CUBE-LRWAN refer to document *2*.

6.1 Firmware overview

This overview refers to the software Expansion Package for STM32Cube (see document *2*) and not to the specific implementation of the LoRa® technology. For more details on how to proceed with the specific LoRa® technology case refer to document *2*).

The AT command processing is found in the source files listed below:

- command.c: contains the definition and handlers of all the commands
- at.c: contains AT driver functions (basic action to provide what to whom)

6.2 LPUART

The AT commands are sent through an UART carrier. In order to optimize the low power, the LPUART of the B-L072Z-LRWAN1 Discovery board is used.

The AT_slave module executes two different tasks:

- LoRa® tasks: it manages the received windows, and it sends data
- Receives commands from the master that schedules LoRa® tasks, and then sends back the requested value and the status of the command.

As the AT_slave is already executing the two tasks described above, the MCU is idle most of the time. The MCU remains waiting either for a command from the master or for a LoRa® task schedule.

So it is important to be in Stop mode in order to optimize the low-level power of the MCU. As commands are received through the UART, the low power UART (LPUART) is used, hence the communication transfer rate is limited to 9600 bauds.

The LPUART is initialized to be enabled in Stop mode, and the wakeup from Stop mode is performed on a Start bit detection. The LPUART handler (vcom_IRQHandler()) enables RXNE (RX not empty) IT, so that when RXNE IT is raised, the character is read and stored in an internal circular buffer.

The buffer of read characters is then processed in the normal thread (not in the IT thread). A command is recognized when the new character received is <CR> or <LF>.

6.3 Compilation switches

This section lists the compilation switches provided to the user to control the compilation process.

Table 37 provides a summary of the main options for the application configuration.

Table 37. Compilation switch options

Switch option		Definition	Location
LoRa stack	LORAMAC _CLASSB_ENABLED	Compile the relevant code for Class B mode	Compiler option setting
Bands	-	Default active region is LoRaMAC_REGION_EU868	lora_app.h
Debug	DEBUGGER_ON	Enable “Led on/off”	sys_conf.h
	VERBOSE_LEVEL	Enable the trace level	sys_conf.h
Command	NO_HELP	Disable the short help on AT commands when using AT+<CMD>?	lora_command.c

Note: Even if Class B mode cannot be activated by AT+CLASS command, the code is proposed in this release. It is up to the user to adapt the current AT command interface to support a Class B compatible application.

Note: When “printf” are enabled, the resulting commands may be interlaced with debug printf().

6.3.1 Debug switches

In \Projects\B-L072Z-LRWAN1\Applications\LoRaWAN\LoRaWAN_AT_Slave\Core\Inc the user can enable the debug mode and /or the trace mode by setting #define DEBUGGER_ON.

The debug mode enables the DBG_GPIO_SET and DBG_GPIO_RST macros. This mode also enables the debugger mode even when the MCU goes in low power.

Note: To do a true low power #define DEBUGGER_ON must be off.

6.4 Footprint

The values in *Table 38* have been measured for the following configuration:

- Compiler: IAR Embedded Workbench® 8.32.4
  - Optimization: optimized for size level high
  - Debug option: off

Table 38. AT_slave footprint

Footprint	Flash memory (bytes)	RAM (bytes)	Description
AT_Slave	11759	969	LoRa finite state machine (Lora.c)
LoRa stack	56957	7391	LoRa stack
Total	68716	8360	Total memory

7 Revision history

Table 39. Document revision history

Date	Revision	Changes
10-Jan-2017	1	Initial release.
25-Aug-2017	2	Updated document title and *Section 3: Overview*. Added *Section 4.8: RF tests* and its subsections. Updated *Figure 1: Terminal emulation mode*. Updated *Table 1: List of acronyms*. Minor text edits across the whole document.
14-Dec-2017	3	Updated *Section 2: Reference documents* and *Section 3: Overview*.
11-Jul-2018	4	Updated *Section 2: Reference documents*, *Section 6.3: Compilation switches*, *Section 6.3.1: Debug switches* and *Section 6.4: Footprint*. Minor text edits across the whole document. Updated *Table 13: LoRa® class command* and its footnote *1*, *Table 37: Compilation switch options* and *Table 38: AT_slave footprint*. Removed former *Section 4.4.5: AT+FCD: frame counter downlink,* former *Section 4.4.6: AT+FCU: frame counter uplink*, former *Section 6.2: Low layer driver* and former *Note* in *Section 4.3.4: AT+NJM: LoRa® network join mode*.
17-Dec-2018	5	Updated *Section 4: AT commands*, *Section 4.2.3: AT+APPSKEY: Application session key* and *Section 4.2.7: AT+NWKSKEY: Network session key*. Added *Section 4.5: Class B mode*, *Section 4.6: Asynchronous events* with their subsections and tables, and *Section 5.4: Class B enable request*. Minor text edits across the whole document. Updated *Figure 1: Terminal emulation mode*. Updated *Table 8: Application session key command*, *Table 12: Network session key command*, *Table 13: LoRa® class command* and its footnotes.
08-Jul-2019	6	Updated *Section 4.8.3: AT+TTX: Start RF Tx test* and *Section 4.8.3: AT+TTX: Start RF Tx test*. Updated *Table 31: Start RF Tx test command* and *Table 32: Start RF Rx test command*.

Table 39. Document revision history

Date	Revision	Changes
15-Feb-2021	7	Updated *Section 4: AT commands*, *Section 4.3.2: AT+SEND: Send binary data*, *Section 4.8.3: AT+TTX: Start RF Tx test*, *Section 4.8.4: AT+TRX: Start RF Rx test*, *Section 4.8.5: AT+TCONF: Configure LoRa RF test*, *Section 4.8.7: AT+CERTIF: Set the module in LoRaWAN certification mode*, *Section 5.1: Join and send*, *Section 5.2: Confirmation*, *Section 5.3: Receiving data*, *Section 5.4: Class B enable request*, *Section 6.3.1: Debug switches* and *Section 6.4: Footprint*. Added *Section 4.1.4: AT+VL: Verbose level*, *Section 4.4.5: AT+BAND: Active region* and *Section 4.8.8: AT+TTH: RF Tx hopping test*. Removed former *Section 4.2.2: AT+APPKEY: Application key*, *Section 4.2.3: AT+APPSKEY: Application session key*, *Section 4.2.7: AT+NWKSKEY: Network session key*, *Section 4.3.1: AT+CFM: confirm mode*, *Section 4.3.2: AT+CFS: confirm status*, *Section 4.3.4: AT+NJM: LoRa® network join mode*, *Section 4.3.5: AT+NJS: LoRa® network join status*, *Section 4.3.6: AT+RECV: last received text data*, *Section 4.3.7: AT+RECVB: last received binary data, Section 4.3.8: AT+SEND: send text data, Section 4.4.7: AT+PNM: Public network mode*, *Section 4.5.2: AT+BFREQ*, *Section 4.5.3: AT+BTIME*, *Section 4.5.4: AT+BGW*, *Section 4.7.2: AT+RSSI: RSSI on reception* and *Section 4.7.3: AT+SNR: Signal to noise ratio*. Updated *Table 6: Application identifier command*, *Table 7: Device address command*, *Table 9: Network ID command*, *Table 10: Join LoRa network command*, *Table 11: Send data command* and its footnotes, *Table 13: LoRa® class command*, *Table 14: Duty cycle settings command*, *Table 26: Asynchronous events*, *Table 28: Version of the firmware command*, *Table 31: Start RF Tx test command*, *Table 32: Start RF Rx test command*, *Table 33: Configure LoRa RF test command*, *Table 35: Set the module in LoRaWAN® Certification mode command*, *Table 37: Compilation switch options* and *Table 38: AT_slave footprint*. Minor text edits across the whole document.