LoRa Alliance®

LoRaWAN® Certification Protocol Specification TS009-1.2.0

NOTICE OF USE AND DISCLOSURE

Elements of LoRa Alliance specifications may be subject to third-party intellectual property rights, including without limitation, patent, copyright or trademark rights (such a third party may or may not be a member of the LoRa Alliance). The Alliance is not responsible and shall not be held responsible in any manner for identifying or failing to identify any or all such third-party intellectual property rights.

This document and the information contained herein are provided on an "AS IS" basis and THE ALLIANCE DISCLAIMS ALL WARRANTIES EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO (A) ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OF THIRD PARTIES (INCLUDING WITHOUT LIMITATION ANY INTELLECTUAL PROPERTY RIGHTS INCLUDING PATENT, COPYRIGHT OR TRADEMARK RIGHTS) OR (B) ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NONINFRINGEMENT. IN NO EVENT WILL THE ALLIANCE BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OF DATA, INTERRUPTION OF BUSINESS, OR FOR ANY OTHER DIRECT, INDIRECT, SPECIAL OR EXEMPLARY, INCIDENTIAL, PUNITIVE OR CONSEQUENTIAL DAMAGES OF ANY KIND, IN CONTRACT OR IN TORT, IN CONNECTION WITH THIS DOCUMENT OR THE INFORMATION CONTAINED HEREIN, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The above notice and this paragraph must be included on all copies of this document.

LoRa Alliance®
5177 Brandin Court
Fremont, CA 94538
United States

Note: LoRa Alliance® and LoRaWAN® are trademarks of the LoRa Alliance, used by permission. All company, brand and product names may be trademarks that are the sole property of their respective owners.

LoRaWAN® Certification Protocol Specification TS009-1.2.0

Authored by the Certification Working Group of the LoRa Alliance Technical Committee

Technical Committee Chair and Vice-Chair:
A.YEGIN (Actility), O.SELLER (Semtech)

Working Group Chair:
D.SMITH (GetWireless LLC)

Editor:
M.LUIS (Semtech), N. BHASKARAN (LoRa Alliance)

Contributors:
H. Al Haj Hassan (Acklio), D.HUNT (LoRa Alliance), M.LUIS (Semtech), C.NFOR (7 Layers), J.PETRUNO (Senet)

Version: 1.2.0
Date: May 2024
Status: Released

Contents

Tables

Figures

1 Conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The tables in this document are normative. The figures and notes in this document are informative.

Referenced document titles are written as *LoRaWAN L2 Specification [TS001]* and referenced section titles within this document are written as “Functional Test Description for LoRaWAN Certification”.

Commands are written ***PackageVersionReq***, bits and bit fields are written `PackageIdentifier`, constants are written RECEIVE_DELAY1, variables are written *N*.

In this document:

- The octet order for all multi-octet fields SHALL be little endian.
- EUI are 8-octet fields and SHALL be transmitted as little endian.
- By default, `RFU` bits are Reserved for Future Use and SHALL be set to 0 by the transmitter of the packet and SHALL be silently ignored by the receiver.

2 Introduction

All messages described in this document are transported as application layer messages on a dedicated port. As such, all unicast messages (uplink or downlink) are encrypted by the LoRaWAN Media Access Control (MAC) layer using the end-device's `AppSKey`.

This protocol specification allows the LoRaWAN Certification Test Tool (LCTT) to fully validate compliance of an end-device to the *LoRaWAN L2 Specification* [TS001] and the *LoRaWAN Regional Parameters Specification* [RP002] for Class A, B, and C end-devices and end-devices that support the Static Context Header Compression (SCHC) protocol.

In order for an end-device to be designated "LoRaWAN Certified^{CM}" it SHALL implement this application layer specification and have Port Field (`FPort)` 224 enabled for the duration of the certification tests. This `FPort` SHALL be disabled on any device in production. Otherwise, it may intentionally or accidentally be activated to harm the device itself or the networks in its radio coverage.

The end-device to be certified SHALL be sent to an Authorized Test House (ATH) with `FPort` enabled for the certification test and will then be returned to the end-device manufacturer for `FPort` to be disabled.

The following flow chart describes the process:

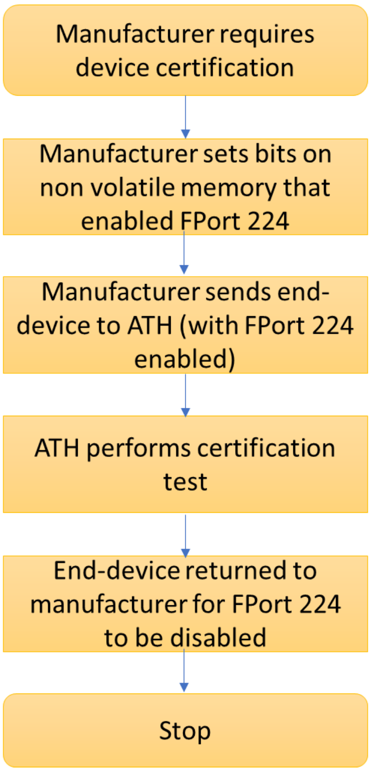
Manufacturer requires device certification
Manufacturer sets bits on non volatile memory that enabled FPort 224
Manufacturer sends end-device to ATH (with FPort 224 enabled)
ATH performs certification test
End-device returned to manufacturer for FPort 224 to be disabled
Stop

Figure 1: Certification process flow chart

When an Over-the-Air Activation (OTAA) device has connected to the Test Control Layer (TCL, refer to Chapter 3, "Functional Test Description for LoRaWAN Certification") after the Join-Request and Join-Accept frame exchange, the device SHOULD send an uplink message as soon as possible.

When an Activation By Personalization (ABP) end-device has connected to the TCL, there SHALL be a way to trigger an uplink message.

2.1 Scope of LoRaWAN Certification

The scope of this specification is limited to validating compliant implementation of the LoRaWAN protocol.

Intended or otherwise, the inevitable variability of performance and quality of the radio implementation among end-devices is too high to allow normalized, practical evaluation. Radio Frequency (RF) performance measurement, whether radiated or conducted, is therefore considered out of scope of the tests described herein. Subsequently, all methodologies describing RF provisioning or adjustments, (e.g., device attenuation, etc.), are intentionally absent. It is the shared responsibility of the Authorized Test Houses and those parties seeking certification to best accommodate the submitted end-device RF characteristics for LoRaWAN protocol certification. This is intended to optimize reliability and consistency of bi-directional communication of the test harness.

2.2 LoRaWAN Certification Process

For details about the LoRaWAN certification process, refer to:
https://lora-alliance.org/lorawan-certification

3 Functional Test Description for LoRaWAN Certification

The list of tests specified on regional certification documents reflect functional requirements of an end-device as defined by the targeted LoRaWAN specification. The tests are conducted in the LCTT, which is comprised of:

- Test Control Layer [TCL]
- LoRaWAN Network Server [NS]
- LoRaWAN gateway [GW]
- Device Under Test [DUT]

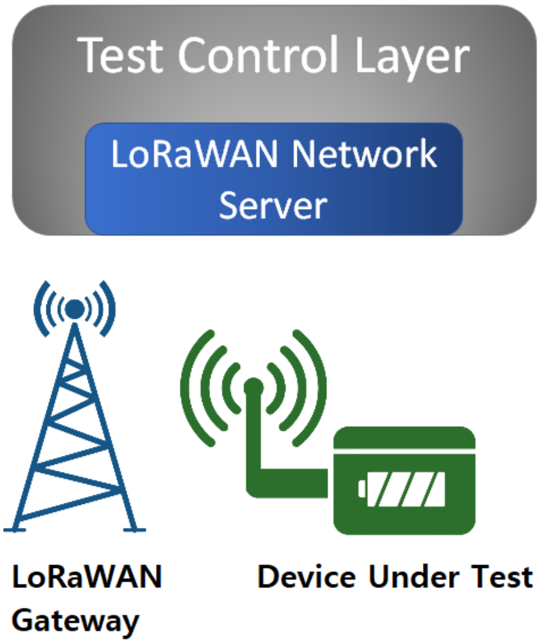

Figure 2: LCTT architecture

The TCL is a framework of automated scripts and tools that manipulate the NS to facilitate the tests. Specifically, the TCL drives events in the LCTT, controlling application and network-control content of downlinks. It also decrypts, inspects, and validates content of uplinks sent by the DUT. This allows test coverage to include:

- Cryptography
- Timing of the DUT receive windows
- Frequency channel usage and data rate adaptation
- Maximum payload length handling

For brevity, this document makes procedural reference to only the TCL, NS, GW, and DUT.

The GW and DUT are collocated in an RF-isolated environment, provisioned as necessary for reliable bi-directional communication. It is nonetheless expected that both the DUT and GW will not receive every frame intended for reception. The TCL SHOULD make a reasonable effort to accommodate this inevitability.

The DUT is required to implement this *LoRaWAN Certification Protocol Specification* in order to provide a way to control the DUT. The RF-isolated environment mentioned above SHOULD mitigate any potential interference.

Testing occurs to certify the DUT for each supported activation method, be it OTAA, ABP, or both.

Between each test section described in specific regional documents, the TCL will return the DUT to a known state.

The TCL SHALL verify the following throughout the course of this certification test suite:

- The DUT uplink frame size SHALL respect the maximum allowed uplink frame size for the data rate currently in use.
- The size of the DUT’s uplink frames SHALL match the expected content to ensure no extraneous and unnecessary content is present.

4 End-Device Certification Description

4.1 Overview

Every LoRaWAN end-device SHALL implement this applicative protocol specification in its application layer. This allows the LCTT to fully validate compliance of the end-device's LoRaWAN MAC layer implementation.

The `FPort` value of 224 is dedicated to the LoRaWAN MAC layer certification protocol.

The DUT SHALL return to its normal application behavior via a command disabling certification `FPort` 224 processing. Additionally, the DUT SHALL be reset with a dedicated command, returning to a join state from which it can then establish a new session.

All defined test commands SHALL be sent by the TCL to the DUT using `FPort` 224 when the end-device is in its normal operation. The DUT SHOULD execute the given command as soon as possible.

4.1.1 Over-The-Air Activated DUT

If the DUT uses OTAA, when the DUT is first powered up, it SHOULD join the network by issuing a Join-Request. The TCL will respond with a Join-Accept. As a best practice of the certification process, upon receipt of the Join-Accept frame, the DUT SHOULD send a (possibly empty) uplink frame. The TCL is then able to send certification test commands on `FPort` 224, if enabled.

4.1.2 Activated by Personalization DUT

A personalized DUT is one that comes with session keys pre-programmed. The personalization information SHALL be supplied to the ATH. When the DUT is first powered on, it SHOULD send an uplink frame. The TCL is then able to send certification test commands on `FPort` 224, if enabled.

4.2 Certification Commands

4.2.1 Downlink Counter

The DUT creates a 16-bit unsigned counter (called `RxAppCnt`) that is incremented each time the DUT receives an applicative downlink frame (`FPort > 0`). An empty downlink frame with a `FCtrl ACK` bit set SHALL be considered as an applicative downlink.

The `RxAppCnt` counter SHALL be initialized to 0 when the DUT is reset or each time the TCL sends a ***DownlinkCntRstReq*** command on `FPort` 224.

4.2.2 Beacon Counter

The DUT creates a 16-bit unsigned counter (called `BeaconCnt`) that is incremented each time the DUT receives a Class B beacon.

The `BeaconCnt` counter SHALL be initialized to 0 when the DUT is reset or each time the TCL sends a ***BeaconCntRstReq*** command on `FPort` 224.

4.2.3 Commands handling

The TCL MAY send the certification application commands at any given time. The DUT SHOULD execute a command as soon as possible.

5 Certification Protocol Commands

The `PackageIdentifier` of the certification protocol transport package is 5. The `PackageVersion` of this package is version 1.

> **Note:** This version of the package is not compatible with any previous version of the certification protocol used with LoRaWAN Specification v1.0.2 and earlier releases.

This package supports all the commands necessary to execute the LoRaWAN end-device certification tests. The `FPort` value is 224 refer to [TS001]. This port SHALL NOT be used for any other purposes.

All certification protocol command messages are exchanged on this port using an application payload and are encrypted using the end-device's `AppSKey`. All certification protocol command messages use the same format:

Certification protocol command	Certification protocol command payload

Table 1: Format of certification protocol command messages

A frame SHALL NOT carry more than one certification protocol command message. The length of the certification protocol command payload can be determined unambiguously as a function of the command.

The following table summarizes the list of certification protocol command messages:

CID	Certification Protocol Command Name	Transmitted by		Short Description
		End-device	Server	
0x00	***PackageVersionReq***		x	The TCL requests the package version implemented by the end-device
0x00	***PackageVersionAns***	x		The DUT SHALL convey the answer to the ***PackageVersionReq*** request
0x01	***DutResetReq***		x	The DUT SHALL reset the Microcontroller Unit (MCU)
0x02	***DutJoinReq***		x	The DUT SHALL reset the LoRaWAN MAC layer and start issuing Join-Request messages
0x03	***SwitchClassReq***		x	The DUT SHALL request for change of its class of operation to A, B, or C
0x04	***AdrBitChangeReq***		x	The DUT SHALL activate/deactivate Adaptive Data Rate (ADR)
0x05	***RegionalDutyCycleCtrlReq***		x	The DUT SHALL activate/deactivate the regional band duty-cycle enforcement
0x06	***TxPeriodicityChangeReq***		x	The DUT SHALL change its uplink periodicity to the provided value
0x07	***TxFramesCtrlReq***		x	The DUT SHALL send all subsequent uplinks of the specified type
0x08	***EchoPayloadReq***		x	The TCL requests the DUT to echo the provided payload, where each byte is incremented by 1
0x08	***EchoPayloadAns***	x		The DUT SHALL convey the answer to the ***EchoPayloadReq*** request
0x09	***RxAppCntReq***		x	The TCL requests the DUT to provide the current applicative RxAppCnt value
0x09	***RxAppCntAns***	x		The DUT SHALL convey the answer to the ***RxAppCntReq*** request
0x0A	***RxAppCntResetReq***		x	The DUT SHALL reset the applicative RxAppCnt value to 0
0x0B-0x1F	***RFU***			
0x20	***LinkCheckReq***		x	The DUT SHALL send a ***LinkCheckReq*** MAC command to the TCL
0x21	***DeviceTimeReq***		x	The DUT SHALL send a ***DeviceTimeReq*** MAC command to the TCL
0x22	***PingSlotInfoReq***		x	The DUT SHALL send a ***PingSlotInfoReq*** MAC command to the TCL; only required for Class B DUTs
0x23-0x3F	***RFU***			
0x40	***BeaconRxStatusIndCtrl***		x	The DUT SHALL activate/deactivate the autonomous ***BeaconRxStatusInd*** transmission
0x41	***BeaconRxStatusInd***	x		The DUT autonomously sends this indication every time a beacon is supposed to have been received
0x42	***BeaconCntReq***		x	The TCL requests the DUT to provide the current BeaconCnt value
0x43	***BeaconCntAns***	x		The DUT SHALL convey the answer to the ***BeaconCntReq*** request

CID	Certification Protocol Command Name	Transmitted by		Short Description
		End-device	Server	
0x44	***BeaconCntRstReq***		x	The DUT SHALL reset the BeaconCnt value to 0
0x44-0x4F	***RFU***			
0x50	***SCHCMsgSendReq***		x	The TCL requests the DUT to send an uplink SCHC message
0x50	***SCHCMsgSendAns***	x		The DUT SHALL convey the answer to the ***SCHCMsgSendReq*** request
0x51	***SCHCACKInd***	x		The DUT SHALL send the TCL an acknowledgement that it has received an SCHC downlink session
0x52	***FragSessionCntReq***		x	The TCL requests the DUT to provide the SessionCntPrev value for the FragIndex requested
0x52	***FragSessionCntAns***	x		The DUT SHALL convey the answer to the ***FragSessionCntReq*** request
0x53 – 0x7C	***RFU***			
0x7D	***TxCwReq***		x	The DUT SHALL set the radio in continuous wave transmission mode
0x7E	***DutFPort224DisableReq***		x	The DUT SHALL disable the processing of data received on FPort 224
0x7F	***DutVersionsReq***		x	The TCL requests the DUT to send its firmware version, LoRaWAN version, and regional parameters version
0x7F	***DutVersionsAns***	x		The DUT SHALL convey the answer to the ***DutVersionsReq*** request
0x80 – 0xFF	***Proprietary***	x	x	Reserved for proprietary end-device command extensions

Table 2: Summary of certification protocol command messages

5.1 Package Version Commands (*PackageVersionReq*, *PackageVersionAns*)

The ***PackageVersionReq*** command has no payload.

The end-device SHALL answer this command with a ***PackageVersionAns*** command with the following payload:

Size (octets)	1	1
Field	`PackageIdentifier`	`PackageVersion`

Table 3: PackageVersionAns

`PackageIdentifier` uniquely identifies the package.

`PackageVersion` corresponds to the version of the package specification implemented by the end-device.

5.2 DUT Reset Command (*DutResetReq)*

The ***DutResetReq*** command has no payload.

This command instructs the DUT to execute/simulate a full DUT MCU reset.

This command allows the verification of the session context storage.

5.3 DUT JoinReq Command (*DutJoinReq)*

The ***DutJoinReq*** command has no payload.

This command instructs the DUT to reset the LoRaWAN MAC layer and start issuing Join-Request frames. The LoRaWAN MAC layer SHALL be reinitialized, such that all RF parameters are restored to default settings and the end-device SHALL then attempt to join the network as part of normal operation.

This command allows testing the various Join-Accept test scenarios.

5.4 LoRaWAN Class Selection Command (*SwitchClassReq*)

The ***SwitchClassReq*** command payload is used to instruct the DUT to switch to the class provided in the command as soon as possible.

Size (octets)	1
Field	`Class`

Table 4: SwitchClassReq

The `Class` field tells the DUT to switch to that class of operation, 0 for Class A, 1 for Class B, and 2 for Class C.

> **Note:** If not already performed, the DUT must execute the following steps prior to switching to Class B:
>
> 1. Optionally, execute the ***DeviceTimeReq*** MAC command (speeds up beacon acquisition).
> 2. Start beacon acquisition.
> 3. Execute the ***PingSlotInfoReq*** MAC command once the beacon has been acquired.

5.5 ADR Control Command (*AdrBitChangeReq*)

The ***AdrBitChangeReq*** command requests the DUT to activate/deactivate the ADR feature.

Size (octets)	1
Field	`ADR`

Table 5: AdrBitChangeReq

The `ADR` field encodes the ADR state. A value of 1 means ADR ON and a value of 0 means ADR OFF.

The TCL can verify the correct operation by checking the ADR bit state of every uplink frame.

5.6 Regional Duty-Cycle Enforcement Command (*RegionalDutyCycleCtrlReq*)

The ***RegionalDutyCycleCtrlReq*** command requests the DUT to activate/deactivate the regional duty-cycle enforcement for regions that require it.

Size (octets)	1
Field	`DutyCycle`

Table 6: RegionalDutyCycleCtrlReq

The `DutyCycle` field encodes the regional duty-cycle enforcement state. A value of 1 means that the regional duty-cycle enforcement is ON and a value of 0 means that the regional duty-cycle enforcement is OFF.

The TCL can verify the correct operation by checking that the uplinks are no longer delayed.

5.7 Application Transmission Periodicity Control Command (*TxPeriodicityChangeReq*)

The ***TxPeriodicityChangeReq*** command payload is used to change the periodicity of uplink frames.

Size (octets)	1
Field	`Periodicity`

Table 7: TxPeriodicityChangeReq

The `Periodicity` field encodes time values in seconds, which will allow all test scenarios to be run.

`Periodicity`	Value
0	Default DUT behavior
1	5
2	10
3	20
4	30
5	40
6	50
7	60
8	120
9	240
10	480
11-255	RFU

Table 8: Periodicity field encoding

The TCL can verify the correct operation by checking that the time between successive physical uplink packets has been changed as requested.

5.8 Uplink Frames Control Command (*TxFramesCtrlReq)*

The ***TxFramesCtrlReq*** command is used to change the frame type to be used by subsequent uplink frames. This command MAY also convey *N* extra octets.

Size (octets)	1	N-1
Field	FrameType	0..N

Table 9: TxFramesCtrlReq

The FrameType field encodes the frame type to be used by the DUT on all subsequent uplink frames.

FrameType	Description	Remarks
0	No change	Allows the TCL to perform a no-operation downlink
1	Unconfirmed	L2 Unconfirmed FType = 2 frames
2	Confirmed	L2 Confirmed FType = 4 frames
3-255	RFU	

Table 10: FrameType values description

The TCL can verify the correct operation by checking that the FType of the MHDR field of subsequent uplink frames has changed as requested.

5.9 Echo Frame Request Commands (*EchoIncPayloadReq, EchoIncPayloadAns*)

The ***EchoIncPayloadReq*** command payload contains the *N* bytes to be echoed, plus one. The *N* value is arbitrary. In case the *N* value is bigger than the application payload buffer, the echoed packet SHALL be clipped to the maximum payload buffer size.

Size (octets)	1	1	…	1
Fields	Octet0	Octet1	…	Octet N-1

Table 11: EchoIncPayloadReq

EchoIncPayloadReq instructs the DUT to respond with a subsequent uplink whose payload content is the downlink's data incremented octet-by-octet, except the first octet, which remains 0x08.

Assume the received payload length is *N*, where *N* is any value between zero and the maximum LoRaWAN region payload size allowed. The bytes composing the command payload are:

[0x08, octet_0, octet_1, …, octet_{N-1}]

EchoIncPayloadAns SHALL convey a payload whose content is as follows:

[0x08, mod(octet_0 + 0x01, 256), mod(octet_1 + 0x01, 256), …, mod(octet_{N-1} + 0x01, 256)]

Where `mod()` indicates modulo arithmetic.

For example, if the DUT receives a payload of `[8 1 5 255]` on `FPort` 224, it will respond with `[8 2 6 0]` on `FPort` 224. This echo functionality is used to validate the DUT cryptography implementation as well as its handling of the maximum payload for both uplinks and downlinks.

The ***EchoIncPayloadAns*** SHALL be clipped to the maximum uplink frame payload size allowed by the *LoRaWAN Regional Specifications*.

5.10 Applicative Rx Counter Commands (*RxAppCntReq, RxAppCntAns, RxAppCntRstReq*)

The ***RxAppCntReq*** command requests the DUT to provide the current `RxAppCnt` value. This command has no payload.

The DUT answers the ***RxAppCntReq*** command with an ***RxAppCntAns*** command.

Size (octets)	2
Field	RxAppCnt

Table 12: RxAppCntAns

The ***RxAppCntRstReq*** command requests the DUT to reset the `RxAppCnt` value to 0. This command has no payload.

The TCL can verify the correct operation of the ***RxAppCntRstReq*** command by issuing an ***RxAppCntReq*** command.

5.11 Link Check Request MAC Command (*LinkCheckReq*)

The ***LinkCheckReq*** command has no payload.

This command instructs the DUT to send a `LinkCheckReq` MAC command.

5.12 DeviceTimeReq MAC Command (*DeviceTimeReq*)

The ***DeviceTimeReq*** command has no payload.

This command instructs the DUT to send a `DeviceTimeReq` MAC command.

5.13 PingSlotInfoReq MAC Command (*PingSlotInfoReq)*

Note: This command is required for only Class B DUTs.

The ***PingSlotInfoReq*** command payload is used to convey the ping slot periodicity.

This command instructs the DUT on the ping slots periodicity to be used when the DUT decides to send a ***PingSlotInfoReq*** MAC command.

Size (octets)	1
Field	`Periodicity`

Table 13: PingSlotInfoReq

The `Periodicity` field follows the same rules as the ones provided by [TS001].

5.14 Class B Beacon Status Indication Control Command (*BeaconRxStatusIndCtrl*)

The ***BeaconRxStatusIndCtrl*** command requests the DUT to activate/deactivate the ***BeaconRxStatusInd*** transmission.

Size (octets)	1
Field	CTRL

Table 14: BeaconRxStatusIndReq

The CTRL field encodes the ***BeaconRxStatusInd*** transmission state. A value of 1 means the transmission is activated and a value of 0 means the transmission is deactivated.

The TCL can verify the correct operation by checking ***BeaconRxStatusInd*** transmission.

5.15 Class B Beacon Status Indication Command (*BeaconRxStatusInd*)

The ***BeaconRxStatusInd*** command is autonomously sent by the DUT every time a beacon is supposed to be received.

Size (octets)	1	2	3	1	2	1	1	4	7
Fields	State	RxCnt	Frequency	DR	RSSI	SNR	Param	Time	GwSpecific

Table 15: BeaconRxStatusInd

The State field encodes the beacon acquisition state. A value of 1 means that the beacon acquisition is ON and a value of 0 means that the beacon acquisition is OFF.

The RxCnt field corresponds to the current BeaconCnt value.

The Frequency field is a 24-bit unsigned integer. The actual channel frequency in Hz is 100 x Frequency, whereby values representing frequencies below 100 MHz are reserved for future use.

The DR field is the data rate index.

The RSSI field is a signed 16-bit value expressed in dBm of the received beacon signal strength.

The SNR field is a signed 8-bit value expressed in dB of the received beacon signal to noise ratio.

The Param field follows the same rules as the ones provided by [TS001].

The Time field follows the same rules as the ones provided by [TS001].

The GwSpecific field follows the same rules as the ones provided by the [TS001].

5.16 Class B Beacon Counter Commands (*BeaconCntReq, BeaconCntAns, BeaconCntRstReq*)

The ***BeaconCntReq*** command requests the DUT to provide the current `BeaconCnt` value. This command has no payload.

The DUT answers the ***BeaconCntReq*** with a ***BeaconCntAns*** command.

Size (octets)	2
Field	`BeaconCnt`

Table 16: BeaconCntAns

The ***BeaconCntRstReq*** command requests the DUT to reset the `BeaconCnt` value to 0. This command has no payload.

The TCL can verify the correct operation of the ***BeaconCntRstReq*** command by issuing a ***BeaconCntReq*** command.

5.17 SCHC Message Commands (SCHCMsgSendReq, SCHCMsgSendAns)

The ***SCHCMsgSendReq*** command requests the DUT to send an uplink SCHC message where the `UDPAppPort` and `DataSize` are specified.. The `UDPAppPort` must be used by the DUT in building the UDP header. The `DataSize` provides the size (in bytes) of the data (all zeros) to add in the message payloads.

Size (octets)	2	2
Fields	`UDPAppPort`	`DataSize`

Table 17: SCHCMessageSendReq

The DUT answers the ***SCHCMsgSendReq*** with a ***SCHCMsgSendAns*** command. The ***SCHCMsgSendAns*** has no payload.

5.18 SCHC Acknowledge Indication Command (SCHCACKInd)

The ***SCHCACKInd*** command SHALL be sent from the DUT to the TCL after it has received the complete downlink session from the TCL. The ***SCHCACKInd*** payload contains the IPv6 headers, UDP headers, and data.

Size (bits)	4	8	20	16	8	8	64	64
Fields	`IPv6 version`	`IPv6 Diffserv`	`IPv6 Flow label`	`IPv6 length`	`IPv6 next header`	`IPv6 Hop limit`	`IPv6 DevPrefix`	`IPv6 DevIID`

Size (bits)	64	64	16	16	16	16	[8-480]
Fields	`IPv6 AppPrefix`	`IPv6 AppIID`	`UDP DevPort`	`UDP AppPort`	`UDP length`	`UDP checksum`	`Data`

Table 18: SCHCACKInd

5.19 Fragmentation Session Counter Commands (Frag*SessionCntReq, FragSessionCntAns*)

This command is related to the Fragmentation Data Block Transport (TS004) specification.

The ***FragSessionCntReq*** command requests the DUT to provide the `SessionCntPrev` value for the `FragIndex` requested.

Size (octets)	1
Field	`Param`

Table 19: FragSessionCntReq

Where:

Bits	7:2	1:0
Param bits	`RFU`	`FragIndex`

Table 20: FragSessionCntReq Param bits

The DUT answers the ***FragSessionCntReq*** command with a ***FragSessionCntAns*** command.

Size (octets)	1	2
Field	`Status`	`SessionCntPrev (conditional)`

Table 21: FragSessionCntAns

Where:

Bits	7:3	2	1:0
`Status` bits	`RFU`	`Session is not supported`	`FragIndex`

Table 22: FragSessionCntAns Status bits

The `SessionCntPrev` field SHALL be present only when `Session is not supported` bit is set to 0.

5.20 Transmit Continuous Wave Request Command (*TxCwReq*)

The ***TxCwReq*** command payload is used to define the timeout, radio frequency, and radio transmission output power.

Size (octets)	2	3	1
Fields	`Timeout`	`Frequency`	`TxPower`

Table 23: TxCwReq

The `Timeout` field is a 16-bit unsigned integer indicating the number of seconds that the DUT will spend in Continuous Wave (CW) transmission mode.

The Frequency field is a 24-bit unsigned integer. The actual channel frequency in Hz is 100 x Frequency, whereby values representing frequencies below 100 MHz are reserved for future use.

The TxPower field is an 8-bit signed integer. The value in dBm represents the output power applied to the CW mode.

> **Example**: If the DUT receives 7 bytes on port 224, [0x7D 0x08 0x00 0xD8 0xB2 0x83 0x0E], (0x7D indicating the command ID), it SHALL enter the CW mode for 8 seconds using frequency 863.1 MHz and 14 dBm.

5.21 DUT Disables FPort 224 (*DutFPort224DisableReq*)

The ***DutFPort224DisableReq*** command has no payload.

instructs the DUT to disable access to FPort 224 and executes a full reset of the DUT.

5.22 DUT Versions Command (*DutVersionsReq, DutVersionsAns*)

The ***DutVersionsReq*** command requests the DUT to provide its firmware version, [TS001] version, and [RP002] version. This command has no payload.

The DUT answers the ***DutVersionsReq*** with a ***DutVersionsAns*** command.

Size (octets)	4	4	4
Fields	FwVersion	LrwanVersion	LrwanRpVersion

Table 24: DutVersionsAns

The version (FwVersion, LrwanVersion, and LrwanRpVersion) fields SHALL be encoded as Major.Minor.Patch.Revision: 1 octet for Major, 1 octet for Minor, 1 octet for Patch and 1 octet for Revision. [Semantic Versioning 2.0.0: https://semver.org/].

> **Note:** In some regions, (e.g., US915), the answer to this request will not fit the allowed frame size at the lowest data rates. As such, the TCL may only issue the ***DutVersionReq*** when the DUT is set with a data rate allowing the ***DutVersionAns*** to fit the corresponding frame size.

6 Glossary

ABP	Activation By Personalization
ADR	Adaptive Data Rate
CW	Continuous Wave
DR	Data Rate
DUT	Device Under Test
FType	Frame Type
GW	LoRaWAN Gateway
L2	Link Layer 2
LCTT	LoRaWAN Certification Test Tool
MCU	Microcontroller Unit
NS	LoRaWAN Network Server
MAC	Media Access Control
MHDR	MAC Header
OTAA	Over-the-Air Activation
RF	Radio Frequency
SCHC	Static Content Header Compression
TCL	Test Control Layer of the Test Harness

7 Bibliography

7.1 References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997

[RFC8174] Leiba, B., “Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words”, BCP 14, RFC8174, May 1997

[TS001]: LoRaWAN L2 Specification, LoRa Alliance

[RP002]: LoRaWAN Regional Parameters, LoRa Alliance