LoRaWAN Certification Protocol Specification TS009-1.0.0

NOTICE OF USE AND DISCLOSURE

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

LoRaWAN Certification Protocol Specification TS009-1.0.0

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 Chairs:
D.HUNT (LoRa Alliance)

Editor:
M.LUIS (Semtech)

Contributors:
N. BHASKARAN (LoRa Alliance), D.HUNT(LoRa Alliance), M.LUIS (Semtech), C.NFOR (7 Layers), J.PETRUNO (Senet)

Version: 1.0.0
Date: October 2020
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 in this document are informative. The notes in this document are informative.

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 MAC layer using the end-device's `AppSKey`.

This protocol specification allows a Certification test harness to fully validate compliance of the end-device to the LoRaWAN Link Layer Specification [TS001-1.0.4] and the LoRaWAN Regional Parameters Specification [RP002-1.0.1] for Class A end devices.

In order for an end-device to be designated "LoRaWAN CertifiedCM" it SHALL implement this application layer specification and have the `FPort` 224 enabled for the duration of the Certification tests. This application SHALL be disabled on any device in production, otherwise it may be 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 the Authorized Test House (ATH) with the `FPort` enabled for the Certification test, then returned to the end-device manufacturer for the `FPort` to be disabled.

Flow chart for process

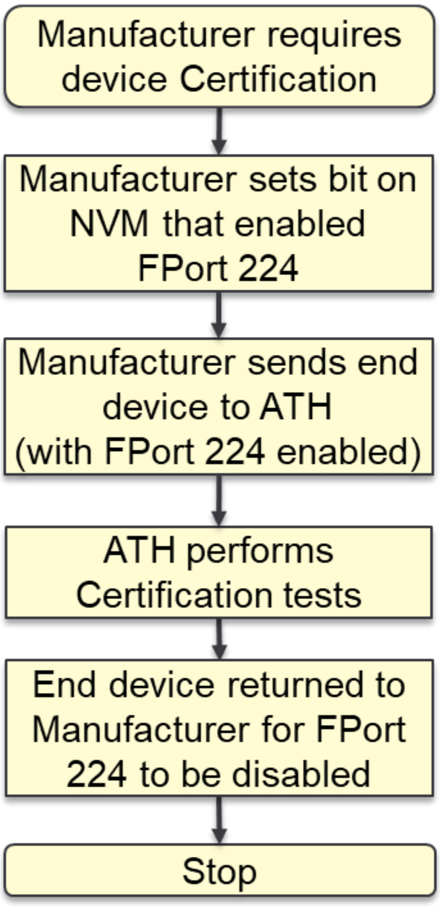


Figure 1: Certification Process Flow Chart

When an OTAA device has connected to the Test Control Layer (**TCL refer to Chapter 3**) after the Join-Request and Join-Accept frame exchange the device SHOULD send an uplink message as soon as possible.

When an ABP device has connected to the (**TCL**) there SHALL be a way to trigger an uplink message.

2.1 Scope of LoRaWAN Certification

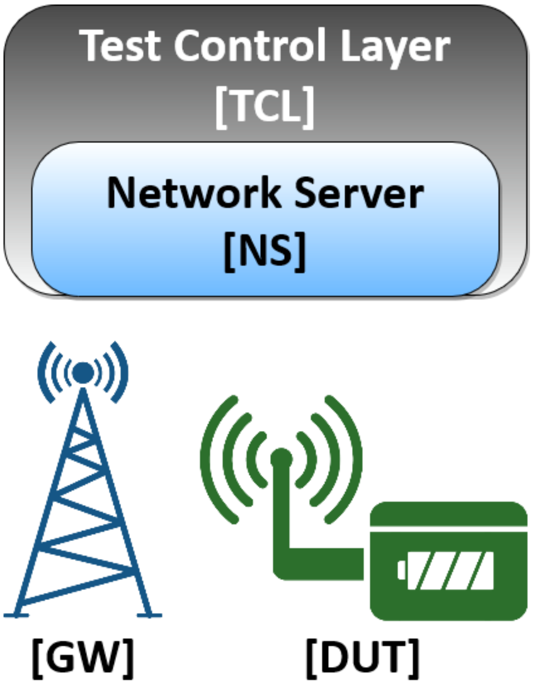
The scope of this specification is limited to validating compliant implementation of the LoRaWAN protocol.
Intended or otherwise, the inevitable variability of performance and quality of the radio implementation among end-devices is too high to allow normalized, practical evaluation. RF performance measurement, whether radiated or conducted, which are 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 of the LoRaWAN Certification Process see:
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 a test harness generally comprised of:



- A Test Control Layer [**TCL**]
- A LoRaWAN Network Server [**NS**]
- A LoRa gateway [**GW**]
- The end-device Under Test [**DUT**]

Figure 2: Test Harness Architecture

Implementation of this harness architecture is expected to vary among test houses. The Test Control Layer [**TCL**] is assumed to be a framework of automated scripts and tools that manipulates the LoRaWAN Network Server **[NS]** to facilitate the tests. Specifically, the **TCL** drives events in the harness, 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 LoRaWAN gateway [**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 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 **DUT** application. The RF-isolated environment mentioned above SHOULD mitigate any potential interference.

Testing occurs to certify the **DUT** for each supported activation method, be it over-the-air activation (**OTAA**), activation-by-personalized (**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 frames 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 test harness to fully validate compliance of the end-device's LoRaWAN MAC layer implementation.

The Port Field (`FPort`) value 224 is dedicated to the LoRaWAN MAC layer certification protocol.

The **DUT** SHALL return to its normal application behaviour by 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 **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

When the **DUT** is first powered up if it uses OTAA it SHOULD join the network by issuing a Join-request, the **TCL** will respond with Join-accept. For best practice operation of the Certification process, upon receipt of the Join-accept frame the **DUT** SHOULD then 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 Test House. 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`) which is incremented each time the **DUT** receives an applicative downlink frame (`FPort > 0`). An empty downlink frame with `FCtrl ACK` bit set SHALL be considered as and applicative downlink.

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

4.2.2 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 6. 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 LW1.0.2 and earlier releases.

This package supports all the commands necessary to execute the LoRaWAN end-device certification tests. The port value is 224 (see [TS008] FPort Assignments). This port SHALL NOT be used for any other purposes.

All certification protocol command messages are exchanged on this port using application payload and 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: Certification protocol command messages format

A frame SHALL NOT carry more than one Certification protocol command message. The length of 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: End-device	Transmitted by: server	Short Description
0x00	***PackageVersionReq***		x	Used by the **TCL** to request the package version implemented by the end-device
0x00	***PackageVersionAns***	x		Conveys the answer to ***PackageVersionReq***
0x01	***DutResetReq***		x	**DUT** SHALL reset the MCU
0x02	***DutJoinReq***		x	**DUT** SHALL start issuing Join-Request messages
0x03	***SwitchClassReq***		x	**DUT** SHALL change its Class of operation to A, B or C
0x04	***AdrBitChangeReq***		x	**DUT** SHALL activate/deactivate ADR
0x05	***RegionalDutyCycleCtrlReq***		x	**DUT** SHALL activate/deactivate the regional band duty-cycle enforcement
0x06	***TxPeriodicityChangeReq***		x	**DUT** SHALL change its uplink periodicity to the provided value
0x07	***TxFramesCtrlReq***		x	All subsequent **DUT** uplinks SHALL be of specified type
0x08	***EchoPayloadReq***		x	**TCL** requests the **DUT** to echo the provided payload where each byte is incremented by 1
0x08	***EchoPayloadAns***	x		Conveys the answer to ***EchoPayloadReq*** request
0x09	***RxAppCntReq***		x	**TCL** requests the **DUT** to provide the current applicative `RxAppCnt` value
0x09	***RxAppCntAns***	x		Conveys the answer to ***RxAppCntReq*** request
0x0A	***RxAppCntResetReq***		x	**DUT** SHALL reset the applicative `RxAppCnt` value to 0
0x0B-0x1F	***RFU***			
0x20	***LinkCheckReq***		x	**DUT** SHALL send a `LinkCheckReq` MAC command to the **TCL**
0x21	***DeviceTimeReq***		x	**DUT** SHALL send a `DeviceTimeReq` MAC command to the **TCL**
0x22	***PingSlotInfoReq***		x	**DUT** SHALL send a `PingSlotInfoReq` MAC command to the **TCL** Only required for Class B **DUT**
0x23-0x7C	***RFU***			
0x7D	***TxCwReq***		x	**DUT** SHALL set the radio in continuous wave transmission mode
0x7E	***DutFPort224DisableReq***		x	**DUT** SHALL disable the processing of data received on `FPort` 224
0x7F	***DutVersionsReq***		x	**TCL** requests the **DUT** to send its firmware version, LoRaWAN version and Regional parameters version
0x7F	***DutVersionsAns***	x		Conveys the answer to ***DutVersionsReq*** request
0x80 - 0xFF	***Proprietary***	x	x	Reserved for proprietary end-device command extensions

Table 2: Certification protocol command messages summary

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.

Field	PackageIdentifier	PackageVersion
Size (octets)	1	1

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.

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.

Instructs the **DUT** to reset the LoRaWAN MAC layer and to start issuing Join-Request frames. The LoRaWAN MAC layer SHALL reinitialize 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 convey the new target end-device Class.

Fields	Class
Size (octets)	1

Table 4: SwitchClassReq fields

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

5.5 ADR Control Command (*AdrBitChangeReq*)

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

Fields	`ADR`
Size (octets)	1

Table 5: AdrBitChangeReq fields

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 every uplink frame ADR bit state.

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

The ***RegionalDutyCycleCtrlReq*** command sent by the **TCL** requests the **DUT** to activate/deactivate the regional duty-cycle enforcement for regions requiring it.

Fields	`DutyCycle`
Size (octets)	1

Table 6: RegionalDutyCycleCtrlReq fields

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 aren't anymore delayed.

5.7 Application Transmission Periodicity Control Command (*TxPeriodicityChangeReq*)

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

Fields	`Periodicity`
Size (octets)	1

Table 7: TxPeriodicityChangeReq fields

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

`Periodicity`	Value [s]
0	Default **DUT** application 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 time between successive physical uplink packets has changed as requested.

5.8 Uplink Frames Control Command (*TxFramesCtrlReq)*

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

Fields	`FrameType`	`0..N`
Size (octets)	1	N-1

Table 9: TxFramesCtrlReq fields

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

`FrameType`	Name	Remarks
0	No change	Allows to perform a no operation downlink from **TCL**
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 `FType` of `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 then the echoed packet SHALL be clipped to the maximum payload buffer size.

Fields	`Octet1`	`Octet2`	…	`Octet N-1`
Size (octets)	1	1	…	1

Table 11: EchoIncPayloadReq fields

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

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

$$[0x08, octet_1, octet_2, …, octet_{N-1}]$$

EchoIncPayloadAns SHALL convey a payload whose content is as follows:

$$[0x08, mod(octet_1 + 0x01, 256), mod(octet_2 + 0x01, 256), …, mod(octet_{N-1} + 0x01, 256)]$$

…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 the `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 maximum Regional Parameters allowed uplink frame payload size.

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

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

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

Fields	RxAppCnt
Size (octets)	2

Table 12: RxAppCntAns fields

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

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

5.11 Link Check Request Command (*LinkCheckReq*)

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

Instructs the **DUT** to send a `LinkCheckReq` MAC command.

5.12 DeviceTimeReq MAC Command (*DeviceTimeReq*)

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

Instructs the **DUT** to send a `DeviceTimeReq` MAC command.

5.13 PingSlotInfoReq MAC Command (*PingSlotInfoReq)*

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

Instructs the **DUT** to send a `PingSlotInfoReq` MAC command.

Fields	`Periodicity`
Size (octets)	1

Table 13: PingSlotInfoReq fields

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

5.14 Transmit Continuous Wave Request Command (*TxCwReq*)

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

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

Table 14: TxCwReq fields

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

*Example: If the **DUT** receives 7 bytes on the 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.*

5.15 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**.

5.16 DUT Versions Command (*DutVersionsReq, DutVersionsAns*)

The ***DutVersionsReq*** command sent by the **TCL** requests the **DUT** to provide its firmware version, Link Layer specification [TS001] version and Regional Parameters specification [RP002] version. This command has no payload.

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

Fields	`FwVersion`	`LrwanVersion`	`LrwanRpVersion`
Size (octets)	4	4	4

Table 15: DutVersionsAns fields

The versions (`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 (i.e: US915) the answer to this request will not fit the allowed frame size at 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
GW	LoRaWAN Gateway
NS	LoRaWAN Network Server
MAC	Media Access Control
OTAA	Over-the-Air-Activation
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

[TS001-1.0.4]: LoRaWAN L2 1.0.4 Specification, LoRa Alliance, October 2020

[RP002-1.0.1]: LoRaWAN Regional Parameters, LoRa Alliance, November 2019

[TS008]: TS008 LoRa Alliance Assigned Value Registries May 2020