NOTICE OF USE AND DISCLOSURE

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

LoRaWAN® Device Identification QR Codes for Automated Onboarding Technical Recommendation (TR005)

Authored by the LoRa Alliance Technical Committee

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

Editors:
A. YEGIN (Actility), J. KNAPP (Semtech)

Contributors:
R. APFELDORFER (Actility), J. KNAPP (Semtech), A. YEGIN (Actility)

Version: 1.0.0
Date: Oct 19, 2020
Status: Final

Contents

Tables

Figures

1 Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

2 Introduction

This document recommends a standard tagging scheme for LoRaWAN® devices to simplify the device onboarding steps onto a LoRaWAN® network.

It provides a low-cost and practical method for a variety of LoRaWAN® devices to be securely onboarded by the device owner. By utilizing industry-accepted norms for information processing, this recommendation enables the LoRaWAN® members to implement a quick, easy, secure, and interoperable method for onboarding a device through optical reading or manual entry.

3 Background

LoRaWAN devices are generally manufactured in bulk and personalized with their unique DevEUI, JoinEUI, security key(s) and DevAddr (in the case of ABP devices), at the time of manufacturing.

In order for devices to be accepted by the network, relevant device information must be shared with each network element prior to device activation. This process is called provisioning.

The process of associating an owner and verifying permission to use a network is called onboarding. During the onboarding process a generic provisioned device is associated with its owner and, optionally, meta-data that eases the management or use of the device. These devices may be offboarded when a user no longer desires to be responsible for the device.

Device attributes directly or indirectly identified by the QR code SHALL be valid when the device is put to use for the first time. Please note that they MAY change throughout the lifecycle of the device, e.g., by changing the JoinEUI over-the-air or using FUOTA to upgrade the firmware and the Device Profile. Addressing issues that may arise from such a change is outside the scope of this document.

4 Onboarding Tag Content

4.1 Character Set

Only the ALPHA and DIGIT characters as defined in IETF RFC 5234 [RFC5234], ".", and ":" SHALL be utilized. ALPHA characters SHALL be upper case.

4.2 Tag Information

Data is organized similarly to URN. A specific sequence of values and optional values are delimited by a ":".

> **Note:** This document has been written to enable the option to adopt this identifier within a standard URN as managed by IETF in the future. Specifically, this document would allow the identifier as described to be prefaced by "URN:DEV:LW:" if and when IANA allocates this URN to the LoRa Alliance.

The preface of the identifier SHALL only consist of "LW:".

The identifier SHALL be, at the minimum, composed of the following mandatory values, and always in this order: SchemaID, JoinEUI, DevEUI, ProfileID.

The mandatory values MAY be followed by one or more optional extensions. The optional extensions are CheckSum, OwnerToken, SerNum, and Proprietary extension and are pre-fixed by a parameter key:

Parameters	Key	Description
CheckSum	**C**	QR Checksum
OwnerToken	**O**	Owner Token
SerNum	**S**	Device Serial Number
Proprietary	**P**	Proprietary Extension

Table 1 – LoRaWAN® parameters keys

The maximum size of the full tag information shall be 128 characters, 48 characters are consumed by the mandatory information which leaves 80 characters for the optional extensions.

4.2.1 SchemaID

The SchemaID is used to indicate the schema that should be applied to the remaining data of the identifier. The SchemaID consists of two characters and is currently defined in the following table.

Schema ID	Description
D0	Device Schema Version 0 – the schema described in this document

Table 2 – Schema IDs

4.2.2 JoinEUI

JoinEUI is the initial JoinEUI value used by the device after shipment (in case the device uses multiple JoinEUIs) that identifies the JS uniquely in the LoRaWAN® Backend Interfaces 1.0 Specification.

The JoinEUI, formerly AppEUI, value uses a hexadecimal representation resulting in 16 characters.

4.2.3 DevEUI

IEEE allocates an Organization Unique Identifier and identifier range to the manufacturer of the subsystem running the LoRaWAN® stack (please refer to **IEEE Registration Authority**).

DevEUI consist of 8 bytes and is described in the LoRaWAN® Link-layer Specification [LW].

The DevEUI value uses a hexadecimal representation resulting in 16 characters.

4.2.4 ProfileID

The profile identifier encodes a Vendor Identifier and a Vendor Profile Identifier as a hexadecimal representation resulting in 8 characters.

ProfileID	VendorID	VendorProfileID
Size (Bytes)	2	2

Table 3 – Product ID

VendorID is assigned by the LoRa Alliance [VID]. The VendorID, 0xFFFF, is to be utilized prior to commercial production of a device.

VendorProfileID is assigned by the device manufacturer for commercial products. Because the ProfileID may be used as an index on a Device Profile database or payload decoding database, the manufacturer SHALL use a distinct VendorProfileID for devices with different Device Profiles [BE] or payload encodings.

4.2.5 Optional Extensions

4.2.5.1 Checksum

The Checksum is used to validate the data integrity. Even though QR codes have their own built-in integrity checks, this explicit checksum is useful when the content of the QR code is presented as plain text. Checksum is generated using the CRC-16-MODBUS [CRC] of the full QR content except the Checksum field itself, and presented in hexadecimal format without the leading “0x”.

4.2.5.2 OwnerToken

The OwnerToken is used to prove the ownership of the end-device (as identified by its DevEUI) to a system that allows the device owner to create and modify settings associated with the end-device. For example: Registering the end-device on the home NS and setting the home NS of the end-device on the JS.

It is RECOMMENDED that OwnerToken is not used for retrieving any confidential information related to the end-device. Using it for such purposes requires the OwnerToken to be protected at the same level as the AppKey/NwkKey.

The OwnerToken SHALL be protected against unauthorized access on the end-device until it is used by the legitimate owner of the device (e.g., not accessible until the end-device is unpacked).

How the OwnerToken is generated, delivered to the systems verifying its value, and whether it is a one-time-use value are outside the scope of this document. See Appendix A for an example.

4.2.5.3 SerNum

The SerNum, serial number, is a unique identifier assigned during the product manufacturing process. The SerNum does not need to be strictly a number and may contain any characters, with the exception of the “:” from the available alphabet indicated in Section 5.1 of this document.

4.2.5.4 Proprietary

Tag content can be extended using proprietary schemes, using any characters, with the exception of the “:” from the available alphabet indicated in Section 5.1 of this document.

The ProfileID field may be used to determine the interpretation of the proprietary extensions and SerNum field.

4.3 Example QR Codes

Given:

SchemaID of D0
JoinEUI of 11-22-33-44-55-66-77-88
DevEUI of AA-BB-CC-DD-EE-FF-00-11
ProfileID of AABB-1122
OwnerToken of AABBCCDDEEFF
SerNum of YYWWNNNNNN
Proprietary of FOOBAR
CheckSum of AF2C

This requires size 4 and can only have ECC=Medium.

Here are the 88 bytes of data:

LW:D0:1122334455667788:AABBCCDDEEFF0011:AABB1122:OAABBCCDDEEFF:SYYWWNNNNNN:PFOOBAR:CAF2C

And the QR code:

Figure 1 QR code example (full)

Example of minimal mandatory:

This requires size 4 and can have ECC=High.

Here are the 48 bytes of data:

LW:D0:1122334455667788:AABBCCDDEEFF0011:AABB1122

And the QR code:

Figure 2 QR code example (basic)

5 QR Code Physical Recommendations

The exact physical tag is not specified but MUST comply to the following requirements.

> **Note:**
>
> - Most QR codes default to Medium level of error correction.
> - Version 4 QR code with level medium correction can contain up to 62 Binary or 90 Alphanumeric characters.

5.1 Requirement 1

The QR code format SHALL be correctly readable with standard readers, even when parts of the QR code are dirty or damaged.

5.2 Requirement 2

The format SHALL allow a QR code with only mandatory parameters to be printed and scanned on as little as a 7x7mm surface and still comply with the interoperability and robustness requirement. Rationale: Some devices have tight space constraints on where to print the QR-code and thus need a very small QR-code size.

5.3 Requirement 3

The QR code format SHALL be feasible to be added to the devices in mass production in a cost-efficient manner.

Appendix A. Example Generation and Use of OwnerToken

This is an example of how OwnerToken may be generated and used. There are possibly other ways of managing the OwnerToken. They are all valid as long as they are compliant with the specification in Section 4.2.5.2.

- At the time of device provisioning on the JS, the JS generates a random OwnerToken and assigns it to the DevEUI of the device. Both the DevEUI and the OwnerToken (along with some other attributes, such as AppKey) are stored on the JS. The JS marks the device as “unclaimed” at that point.

- The OwnerToken is provided to the manufacturer to generate the QR code to be printed and placed on the end-device. There are other information elements needed for the QR that are not mentioned here for the sake of brevity.

- Device with the QR code is placed inside a box and made available to the final owner through a chain of events. The QR code cannot be readily seen from outside the box during this journey.

- The owner unboxes the device. The owner uses the mobile app provided by her LoRaWAN operator in order to provision the newly-acquired device with her own account.

- Mobile app reads the QR code on the device and sends that to a user account management server in the operator domain.

- The server contacts the JS as identified by the JoinEUI which is encoded on the QR to attempt to claim the device as identified by the DevEUI in order to set its home NS while presenting the OwnerToken as the proof of ownership. The interface allowing this interaction is currently outside the scope of LoRa Alliance.

- The JS verifies the device identified by the DevEUI is not claimed and the received OwnerToken matches the stored value. If this is the case, the JS marks the device as “claimed”, and sets the home NS to the value received in the request.

In this example, one-time claim is illustrated. Subsequent unclaim/claim procedures can be implemented by the JS generating and providing a new OwnerToken back to the current owner to be shared with the future owner of the device.

Glossary

AS	Application Server
JS	Join Server
Hex	Hexadecimal
NS	Network Server
QR code	A machine-readable optical label that contains information about the item to which it is attached

6 Bibliography

6.1 References

[L2] LoRaWAN™ 1.0.3 Specification, LoRa Alliance, March 20, 2018

[BE] LoRaWAN™ Backend Interfaces 1.0 Specification, LoRa Alliance, October 11, 2017

[CRC] Modicon Modbus Protocol Reference Guide, Modicon Inc., June, 1996. (http://modbus.org/docs/PI_MBUS_300.pdf)

[RFC5234] Augmented BNF for Syntax Specifications: ABNF, IETF 5234, IETF, January, 2008.

[VID] https://lora-alliance.org/resource-hub/lora-alliance-vendor-id-qr-code

7 NOTICE OF USE AND DISCLOSURE

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