QR Code Requirements

18-01000 Revision 7

Zigbee Document	18-01000
Date of release	June 2018
Sponsored by	Zigbee Alliance
Accepted by	This document has been accepted for release by the Zigbee Alliance Board of Directors on June 14, 2018
Abstract	This document is a specification for a best practice to use QR Codes for Dotdot & Zigbee products
Keywords	Zigbee, Dotdot, Commissioning, QR Code

Notice of Use and Disclosure

35

36 # 1. Table of Contents

69

70

71 # 2. Glossary

Term	Description
QR-Code	A machine-readable optical label that contains information about the item to which it is attached.
Parameter or Field	An individual piece of information stored in a QR-code
Install Code[1]	Random number specific to a device used to derive the unique Trust Center Link Key for joining a Zigbee 3.0 device to a network. Includes a CRC.
AN	Alphanumeric with only upper case
QA	QR-Code alphanumeric[2] – 0-9, A-Z, *space*, ':', '$', '%', '*', '+','-','.','/'
Dec	Decimal
Hex	Hexadecimal
MS or ms or Mfg	Manufacturer Specific
M, O, X, n/a	Mandatory, Optional, Prohibited, Not Applicable

72 # 3. Reference documents

Name	Doc Number
Base Device Behavior	13-0402
Zigbee 3.0 Cluster List	16-02867
Manufacturer Code Database	05-3874

73

74 # 4. Document Version

Rev	Date	Author	Notes
000	01/26/2018	Emmanuel Pauchard	First Draft in Progress
001	02/06/2018	Cam Williams	Updated from TG minutes
002	02/16/2018	Cam Williams	Updated from call and emails
003	03/20/2018	Emmanuel Pauchard	San Diego update
004	06/04/2018	Cam Williams	Update after month of feasibility study
005	06/05/2018	Cam Williams	Added local protocol parameter overrides global
006	06/07/2018	Cam Williams	Clarified that CRC16 included in Install Code parameter
007	6/20/2018	Victor Berrios	Updated BoD approval status, removed "DRAFT" watermark. Prepared document for release.

75

76 # 5. Purpose and Scope

77 The purpose of this document is to provide requirements in a specification to be used as a best practice for use of
78 QR Codes on products. This is a living document and should be maintained as a useful specification on the format of
79 QR Codes and related scannable labeling for Dotdot and Zigbee products.

[1]Formatted as described in the Base Device Behavior specification 13-0402 without spaces
[2] Reference: ISO/IEC 18004:2015, section 7.3.4: Alphanumeric mode

6. Executive Summary

Zigbee 3.0 requires that all devices support Install Codes and support network joining using these Install Codes. The specification is also clear on the format that this Install Code should take when written on the device in a human-readable form. However, the specification does not currently mandate the format that the Install code should take if another solution than writing on the product is chosen, esp. any machine-readable form.

This document will list requirements that apply to the devices using a QR-code to carry the Install Code of the device and will list the additional fields that can be included in the QR-code.

7. Stakes & Outcomes

Many Zigbee 3.0 devices will use QR-codes to carry Install Codes along with other fields. If the standard is not clear on the format of this QR-code, then limited cross-vendor interoperability will be achieved and user experience during network commissioning might degrade. On the other hand, if all devices use the same QR-code format then smartphone commissioning application design will be eased and user experience made more consistent.

8. Needs & Constraints

This section lists the needs and constraints that the proposed QR-code format should solve.

8.1. QR-NEED-1: Content: install code

The QR-code SHALL include the Install Code of the device, including its CRC as defined in the Base Device Behavior, section 10.1.1.

8.2. QR-NEED-2: Content: EUI64

The QR-code SHALL include the EUI64 of the device.

8.3. QR-NEED-3: Fields Extensibility

The QR-code format SHALL be extensible (ie: allow additional fields, either future standard fields described in this document or manufacturer specific fields)

Rationale: this makes the QR-code format future-proof as we can always add new parameters later-in; also having a way for manufacturers to set up specific fields in the same QR code would lead to easier user experience (only 1 qr code to scan to gather different information).

8.4. QR-NEED-4: Fields Flexibility

The QR-code format SHALL NOT rely on fixed positions for fields (ie: field order should not matter).

Rationale: because of the extensibility requirement and the compatibility with other standards, we cannot guarantee that the order of fields will always be the same for all zigbee devices.

8.5. QR-NEED-5: Compatibility with other standards

Optional qualifying notation is supported to differentiate between protocols.

Rationale: in the case where a device is compatible with several standards that require security material or other parameters it is beneficial for the user experience to have all parameters in the same QR-code, and thus namespaces for each standard should be defined.

8.6. QR-NEED-6: Interoperability

The QR-code format SHALL follow the standard QR-code definition.

Rationale: must be readable by the typical QR-code reader library already available.

8.7. QR-NEED-7: Coding efficiency

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.

8.8. QR-NEED-8: Sufficient data

The QR-code format SHALL be able to store at least the mandatory items.

The number of mandatory items SHALL be minimized, to still comply with the Coding efficiency and Robustness requirement.

Should allow for optional extensions, while still allowing for minimum space.

8.9. QR-NEED-9: Robustness

The QR-code format SHALL be correctly readable with standard readers (incl. smartphones), in normal home and industry lighting operational conditions, incl. shiny and curved surfaces the QR codes are put on.

8.10. QR-NEED-10: Apply suitable level of ECC

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

8.11. QR-NEED-11: Suitable for mass production

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

9. Proposed Solution

This section describes the solution proposed to cover the user needs and constraints, and the requirements that define this solution.

9.1. Format

The following is the simple BNF grammar for a reader to parse a QR Code. Spaces are allowed in <value>.

<qr code string> ::= < parameter list>[%<parameter list>]…

<parameter list> ::= <parameter>[$<parameter>]…

<parameter> ::= <key>[:<value>]

<key> ::= <alpha>[<alpha>]…

<alpha> ::= *capital letters A-Z*

<value> ::= *This is specific to the <key>. To represent characters "*", "$", %", and ":", are coded for as "**", "*$", "*%", "*:", respectively. Spaces are allowed, but discouraged.*

9.1. Semantic

A <key> is a name made up of one or more capital letters. A <key> has a type that is listed in the following table with the following rules.

Key Type	Description
Interface	This type of key defines a type of interface or protocol (e.g. Zigbee, Thread, WiFi, etc). **It SHALL be the first <key> in a <parameter list>**, and defines the interface associated with the parameters in the <parameter list>.As a shortcut, the value for this key MAY also be used to define a value for the interface. See the table below for valid Interface keys. Each Interface is listed only once in the string. If there are multiple instances of the same interfaces on the product, then each MAY be listed. Mapping each interface instance to a network node within the device, is not defined here, and is manufacturer specific.
Parameter	This type of key defines a parameter for an interface and has a fixed data type. Each key SHALL be unique to the interface, though many interfaces use the same key to represent the same parameter type.See the tables below for each interface. Keys for the Manufacter Specific Interface are not specified here, and are up to the manufacturer to manage.

9.2. Interface Keys

Below is a list of interface keys. After an interface key, comes a list of parameters for the interface. The interface key may have a value that is a short cut value for one main parameter.

Interface	Key	Short Cut Value	Format	# Chars	Value Description
Zigbee	**Z**	MAC Address	Hex	16	MAC Address[3]
Thread	**T**	MAC Address	Hex	16	MAC Address[3]
Wi-Fi	**W**	MAC Address	Hex	12	MAC Address[3]
Ethernet	**E**	MAC Address	Hes	12	MAC Address[3]
Bluetooth	**B**	MAC Address	Hex	12	MAC Address[3]
Global	**G**	n/a	n/a	n/a	n/a
Mfg Spefic	**M**	Zigbee Mfg Code	Hex	4	16-bit Zigbee Manufacturer Code[4]

9.1. Parameter Keys

Parameter keys SHALL be unique to the interface, though the same key name MAY be resused by many interfaces to represent the same or similar interface parameters.

9.1.1. Zigbee Parameter Keys

Parameter	Key	Format	# Chars	Value Description	M/O
MAC Address	**A**	Hex	16	EUI64 MAC Address	M[5]
Install Code[6]	**I**	Hex	36	Install Code with CRC	M
Certification #	**C**	QA	19	Zigbee unique certificate number	O

[3] Using MAC Canonical formatas defined in IEEE 802-2001
[4] See document 05-3874, Manufacturer Code Database
[5] The MAC Address is mandatory using this parameter key, the Zigbee interface key value, or the global interface MAC Address key value.
[6]Formatted as described in the Base Device Behavior specification13-0402 , without spaces

9.1.1. Thread Parameter Keys

Parameter	Key	Format	# Chars	Value Description	M/O
MAC Address	**A**	Hex	16	EUI64 MAC Address	M[7]
Passcode	**P**	AN	Variable	Thread Passcode	M

9.1.2. Wi-Fi Parameter Keys

Parameter	Key	Format	# Chars	Description	M/O
MAC Address	**A**	Hex	12	EUI48 MAC Address	M[5]
Wi-Fi SSID	**S**	AN	Variable	WiFi SSID of network	TBD
Password	**P**	AN	Variable	WiFi password	TBD

9.1.1. Ethernet Parameter Keys

Parameter	Key	Format	# Chars	Description	M/O
MAC Address	**A**	Hex	12	EUI48 MAC Address	M[5]

9.1.2. Bluetooth Parameter Keys

Parameter	Key	Format	# Chars	Description	M/O
MAC Address	**A**	Hex	12	EUI48 MAC Address	M[5]
PIN	**P**	AN	Variable	Bluetooth PIN	TBD

9.1.3. Global Parameter Keys

Global Parameters are used for items that are common to the entire product or as a short cut to represent many interfaces. For example, the MAC Address would be the same for a product that with a 802.15.4 chip that could support Zigbee and/or Thread. Both the Zigbee and Thread

[7] The MAC Address is mandatory using this parameter key, the interface key value, or the global interface MAC Address key value.

interfaces would have the same MAC Address which would then be a global parameter. An protocol Interface parameter SHALL override a global interface parameter.

Parameter	Key	Format	# Chars	Description	M/O
MAC Address	**A**	Hex	Variable	MAC Address[8] of one or more interfaces	O
GTIN	**G**	Dec	Variable	Global Trade Item Number	O
SKU	**K**	Dec	Variable	Stock Keeping Unit	O
Serial Number	**S**	AN	Variable	Mfg serial number of product unit	O
Model Number	**M**	AN	Variable	Mfg model number of product	O
Description	**D**	AN	Variable	Mfg description of product	O

9.1.4. Manufacturer Specific Parameter Keys

9.2. Parameters for the Manufacturer Specific Interface are not defined here and are a specific to the manufacturer, except for the Zigbee Manufacturer Code[9], which SHALL be present, as the value of the Interface Key.Examples

Global is gray. Zigbee is red. Thread is orange. WiFi is blue. Manufacturer Specific is green. Delimiters are black.

Zigbee End Device:

Z$**A:**0123456789ABCDEF$**I:**023047432043AF3456FEB234524234234567

with shortcut:

Z:0123456789ABCDEF$**I:**023047432043AF3456FEB234524234234567

Zigbee End Device with Manufacturer Specific Data:

Z:0123456789ABCDEF$**I:**023047432043AF3456FEB234524234234567**%M:**10CD$**A:**243$**B:**EDF

Thread End Device with Serial Number, Model NumberManufacturer Specific Data:

G$S:**76E56345W$M:**BOX**%T:**0123456789ABCDEF$**P:**ABC**%M:**10CD$**P:**243$**B:**EDF

[8] This is used as a short cut when a single node is used for multiple interfaces and uses the Canonical format as defined in IEEE 802-2001

[9] See document 05-3874, Manufacturer Code Database

Zigbee or Thread dual boot End Device with one MAC Address (same):

G$A:0123456789ABCDEF%Z$I:023047432043AF3456FEB234524234234567%T$P:ABC

Zigbee or Thread dual boot End Device with one MAC Address and single WiFi chip with its own MAC Address:

G$A:0123456789ABCDEF%Z$I:023047432043AF3456FEB234524234234567%T$P:ABC%W:0123456789AB$S:NET$P:NET

Zigbee/Thread Gateway with each on separate chips:

Z:0123456789ABCDEF$I:023047432043AF3456FEB234524234234567%T:ABCDEF0123456789$P:ABC

WiFi/Thread/Thread Gateway with 2 Thread chips:

W:0123456789AB$S:NET$P:NET%T:0123456789ABCDEF$P:ROB%T:ABCDEF0123456789$P:BOB