Quuppa

Quuppa Tag Emulation for Android Devices

January 2025

Contents

Terms of Use

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE SPECIFICATIONS ARE PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE, AND QUUPPA EXPRESSLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY OTHER WARRANTY. THE ENTIRE RISK FOR USING THE SPECIFICATIONS AND PERFORMANCE THEREOF REMAINS WITH YOU.

IN NO EVENT SHALL QUUPPA BE LIABLE TO YOU OR TO ANY OTHER PARTY FOR THE COST OF PROCURING ANY GOODS OR SERVICES, LOST PROFITS, LOSS OF USE, LOSS OF DATA OR ANY INCIDENTAL, CONSEQUENTIAL, INDIRECT, OR SPECIAL DAMAGES OF ANY KIND, WHETHER UNDER CONTRACT, TORT, WARRANTY OR OTHERWISE AND ARISING IN ANY WAY OUT OF THESE TERMS OF USE, WHETHER OR NOT QUUPPA HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

Introduction

This document describes how to emulate a Quuppa Tag using Android Devices. It's assumed that the reader is familiar with Bluetooth® Low Energy wireless technology and its use on Android.

Quuppa Tag Emulation

Quuppa Tag emulation can be implemented on any device that supports the Bluetooth Low Energy Advertising State as defined in the Bluetooth Core Specification version 4.0 and higher. For example a smartphone running Android 5.0 and Bluetooth version 4.1 can perform Quuppa Tag emulation. A device emulating a Quuppa Tag will send Quuppa Direction Finding Packets (DF packets) and can send Quuppa Data packets. Quuppa Data packets are defined in the generic Quuppa Tag emulation specification.

A *Quuppa Intelligent Locating System*™ can accurately track the position of devices emitting DF Packets. For more information about Bluetooth Low Energy on Android OS, see: https://developer.android.com/guide/topics/connectivity/bluetooth/ble-overview

Advertising parameters for Android Applications Emulating Quuppa Tags

Generic documentation about Bluetooth Low Energy advertising set parameters for Android is available here:
https://developer.android.com/reference/android/bluetooth/le/AdvertisingSetParameters

The transmit power of an app emulating a Quuppa tag is set to **TX_POWER_HIGH**.

The advertising interval is set to **INTERVAL_LOW**.

If the emulating app can detect the movement state of the device, the app can optimize the air interface usage of the device by switching the advertising interval to **INTERVAL_HIGH** and updating the header of the Quuppa DF packet to the value **0x18** when the device is stationary. However, this is optional as the default INTERVAL_LOW and header value **0x19** can be used for both moving and stationary devices. The encoding of the header value and mapping to advertise mode is shown in Table 2.

Quuppa Direction Finding Packet

The advertising data of a Quuppa DF packet is outlined in the following section. A complete Bluetooth Low Energy Link Layer packet formatted as a Quuppa DF packet is shown in Appendix A.

Advertising Data

The advertising data of the Quuppa DF packet is shown in Figure 1.

LSB *MSB*

Advertising Data					
AD Structure 1			AD Structure 2		
Length (1 octet)	AD Type (1 octet)	AD Data (1 octet)	Length (1 octet)	AD Type (1 octet)	AD Data (26 octets)
0x02	0x01 (Flags)	Flags data	0x1B	0xFF (Man. Specific)	Manufacturer Specific data

Figure 1. Advertising data of Quuppa DF packet

The advertising data consist of two advertising data (AD) structures. The first AD structure is the flags AD structure. The content of the flags data depends on the device features and is defined in the Bluetooth Core Specification v4.1.[1] The content of the flags data field is not relevant for the Quuppa system but the flags AD structure must be present as the first AD structure in the packet.

The second AD structure contains Manufacturer Specific Data. The content of the manufacturer specific AD data is outlined in the next section.

Manufacturer Specific Data

Documentation about how to add manufacturer data to the Bluetooth advertising packet in Android is available here:
https://developer.android.com/reference/android/bluetooth/le/AdvertiseData.Builder

The Manufacturer specific data of the Quuppa DF packet is shown in Figure 2.

LSB *MSB*

Company ID (2 octet)	Quuppa Packet ID (1 octet)	Quuppa Device Type (1 octet)	Header (1 octet)	Quuppa Tag ID (6 octets)	Checksum (1 octet)	DF Field (14 octets)
0x00C7	0x01					

Figure 2. Manufacturer specific data of Quuppa DF packet

[1] Note, that for example Android 5.0 devices add the flags AD structure to the beginning of advertising data automatically. If the flags do not appear automatically, then the app needs to set the advertising type as connectable.

Company ID

The Company ID needs to be set to 0x00C7.

Quuppa Packet ID

The Quuppa Packet ID needs to be set to 0x01.

Quuppa Device Type

The Quuppa device type is used to define the type of the device emulating the Quuppa Tag. It's exposed through the Quuppa Web Services API. Table 1 shows the encoding of the Quuppa device type field.

Quuppa Device Type	Quuppa Device Type Name
0x21	Android smartphone
0x22	Android tablet

Table 1. Encoding of the Quuppa Device Type field

Header

The header carries information about the movement status of the device. Table 2 shows the encoding of the header field.

Header	Movement status	Advertising interval
0x19	Moving at walking speed	**INTERVAL_HIGH**
0x18	Stationary	**INTERVAL_LOW**

Table 2. Encoding of the Header field. Please note that the device should report Stationary status only when it really is stationary. If a moving device reports that it is stationary, the QPE will not be able to track the movement. If the app cannot detect whether the device is stationary or not the app should always use the header value 0x19 and use INTERVAL_HIGH.

Checksum

The checksum is calculated using a CRC algorithm with following parameters:

- Width: 8-bit
- Polynomial: 0x97
- Initial value: 0x00
- Final XOR value not used (0x00)
- No input or output reflection

The checksum is calculated over the Quuppa Device Type, Header, and Quuppa Tag ID fields as shown below[2]:

CRC-8[Quuppa Device Type, Header, Quuppa Tag ID (MSB) … Quuppa Tag ID (LSB)]

[2] An example implementation of the checksum calculation can be found in the source code of the Android Quuppa Tag Library. https://github.com/quuppalabs/android-quuppa-taglib

Example result for a checksum calculation is shown below:

CRC-8[0x21, 0x19, 0x55, 0x44, 0x33, 0x22, 0x11, 0x00] = 0x38

DF Field

The last field of the manufacturer specific data in the DF packet is the DF field. The DF field consists of 14 fixed octets as defined in Table 3.

Octet #	DF Field content
OCTET 0	0x67
OCTET 1	0xF7
OCTET 2	0xDB
OCTET 3	0x34
OCTET 4	0xC4
OCTET 5	0x03
OCTET 6	0x8E
OCTET 7	0x5C
OCTET 8	0x0B
OCTET 9	0xAA
OCTET 10	0x97
OCTET 11	0x30
OCTET 12	0x56
OCTET 13	0xE6

Table 3. Content of the DF field of the Quuppa DF packet

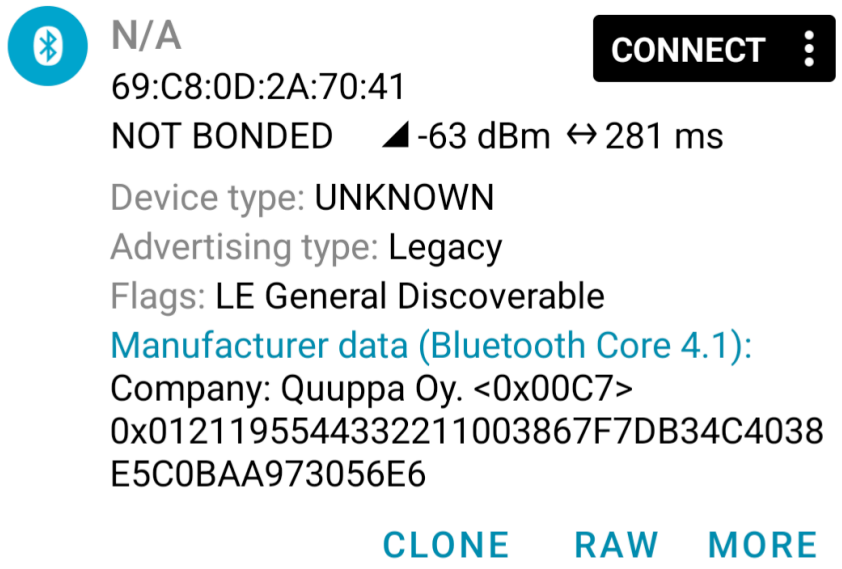
Appendix A

A Bluetooth Low Energy Link Layer packet formatted as a Quuppa DF packet for a moving Android smartphone. The values in *Italics* are not fixed and are given here as an example. Each Quuppa Tag should set the Quuppa Device Type and Header based on the used parameters, use a unique Quuppa Tag ID, and compute the Checksum accordingly.

Octet #	Field	Contents	Example
0	Preamble	Bluetooth Low Energy ADV PDU Preamble	
1	Access Address	Bluetooth Low Energy ADV PDU Access Address	
2	Access Address		
3	Access Address		
4	Access Address		
5	BLE header (LSB)	Bluetooth Low Energy ADV PDU Header	
6	BLE header (MSB)		
7	AdvA (LSB)	Advertiser's public or random Bluetooth Low Energy address	
8	AdvA		
9	AdvA		
10	AdvA		
11	AdvA		
12	AdvA (MSB)		
13	AD data Length	0x02	
14	AD Type: Flags	0x01	
15	Flags Data	Depends on the device features. Defined in the Bluetooth core spec. Content is not relevant for the Quuppa system but the flags field must be present in the packet as the first AD structure.	
16	AD data Length	0x1B	
17	AD Type	Manufacturer specific data, Value=0xFF	
18	Company ID (LSB)	0xC7	
19	Company ID (MSB)	0x00	
20	Quuppa Packet ID	0x01	
21	Quuppa Device Type	Device type identifier of a Quuppa Tag	*0x21*
22	Header	Device's movement status	*0x19*
23	Quuppa Tag ID (MSB)	6-byte identifier for a Quuppa Tag	*0x55*
24	Quuppa Tag ID		*0x44*
25	Quuppa Tag ID		*0x33*
26	Quuppa Tag ID		*0x22*
27	Quuppa Tag ID		*0x11*
28	Quuppa Tag ID (LSB)		*0x00*
29	Checksum	CRC-8, polynomial 0x97 over bytes 21-28	*0x38*
30	DF field	0x67	
31	DF field	0xF7	
32	DF field	0xDB	
33	DF field	0x34	
34	DF field	0xC4	
35	DF field	0x03	
36	DF field	0x8E	
37	DF field	0x5C	
38	DF field	0x0B	
39	DF field	0xAA	
40	DF field	0x97	
41	DF field	0x30	
42	DF field	0x56	
43	DF field	0xE6	
44	CRC	3-byte CRC according to BLE specification	
45	CRC		
46	CRC		

Appendix B

Screenshots of the example Quuppa DF packet (shown in Appendix A) captured with the nRF connect Android app. Please verify that your app can produce the example packet correctly and especially that your CRC calculation produces the same CRC with the given input and that the content of the packet matches otherwise also. Remember the value of the flags AD structure (type 0x01) doesn't matter but it has to be included as the first AD structure in the packet.



Raw data:

```
0x0201021BFFC700012119554433221100386
7F7DB34C4038E5C0BAA973056E6
```

Details:

LEN.	TYPE	VALUE
2	0x01	0x02
27	0xFF	0xC700012119554433221100386 7F7DB34C4038E5C0BAA973056 E6

LEN. - length of AD packet (Type + Data) in bytes,
TYPE - the data type as in Assigned Numbers.pdf,
chapter 2.3: Common Data Types.

Revision History

Version:	Date:	By:	Notes
0.9	13.01.2022	AK	Draft created.
1.0	16.02.2022	JT	Edited text and prepared for release.
2.0	28.1.2025	AK	Updated to refer to the latest Android Bluetooth APIs. Header values updated to report TX rate level 0.