Quick Start Guide

Getting started with ST Quuppa tag emulation

Introduction

This document provides information about the BlueNRG-LP ST Quuppa tag emulation SW provided within the STSW-QUUPPA-ETAG SW package.
The STSW-QUUPPA-ETAG evaluation SW package provides the ST Quuppa tag emulation library and associated demonstration application which allow to build a Quuppa tag emulation device supporting the following features:

- location-tracking capability
- multiple sensors data provisioning
- framework for some custom back channel commands to be used through the Quuppa's proprietary positioning system.

ST Quuppa tag emulation library is built following the specification of Quuppa Tag Emulation using Bluetooth Wireless Technology and the specification of Quuppa Tag Back Channel using Bluetooth Wireless Technology. It also allows to support a set of Quuppa preconfigured profiles which define specific tag features.

ST Quuppa tag emulation is based on Bluetooth Low Energy Advertising State as defined in Bluetooth Core Specification version 4.0 and higher. It provides support for specific standard manufacturing advertising packets (direction finding and data packets) and it works on standard Bluetooth LE channels.

This document describes the STSW-QUUPPA-ETAG SW package components and related HW and SW setup instructions.
It also provides the fundamental information about the Quuppa tag emulation and back channel capability and some details about the ST Quuppa tag emulation solution.

The following BlueNRG-LP kits are supported:

- STEVAL-IDB011V1 (QFN48 package) development platform

快速入门指南

开始使用 ST Quuppa 标签模拟

介绍

本文档提供有关BlueNRG-LP的信息STSW-QUUPPA-ETAG内提供 ST Quuppa 标签仿真软件SW 封装。

STSW -QUUPPA-ETAG评估软件包提供了 ST Quuppa 标签仿真库和相关的演示应用程序,允许构建支持以下功能的 Quuppa 标签仿真设备:

- 位置跟踪能力
- 多传感器数据配置
- 一些自定义反向通道命令的框架,可通过 Quuppa 的专有定位系统。

ST Quuppa 标签仿真库遵循使用蓝牙无线技术的 Quuppa 标签仿真规范和使用蓝牙无线技术的 Quuppa 标签反向通道规范构建。它还允许支持一组定义特定标签功能的 Quuppa 预配置文件。

ST Quuppa 标签模拟基于蓝牙核心规范 4.0 版及更高版本中定义的蓝牙低功耗广告状态。它支持特定标准制造广告包（测向和数据包）,并可在标准蓝牙低功耗通道上运 。

本文档描述了STSW-QUUPPA-ETAG SW包组件以及相关的硬件和软件设置说明。

它还提供了有关 Quuppa 标签模拟和反向通道功能的基本信息以及有关 ST Quuppa 标签模拟解决方案的一些细节。

以下BlueNRG-LP支持的套件:

- STEVAL-IDB011V1 (QFN48封装)开发平台

Contents

内容

1 HW/SW setup

The STSW-QUUPPA-ETAG SW package is delivered as a zip file and it provides the following components:

- ST Quuppa tag emulation library folder with ST Quuppa tag emulation library, header files and demonstration application source and header files (`Projects\BLE_Examples\BLE_Quuppa_Tag_Library`).
- ST Quuppa tag emulation folder with demonstration application IDE projects IAR, KEIL and WiSE-Studio (`Projects\BLE_Examples\BLE_Quuppa_Tag_with_Lib`).
- ST Quuppa tag emulation release notes and APIs html documentation (`Docs\st_quuppa_tag_emulation_release_notes_html`, `Docs\st_quuppa_tag_emulation_apis_html`, `Docs\index_st_tag.html`).

  NOTE: `Docs\index_st_tag.html` is the entry point to ST Quuppa tag emulation html documentation.

User is requested to unzip/extract the en.STSW-QUUPPA-ETAG.zip file under his local STSW-BNRGLP-DK SW package installation folder

1.1 How to configure a ST Quuppa tag emulation?

The following instructions must be followed in order to configure a BlueNRG-LP, STEVAL-IDB011V1 kit as an ST Quuppa tag emulation device:

1) Open ST Quuppa tag emulation IDE project (IAR or KEIL or WiSE-Studio) available on `Projects/BLE_Examples/BLE_Quuppa_Tag_with_Lib` folder.
2) Power on BlueNRG-LP, STEVAL-IDB011V1 using USB (connect to a PC USB port) or by battery.
3) Build and download the ST Quuppa tag emulation on selected platform using the selected IDE download option, or just drag and drop the `BLE_Quuppa_Tag_with_Lib.hex` on the STEVAL-IDB011V1 available on Windows navigation tree, or use the RF-Flasher Utility tool.
4) If STEVAL-IDB011V1 is connected to a PC USB port, open a serial communication terminal on PC (as HyperTerminal or TeraTerm), in order to get some debug messages.
   Configuration is:
   - 115200 baud rate
   - 8 bits data
   - 1 start bit
   - 1 stop bit
   - no parity
   - no HW flow control

   NOTE: ST_QUUPPA_TAG_PROFILE_ID_BADGE is the current default Quuppa profile selected on `st_quuppa_tag_main.h` header file.

On power up, user gets a welcome message on connected hype terminal and several information messages allowing to follow device states transitions:

```
BlueNRG-LP ST Quuppa Tag Application (version: 1.0.0)
aci_gatt_srv_init() --> SUCCESS
aci_gap_init() --> SUCCESS
STORAGE state: press botton or trigger acceleration!
```

1 硬件/软件设置

STSW -QUUPPA-ETAG SW 包以 zip 文件形式提供,它提供以下组件:

- o ST Quuppa 标签仿真库文件夹,其中包含 ST Quuppa 标签仿真库、头文件以及演示应用程序源和头文件（项目\BLE_Examples\BLE_Quuppa_Tag_Library）。
- o ST Quuppa 标签仿真文件夹,带有演示应用程序 IDE 项目 IAR、KEIL 和 WiSE-Studio（项目\BLE_Examples\BLE_Quuppa_Tag_with_Lib）。
- o ST Quuppa 标签模拟发说明和 API html 文档（Docs\st_quuppa_tag_emulation_release_notes_html，文档\st_quuppa_tag_emulation_apis_html,文档\index_st_tag.html)。

注意:Docs\index_st_tag.html 是 ST Quuppa 标签模拟 html 文档的入口点。

要求用户解压/提取en.STSW-QUUPPA-ETAG.zip将其归档到本地STSW-BNRGLP-DK SW包安装文件夹

1.1 如何配置 ST Quuppa 标签模拟?

要配置BlueNRG-LP,必须遵循以下说明， STEVAL- IDB011V1套件作为 ST Quuppa 标签仿真设备:

1) 打开 Projects/BLE_Examples/BLE_Quuppa_Tag_with_Lib 文件夹中的 ST Quuppa 标签仿真 IDE 项目（IAR 或 KEIL 或 WiSE-Studio）。
2) 启动BlueNRG-LP， STEVAL-IDB011V1使用 USB（连接到 PC USB 端口）或使用电池。
3)使用选定的 IDE 下载选项在选定的平台上构建和下载 ST Quuppa 标签仿真,或者只需将 BLE_Quuppa_Tag_with_Lib.hex 拖放到 STEVAL-IDB011V1上可以在 Windows 导航树中找到,或者使用 RF-Flasher 实用工具。

4)如果STEVAL-IDB011V1连接到PC的USB端口,在PC上打开串通信终端（如超级终端或TeraTerm）,以获取一些调试信息。

配置为： - 115200 波特率
- 8位数据
- 1个起始位
- 1 个停止位
- 没有奇偶校验
- 无硬件流量控制

注意： ST_QUUPPA_TAG_PROFILE_ID_BADGE 是当前在 st_quuppa_tag_main.h 头文件上选择的默认 Quuppa 配置文件。

通电后,用户会在连接的炒作终端上收到一条欢迎消息和几条允许跟踪设备状态转换的信息消息:

```
BlueNRG-LP ST Quuppa 标签应用程序（版本:1.0.0）
aci_gatt_srv_init() --> 成功
aci_gap_init() --> 成功
存储状态:按按钮或触发加速!
```

ST Quuppa tag emulation is then ready to be used within a Quuppa positioning tracking system.

2 ST Quuppa tag emulation

The ST Quuppa tag emulation demonstration example emulates a Quuppa Tag using standard Bluetooth LE specification. It configures a BlueNRG-LP ST device in order to send Quuppa Direction Finding (DF) packets used for position tracking and Quuppa Data packets which are used for data transfer.

A system employing Quuppa Intelligent Locating Technology can accurately detect and track the position of ST Quuppa tag emulation as well as receive and expose data transmitted by it.

For more information about the Quuppa emulation system and related functionalities please refer to the documentation on Quuppa web site.

NOTE:

In order to evaluate the ST Quuppa tag emulation, user could simply verify proper Direction Finding and Data packets format and contents or he could use the solution within a specific Quuppa positioning system.

The Quuppa Development Kit could be also used. In this case, user could refer to the Quuppa Development Kit User Manual for step-by-step instructions about planning, installing, deploying, and configuring a Quuppa Location System with locators and track the ST Quuppa tag emulation.

2.1 ST Quuppa tag emulation configuration

The ST Quuppa tag emulation works on standard Bluetooth LE channel 37 and it could be tracked by the Quuppa Positioning Engine (QPE) localization systems provided on Quuppa Location System.

It uses a specific public address as Quuppa tag ID which could be customized by user. The ST Quuppa developer ID is 0x0081.

A specific `ST_Quuppa_Tag_Init()` API allows to provide the following initialization parameters to the ST Quuppa tag emulation library:

- ST Quuppa tag emulation public address also used as tag ID;
- ST Quuppa tag emulation device name;
- ST Quuppa tag emulation profile number used for selecting the profile number within the supported range of profiles;
- ST Quuppa tag emulation DF packet Device Type;
- ST Quuppa tag emulation developer specific payload data used on generic developer specific payload data response;
- ST Quuppa tag emulation unsupported response payload data used on unsupported response payload.

User can customize such parameters according to the allowed values through the `st_quuppa_tag_main.h` header file.

The `ST_Quuppa_Tag_Init()` API must be called at initialization phase with the selected parameters. The user must call the related ST Quuppa tag emulation state machine API `ST_Quuppa_Tag_SM()` on main `while{1}` loop.

The predefined supported profiles define the following configurations which impact how the device state machine works and some specific device RF capabilities.

ST Quuppa 标签仿真随后即可在 Quuppa 定位跟踪系统中使用。

2 ST Quuppa 标签模拟

ST Quuppa 标签模拟演示示例使用标准蓝牙 LE 规范模拟 Quuppa 标签。它配置了BlueNRG-LP ST 设备以发送用于位置跟踪的 Quuppa 测向 (DF) 数据包和用于数据传输的 Quuppa 数据包。

采用Quuppa智能定位技术的系统可以准确检测和跟踪ST Quuppa标签仿真的位置以及接收和显示其传输的数据。

有关 Quuppa 仿真系统和相关功能的更多信息,请参阅 Quuppa 网站上的文档。

笔记:

为了评估 ST Quuppa 标签仿真,用户可以简单地验证正确的测向和数据包格式和内容,或者可以在特定的 Quuppa 定位系统中使用该解决方案。

也可以使用 Quuppa 开发套件。在这种情况下,用户可以参考 Quuppa 开发套件用户手册,了解有关规划、安装、部署和配置带有定位器的 Quuppa 定位系统以及跟踪 ST Quuppa 标签模拟的分步说明。

2.1 ST Quuppa标签模拟配置

ST Quuppa 标签仿真在标准蓝牙 LE 通道 37 上运 ,并可通过 Quuppa 定位引擎 (QPE) 定位系统进跟踪

由 Quuppa 定位系统提供。
它使用特定的公共地址作为 Quuppa 标签 ID,用户可以自定义该地址。ST Quuppa 开发者 ID 为 0x0081。

特定的 ST_Quuppa_Tag_Init() API 允许向 ST Quuppa 标签仿真库提供以下初始化参数:

- o ST Quuppa 标签模拟公共地址也用作标签 ID;o ST Quuppa 标签模拟设备名称;
- o ST Quuppa 标签仿真配置文件编号,用于在支持的配置文件范围内选择配置文件编号;
- o ST Quuppa 标签模拟 DF 包设备类型;
- o ST Quuppa 标签仿真开发人员特定有效载荷数据用于
  通用开发人员特定的有效载荷数据响应;
- o ST Quuppa 标签模拟不支持的响应有效负载数据用于不支持的响应有效负载。

用户可以通过st_quuppa_tag_main.h头文件根据允许的值自定义这些参数。

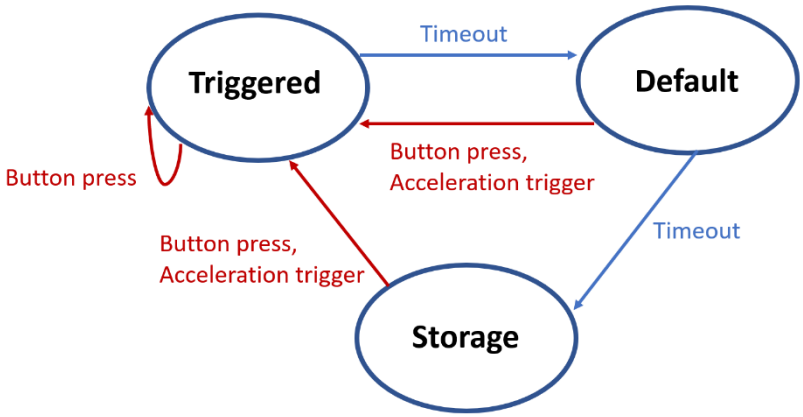
必须在初始化阶段使用所选参数调用 ST_Quuppa_Tag_Init() API。用户必须在主 while{1} 循环中调用相关的 ST Quuppa 标签模拟状态机 API ST_Quuppa_Tag_SM()。

预定义的支持配置文件定义了以下配置,这些配置会影响设备状态机的工作方式和某些特定的设备 RF 功能。

The ST Quuppa tag emulation library implements the following general states defined on Quuppa Emulation Specification:

- Triggered
- Default
- Storage

Figure 1 ST Quuppa tag emulation state machine



On power up, ST Quuppa tag emulation starts according to the start state defined by the specific selected profile (refer to Section 2.2 ST Quuppa tag emulation profiles). The following states and related transitions are supported:

- Storage state with no advertising and low power mode: device kit button (PUSH1) or motion detection through device kit accelerometer allow to move to Triggered state.
- Triggered state: device has an increased direction finding advertising rate in order to improve quality on detecting position of moving tags. After a specific timeout, device moves to Default state.
- Default State: device has a reduced advertising rate which allows to track position of a stationary tag. Device kit button (PUSH1) or motion detection through device kit accelerometer allow to move to Triggered state if supported by the profile. After a specific timeout, device moves to Storage state, if this state is supported by the current profile.

NOTES:

1. Storage and Triggered states are not supported by all profiles.

2.2 ST Quuppa tag emulation profiles

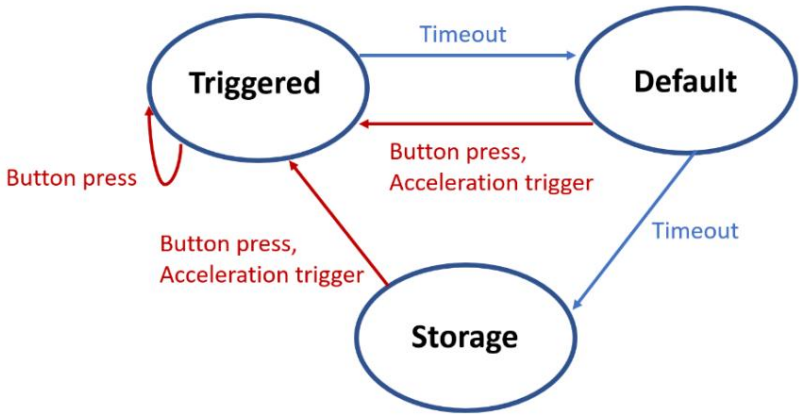
The ST Quuppa tag emulation supports the following Quuppa profiles defined on Tag Emulation Pro Tag profiles specification:

1. ASSET TAG
2. ID BADGE
3. ID BADGE FAST ALARM
4. FORKLIFT/VEHICLE
5. FORKLIFT/VEHICLE FAST RX RATE

ST Quuppa 标签模拟库实现了 Quuppa 模拟规范上定义的以下一般状态：

- o 触发
- o 默认
- o 储存

图1 ST Quuppa标签模拟状态机



通电后，ST Quuppa 标签仿真将根据特定选定配置文件定义的起始状态启动（请参阅第 2.2 节 ST Quuppa 标签仿真配置文件）。支持以下状态和相关转换：

- o 无广告和低功耗模式的存储状态：设备套件按钮（PUSH1）或通过设备套件加速度计的运动检测允许移动到触发状态。
- o 触发状态：设备具有增加的测向广告率，以提高检测移动标签位置的质量。

  经过特定超时后，设备将进入默认状态。
- o 默认状态：设备具有较低的广告费率，允许跟踪固定标签的位置。如果配置文件支持，设备套件按钮 (PUSH1) 或通过设备套件加速度计进的运动检测允许移动到触发状态。在特定超时后，设备移动到

  转换为存储状态（如果当前配置文件支持此状态）。

笔记：

1. 并非所有配置文件都支持存储和触发状态。

2.2 ST Quuppa 标签仿真配置文件

ST Quuppa 标签仿真支持以下 Quuppa 配置文件，定义在
Tag Emulation Pro 标签配置文件规格：

1. 资产标签
2. 身份证
3. ID徽章快速报警
4. 叉车/车辆
5. 叉车/车辆快速接收率

6. DEMO TAG
7. POWERSAVE

At tag initialization time, a specific profile can be selected, through the associated profile number initialization parameter, by using one of the following supported values:

- ST_QUUPPA_TAG_PROFILE_ASSET_TAG_NUM
- ST_QUUPPA_TAG_PROFILE_ID_BADGE_NUM
- ST_QUUPPA_TAG_PROFILE_ID_BADGE_FAST_ALARM_NUM
- ST_QUUPPA_TAG_PROFILE_FORKLIFT_VEHICLE_NUM
- ST_QUUPPA_TAG_PROFILE_FORKLIFT_VEHICLE_FAST_ALARM_NUM
- ST_QUUPPA_TAG_PROFILE_DEMO_TAG_NUM
- ST_QUUPPA_TAG_PROFILE_POWERSAVE_NUM

ST_QUUPPA_TAG_PROFILE_ID_BADGE_NUM is the default profile number selected on `st_quuppa_tag_main.h` file.

A profile can be also set at runtime through a specific custom Developer Specific request packet allowing to specify one of the predefined supported profiles (refer to Section 3.8 Developer Specific Set Profile Number request/response example).

The following configurations setting are used based on the selected profile, and they determine the tag behavior:

- DF packet TX rate (advertising interval)
- DF RX ON packet rate (when the specific RX_ON bit is set to 1 on DF packet)
- TX power used for transmitting packets
- Device Info packet TX rate (advertising interval) used for periodically transmitting such packets
- State transition timeout for moving from a state to another according to the device state machine transition rules.

Table 1 ASSET TAG profile configuration parameters

	Triggered	Default	Storage
DF packet TX rate	3 HZ	0.1 Hz	NA
DF RX ON packet rate	0.5 HZ	0.2 Hz	NA
TX power rate	0 dBm	0 dBm	NA
Device Info Packet TX rate	0.5 Hz	0.1 Hz	NA
State transition timeout	20 sec	NA	NA

NA: Not applicable

6. 演示标签
7. 省电

在标签初始化时,可以通过关联的配置文件编号初始化参数使用以下支持的值之一来选择特定的配置文件:

o ST_QUUPPA_TAG_PROFILE_ASSET_TAG_NUM o
ST_QUUPPA_TAG_PROFILE_ID_BADGE_NUM o
ST_QUUPPA_TAG_PROFILE_ID_BADGE_FAST_ALARM_NUM o
ST_QUUPPA_TAG_PROFILE_FORKLIFT_VEHICLE_NUM o
ST_QUUPPA_TAG_PROFILE_FORKLIFT_VEHICLE_FAST_ALARM_
编号
ST_QUUPPA_TAG_PROFILE_DEMO_TAG_NUM
ST_QUUPPA_TAG_PROFILE_POWERSAVE_NUM

ST_QUUPPA_TAG_PROFILE_ID_BADGE_NUM 是在 st_quuppa_tag_main.h 文件上选择的默认配置文件编号。

还可以在运时通过特定的自定义开发人员特定请求包来设置配置文件,以指定预定义的支持配置文件之一(请参阅第 3.8 节开发人员特定设置配置文件编号请求/响应示例)。

根据所选配置文件使用以下配置设置,它们决定标签为:o DF 数据包 TX 速率(广告间隔)

o DF RX ON 数据包速率(当 DF 上的特定 RX_ON 位设置为 1 时)
数据包)o
用于传输数据包的 TX 功率
o 设备信息包 TX 速率(广告间隔)用于定期
传输此类数据包
o 根据以下条件从一个状态转换到另一个状态的状态转换超时:
设备状态机转换规则。

表1 ASSET TAG配置文件配置参数

	已触发	默认	贮存
DF 数据包 发送速率	3赫兹	0.1 赫兹	那
DF RX ON 数 据包速率	0.5赫兹	0.2 赫兹	那
发射功率	0 分贝	0 分贝	那
设备信息 数据包发送速 率	0.5 赫兹	0.1 赫兹	那
状态转 换超时	20 秒	那	那

NA:不适用

Table 2 ID BADGE profile configuration parameters

	Triggered	Default	Storage
DF packet TX rate	4 Hz	0.1 Hz	OFF
DF RX ON packet rate	0.5 Hz	0.1 Hz	OFF
TX power rate	0 dBm	0 dBm	OFF
Device Info Packet TX rate	0.5 Hz	0.1 Hz	OFF
State transition timeout	20 sec	2 hours	NA

NA: not applicable

Table 3 ID BADGE FAST ALARM profile configuration parameters

	Triggered	Default	Storage
DF packet TX rate	4 Hz	0.1 Hz	OFF
DF RX ON packet rate	4 Hz	0.1 Hz	OFF
TX power rate	0 dBm	0 dBm	OFF
Device Info Packet TX rate	0,5 Hz	0.1 Hz	OFF
State transition timeout	20 sec	2 hours	NA

NA: not applicable

Table 4 FORKLIFT/VEHICLE profile configuration parameters

	Triggered	Default	Storage
DF packet TX rate	9 Hz	0.1 Hz	OFF
DF RX ON packet rate	0.5 Hz	0.1 Hz	OFF
TX power rate	0 dBm	0 dBm	OFF
Device Info Packet TX rate	0.5 Hz	0.1 Hz	OFF
State transition timeout	20 sec	2 hours	NA

NA: not applicable

表2 ID BADGE 配置文件配置参数

	已触发	默认	贮存
DF 数据包 发送速率	4赫兹	0.1 赫兹	离开
DF RX ON 数 据包速率	0.5 赫兹	0.1 赫兹	离开
发射功率	0 分贝	0 分贝	离开
设备信息 数据包发送速 率	0.5 赫兹	0.1 赫兹	离开
状态转 换超时	20 秒	2小时	那

NA:不适用

表 3 ID BADGE FAST ALARM 配置文件配置参数

	已触发	默认	贮存
DF 数据包 发送速率	4赫兹	0.1 赫兹	离开
DF RX ON 数 据包速率	4赫兹	0.1 赫兹	离开
发射功率	0 分贝	0 分贝	离开
设备信息 数据包发送速 率	0.5赫兹	0.1 赫兹	离开
状态转 换超时	20 秒	2小时	那

NA:不适用

表 4 FORKLIFT/VEHICLE 配置文件配置参数

	已触发	默认	贮存
DF 数据包 发送速率	9赫兹	0.1 赫兹	离开
DF RX ON 数 据包速率	0.5 赫兹	0.1 赫兹	离开
发射功率	0 分贝	0 分贝	离开
设备信息 数据包发送速 率	0.5 赫兹	0.1 赫兹	离开
状态转 换超时	20 秒	2小时	那

NA:不适用

Table 5 FORKLIFT/VEHICLE FAST ALARM profile configuration parameters

	Triggered	Default	Storage
DF packet TX rate	9 HZ	0.1 Hz	OFF
DF RX ON packet rate	5 Hz	0.1 Hz	OFF
TX power rate	0 dBm	0 dBm	OFF
Device Info Packet TX rate	0.5 Hz	0.1 HZ	OFF
State transition timeout	20 sec	2 hours	NA

NA: not applicable

Table 6 DEMO TAG profile configuration parameters

	Triggered	Default	Storage
DF packet TX rate	9 Hz	0.2 Hz	NA
DF RX ON packet rate	5 Hz	02 Hz	NA
TX power rate	0 dBm	0 dBm	NA
Device Info Packet TX rate	0.5 Hz	0.1 Hz	NA
State transition timeout	4 sec	NA	NA

NA: not applicable

Table 7 POWERSAVE profile configuration parameters

	Triggered	Default	Storage
DF packet TX rate	NA	0.1 Hz	NA
DF RX ON packet rate	NA	0.1 Hz	NA
TX power rate	NA	0 dBm	NA
Device Info Packet TX rate	NA	0.1 Hz	NA
State transition timeout	NA	NA	NA

NA: not applicable

表 5 叉车/车辆快速报警配置文件配置参数

	已触发	默认	贮存
DF 数据包 发送速率	9赫兹	0.1 赫兹	离开
DF RX ON 数 据包速率	5赫兹	0.1 赫兹	离开
发射功率	0 分贝	0 分贝	离开
设备信息 数据包发送速 率	0.5 赫兹	0.1赫兹	离开
状态转 换超时	20 秒	2小时	那

NA:不适用

表6 DEMO TAG配置文件配置参数

	已触发	默认	贮存
DF 数据包 发送速率	9赫兹	0.2 赫兹	那
DF RX ON 数 据包速率	5赫兹	02 赫兹	那
发射功率	0 分贝	0 分贝	那
设备信息 数据包发送速 率	0.5 赫兹	0.1 赫兹	那
状态转 换超时	4 秒	那	那

NA:不适用

表 7 POWERSAVE 配置文件配置参数

	已触发	默认	贮存
DF 数据包 发送速率	那	0.1 赫兹	那
DF RX ON 数 据包速率	那	0.1 赫兹	那
发射功率	那	0 分贝	那
设备信息 数据包发送速 率	那	0.1 赫兹	那
状态转 换超时	那	那	那

NA:不适用

2.3 ST Quuppa tag emulation packets

ST Quuppa tag emulation supports the following advertising packets:

- Quuppa Direction Finding (DF) packets used for position tracking
- Quuppa Data packets which are used for data transfer.

Quuppa Direction Finding packet is a standard non-connectable Bluetooth Low Energy manufacturing advertising packet with specific information on packet payload as follow:

Figure 2 Direction Finding payload data

LSB MSB

Company ID (2 bytes)	Packet ID (1 byte)	Device Type (1 byte)	Header (1 byte)	Tag ID (1 byte)	Checksum (1 byte)	DF field (14 bytes)
0x00C7	0x01					

The header field allows to define some important parameters (TX power and TX rate) impacting the position detection and the interaction with Quuppa positioning system (Back Channel support and RX ON field)

Figure 3 Direction Finding payload header field

LSB MSB

TX Rate (2 bits)	TX Power (2 bits)	Tag ID Type (2 bits)	Back Channel Support (1bit)	RX ON (1 bit)

Quuppa Data packet is a standard non-connectable Bluetooth Low Energy manufacturing advertising packet with specific information on packet payload as follow:

Figure 4 Data packet payload

LSB MSB

Company ID (2 bytes)	Packet ID (1 byte)	Header (1 byte)	Tag ID (6 bytes)	Payload (16 Bytes)	
				Payload Type (1 byte)	Payload Data (15 bytes)
0x00C7	0xF0				

There are 2 types of Quuppa Data packet payloads: Device Info and Developer Specific.

These packets are exchanged through the Quuppa Tag Back Channel mechanism which uses the Quuppa Request (REQ) non-connectable advertising packets to carry data from Quuppa Locators to Quuppa tags and Quuppa Response (RSP) non-connectable advertising packets to carry response data from Quuppa Tags to Quuppa Locators (refer to Section 3 ST & Quuppa Back channel).

Each time the device sent a DF packet with RX_ON = 1 on header field, it scans for possible REQ packet from locator and then it replies with a RESP packet.

2.3 ST Quuppa 标签模拟包ST Quuppa 标签模拟支持以下广告包：

- Quuppa 测向 (DF) 数据包用于位置跟踪
- Quuppa 数据包,用于数据传输。

Quuppa 测向包是标准的非连接蓝牙低
能源制造广告包,其数据包负载具体信息如下：

图 2 测向载荷数据

最低有效位 最高有效位

公司 ID (2字节)	包 ID (1字节)	设备 类型 (1字节)	标头（1 个 字节）	标签 ID (1字节)	校验和（1字节）	DF 字段 (14字节)
0x00C7	0x01					

标头字段允许定义一些影响位置检测和与 Quuppa 定位交互的重要参数（TX 功率和 TX 速率）

系统（反向通道支持和 RX ON 字段）

图 3 测向载荷头字段

最低有效位 最高有效位

发送速率 (2 位)	发射功率 (2 位)	标签 ID 类型 (2 位)	反向通道支持 (1bit)	接收开启 (1 位)

Quuppa 数据包是标准的非可连接蓝牙低功耗制造广告包,其数据包有效负载的具体信息如下：

图4 数据包载荷

最低有效位 最高有效位

公司 ID (2字节)	数据包 ID (1字节)	标头（1 个 字节）	标签 ID (6字节)	有效载荷 (16字节)	
				有效载荷类型 (1字节)	有效载荷数据 (15字节)
0x00C7	0xF0				

Quuppa 数据包有效负载有两种类型:设备信息和开发人员特定。

这些数据包通过 Quuppa 标签反向通道机制进交换,该机制使用 Quuppa 请求（REQ)不可连接广告数据包将数据从 Quuppa 定位器传送到 Quuppa 标签,并使用 Quuppa 响应（RSP)不可连接广告数据包将响应数据从 Quuppa 标签传送到 Quuppa 定位器（参见第 3 节 ST 和 Quuppa 反向通道）。

每次设备发送头字段上带有 RX_ON = 1 的 DF 数据包时,它都会扫描来自定位器的可能的 REQ 数据包,然后使用 RESP 数据包进回复。

The following REQ/RESP packets are supported from ST Quuppa tag emulation:

1) Device Info REQ/RESP payloads which allow to request information of the ST Quuppa tag emulation
2) Back Channel Info REQ/RSP payloads which allow to request information related to the back channel of the ST Quuppa tag emulation
3) Developer Specific REQ/ RESP payloads which allow to request information of the ST Quuppa tag emulation.
    a. A generic Developer Specific REQ/ RESP framework is provided with default preconfigured date sent as developer data.
    b. A 'Set Profile Number" ST custom Developer Specific REQ/ RESP framework is provided in order to configure the specific profile number.
4) Unsupported Request RSP payload sent when Quuppa Tag receives a REQ packet that it does not support.

Device Info REQ/RESP payloads

Table 8 Device Info REQ payload data format

Byte	Field	Description
0	REQ Type	Device Info. Value=0x01
1	RFU	Value=0x00
2	RFU	Value=0x00
3	RFU	Value=0x00
4	RFU	Value=0x00
5	RFU	Value=0x00
6	RFU	Value=0x00
7	RFU	Value=0x00
8	RFU	Value=0x00
9	RFU	Value=0x00
10	RFU	Value=0x00
11	RFU	Value=0x00
12	RFU	Value=0x00
13	RFU	Value=0x00
14	RFU	Value=0x00
15	RFU	Value=0x00d

Table 9 Device Info RSP payload data format

Byte	Field	Description
0	RSP Type	Device Info. Value=0x01
1	Version	Device Info version. Value =0x00
2	Supported features	
3	Status	
4	Battery Voltage (LSB)	Battery voltage, 1 mV/bit
5	Battery Voltage (MSB)	
6	Temperature (LSB)	Temperature of the Quuppa

ST Quuppa 标签模拟支持以下 REQ/RESP 数据包：

1)设备信息 REQ/RESP 有效载荷,允许请求 ST 的信息
Quuppa 标签模拟

2)反向通道信息 REQ/RSP 有效载荷,允许请求与 ST Quuppa 标签模拟反向通道相关的信息

3)开发者特定的 REQ/RESP 有效载荷,允许请求以下信息
ST Quuppa 标签模拟。

a. 提供通用的开发者专用 REQ/RESP 框架
默认预配置日期作为开发人员数据发送。

b. “设置配置文件编号” ST 自定义开发人员特定 REQ/ RESP
提供框架以便配置特定的配置文件号码。

4)Quuppa Tag 收到 REQ 时发送了不支持的请求 RSP 有效负载
它不支持的数据包。

设备信息 REQ/RESP 有效载荷

表 8 设备信息 REQ 载荷数据格式

字节	场地	描述
0	请求类型	设备信息值=0x01
1	拉夫联	值=0x00
2	拉夫联	值=0x00
3	拉夫联	值=0x00
4	拉夫联	值=0x00
5	拉夫联	值=0x00
6	拉夫联	值=0x00
7	拉夫联	值=0x00
8	拉夫联	值=0x00
9	拉夫联	值=0x00
10	拉夫联	值=0x00
11	拉夫联	值=0x00
12	拉夫联	值=0x00
十三	拉夫联	值=0x00
14	拉夫联	值=0x00
15	拉夫联	值=0x00d

表 9 设备信息 RSP 有效载荷数据格式

字节	场地	描述
0	RSP 类型	设备信息值=0x01
1	版本	设备信息版本。 值 =0x00
2	支持的功能	
3	地位	
4	电池电压 (LSB)	电池电压,1 mV/位
5	电池电压 (MSB)	
6	温度（LSB）	Quuppa 的温度

Byte	Field	Description
		Tag, 0.1 K/bit
7	Temperature (MSB)	
8	Developer ID (LSB)	Developer ID (0x81)
9	Developer ID (MSB)	Develop ID (0x00)
10	FW version (LSB)	minor
11	FW version (MSB)	major
12	HW version (LSB)	minor
13	HW version (MSB)	major
14	Pressure (LSB)	Air pressure, 1 Pa/bit and offset 50000 Pa
15	Pressure (MSB)	

Supported Features is used to inform about general HW features of ST Quuppa tag emulation as follow:

Table 10 Device Info RSP payload data: supported features field

Bit number	Field	Description
0	Battery Alarm ‘	0’=not supported, ‘1’ =supported
1	Button 1	0’=not supported, ‘1’ =supported
2	Button 2	0’=not supported, ‘1’ =supported
3	Batt. Voltage Meter	0’=not supported, ‘1’ =supported
4	Temperature Meter	0’=not supported, ‘1’ =supported
5	Pressure Meter	0’=not supported, ‘1’ =supported
6	RFU	0
7	RFU	0

Table 11 Device Info RSP payload data: status field

Bit number	Field	Description
0	Battery Alarm ‘	‘0’=battery OK, ‘1’ =Battery low
1	Button 1	‘0’ =not pressed, ‘1’=pressed
2	Button 2	‘0’ =not pressed, ‘1’=pressed
3	RFU	0
4	RFU	0
5	RFU	0

字节	场地	描述
		天,0.1K/位
7	温度（最高有效位）	
8	开发者 ID (LSB)	开发者 ID（0x81）
9	开发者 ID (MSB)	开发ID（0x00）
10	固件版本 (LSB)	次要的
11	固件版本 (MSB)	主要的
12	硬件版本 (LSB)	次要的
十三	硬件版本 (MSB)	主要的
14	压力（LSB）	气压,1 Pa/位,偏移50000 Pa
15	压力（最高有效位）	

支持的功能用于告知 ST Quuppa 标签仿真的一般硬件功能如下：

表 10 设备信息 RSP 有效载荷数据:支持的功能字段

位数	场地	描述
0	电池警报	0 =不支持， 1 = 支持
1	按钮 1	0 =不支持， 1 = 支持
2	按钮 2	0 =不支持， 1 = 支持
3	电池电压表	0 =不支持，‘1’ = 支持
4	温度计	0 =不支持， 1 = 支持
5	压力表	0 =不支持， 1 = 支持
6	拉夫联	0
7	拉夫联	0

表 11 设备信息 RSP 有效载荷数据:状态字段

位数	场地	描述
0	电池警报	‘0’ =电池正常，‘1’ = 电池电量低
1	按钮 1	‘0’ = 未按下， ‘1’ = 按下
2	按钮 2	‘0’ = 未按下， ‘1’ = 按下
3	拉夫联	0
4	拉夫联	0
5	拉夫联	0

Bit number	Field	Description
6	RFU	0
7	RFU	0

Back Channel Info REQ/RSP payloads format

Table 12 Back Channel Info REQ payload data format

Byte	Field	Description
0	REQ Type	Back Channel Information. Value=0x02
1	RFU	Value=0x00
2	RFU	Value=0x00
3	RFU	Value=0x00
4	RFU	Value=0x00
5	RFU	Value=0x00
6	RFU	Value=0x00
7	RFU	Value=0x00
8	RFU	Value=0x00
9	RFU	Value=0x00
10	RFU	Value=0x00
11	RFU	Value=0x00
12	RFU	Value=0x00
13	RFU	Value=0x00
14	RFU	Value=0x00
15	RFU	Value=0x00d

Table 13 Back Channel Info RSP payload data format

Byte	Field	Description
0	RSP Type	Back Channel Information. Value=0x02
1	Back channel Spec number Minor	Minor
2	Back channel Spec number Minor	Major
3	Developer ID (LSB)	Developer ID (0x81)
4	Developer ID (MSB)	Develop ID (0x00)
5	RFU	Value=0x00
6	RFU	Value=0x00
7	RFU	Value=0x00
8	RFU	Value=0x00
9	RFU	Value=0x00
10	RFU	Value=0x00
11	RFU	Value=0x00

位数	场地	描述
6	拉夫联	0
7	拉夫联	0

反向通道信息 REQ/RSP 有效载荷格式

表 12 反向通道信息 REQ 有效载荷数据格式

字节	场地	描述
0	请求类型	反向通道信息。 值=0x02
1	拉夫联	值=0x00
2	拉夫联	值=0x00
3	拉夫联	值=0x00
4	拉夫联	值=0x00
5	拉夫联	值=0x00
6	拉夫联	值=0x00
7	拉夫联	值=0x00
8	拉夫联	值=0x00
9	拉夫联	值=0x00
10	拉夫联	值=0x00
11	拉夫联	值=0x00
12	拉夫联	值=0x00
十三	拉夫联	值=0x00
14	拉夫联	值=0x00
15	拉夫联	值=0x00d

表 13 反向通道信息 RSP 有效载荷数据格式

字节	场地	描述
0	RSP 类型	反向通道信息。 值=0x02
1	后通道规格编号 次要的	次要的
2	后通道规格编号 次要的	主要的
3	开发者 ID (LSB)	开发者 ID（0x81）
4	开发者 ID (MSB)	开发ID（0x00）
5	拉夫联	值=0x00
6	拉夫联	值=0x00
7	拉夫联	值=0x00
8	拉夫联	值=0x00
9	拉夫联	值=0x00
10	拉夫联	值=0x00
11	拉夫联	值=0x00

Byte	Field	Description
12	RFU	Value=0x00
13	RFU	Value=0x00
14	RFU	Value=0x00
15	RFU	Value=0x00

Unsupported Request RSP payload

Table 14 Unsupported Request RSP payload data format

Byte	Field	Description
0	RSP type	Unsupported REQ. Value=0x00
1	Back channel Spec number	Minor
2	Back channel Spec number	Major
3	Developer ID (LSB)	Developer ID (0x81)
4	Developer ID (MSB)	Developer ID (0x00)
5	DEVELOPER DATA 1	Developer specific data
6	DEVELOPER DATA 2	Developer specific data
7	DEVELOPER DATA 3	Developer specific data
8	DEVELOPER DATA 4	Developer specific data
9	DEVELOPER DATA 5	Developer specific data
10	DEVELOPER DATA6	Developer specific data
11	DEVELOPER DATA 7	Developer specific data
12	DEVELOPER DATA 8	Developer specific data
13	DEVELOPER DATA 9	Developer specific data
14	DEVELOPER DATA 10	Developer specific data
15	DEVELOPER DATA 11	Developer specific data

Developer Specific REQ/RSP payload format

Table 15 Developer Specific REQ/RSP payloads data format

Byte	Field	Description
0	REQ/RSP Payload Type	Developer Specific. Value = 0xFF
1	Developer ID (LSB)	Developer ID (0x81)
2	Developer ID (MSB)	Developer ID (0x00)
3	DEVELOPER DATA 0	Developer specific data (Default value)
4	DEVELOPER DATA 1	‘’
5	DEVELOPER DATA 2	‘’
6	DEVELOPER DATA 3	‘’
7	DEVELOPER DATA 4	‘’

字节	场地	描述
12	拉夫联	值=0x00
十三	拉夫联	值=0x00
14	拉夫联	值=0x00
15	拉夫联	值=0x00

不支持的请求 RSP 负载

表 14 不支持的请求 RSP 有效载荷数据格式

字节	场地	描述
0	RSP 类型	不支持的 REQ。 值=0x00
1	后通道规格编号	次要的
2	后通道规格编号	主要的
3	开发者 ID (LSB)	开发者 ID（0x81）
4	开发者 ID (MSB)	开发者 ID (0x00)
5	开发者数据 1	开发人员特定数据
6	开发者数据 2	开发人员特定数据
7	开发者数据 3	开发人员特定数据
8	开发者数据 4	开发人员特定数据
9	开发者数据 5	开发人员特定数据
10	开发者数据6	开发人员特定数据
11	开发者数据 7	开发人员特定数据
12	开发者数据 8	开发人员特定数据
十三	开发者数据 9	开发人员特定数据
14	开发者数据 10	开发人员特定数据
15	开发者数据 11	开发人员特定数据

开发人员特定的 REQ/RSP 有效负载格式

表 15 开发人员特定的 REQ/RSP 有效载荷数据格式

字节	场地	描述
0	REQ/RSP 有效载荷类型	特定于开发人员。 值 = 0xFF
1	开发者 ID (LSB)	开发者 ID（0x81）
2	开发者 ID (MSB)	开发者 ID (0x00)
3	开发者数据 0	开发人员特定数据 （默认值）
4	开发者数据 1	“”
5	开发者数据 2	“”
6	开发者数据 3	“”
7	开发者数据 4	“”

Byte	Field	Description
8	DEVELOPER DATA 5	‘’
9	DEVELOPER DATA 6	‘’
10	DEVELOPER DATA 7	‘’
11	DEVELOPER DATA 8	‘’
12	DEVELOPER DATA 9	‘’
13	DEVELOPER DATA 10	‘’
14	DEVELOPER DATA 11	‘’
15	DEVELOPER DATA 12	‘’

Custom ST Quuppa tag emulation Developer Specific “Set Profile Number” REQ and RSP payloads allow to specify, at run-time, which profile has to be configured within the list of predefined supported profiles:

Table 16 Developer Specific set profile number REQ/ RESP

Byte	Field	Description
0	REQ/RSP Type	Developer Specific. Value = 0xFF
1	Developer ID (LSB)	Developer ID (0x81)
2	Developer ID (MSB)	Developer ID (0x00)
3	ST Set Profile Number REQ	0xAD (ST tag custom id for requesting a specific profile set within the supported preconfigured profiles)
4	ST Profile Number	Value in the range of supported profiles [1-7]
5	DEVELOPER DATA 2	Default value (not used)
6	DEVELOPER DATA 3	‘’
7	DEVELOPER DATA 4	‘’
8	DEVELOPER DATA 5	‘’
9	DEVELOPER DATA 6	‘’
10	DEVELOPER DATA 7	‘’
11	DEVELOPER DATA 8	‘’
12	DEVELOPER DATA 9	‘’
13	DEVELOPER DATA 10	‘’
14	DEVELOPER DATA 11	‘’
15	DEVELOPER DATA 12	‘’

3 ST & Quuppa Back Channel

Quuppa Tag Back Channel is a mechanism that allows the user of the Quuppa Intelligent Locating Technology ™ to write/read data to/from Quuppa Tags using Quuppa Web APIs available through the Quuppa Positioning Engine (QPE) server.

字节	场地	描述
8	开发者数据 5	“”
9	开发者数据 6	“”
10	开发者数据 7	“”
11	开发者数据 8	“”
12	开发者数据 9	“”
+三	开发者数据 10	“”
14	开发者数据 11	“”
15	开发者数据 12	“”

自定义 ST Quuppa 标签模拟开发人员特定的“设置配置文件编号” REQ 和 RSP 有效负载允许在运行时指定必须在预定义支持配置文件列表中配置哪个配置文件：

表 16 开发人员特定设置配置文件编号 REQ/ RESP

字节	场地	描述
0	REQ/RSP 类型	特定于开发人员。 值 = 0xFF
1	开发者 ID (LSB)	开发者 ID（0x81）
2	开发者 ID (MSB)	开发者 ID (0x00)
3	ST 设置配置文件编号 REQ 0xAD（ST 标签自定义 ID	在支持的预配置配置文件中请求特定的配置文件集）
4	ST 轮廓编号	受支持的配置文件范围内的 值 [1-7]
5	开发者数据 2	默认值（未使用）
6	开发者数据 3	“”
7	开发者数据 4	“”
8	开发者数据 5	“”
9	开发者数据 6	“”
10	开发者数据 7	“”
11	开发者数据 8	“”
12	开发者数据 9	“”
+三	开发者数据 10	“”
14	开发者数据 11	“”
15	开发者数据 12	“”

3 ST 和 Quuppa 反向通道

Quuppa 标签反向通道是一种允许 Quuppa 用户
智能定位技术™ 使用 Quuppa 标签写入/读取数据
可通过 Quuppa 定位引擎 (QPE) 访问 Quuppa Web API
服务器。

The Quuppa Web APIs provide a mechanism for getting several data of a project and/or Tag using HTTP requests. The requested data is returned in an easy-to-handle JSON format.

The mechanism uses then Quuppa Request (REQ) packets to carry data from Quuppa locators to Quuppa tags and Quuppa Response (RSP) packets to carry response data from Quuppa Tags to Quuppa Locators.

In particular, the Quuppa Web APIs support the Send Back Channel Request to Tag API (sendQuuppaRequest) and Retrieve Tag Replies to Back Channel Requests API (getQuuppaRequestResponse) for interacting with the specific tag.

3.1 sendQuuppaRequest Back Channel API

sendQuuppaRequest API allows to send Back Channel request to tag(s). It requires Quuppa back channel enabled.

Request packet is as follow:

/qpe/sendQuuppaRequest?tag=<tag_1,...,tag_n>&requestData=<requestBytes>[&time=timeOutMs][&humanReadable]

Request packet parameters:

`Tag`: it defines the tag id(s) to which the request is sent [Required field].

`requestData`: it defines the data payload of the request (between 1 and 16 bytes). It can be in hex format ('0x00 0x0c' or '0x000c') or in decimal format ('0 10') [Required field].

`time`: it defines how long (in ms) the system tries to command the tag (default 60000) [Optional field].

`humanReadable`: if set to true ('&humanReadable=true'), the output is formatted (indented and line-wrapped) for easier reading [Optional field].

Response is a JSON object containing the following fields in addition to base status parameters defined on Section 3.3 Base response format back channel.

Table 17 JSON object response: JSON tags array

Key	Data Type	Value
tags	JSON array	Array containing information about each of the tags to which the request was sent

Table 18 JSON object response: Tags JSON object

Key	Data type	Value
id	String	Id of the tag
Name	String	Name of the tag
sequenceNumber	Number	Running number between 0 and 15 that can be used to match the request to replies

Quuppa Web API 提供了一种使用 HTTP 请求获取项目和/或标签的多项数据的机制。请求的数据以易于处理的 JSON 格式返回。

该机制使用 Quuppa 请求（REQ)数据包将数据从 Quuppa 定位器传送到 Quuppa 标签,并使用 Quuppa 响应（RSP)数据包将响应数据从 Quuppa 标签传送到 Quuppa 定位器。

具体来说,Quuppa Web API 支持将反向通道请求发送到标签 API（sendQuuppaRequest)和检索反向通道请求的标签回复 API（getQuuppaRequestResponse）,以便与特定标签进交互。

3.1 sendQuuppaRequest 反向通道 API

sendQuuppaRequest API 允许向标签发送反向通道请求。它需要启用 Quuppa 反向通道。

请求数据包如下：/qpe/
sendQuuppaRequest?tag=<tag_1,...,tag_n>&requestData=<requestBytes>[&time=timeOutMs][&humanReadable]

请求包参数：

标签:定义发送请求的标签 ID [必填字段]。
requestData:定义请求的数据有效负载（1 到 16 个字节之间）。它可以是十六进制格式（“0x00 0x0c”或“0x000c”)或十进制格式（“0 10”）。
[必填字段]。
时间:它定义系统尝试命令标签的时间（以毫秒为单位）(默认
60000)[可选字段]。
humanReadable:如果设置为 true（ &humanReadable=true ）,则输出将被格式化（缩进和换)以便于阅读[可选字段]。

响应是一个 JSON 对象,除了第 3.3 节基本响应格式反向通道中定义的基本状态参数外,还包含以下字段。

表 17 JSON 对象响应:JSON 标签数组

关键	数据类型	价值
标签	JSON 数组	包含有关发送请求的每个标签的信息的数组

表 18 JSON 对象响应:标签 JSON 对象

密钥	数据类型	价值
ID	细绳	标签的 ID
姓名	细绳	标签名称
序列号	数字	0 到 15 之间的连续数字,可用于将请求与回复进匹配

Example

Request:

/qpe/sendQuuppaRequest?tag=010080e18002&requestData=0x01 0xaa 0x03&humanReadable=true

Response:


```
{
"code": 0,
"command": "/qpe/sendQuuppaRequest?tag=010080e18002&requestData=0x01 0xaa
                0x03&humanReadable=true",
"message": "Commanding 1 tag(s)",
"responseTS": 1446469174514,
"status": "Ok",
"tags": [{
   "id": "010080e18002",
   "name": null,
   "sequenceNumber": 10
}],
"version": "1.0"
}
```


3.2 getQuuppaRequestResponse Back Channel API

getQuuppaRequestResponseAPI allows to Retrieve tag replies to Back Channel requests. It requires Quuppa back channel module enabled.

Request packet is as follow:

`/qpe/getQuuppaRequestResponse?[tag=<tag_1,...,tag_n>][&humanReadable]`

Request packet parameters:

`Tag`: it defines the tag id(s) tag id(s) from which the responses are retrieved [Optional field].

`humanReadable`: if set to true ('&humanReadable=true'), the output is formatted (indented and line-wrapped) for easier reading [Optional field].

Response is a JSON object containing the following fields in addition to base status parameters defined on Section 3.3 Base response format back channel.

Table 19 JSON object response: JSON tags array

Key	Data type	Value
tags	JSON array	Array containing information about each of the tags to which the request was sent

例子

要求：

/qpe/sendQuuppaRequest?tag=010080e18002&requestData=0x01 0xaa 0x03&humanReadable=true

回复：

{
“代码”:0，
“命令”：“/qpe/sendQuuppaRequest?tag=010080e18002&requestData=0x01 0xaa
0x03&humanReadable=true”，
message ： 正在命令 1 个标签 ，
“响应TS”:1446469174514，
“状态”：“确定”，
“标签”:[{
“id”：“010080e18002”，
“名称”:空，
“序列号”:10
}]，
“版本”：“1.0”
}

3.2 getQuuppaRequestResponse 反向通道 API

getQuuppaRequestResponseAPI 允许检索对 Back Channel 请求的标签回复。
它需要启用 Quuppa 反向通道模块。

请求数据包如下：/qpe/
getQuuppaRequestResponse?[tag=<tag_1,...,tag_n>][&humanReadab
这]

请求包参数：

标签:它定义从中检索响应的标签 id [可选字段]。

humanReadable:如果设置为 true（ &humanReadable=true ），则输出将被格式化（缩进和换 ）以便于阅读[可选字段]。

响应是一个 JSON 对象,除了第 3.3 节基本响应格式反向通道中定义的基本状态参数外,还包含以下字段。

表 19 JSON 对象响应:JSON 标签数组

关键	数据类型	价值
标签	JSON 数组	包含有关发送请求的每个标签的信息的数组

Table 20 JSON object response: Tags JSON object

Key	Data type	Value
id	String	Id of the tag
Name	String	Name of the tag
sequenceNumber	Number	Running number between 0 and 15 that can be used to match the request to replies
replyTS	Number	Timestamp when the response was received
requestTS	Number	Timestamp when the request was sent to the tag
requestData	String	The original request payload data in hex format
replyData	String	The response payload data in hex format

Example

Request:
/qpe/getQuuppaRequestResponse?humanReadable&tag=010080e18002

Response:


```
{
"code": 0,
"command":
"http://localhost:8080/qpe/getQuuppaRequestResponse?humanReadable&tag=010080e1
  8002",
"message": "QuuppaRequests",
"responseTS": 1440409450479,
"status": "Ok",
"tags": [
{
"id": "010080e18002",
"status": "Reply Ok",
"name": "RequestTags_0002",
"quuppaRequests": [
{
"reply TS": 1440409419456,
"reply Data": "0x01",
"requestTS": 1440409418330,
"sequenceNumber": 0,
"requestData": "0x01"
},
{
"reply TS": 1440409432683,
"reply Data": "0x01",
"requestTS": 1440409432094,

"sequenceNumber": 1,
"requestData": "0x01"
},
{
"reply TS": 1440409447347,
"reply Data": "0x05",
"requestTS": 1440409446236,
"sequenceNumber": 2,
"requestData": "0x05"
},
```

表 20 JSON 对象响应:标签 JSON 对象

密钥	数据类型	价值
ID	细绳	标签的 ID
姓名	细绳	标签名称
序列号	数字	0 到 15 之间的连续数字,可用于将请求与回复进匹配
回复TS	数字	收到响应的时间戳
请求TS	数字	向标签发送请求的时间戳
请求数据	细绳	十六进制格式的原始请求有效负载数据
回复数据	细绳	十六进制格式的响应有效负载数据

例子

要求:

/qpe/getQuuppaRequestResponse?humanReadable&tag=010080e18002

回复:

{
“代码”:0,
“命令”:“http://
localhost:8080/qpe/getQuuppaRequestResponse?humanReadable&tag=010080e1
8002”,
“消息”:“QuuppaRequests”,
“响应TS”:1440409450479,
“状态”:“确定”,
标签 :[
{
“id”:“010080e18002”,
status : 回复确定 ,
“名称”:“RequestTags_0002”,
“quuppa请求”:[
{
“回复 TS”:1440409419456,
“回复数据”:“0x01”,
“请求TS”:1440409418330,
“序列号”:0,
“请求数据”:“0x01”
},
{
“回复 TS”:1440409432683,
“回复数据”:“0x01”,
“请求TS”:1440409432094,

“序列号”:1,
“请求数据”:“0x01”
},
{
“回复 TS”:1440409447347,
“回复数据”:“0x05”,
“请求TS”:1440409446236,
“序列号”:2,
“请求数据”:“0x05”
},

null,
null,
null,
null,
null,
null,
null,
null,
null,
null,
null,
null,
null
]
}
],
"version": "1.0"
}

3.3 Base status parameters response format

Base status parameters responses are defined as on following table:

Table 21 Base status parameters: response format

Param name	Param format	Param value
code	number	Status code, 0 if command successful, otherwise refer to Table 15 Base response parameters: status field
status	string	Status string. Refer to Table 15 Base response parameters: status field
message	string	Custom status message trying to explain error situations
version	string	version string Version of this response, in form of major, minor. Higher major versions are NOT backwards compatible to low er major versions, but higher minor
command	string	Command that was used to create this result
responseTS	number	Timestamp when this response was generated.

Table 22 Base status parameters: status code

Code	Status string
0	Ok
1	General Failure
2	Not Implemented
3	License Does Not Allow
4	Invalid Parameter
5	Required Parameter Missing

无效的，
无效的，
无效的，
无效的，
无效的，
无效的，
无效的，
无效的，
无效的，
无效的，
无效的，
无效的，
无效的
]
}
],
“版本”：“1.0”
}

3.3 底座状态参数响应格式

基础状态参数响应定义如下表：

表 21 基站状态参数:响应格式

参数名称	参数格式	参数值
代码	数字	状态代码,命令成功则为 0,否则参考表 15 基本响应参数:状态字段
地位	细绳	状态字符串。参考表15 基本响应参数:状态字段
信息	细绳	自定义状态消息试图解释错误情况
版本	细绳	版本字符串此响应的版本,以主要版本、次要版本的形式呈现。较高的主版本不向后兼容较低的主版本,但较高的次版本
命令	细绳	用于创建此结果的命令
响应	数字	生成此响应的时间戳。

表 22 基础状态参数:状态码

代码	状态字符串
0	好的
	一般故障
1	未实施
2	许可证不允许
3	无效参数
4 5	缺少必需参数

Code	Status string
6	Too Many Parameters
7	Object Not Found
8	Group Empty
9	Already Commanded
10	IO Failure
11	PENotStarted
12	Not Supported
13	Unknow nTag Listed
14	Unknow nLocator Listed
15 onwards	RFU

Example


```
{
  "code": 0,
  "command": "http://localhost:8080/qpe/commandTag?tag=010080e18002&id=superMarketActiv
   ation",
  "message": "Commanding 1 tag(s)",
  "responseTS": 1430142139012,
  "status": "Ok",
  "version": "1.0"
}
```


3.4 Device Info request/response example

Request for Device Info (request type value: 0x01)


```
{
  "code": 0,
  "command":
 "http://localhost:8080/qpe/sendQuuppaRequest?tag=010080e18002&requestData=0x01&human
   Readable=true",
  "message": "Commanding 1 tag(s)",
  "responseTS": 1642753883483,
  "status": "Ok",
  "tags": [{
    "sequenceNumber": 0,
    "name": "ST_Tag_BLE_New_version",
    "id": "010080e18002"
  }],
  "version": "1.0”
}
```


Response for Device Info (request type value: 0x01)


```
{
  "code": 0,
  "command":
  "http://localhost:8080/qpe/getQuuppaRequestResponse?humanReadable&tag=010080e18002",
```

代码6	状态字符串
7	参数过多
8	未找到对象
9	空组
	已下令
10	我失败了
11	未启动PE
12	不支持
13	列出未知 nTag
14	列出未知定位器
15 及以上	拉夫联

例子

```
{
  “代码”:0,
  “命令”：“http://localhost:8080/qpe/commandTag?tag=010080e18002&id=superMarketActivation”

   message ： 正在命令 1 个标签 ，
  “响应TS”:1430142139012,
  “状态”：“确定”，
  “版本”：“1.0”
}
```

3.4 设备信息请求/响应示例

请求设备信息（请求类型值:0x01）

```
{
  “代码”:0,
  “命令”：
 “http：//本地主机:8080 / qpe / sendQuuppaRequest?tag = 010080e18002&requestData = 0x01&human
  可读=true ，
   message ： 正在命令 1 个标签 ，
  “响应”:1642753883483,
  “状态”：“确定”，
  “标签”:[{
    “序列号”:0,
    “名称”：“ST_Tag_BLE_New_version”，
    “id”：“010080e18002”
  }],
  “版本”：“1.0”
}
```

设备信息响应（请求类型值:0x01）

```
{
  “代码”:0,
  “命令”：
  “http://localhost:8080/qpe/getQuuppaRequestResponse?humanReadable&tag=010080e18002”，
```

```
  "message": "QuuppaRequests",
  "responseTS": 1642753889840,
  "status": "Ok",
  "tags": [{
    "quuppaRequests": [
      {
        "sequenceNumber": 0,
        "requestTS": 1642753883483,
        "replyTS": 1642753888317,
        "replyData": "0x01 0x01 0x32 0x00 0x00 0x00 0xae 0x0b 0x81 0x00 0x00 0x01 0x00 0x01
                      0xb4 0xcb",
        "requestData": "0x01"
      },
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null
    ],
    "name": "ST_Tag_BLE_New_version",
    "id": "010080e18002",
    "status": "ReplyOk"
  }],
  "version": "1.0"
}
```


3.5 Back Channel Info request/response example

Request for Back Channel (request type value: 0x02)


```
{
  "code": 0,
  "command":
"http://localhost:8080/qpe/sendQuuppaRequest?tag=010080e18002&requestData=0x02&human
   Readable=true",
  "message": "Commanding 1 tag(s)",
```

“消息”：“QuuppaRequests”，“响应”：
1642753889840，“状态”：“Ok”，“标
签”：

[{“quuppaRequests”：[

{ “sequenceNumber”：0，
“requestTS”：1642753883483，
“replyTS”：1642753888317，
“replyData”：“0x01 0x01 0x32 0x00 0x00 0x00 0xae 0x0b 0x81 0x00 0x00 0x01 0x00 0x01
0xb4 0xcb ，
“requestData”： “0x01” }，

null，
null，
null，
null，
null，

无效的，
无效的，
无效的，
空，
空，
空，
空，
空，
无效的
],
“名称”：“ST_Tag_BLE_New_version”， “id”：
“010080e18002”， “状态”：
“ReplyOk” }]，“版本”：

“1.0”
}

3.5 反向通道信息请求/响应示例

请求反向通道（请求类型值：0x02）{

“代码”：0，
“命令”：
“http：//本地主机：8080 / qpe / sendQuuppaRequest?tag = 010080e18002&requestData = 0x02&human
可读=true ，
message ： 命令 1 个标签 ，

```
  "responseTS": 1642753923849,
  "status": "Ok",
  "tags": [{
    "sequenceNumber": 1,
    "name": "ST_Tag_BLE_New_version",
    "id": "010080e18002"
  }],
  "version": "1.0"
}
```


Response for Back Channel (request type value: 0x02)


```
{
  "code": 0,
  "command":
  "http://localhost:8080/qpe/getQuuppaRequestResponse?humanReadable&tag=010080e18002",
  "message": "QuuppaRequests",
  "responseTS": 1642753930684,
  "status": "Ok",
  "tags": [{
    "quuppaRequests": [
      ….,
      {
        "sequenceNumber": 1,
        "requestTS": 1642753923849,
        "replyTS": 1642753924339,
        "replyData": "0x02 0x01 0x01 0x81 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
                      0x00 0x00",
        "requestData": "0x02"
      },
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null
    ],
    "name": "ST_Tag_BLE_New_version",
```

“responseTS”:1642753923849，“状态”：
“Ok”，“标签”：

[{“sequenceNumber”:1，“名
称”：“ST_Tag_BLE_New_version”，
“id”：“010080e18002”
}]，
“版本”：“1.0”
}

反向通道的响应（请求类型值:0x02）{“code”:0，“command”：

“http://localhost:8080/qpe/getQuuppaRequestResponse?humanReadable&tag=010080e18002”，“消息”：“QuuppaRequests”，
“responseTS”:1642753930684，“状态”：
“Ok”，“标签”:[{“quuppaRequests”:[

…,

{ “sequenceNumber”:1，
“requestTS”:1642753923849，
“replyTS”:1642753924339，
“replyData”：“0x02 0x01 0x01 0x81 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00”， “requestData”：
“0x02” },空，

无效的，
空,空，
空,空，
空,空，
空,空，
空,空，

无效的，
无效的，
无效的，
无效的

]， 名称 ： ST_Tag_BLE_New_version ，

```
    "id": "010080e18002",
    "status": "ReplyOk"
  }],
  "version": "1.0"
}
```


3.6 Unsupported request/response example

Request for Unsupported (request type value: 0xAA as example)


```
{
  "code": 0,
  "command":
"http://localhost:8080/qpe/sendQuuppaRequest?tag=010080e18002&requestData=0xAA%200x81%200x00&humanReadable=true",
  "message": "Commanding 1 tag(s)",
  "responseTS": 1642754342912,
  "status": "Ok",
  "tags": [{
    "sequenceNumber": 5,
    "name": "ST_Tag_BLE_New_version",
    "id": "010080e18002"
  }],
  "version": "1.0"
}
```


Response for Unsupported (request type value: 0xAA as example)


```
{
  "code": 0,
  "command":
  "http://localhost:8080/qpe/getQuuppaRequestResponse?humanReadable&tag=010080e18002",
  "message": "QuuppaRequests",
  "responseTS": 1642754385215,
  "status": "Ok",
  "tags": [{
    "quuppaRequests": [
      ….,
      {
        "sequenceNumber": 5,
        "requestTS": 1642754342911,
        "replyTS": 1642754349333,
        "replyData": "0x00 0x01 0x01 0x81 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09
                      0x0a 0x0b",
        "requestData": "0xaa 0x81 0x00"
      },
      null,
      null,
```

“id”：“010080e18002”，“状
态”：“ReplyOk”}]，“版

本”：“1.0”
}

3.6 不支持的请求/响应示例

不支持的请求（请求类型值:0xAA为例）{

“代码”:0，
“命令”：
“http://localhost:8080/qpe/sendQuuppaRequest?tag=010080e18002&requestData=0xAA%200x8 1%200x00&humanReadable=true”，

“消息”：“命令 1 个标签”，“响应 TS”：
1642754342912，“状态”：“确定”，“标签”：

[{“sequenceNumber”:5，“名
称”：“ST_Tag_BLE_New_version”，
“id”：“010080e18002”
}]，
“版本”：“1.0”
}

不支持的响应（请求类型值:0xAA 作为示例）{“code”:0，“command”：

“http://localhost:8080/qpe/getQuuppaRequestResponse?humanReadable&tag=010080e18002”，“消息”：“QuuppaRequests”，
“responseTS”:1642754385215， “状
态”：“Ok”，

“标签”：
[{“quuppaRequests”:[
…,

{ “sequenceNumber”:5，
“requestTS”:1642754342911，“replyTS”：
1642754349333，“replyData”：“0x00
0x01 0x01 0x81 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0a 0x0b”，“requestData”： “0xaa 0x81 0x00”},空，

无效的，

```
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null
    ],
    "name": "ST_Tag_BLE_New_version",
    "id": "010080e18002",
    "status": "ReplyOk"
  }],
  "version": "1.0"
}
```


3.7 Developer Specific request/response example

Request for Developer Data (request type value: 0xFF)


```
{
  "code": 0,
  "command":
"http://localhost:8080/qpe/sendQuuppaRequest?tag=010080e18002&requestData=0xFF%200x81%200x00&humanReadable=true",
  "message": "Commanding 1 tag(s)",
  "responseTS": 1642753992164,
  "status": "Ok",
  "tags": [{
    "sequenceNumber": 2,
    "name": "ST_Tag_BLE_New_version",
    "id": "010080e18002"
  }],
  "version": "1.0"
}
```


Response for Developer Data (request type value: 0xFF)


```
{
  "code": 0,
  "command":
  "http://localhost:8080/qpe/getQuuppaRequestResponse?humanReadable&tag=010080e18002",
  "message": "QuuppaRequests",
  "responseTS": 1642754000072,
  "status": "Ok",
  "tags": [{
    "quuppaRequests": [
      …,
```

无效的，

无效的，

空,空，

空,空，

空，

无效的

],

“名称”：“ST_Tag_BLE_New_version”， “id”：

“010080e18002”， “状态”：

“ReplyOk”}]，“版本”：

“1.0”

}

3.7 开发人员特定的请求/响应示例

请求开发者数据（请求类型值:0xFF）{

“代码”:0，

“命令”：

“http://localhost:8080/qpe/sendQuuppaRequest?tag=010080e18002&requestData=0xFF%200x8 1%200x00&humanReadable=true”，

“消息”：“命令 1 个标签”，“响应 TS”：

1642753992164，“状态”：“确定”，“标签”：

[{“sequenceNumber”：

2，“名称”：

“ST_Tag_BLE_New_version”，

“id”：“010080e18002”

}]，

“版本”：“1.0”

}

开发者数据的响应（请求类型值:0xFF）{“code”:0，“command”：

“http://localhost:8080/qpe/getQuuppaRequestResponse?humanReadable&tag=010080e18002”，“消息”：“QuuppaRequests”，

“responseTS”:1642754000072， “状态”：

“Ok”，

“标签”：

[{“quuppaRequests”:[

…，

```
      {
        "sequenceNumber": 2,
        "requestTS": 1642753992164,
        "replyTS": 1642753997612,
        "replyData": "0xff 0x81 0x00 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0a
                      0x0b 0x0c",
        "requestData": "0xff 0x81 0x00"
      },
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null
    ],
    "name": "ST_Tag_BLE_New_version",
    "id": "010080e18002",
    "status": "ReplyOk"
  }],
  "version": "1.0"
}
```


3.8 Developer Specific Set Profile Number request/response example

Request for Developer Data with REQUEST (0xAD) for Profile Number setting to ID BADGE (0x02)


```
{
  "code": 0,
  "command":
"http://localhost:8080/qpe/sendQuuppaRequest?tag=010080e18002&requestData=0xFF%200x81%200x00%200xAD%200x02&humanReadable=true",
  "message": "Commanding 1 tag(s)",
  "responseTS": 1642754368939,
  "status": "Ok",
  "tags": [{
    "sequenceNumber": 4,
    "name": "ST_Tag_BLE_New_version",
    "id": "010080e18002"
  }],
```

{ “sequenceNumber”:2，
“requestTS”:1642753992164，“replyTS”：
1642753997612，“replyData”：“0xff
0x81 0x00 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0a 0x0b 0x0c”，“requestData”： “0xff 0x81 0x00”},空，

无效的，
无效的，
空,空，
空,空，
空,空，
空,空，
空,空，

无效的，
无效的

], “名称”：“ST_Tag_BLE_New_version”， “id”：
“010080e18002”， “状态”：
“ReplyOk” }],

“版本”：“1.0”
}

3.8 开发者特定设置配置文件编号请求/响应示例

使用 REQUEST（0xAD)请求开发者数据,将个人资料编号设置为 ID BADGE（0x02）{ “code”:0，

“命令”：
“http://localhost:8080/qpe/sendQuuppaRequest?tag=010080e18002&requestData=0xFF%200x81%200x00%200xAD%200x02&humanReadable=true”， “消息”：“命令 1
个标签”，“responseTS”:1642754368939， “状
态”：“确定”，

“标签”：
[{ “序列号”:4，“名称”：
“ST_Tag_BLE_New_version”，“id”：“010080e18002”

}]，

```
  "version": "1.0"
}
```


Response for Developer Data with REQUEST (0xAD) for Profile Number setting to ID BADGE (0x02)


```
{
  "code": 0,
  "command":
  "http://localhost:8080/qpe/getQuuppaRequestResponse?humanReadable&tag=010080e18002",
  "message": "QuuppaRequests",
  "responseTS": 1642754385215,
  "status": "Ok",
  "tags": [{
    "quuppaRequests": [
      …,
      {
       "sequenceNumber": 4,
       "requestTS": 1642754213330,
       "replyTS": 1642754222405,
       "replyData": "0xff 0x81 0x00 0x00 0x01 0x02 0xad 0x02 0x05 0x06 0x07 0x08 0x09 0x0a
                     0x0b 0x0c",
       "requestData": "0xff 0x81 0x00 0xad 0x02"
      },
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null,
      null
    ],
    "name": "ST_Tag_BLE_New_version",
    "id": "010080e18002",
    "status": "ReplyOk"
  }],
  "version": "1.0"
}
```

“版本”：“1.0”

}

开发者数据响应（0xAD)请求将配置文件编号设置为 ID
徽章 (0x02) {

“代码”:0，
“命令”：
“http://localhost:8080/qpe/getQuuppaRequestResponse?humanReadable&tag=010080e18002”，“消息”：“QuuppaRequests”，
“responseTS”:1642754385215，“状态”：
“Ok”，“标签”:[{“quuppaRequests”:[

…，

{ “sequenceNumber”:4，
“requestTS”:1642754213330，“replyTS”：
1642754222405，“replyData”：“0xff
0x81 0x00 0x00 0x01 0x02 0xad 0x02 0x05 0x06 0x07 0x08 0x09 0x0a 0x0b 0x0c”，“requestData”：“0xff 0x81 0x00 0xad 0x02” }，
null,null，

无效的，
无效的，
空,空，
空,空，
空,空，
空，

无效的

],
“名称”：“ST_Tag_BLE_New_version”， “id”：
“010080e18002”， “状态”：
“ReplyOk” }]，“版本”：

“1.0”

}

4 Acronyms and abbreviations

Table 23 List of acronyms

Acronym	Description
Bluetooth LE	Bluetooth Low Energy
DF	Direction Finding
DK	Development kit
QPE	Quuppa Positioning Engine
REQ	Request
RSP	Response
SW	Software
USB	Universal serial bus

5 References

Table 24 References

What	Where	Description
STSW-QUUPPA-ETAG	www.st.com/content/st_com/en/products/embedded-software/evaluation-tool-software/stsw-quuppa-etag.html	ST Quuppa tag emulation SW web page
BlueNRG-LP Bluetooth Low Energy wireless System on Chip	www.st.com/bluenrg-lp	BlueNRG-LP device web page
STEVAL-IDB011V1	www.st.com/bluenrg-lp, Tools and Software, Solution Evaluation Tools section	STEVAL-IDB011V1 platform web page
STSW-BNRGLP-DK	www.st.com/bluenrg-lp, Tools and Software, Evaluation Tool Software section	BlueNRG-LP DK SW package web page
UM2735	www.st.com/resource/en/user_manual/dm00711446-bluenrglp-development-kits-stmicroelectronics.pdf	BlueNRG-LP Development Kits user manual
Quuppa Tag Emulation	Quuppa customer portal	Specification of Quuppa Tag Emulation using Bluetooth Wireless Technology
Quuppa Tag Back Channel	Quuppa customer portal	Specification of Quuppa Tag Back Channel using Bluetooth Wireless Technology
Quuppa Web APIs	Quuppa customer portal	APIs available through the Quuppa Positioning Engine (QPE) server.
Quuppa Tag Emulation Pro profiles	Quuppa customer portal	Specification of Quuppa Profiles specification
Quuppa Development Kit Quick Start Guide	www.quuppa.com	Quick start guide about how to setup a Quuppa Positioning tracking system with Quuppa development kit HW and SW components

4 首字母缩略词和缩写

表 23 缩略词列表

缩写	描述
蓝牙 LE	低功耗蓝牙
自由度	测向
丹麦	开发套件
QPE	Quuppa 定位引擎
请求	要求
定期存货计划	回复
西南	软件
USB	通用串总线

5 参考

表 24 参考文献

什么	其中	描述
STSW- 库帕- 埃塔格	www.st.com/content/st_com/en/products/embedd ed-软件/评估工具软件/stsw-quuppa-etag.html	ST Quuppa 标签模 拟 SW 网页
BlueNRG-LP 蓝牙 低能耗无线 系统开启 芯片	www.st.com/bluenrg-lp	BlueNRG-LP 设备网页 页
STEVAL- IDB011V1	www.st.com/bluenrg-lp,工具和软件,解决方案评估工具部分	STEVAL-IDB011V1平台网 页
STSW- BNRGLP-DK	www.st.com/bluenrg-lp,工具和软件,评估工具软件部分	BlueNRG-LP DK SW 软件包 网页
UM2735	www.st.com/resource/en/user_manual/dm007114 46-bluenrglp-开发套件- stmicroelectronics.pdf	BlueNRG-LP 开发套件用户手册
库帕日 仿真	Quuppa 客户门户	Quuppa 使用蓝牙无线技术的标签模 拟规范 Quuppa 规范
库帕日 后退 渠道	Quuppa 客户门户	使用蓝牙无线技术标记反向通道
库帕网 蜜蜂	Quuppa 客户门户	可通过 Quuppa 定位引擎 (QPE) 服务器提供的 API。
库帕日 仿真 专业资料	Quuppa 客户门户	Quuppa 规格 型材规格
库帕 发展 套件快速 入门指南	www.quuppa.com	关于如何设置 Quuppa 的快速 入门指南 采用 Quuppa 开发套件 HW 和定位跟踪系统 软件组件

6 Revision history

Table 25 Revision history

Date	Revision	Change
10-February-2022	1	Initial release

6 修订历史

表 25 修订历史

日期	修订	改变
2022 年 2 月 10 日	1	初始版本

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

重要通知 请仔细阅读

STMicroelectronics NV 及其子公司（“ST”）保留随时对 ST 产品和/或本文档进更改、修正、增强、修改和改进的权利，恕不另通知。购买者应在下订单前获取有关 ST 产品的最新相关信息。ST 产品根据订单确认时有效的 ST 销售条款和条件进销售。

购买者对 ST 产品的选择、挑选和使用负全部责任，ST 对购买者产品的应用协助或设计不承担任何责任。

ST 在此不授予任何明示或暗示的知识产权许可。

如果转售的 ST 产品的规定与此处所述的信息不同，则 ST 对该产品授予的任何保证将失效。

ST 和 ST 徽标是 ST 的商标。有关 ST 商标的更多信息，请参阅www.st.com/trademarks。所有其他产品或服务名称均归其各自所有者所有。

本文档中的信息取代并替换了该文档之前任何版本中提供的信息。