AN13737

Software configuration to interact with mobile phones

Rev. 1.0 — 18 October 2022

Application note
CONFIDENTIAL

Document information

Information	Content
Keywords	SR1XX, SR040, iOS, Android
Abstract	This document is a reference to make NXP UWB accessory ranging with mobile phones.

1 Revision history

Revision history

Rev	Date	Description
1.0	2022-10-18	Initial version

2 Introduction

This document is intended as guideline for showcasing how NXP UWB Trimension™ chipsets can be configured to interoperate with UWB enabled phones.

This document references to:

- configure Out Of Band data (receive and send handlers)
- configure NXP UWB accessory behavior
- start ranging on accessory side

To illustrate interaction with UWB equipped phones, NXP delivers related software in the form of specific release packages (based on NXP UWB SDK release). Those packages include example application (refer to *demo_NearbyInteraction* application) demonstrating the Bluetooth LE Out of Band communication to establish UWB TWR ranging session between the NXP UWB Trimension™ based device and a mobile phone (iOS or Android).

3 Out Of Band communication

3.1 Out Of Band communication between phone and accessory

For the purpose of the demo, OoB communication with phones is insured via Bluetooth LE. However OoB protocol is not limited to Bluetooth LE, it can be replaced by another physical link.

3.2 Out Of Band BLE Protocol

Out of Band protocol is defined by the application maker.

Depending on customer needs, BLE characteristics of accessory can be changed inside **gatt_uuid128.h** file part of the released package.

It allows to change the following Bluetooth LE parameters:

1. serviceUUID (by changing uuid_service_qpps)
2. rxCharacteristicUUID (by changing uuid_qpps_characteristics_rx)
3. txCharacteristicUUID (by changing uuid_qpps_characteristics_tx)

3.3 Out Of Band default Bluetooth LE settings

By default, here are the default values:

1. serviceUUID: 6E400001-B5A3-F393-E0A9-E50E24DCCA9E
2. rxCharacteristicUUID: 6E400002-B5A3-F393-E0A9-E50E24DCCA9E
3. txCharacteristicUUID: 6E400003-B5A3-F393-E0A9-E50E24DCCA9E

3.4 Receive BLE messages from phone application after connection established

When phone application send data over Bluetooth LE, application needs to parse received data and do the processing.

In the code example BLE message is received inside Middleware code below (in file **TLV_Mng.c**)

```
static void handleTLV(uint8_t *data)
{
    tUWBAPI_STATUS uwb_status = UWBAPI_STATUS_FAILED;
    (void)phOsalUwb_LockMutex(mTlvMutex);

    switch (*data) {
```

data is the buffer containing the BLE message. The switch() function is handling the received command based on the first byte of the message as specified in chapter Section 4. It can be updated to handle more messages coming from the phone application.

3.5 Send BLE message to phone application

When command is processed, a response needs to be send over BLE.

In the code example BLE message is sent using Middlware API below (in file **TLV_Mng.c**).

```
tlvSendRaw(uint8_t *buf, uint16_t size)
```

buf is the buffer containing the message to be sent over BLE link

size is the length of the message to be sent over BLE link

4 Interaction scenario

The expected flow is as described below:

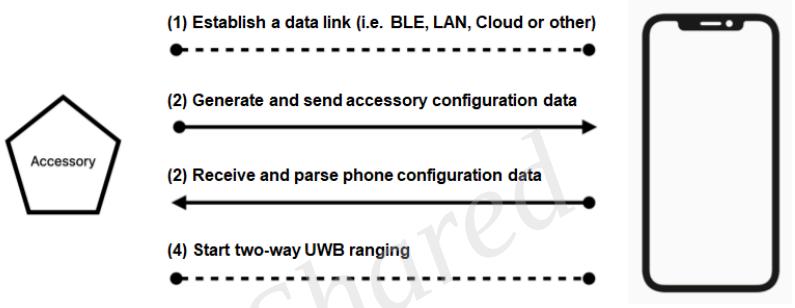

Figure 1. Interaction scenario

The flow to interact with iOS based phones or Android based phones is similar, however there is some differences in the communication parameters.

The demo example defines the following Out of Band command set:

Table 1. Out of Band commands (from phone to accessory)

Command	Code	Parameter
Initialize iOS command	0x0A	none
Initialize Android command	0xA5	none
Configure and start command	0x0B	Phone configuration data
Stop command	0x0C	none

Table 2. Out of Band responses (from accessory to phone)

Response	Code	Parameter
Initialize response	0x01	Accessory configuration data
Start response	0x02	none
Stop response	0x03	none

4.1 Interaction with iOS phones

4.1.1 Introduction

Communication parameters are defined by Apple in Nearby Interaction Specification: https://developer.apple.com/nearby-interaction/specification/.

The demo example scenario follows the following flow:

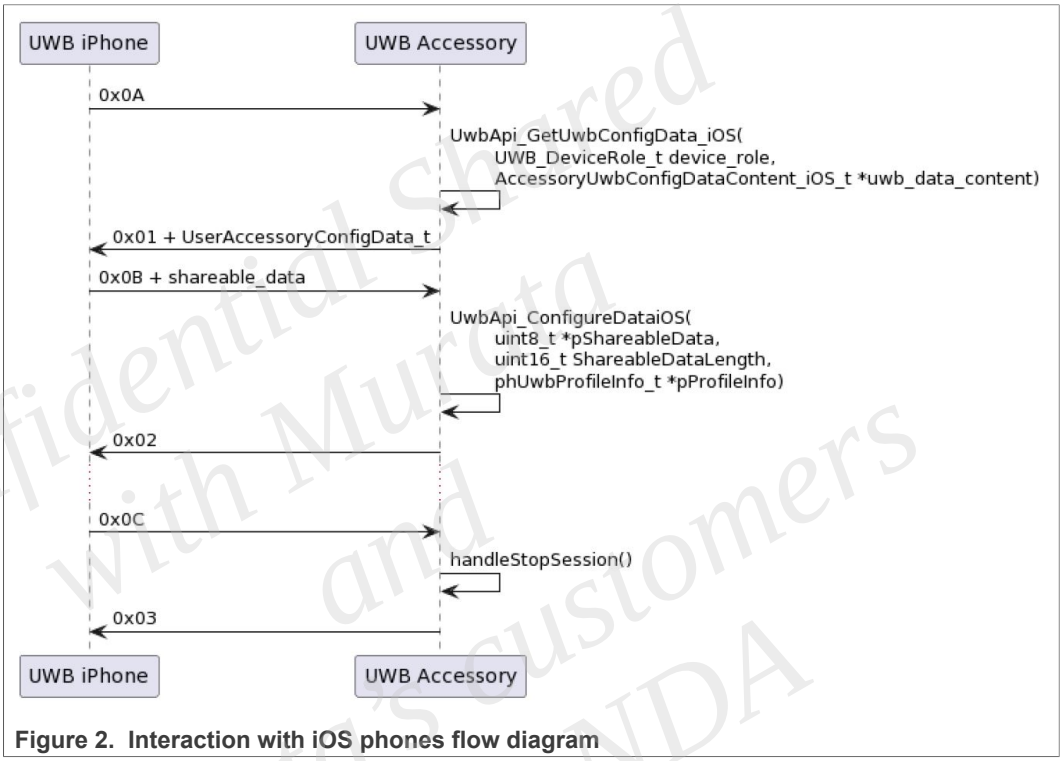

Figure 2. Interaction with iOS phones flow diagram

4.1.2 Accessory configuration data

When receiving the dedicated OoB command (code 0x0A in the demo example), the accessory compulse the configuration data and send it as a response (code 0x01 in the demo example) to the phone. The Accessory configuration data is defined as below:

Table 3. Accessory configuration data

Parameter	Size (bytes)	description
Major Version	2	Please refer to **Nearby Interaction Accessory Protocol Specification**
Minor Version	2	
PreferredUpdateRate	1	
RFU	10	
UWBConfigData Length	1	
UWBConfigData	variable	

This data field is partially filed from the application (*MajorVersion*, *MinorVersion*, *PreferedUpdateRate*), while the rest (UWBConfigData and UWBConfigDataLength) is retrieved from the NXP UWB Trimension™ device thanks to `UwbApi_GetUwbConfigData_iOS()` API.

4.1.3 Phone configuration data (Shareable data)

Apple shareable configuration data are received by the accessory via dedicated Out of Band command (code 0x0B in the demo example).

This data should not be modified by the customer and are directly sent to UWB chipset thanks to `UwbApi_ConfigureData_iOS()` API.

This API takes as parameter a profile structure which must be filed by the application with the following arguments:

- Device role: accessory UWB role (Initiator or Responder)
- Device type: accessory UWB type (Controller or Controlee)

In the example, when the accessory as properly configured and start the UWB ranging session according to the received Apple shareable configuration data, it sends back proper response to the phone (code 0x02 in the demo example)

4.1.4 Phone application

From phone side, the application must use **NINearbyAccessoryConfiguration** API defined in iOS (see https://developer.apple.com/documentation/nearbyinteraction/ninearbyaccessoryconfiguration).

As reference, the provided demo example supports interaction with the Spatial Interactions with Third-Party Accessories sample code offered by Apple (see https://developer.apple.com/documentation/nearbyinteraction/implementing_spatial_interactions_with_third-party_accessories)

The demo example also supports interaction with NXP Timensions AR application freely available on AppStore (see https://apps.apple.com/us/app/nxp-trimensions-ar/id1606143205).

4.2 Interaction with Android phones

4.2.1 Introduction

The demo example scenario follows the following flow:

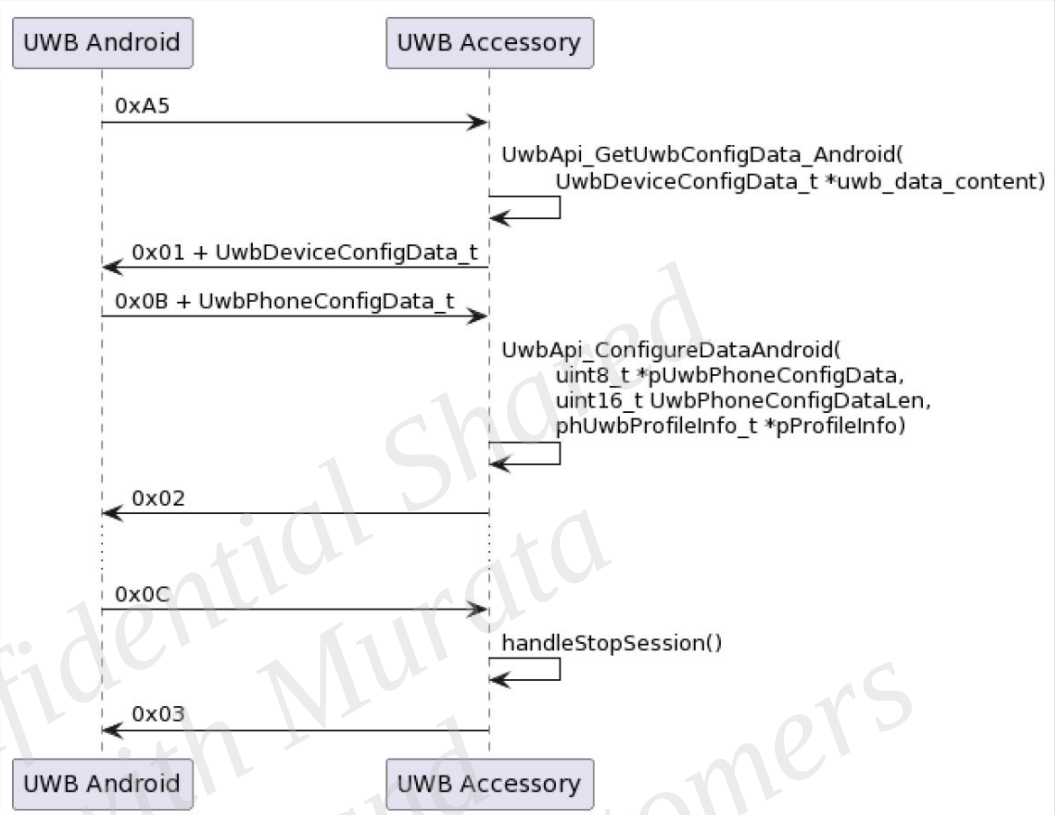

Figure 3. Interaction with Android phones flow diagram

4.2.2 Accessory configuration data (UWB device config data)

When receiving the dedicated OoB command (code 0xA5 in the demo example), the accessory compulse the configuration data and send it as a response (code 0x01 in the demo example) to the phone. The Accessory configuration data is defined as below:

Table 4. UWB device config data

Parameter	Size (bytes)	description
Spec major version	2	Supported spec version
Spec minor version	2	
Chip_id	2	NXP UWB chipset identifier (SR040, SR150 …)
Chip FW version	2	NXP UWB chipset FW version
MW version	3	Accessory UWB MW version
Supported profile IDs	1	List of UWB configuration profiles supported by the accessory
Supported device ranging roles	1	Set of UWB ranging roles supported by the accessory
Device MAC address	2	Accessory UWB MAC address

This data field is entirely retrieved from the NXP UWB Trimension™ device thanks to `UwbApi_GetUwbConfigData_Android()` API.

With this configuration data, the accessory informs the UWB Phone about the supported UWB Profile IDs (see https://developer.android.com/guide/topics/connectivity/uwb) and UWB Ranging roles , bit value (1 or 0) indicating whether the specific profile ID or ranging role is supported by the device:

- supported_profile_ids indicates the set of supported UWB_CONFIG_ID_X profile IDs
  - Bit positions corresponds maps to UWB_CONFIG_ID_X profile ID
  - Example → supported_profile_ids = 0x0000000E (0b00000000000000000000000000001110) indicates that Google defined UWB_CONFIG_1, UWB_CONFIG_2, UWB_CONFIG_3 are supported by the UWB device
- supported_device_ranging_roles indicates the set of UWB ranging role configurations supported by the accessory
  - 0bxxxxxxx1 → UWB device supports Controller device role
  - 0bxxxxxx1x → UWB device supports Controlee device role
  - Example → supported_device_ranging_roles = 0x03 (0b00000011) indicates that both Controller and Controlee roles are supported by the accessory

The UWB phone then decides the UWB Profile ID and UWB Ranging role to be used among the reported supported values.

4.2.3 Phone configuration data

Android phone configuration data are received by the accessory via dedicated Out of Band command (code 0x0B in the demo example). The Phone configuration data is defined as below:

Table 5. Phone configuration data

Parameter	Size (bytes)	description
Spec major version	2	Used spec version
Spec minor version	2	
Session_id	2	UWB session identifier
Preamble_id	2	UWB preamble code index
Channel number	3	UWB channel number
Profile IDs	1	Used UWB_CONFIG_ID profile
Device ranging role	1	Accessory UWB ranging role
Phone MAC address	2	Phone MAC address

This data should not be modified by the customer and are directly sent to UWB chipset thanks to `UwbApi_ConfigureData_Android()` API.

In the example, when the accessory as properly configured and start the UWB ranging session according to the received phone configuration data, it sends back proper response to the phone (code 0x02 in the demo example).

4.2.4 Phone application

From phone side, the application must use **androidx.core.uwb** API defined in Android (see https://developer.android.com/reference/kotlin/androidx/core/uwb/package-summary). The **androidx.core.uwb** API is available from Android 13 release, however

UWB Jet pack API allows deploying similar application on Android 12 (see https://developer.android.com/jetpack/androidx/releases/core-uwb).

For demonstration purpose, the provided demo example supports interaction with UWB phones running the offered Android "UWB Mobile Kit" application (refer to the related release package).

5 Abbreviations

Table 6. Abbreviations

Abbreviation	Meaning
BLE	Bluetooth Low Energy
OoB	Out of Band
LAN	Local Architecture Network
UWB	Ultra Wide Band
OS	Operating System

6 Legal information

6.1 Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

6.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

6.3 Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Tables

Figures

Contents

For more information, please visit: http://www.nxp.com

Date of release: 18 October 2022
Document identifier: AN13737