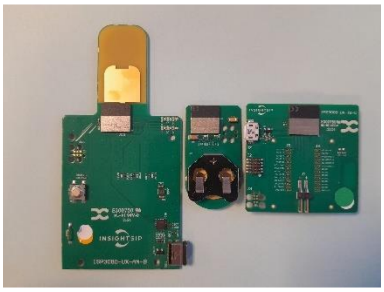
ISP3080

Application note AN240101

Use of ISP3080-UX Development Kit



Introduction

Scope

This document gives details on hardware and software for using and testing Insight SiP Bluetooth Low Energy and Ultra-Wideband module ISP3080-UX.

Contents

1. Recommended Documentation ... 3

2. Hardware kit content ... 4
2.1. Global description ... 4
2.2. ISP3080-UX module ... 5
2.3. ISP3080-UX-TB test board ... 6
2.4. ISP130603 interface board ... 7
2.5. ISP3080-UX-AN ... 8
2.6. ISP3080-UX-TG (ISP3086) ... 9

3. Getting started with the kit ... 10
3.1. Software Developpement kit (SDK) content ... 10
3.2. Setup ... 11
3.3. Qorvo's basic examples ... 12
3.4. TWR_Demo ... 13

4. Two-Way Ranging (TWR) Demonstrator ... 18
4.1. Description ... 18
4.2. Running the demo ... 19
4.3. Tag current consumption ... 21

5. UWB Calibration ... 23
5.1. OTP Map ... 24
5.2. 38.4 MHz Crystal Trimming ... 26
5.3. Antenna delay ... 26
5.4. Output Power ... 28

Document Revision History

Revision	Date	Ref	Change Description
R0	9/1/2024	jf cb	Initial release
R1	14/8/2024	cb jf	Add UWB calibration chapter

1. Recommended Documentation

The following Nordic Semiconductor documents are required to understand the complete setup and programming methods.

Nordic Semiconductor Documents

- nRF52833 Development kit User Guide, hardware section should be partially ignored – Insight SiP development kit hardware replaces Nordic Semiconductor hardware.
- nRF52833 Product Specification– make sure you have the latest document version updated.

To access documentation, information, go to:

- Official Nordic Semi website http://www.nordicsemi.com
- The Nordic Semiconductor Tech Docs is a "comprehensive library" containing technical documentation for current and legacy solutions and technologies https://docs.nordicsemi.com/
- Find documentation about nRF Connect SDK here https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/index.html
- Find documentation about nRF SDK here: https://infocenter.nordicsemi.com/topic/struct_sdk/struct/sdk_nrf5_latest.html
- Ask any Nordic related question and get help https://devzone.nordicsemi.com/questions
- For any question, you can also open a case here https://devzone.nordicsemi.com/support/add

Qorvo Documents

- QM33110W Data sheet
- QM33110W User Manual

To access documentation, information, go to: https://www.qorvo.com/products/p/QM33110W#documents

Insight SiP documents

The following documents are available on Insight SIP website or/and on request:

- AN240101_R1 App Note – this document.
- ISP3080-UX data sheet.
- ISP3080-UX-TB schematic.
- ISP3080-UX-TG and ISP3080-UX-AN schematics
- ISP130603 Interface Board schematic.

2. Hardware kit content

2.1. Global description

.
The picture below shows the Evaluation Kit hardware content for the ISP3080-UX module.

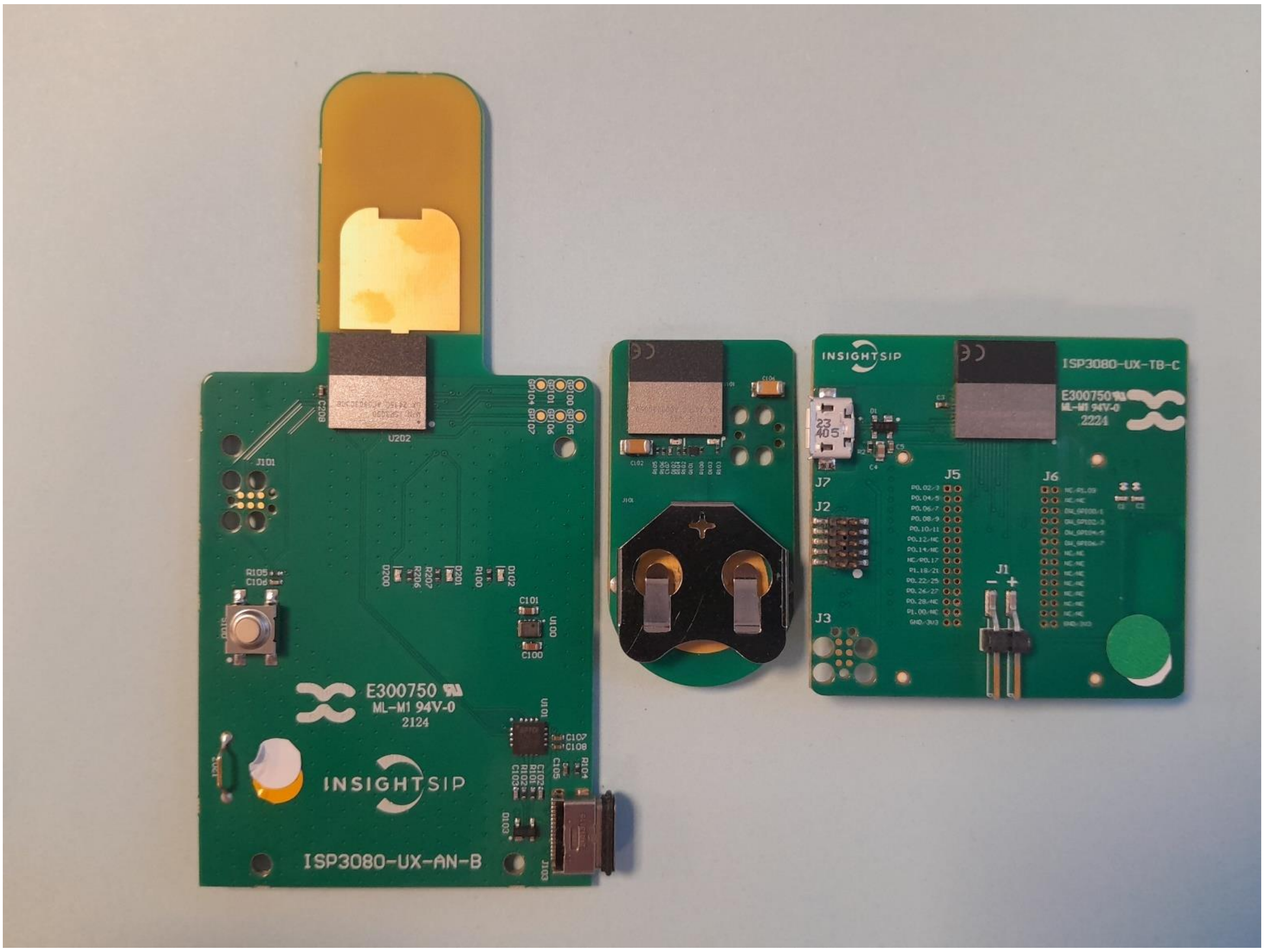


Figure 1: From left to right: Anchor, Tag, Test board

The kit also includes an interface board ISP130603H that is shown in ISP130603 Interface Board section.

2.2. ISP3080-UX module

The ISP3080-UX module is based on Qorvo QM33110 single-chip UWB transceiver and nRF52833 Nordic Semiconductor 2.4GHz wireless System on Chip (SoC).

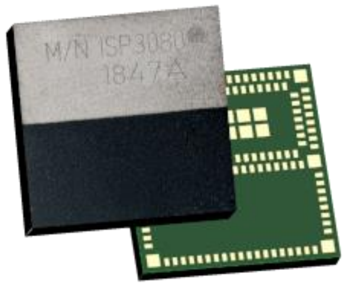


Figure 2: ISP3080-UX module

It integrates a 32-bit ARM Cortex™ M4 CPU,512 kB flash memory, 64 kB RAM as well as analog and digital peripherals. Despite the small size of 12 x 12 x 1.5 mm, the module integrates decoupling capacitors, 38.4 MHz crystal for UWB, 32 MHz and 32.768kHz crystals for BLE, DC-DC converters, RF matching circuits and two antennas in addition to the wireless SoCs. Low power consumption and advanced power management enables battery lifetimes up to several months on a coin cell battery.

For more details, see the ISP3080-UX datasheet.

2.3. ISP3080-UX-TB test board

The ISP3080-UX-TB test board consists of a module mounted on a PCB for prototyping and testing purposes. It has dimensions of 47 x 45 mm.

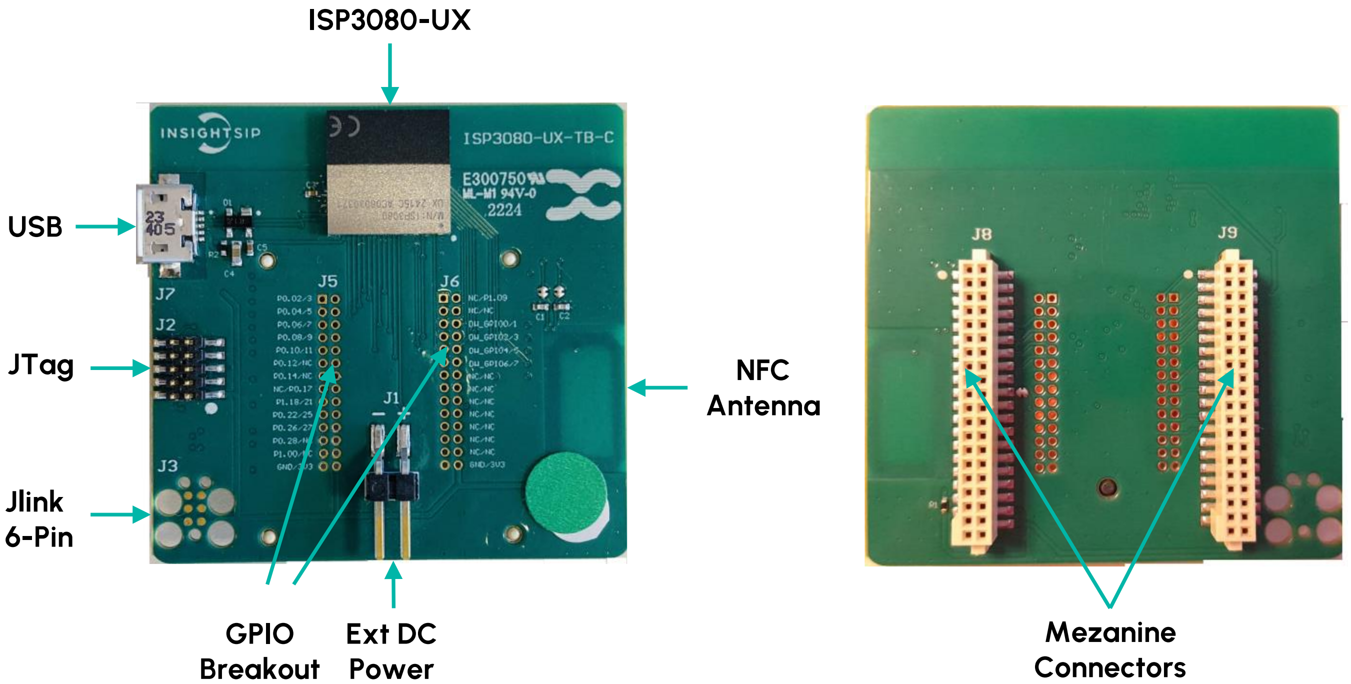


Figure 3: ISP3080-UX-TB

It encloses:

- ISP3080-UX module.
- 2 mezzanine connectors on the bottom side of the board (for connection to an interface board)..
- 2 x footprints for optional 2x14 pin/1.27mm pitch connector for access to the module pins.
- JTAG footprint for programming using 6 pin Segger Jlink Adapter.
- https://www.segger.com/products/debug-probes/j-link/accessories/adapters/6-pin-needle-adapter/
- 2x5 pin header for programming using Segger Jlink interface contained in Nordic Evaluation Board.
- 2-pin header for power supply when using 6 pin JTAG or Nordic JTAG programming options.
- USB connector connected to the module's USB interface.
- NFC Antenna on PCB.

2.4. ISP130603 interface board

The ISP130603 board is an interface board that has dimensions of 100x70mm². It can interface with test boards and is responsible for providing:

- Power supply.
- Programming and debugging capabilities.
- Access to all Ios

Note that in the case of ISP3080-UX only 3V power supply is used.

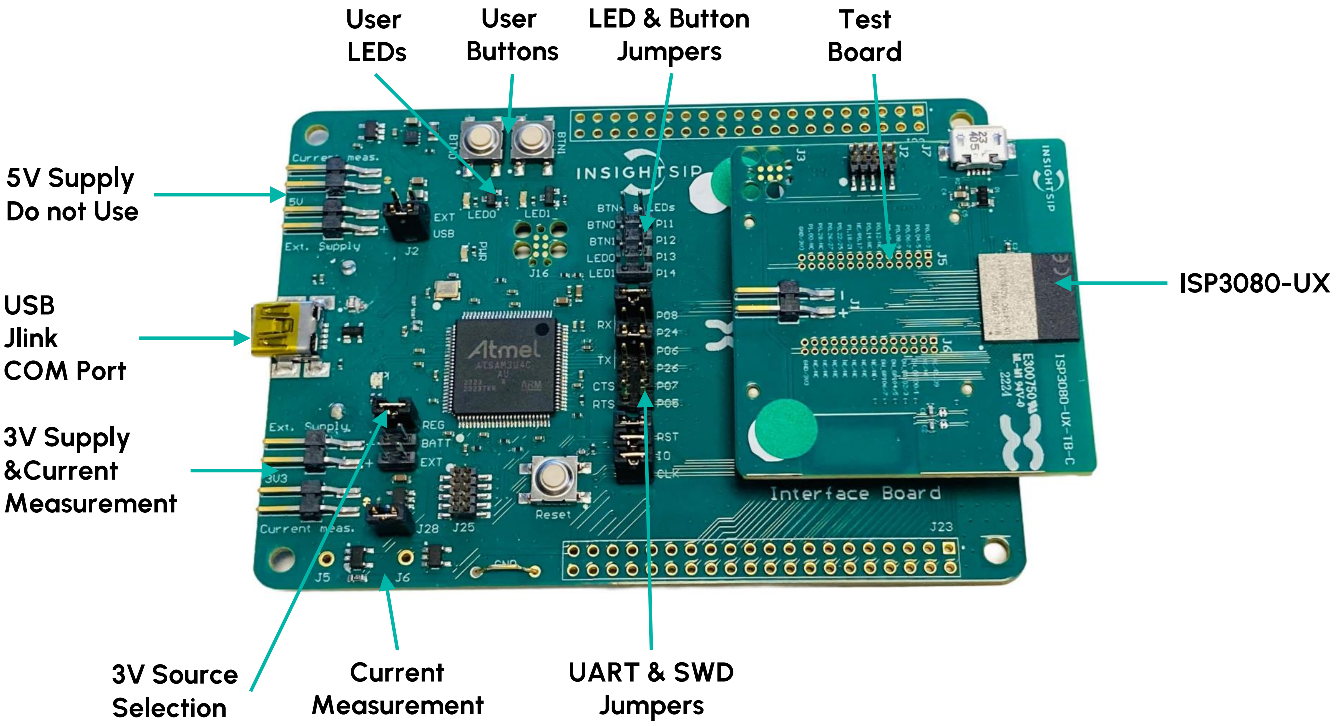


Figure 4: ISP130603 mounted with an ISP3080-UX-TB

2.5. ISP3080-UX-AN

The ISP3080-UX-AN consists of a PCB integrating an ISP3080-UX module and a USB C socket for connection to a PC port com. It enables the communication with a TAG through UWB standards. The board dimensions are 105x50mm².
It can also be referred to as "Anchor".

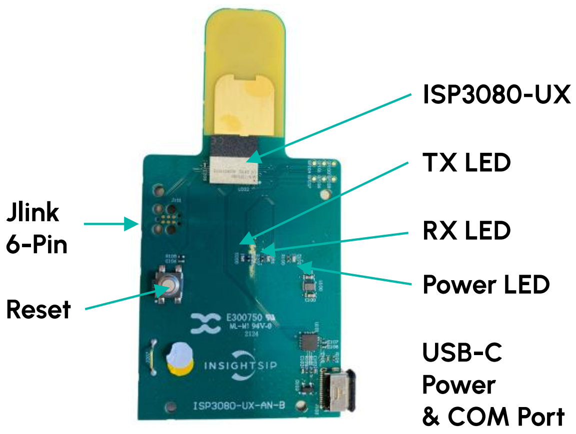


Figure 5: ISP3080-UX-AN

It encloses:

- ISP3080-UX module.
- 3 mini-LEDs (PWR ON and UWB TX/RX LEDs).
- USB-C connector.
- JTAG footprint for programming using 6 pin Segger Jlink Adapter. https://www.segger.com/products/debug-probes/j-link/accessories/adapters/6-pin-needle-adapter/
- FTDI USB-to-Serial adapter.
- UWB Antenna on PCB.
- 3V Regulator.

2.6. ISP3080-UX-TG

The ISP3080-UX-TG consists of a small 44x22mm² PCB integrating an ISP3080-UX module powered by a coin cell.
It can also be referred as "Tag".

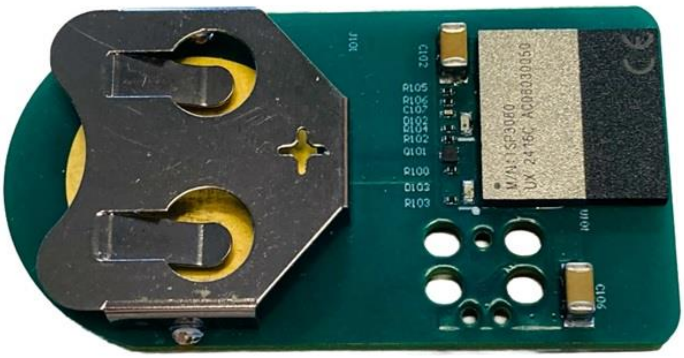


Figure 6: ISP3080-UX-TG

To start the tag a CR2032 battery should be placed in the battery compartment, plus side upwards:

It encloses:

- ISP3080-UX module.
- 2 x user programmable mini-LEDs.
- Battery holder.
- JTAG footprint for programming using 6 pin Segger Jlink Adapter.
  https://www.segger.com/products/debug-probes/j-link/accessories/adapters/6-pin-needle-adapter/
  Note that the contacts for the 6-pin interface are on the backside of the PCB

3. Getting started with the kit

3.1. Software Developpement kit (SDK) content

Insight SiP provides an SDK for the ISP3080-UX available here:
https://github.com/insightsip

This SDK is a nRF52833 port of the Release 6.0C of the Qorvo/Decawave API software. For reference this software can be found on the DWM3000EVB page:
For more information see https://www.qorvo.com/products/p/DWM3000EVB#documents.

This SDK needs the Segger Embedded Studio (SES) IDE to build the projects. We recommend using SES version 5.42a as newer version might be incompatible with the nRF SDK. SES can be freely downloaded at https://www.segger.com/downloads/embedded-studio.

The ISP3080-UX SDK directory tree is shown below:

```
ISP3080-SDK
    ├─ TWR_Demo                          // Insight SiP's TWR demo
    │     └──── application              // application SES project
    │     └──── secure_bootloader        // bootloader SES project
    │     └──── vault                    // bootloader private key
    └─ DW3XXX_API_rev9p3
          └─ API
                ├─ Shared                // Qorvo's UWB chip driver
                ├─ Src                   // Qorvo's code examples
                └─Build_Platforms
                     └─ nRF52833-DK      // SES project for Qorvo's examples
                          └─ sdk         // Adapted nRF SDK v17.0.2
```

3.2. Setup

To start working with the ISP3080-UX module, plug the test board (ISP3080-UX-TB) on the interface board (ISP130603) using the two mezzanine connectors.

Make sure that:

- J2 jumper is on ***USB*** position.
- J4 jumper is on ***REG*** position.
- All 3 ***SWD*** jumpers (***RST***, ***IO*** and ***CLK***) are present.

Plug the USB connector to your computer.

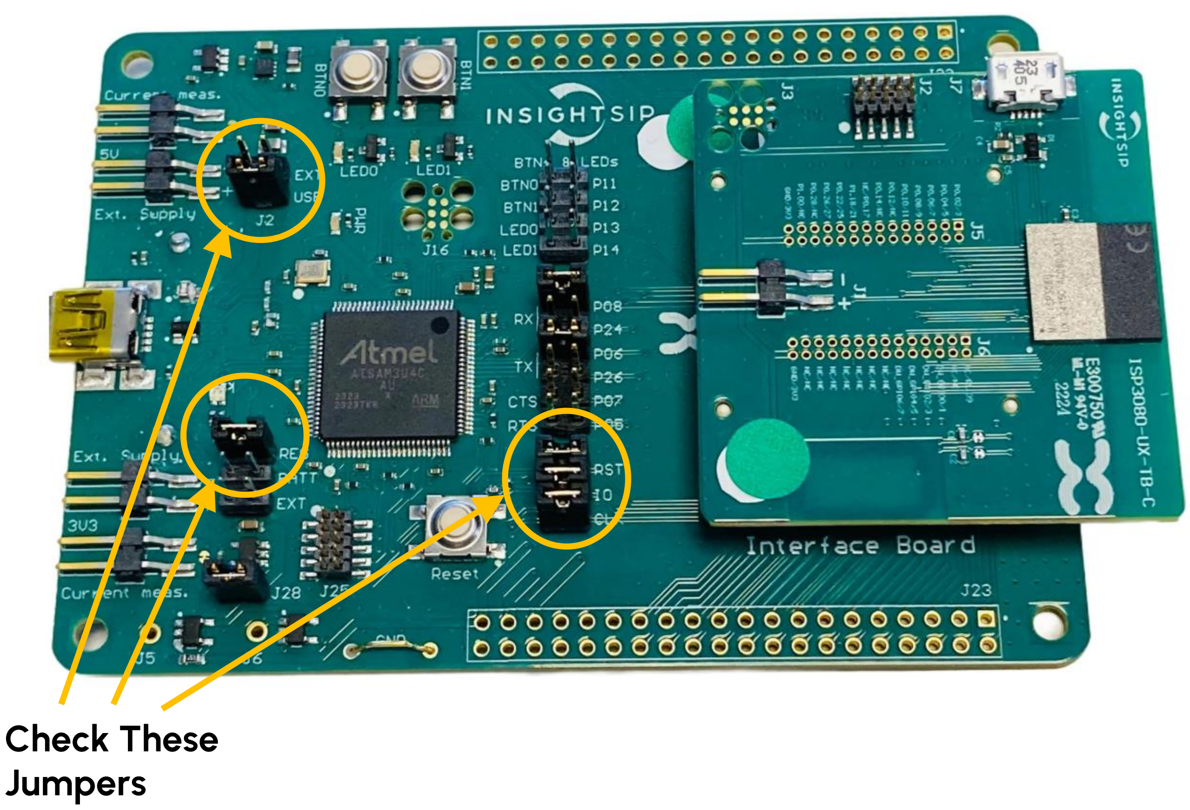


Figure 7: Setup on Interface Board

On the computer open ***Device Manager*** and check that the following devices appear in the list:

- ***J-Link driver*** in ***Universal Serial Bus Controllers***.
- ***J-Link CDC UART Port*** in ***Ports***.

If they do not appear install the J-Link drivers on the computer. They can be downloaded here: https://www.segger.com/downloads/jlink

3.3. Qorvo's basic examples

To use any of the Qorvo examples, open the dw3000_api.emProject file using SES. The file is located in "ISP3080-SDK"\DW3XXX_API_rev9p3\API\Build_Platforms\nRF52833-DK.

Edit the example_selection.h and uncomment the example to build. The file is located in "ISP3080-UWB"\Software\ISP3080-SDK\DW3XXX_API_rev9p3\API\Src

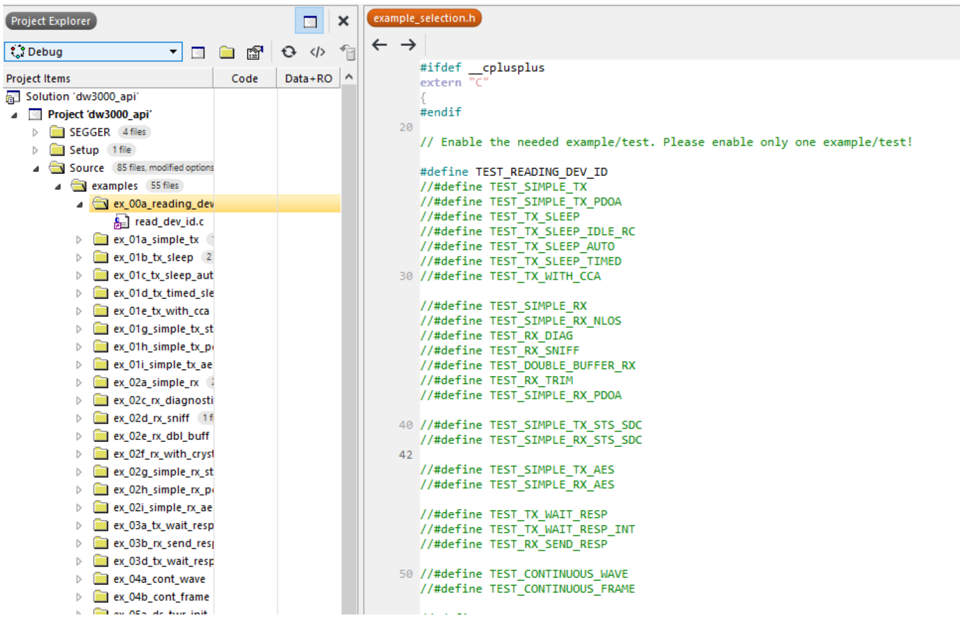


Figure 8: Example_selection.h

Make sure to uncomment only one line.

Click "Build and Run" to build the selected project and load it to the module.

3.4. TWR_Demo

3.4.1. description

The Two-way ranging (TWR) demonstrator folder is divided into three subfolders:

- Application folder.
- Secure bootloader: Used for Device Firmware Update (DFU) operations.
- Vault: Contains keys for bootloader.

The content of the application folder is:

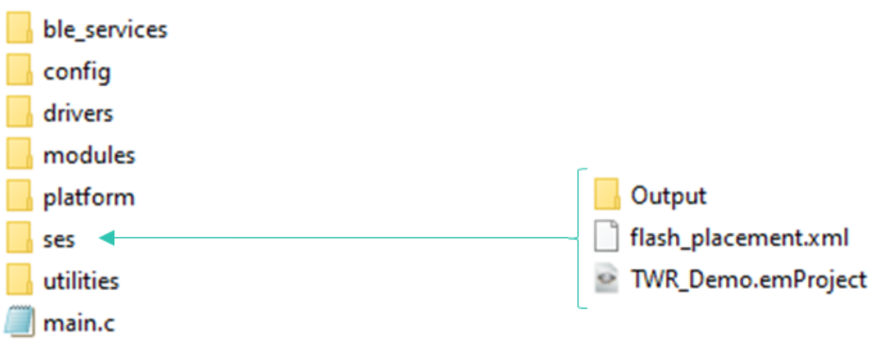


Figure 9: TWR_Demo repertory content

It contains:

- **ble_services**: files containing all the custom BLE services used by the TWR_Demo.
- **config**: contains the sdk_config.h file.
- **drivers**: files containing all the "sensor" drivers.
- **module**: files containing high level features.
- **ses**: Contains the SES project.

Open the TWR_Demo/application/ses/TWR_Demo.emProject using SES to access the sub-projects:

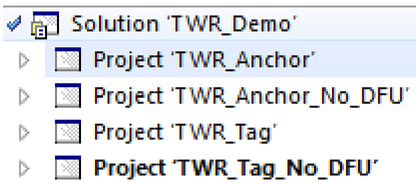


Figure 10: TWR_Demo project explorer

There are four different sub-projects available. Switch between sub-project is possible by simply right-clicking on the project and selecting "Set as active project" as shown below:

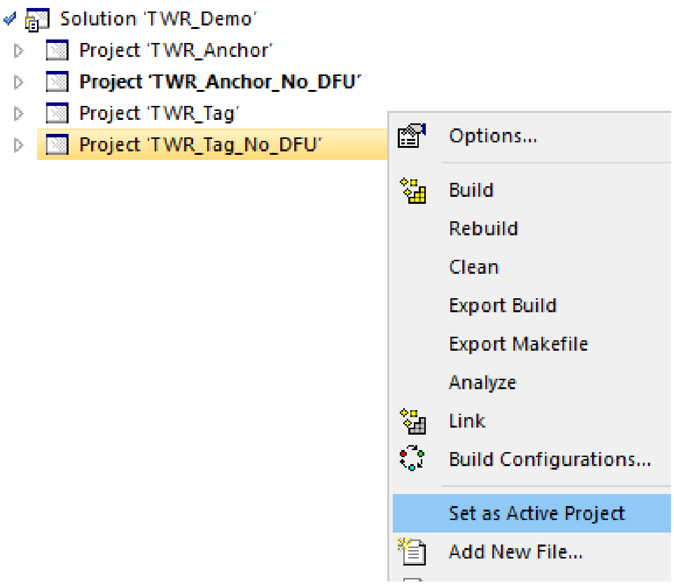


Figure 11: Project selection in SES

3.4.2. Using the "no DFU" version

To get started open the TWR_Demo/application/ses/TWR_Demo.emProject file using SES.
The SES project is divided into four sub-projects, the ones that are used in this chapter are:

- **TWR_Tag_No_DFU**: It is used to generate firmware that can be loaded on the tag. This version of the firmware is very power optimized which enable the use of coin cell battery as power source.
- **TWR_Anchor_No_DFU**: It is used to generate firmware that can be loaded on the anchor. This version of the firmware will enable UWB Reception most of the time thus is power hungry.

Select one of them.

Click "Build and Run" to build the selected project and directly load it to the module.

<u>Note</u>: It is recommended to erase.all the flash first before loading a new file.

3.4.3. Using the "DFU" version

In the DFU version the application is loaded from a smartphone/tablet to the module via BLE transfer. In this case a bootloader is flashed on the module instead of the application. The application can then be downloaded later to the module using the adequate tool on the smartphone/tablet.

Step 1; Build the DFU package of the application.

To get started open the TWR_Demo/application/ses/TWR_Demo.emProject file using SES.
The SES project is divided into four sub-projects, the ones that is used in this chapter are:

- **TWR_Tag**: It is used to generate firmware that can be loaded on the tag. This version of the firmware is very power optimized which enable the use of coin cell battery as power source.
- **TWR_Anchor**: It is used to generate firmware that can be loaded on the anchor. This version of the firmware will enable UWB Reception most of the time thus is power hungry.

Select one of them.

Click "Build TWR_XXX" or press F7. In this case don't load it to the module.

Instead, we need to build a DFU package using nrfutil. Download it here:

https://www.nordicsemi.com/Products/Development-tools/nrf-util

Install the necessary tools using the command:

.\nrfutil install nrf5sdk-tools.

Build the DFU package using the command (it assumes that nrfutil.exe is located in the vault folder, if not adapt paths as necessary):

.\nrfutil.exe pkg generate --application ..\application\ses\Output\Release\Exe\TWR_Tag.hex --application-version-string "1.0.0" --hw-version 52 --sd-req 0x102 --key-file .\priv.pem TWR_Tag.zip

Or:

.\nrfutil.exe pkg generate --application ..\application\ses\Output\Release\Exe\TWR_Anchor.hex --application-version-string "1.0.0" --hw-version 52 --sd-req 0x102 --key-file .\priv.pem TWR_Anchor.zip

The generated package (TWR_Tag.zip) must now be copied to the smartphone/tablet.

Step 2; Build and load the the bootloader.

Open the TWR_Demo/secure_bootloader/ses/TWR_Demo.emProject file using SES and select the needed sub-project.

Note: The sub-project here is not important. It just add LEDs blink for the Tag version.

The bootloader uses the "micro_ecc" cryptography. The library is not present in this SDK.

To install it follow the steps:

- Download it at https://github.com/kmackay/micro-ecc.

- Extract it in "ISP3080-SDK"\DW3XXX_API_rev9p3\API\Build_Platforms\nRF52833-DK\sdk\external\micro-ecc.
- Run build_all.bat located in micro-ecc folder.

Now the secure_bootloader project can be built. Click "Build and Run" to build the selected project and directly load it to the module.

The module (Tag or Anchor) will start in DFU mode.

Step 3: Transfer over BLE

Install ***nRF Connect*** (https://play.google.com/store/apps/details?id=no.nordicsemi.android.mcp) on a smpartphone/tablet and open it.

When no valid application is present on the module will advertise with the name "DfuTarg". Connect to it.

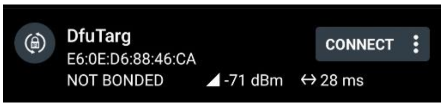
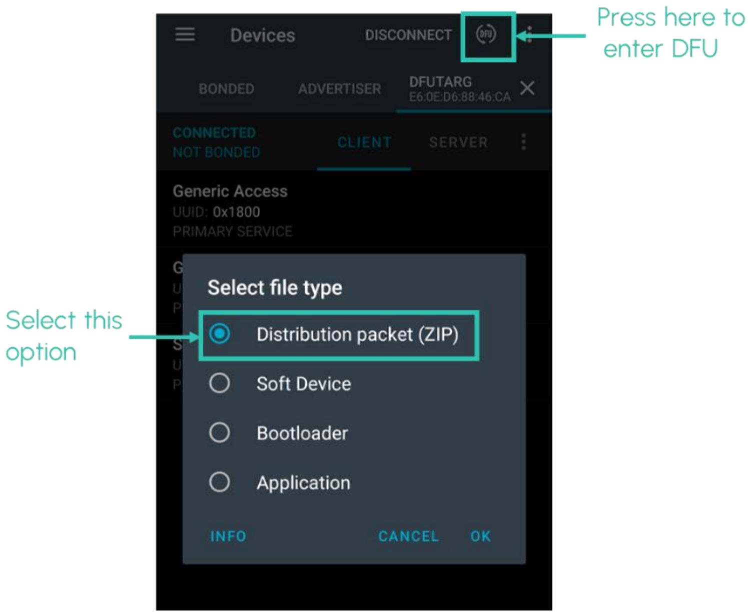
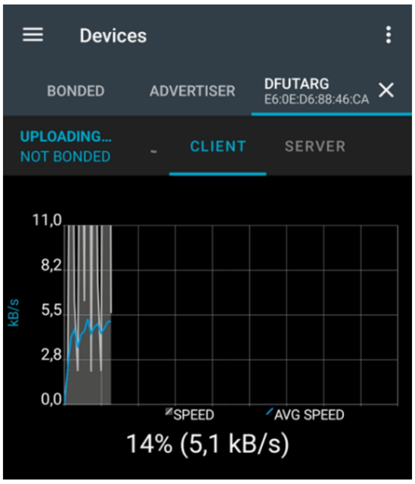


Figure 12: DFU advertisement

Press DFU icon at the top of the screen and select "Distribution packet (ZIP)".



Choose the previously built DFU package and transfer will automatically start.



The module will reset and start the updated application.

4. Two-Way Ranging (TWR) Demonstrator

4.1. Description

This paragraph shows you how to set up the Insight SiP's Range application between the Anchor and the Tag. The range is calculated using UWB and results are sent via the Bluetooth link to a smartphone/tablet.

In this demo, the Anchor and the Tag operate as a pair. The Tag is the one initiating the ranging exchange and the Anchor is listening for the tag messages and responding to it. At the end of the ranging exchange the Tag board can then calculate the time of flight thus the range between the Tag and the Anchor.

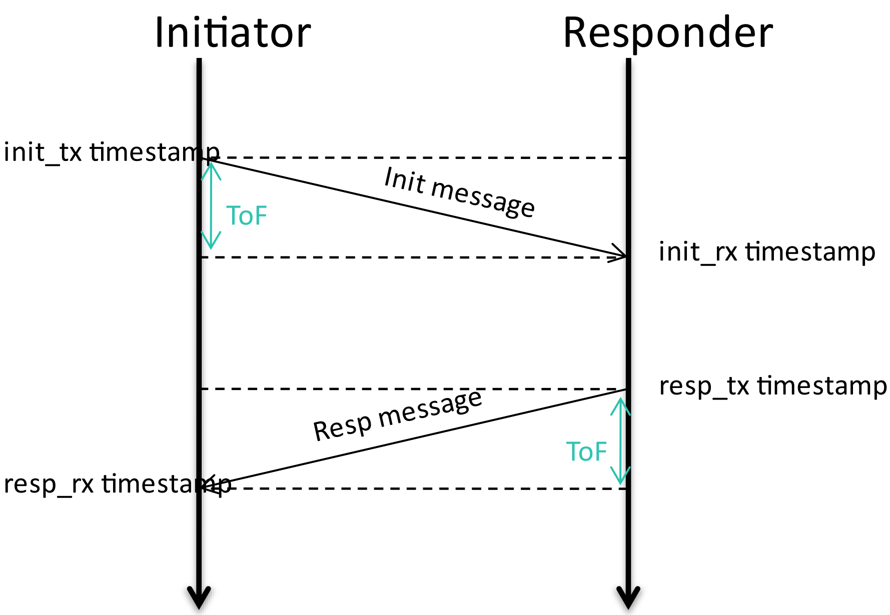


Figure 13: Two-Way Ranging

The initiator can calculate the time of flight (ToF) using the formula:

$$ToF = \frac{(resp_{rx} - init_{tx} - resp_{tx} + init_{rx})}{2}$$

Where:

- Resp_rx is the timestamp of the "resp" message reception.
- Init_tx is the timestamp of the "init" message transmission.
- Resp_tx is the timestamp of the "resp" message transmission.
- Init_rx is the timestamp of the "init" message reception.

The range is then:

$$Range = ToF * SpeedOfLight$$

The demonstration requires the use of an Android device. The Android application is available on Google Play (https://play.google.com/store/apps/details?id=com.insightsip.demouwb)

On Google Play, search "uwb ranging app" and download the App. The android App is a demonstration App that is provided "as is" to demonstrate the module capabilities.

Both Tag and Anchor boards come pre-flashed with the TWR_Demo firmware operating on Channel 5 with nominal output power.

4.2. Running the demo

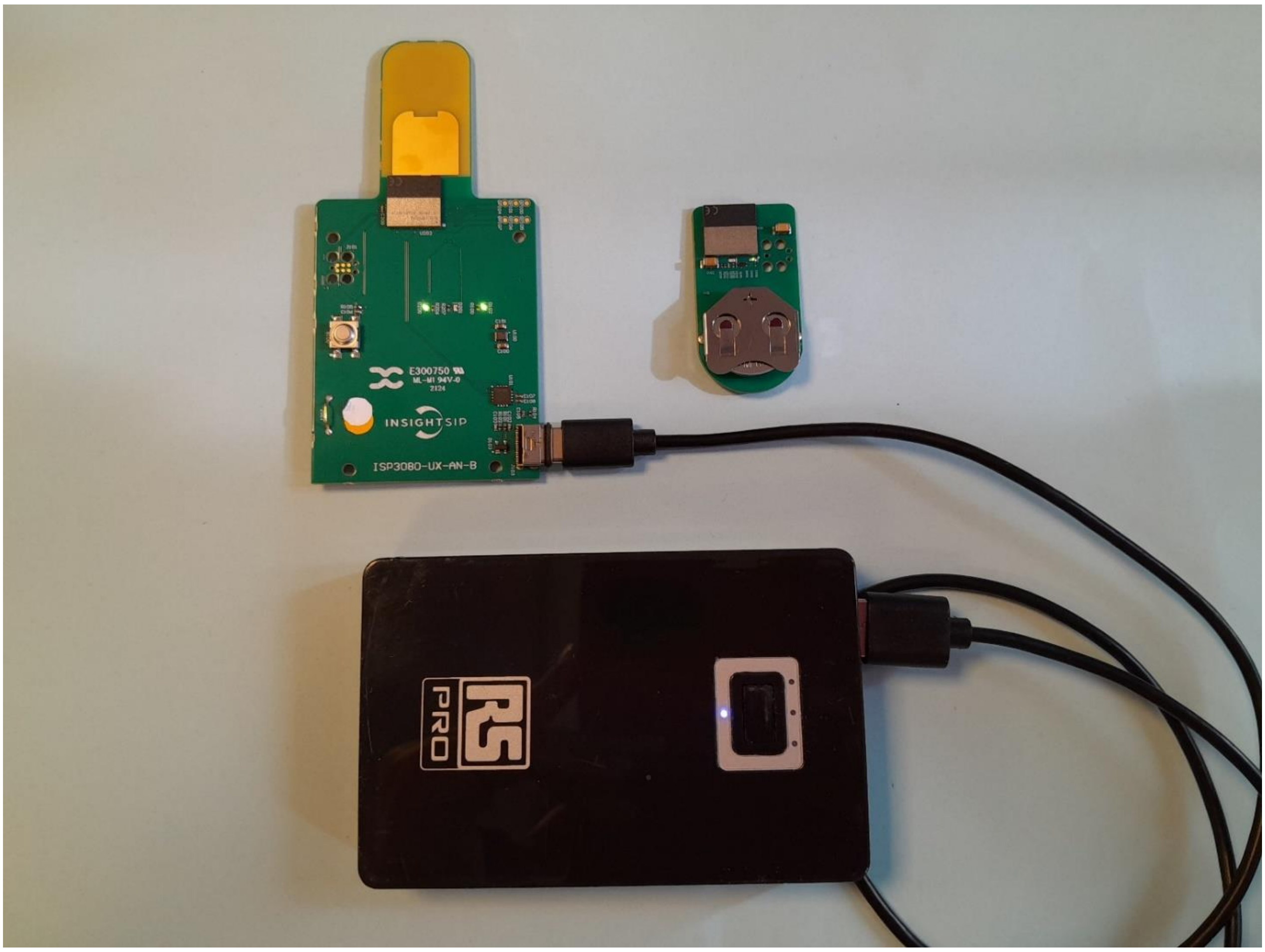


Figure 14: ISP3080-UX Ranging Demo setup

1. Power up both boards.
    - Insert a CRC2032 coin cell battery in the Tag.
    - Plug the Anchor into a USB power supply.
2. The demo immediately starts TWR operations. Check activity using LEDs.
    - On Tag, the green LED blinks for each successful TWR operation and the red LED blinks if the TWR operation fails or timeouts.
    - On Anchor there 3 green LEDs.. One for power supply, one for UWB RX activity and one for TX activity.
3. Start "Ranging Demo" application on your Android, you can establish a BLE connection with the Tag, the Anchor or both by clicking the "connect" buttons.

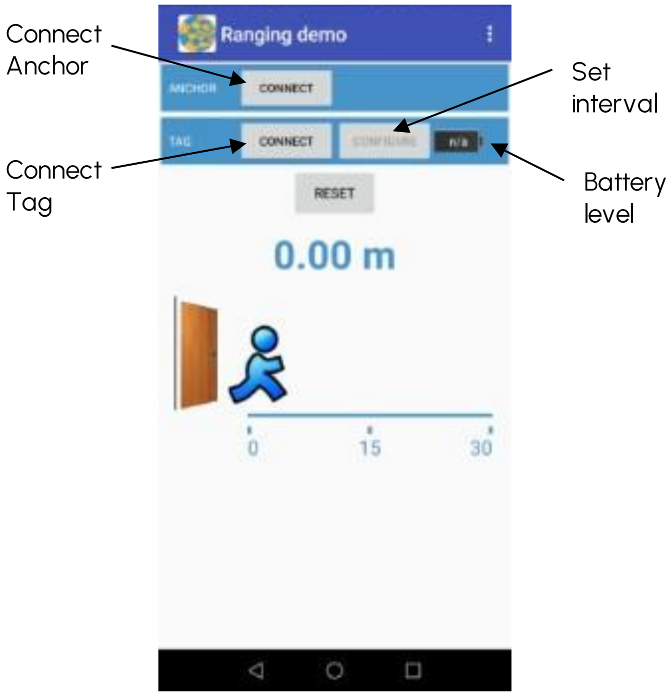


Figure 15: Ranging demo application

4. When BLE connection is established, the range between the Anchor and the Tag should be displayed on the App.

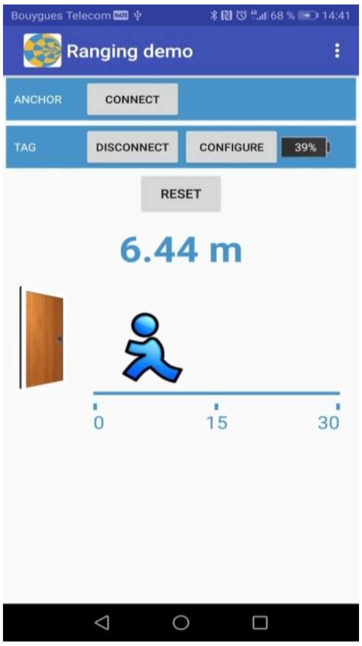


Figure 16: Ranging Demo application with displayed results

4.3. Tag current consumption

Since the Tag is powered by a CR2032 coin cell current should be minimized as much as possible to ensure good lifetime.

This is the current curve of a TWR event:

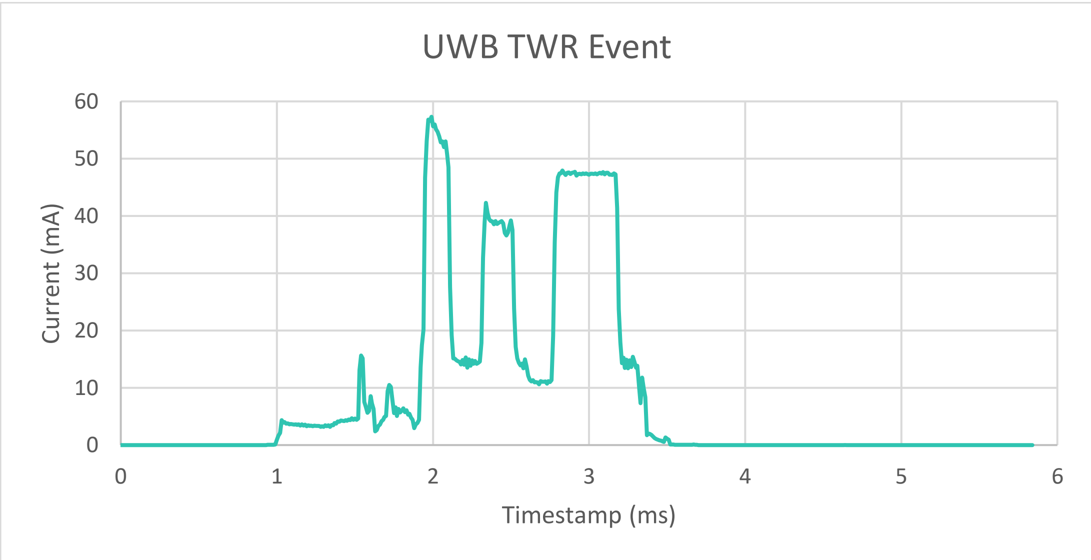


Figure 17: TWR current curve

The charge used by this event is around 50uC.

The event consists of 3 current pulses:

- Self RX calibration that must be performed each time the QM33110w wakes-up from sleep mode.
- Transmit of the initiator message
- Reception of the response message

This is the current curve of a BLE advertisement event:

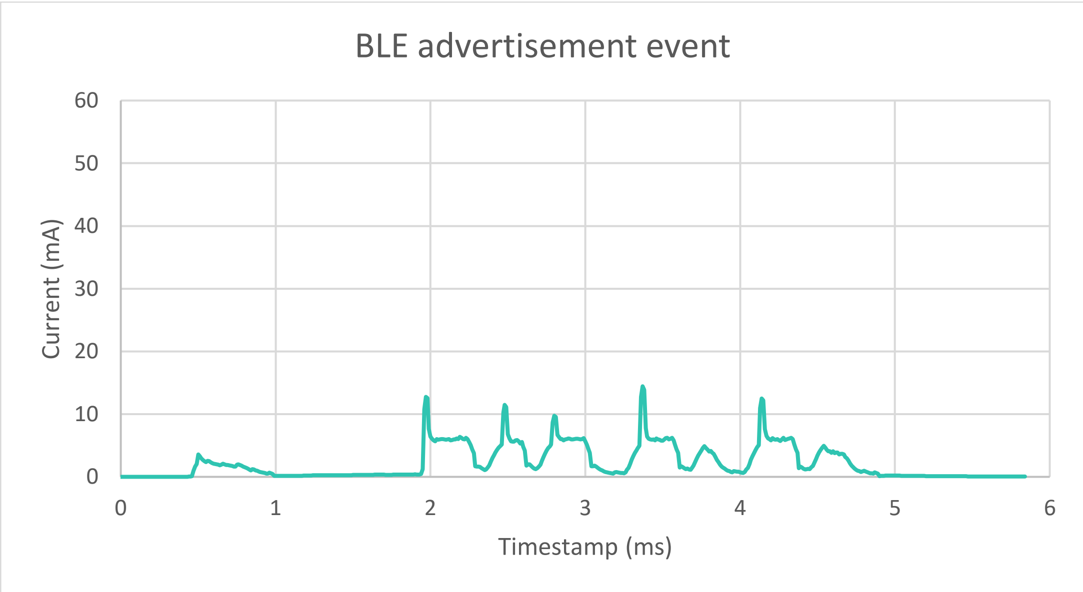


Figure 18: BLE advertisement current curve

The charge used by this event is around 13uC

Assuming 3 TWR events per seconds, 3 BLE Advertisements per seconds and a sleep current of ~5uA, the average current consumption of the TAG is 194uA. With a CR2023 (210mAh) the lifetime is 1082 hours or 45 days.

Note: To protect the CR2032 from the current peaks during TWR event the Tag implements a loading circuit that stores charge in a high value capacitor. The measurements were done by bypassing this circuit.

5. UWB Calibration

The module has 3 characteristics that require calibration:

- 38.4 MHz crystal trimming
- Antenna delay
- Output power to meet regulatory requirements

During production testing these 3 parameters have been tested and values written to an OTP memory inside the QM33110W device. Engineering deliveries only have the crystal trim values stored in the OTP.

5.1. OTP Map

The OTP Map of QM33110W taken from the Qorvo user manual is shown below:

QM33100 User Manual

Table 17: OTP memory map

Address	Size (Used Bytes)	Byte [3]	Byte [2]	Byte [1]	Byte [0]	Programmed By
0x00	4	64 bit EUID				Customer
0x01	4					
0x02	4	Alternative 64bit EUID (Selected via reg/SR register)				Customer
0x03	4					
0x04	4	LDOTUNE_CAL				Prod Test
0x05	4					
0x06	4	{"0001,0000,0001", "CHIP ID 5 nibbles (20 bits)"}				Prod Test
0x07	4	{"0001" , "LOT ID – 7 nibbles (28bits)"}				Prod Test
0x08	4	-	Vbat @ 3.0 V [23:16]	Vbat @ 3.62 V [15:8]	Vbat @ 1.62 V [7:0]	Prod Test
0x09	2				Temp @ 22 °C [7:0]	Prod Test
0x0A	0	BIASTUNE_CAL				Prod Test
0x0B	4	Antenna Delay – RFLoop				Prod Test
0x0C	4	AoA Iso CH9 RF2->RF1	AoA Iso CH9 RF1->RF2	AoA Iso CH5 RF2 -> RF1	AoA Iso CH5 RF1->RF2	Prod Test
0x0D	0	W.S. Lot ID [3]	W.S. Lot ID [2]	W.S. Lot ID [1]	W.S. Lot ID [0]	Prod Test
0x0E	0			W.S. Lot ID [5]	W.S. Lot ID [4]	Prod Test
0x0F	0		W.S. Wafer Number	W.S. Y Loc	W.S. X Loc	Prod Test
0x10	4					Customer
0x11	4					Customer
0x12	4					Customer
0x13	4					Customer
0x14	4					Customer
0x15	4					Customer
0x16	4					Customer
0x17	4					Customer
0x18	4					Customer
0x19	4					Customer
0x1A	4					Customer
0x1B	4					Customer
0x1C	4					Customer
0x1D	4					Customer
0x1E	2				XTAL_Trim[6:0]	Customer
0x1F					OTP Revision	Customer
0x20	4	RX_TUNE_CAL: DGC_CFG0				Prod Test
0x21	4	RX_TUNE_CAL: DGC_CFG1				Prod Test
0x22	4	RX_TUNE_CAL: DGC_CFG2				Prod Test
0x23	4	RX_TUNE_CAL: DGC_CFG3				Prod Test

67 of 285

Figure 19: QM33110 OTP memory map

Free-to-use customer address are used to put antenna delay and output power values obtained by calibration.

OTP Address	Size Used (Bytes)	Byte (3)	Byte (3)	Byte (3)	Byte (3)	Usage*
0x10	4	Rx Antenna Delay CH 5		Tx Antenna Delay CH 5		Prod
0x11	4	Rx Antenna Delay CH 9		Tx Antenna Delay CH 9		Prod
0x12	4					RFU
0x13	4					RFU
0x14	4	Sts_Pwr_Ch5	Shr_Pwr_Ch5	Phr_Pwr_Ch5	Dta_Pwr_Ch5	Prod
0x15	4	Sts_Pwr_Ch9	Shr_Pwr_Ch9	Phr_Pwr_Ch9	Dta_Pwr_Ch9	Prod
0x16	4					RFU
0x17	4					RFU
0x18	4					RFU
0x19	4					RFU
0x1A	4					RFU
0x1B	4					RFU
0x1C	4					RFU
0x1D	4					RFU
0x1E	1				Xtal_Trim (0:5)	Eng & Prod
0x1F	1				OTP Rev	Prod

*Usage:

- Prod = Production units from Date Code 2430 on
- Eng = Units with Date Code < 2430
- RFU = Reserved for Future Use

Note the values stored in OTP are for the standard operating settings with PRF 64 MHz. For other customer settings regulatory compliance must be obtained independently by the customer.

5.2. 38.4 MHz Crystal Trimming

During the production test the 38.4 MHz crystal is tuned to within 2 ppm of its nominal frequency at 25°C. The trim value is stored in the 6 lowest bits of OTP register 0x1E (range 0 to 63)
The programmed value will be read automatically from the OTP register and applied to the crystal. If required, the value in the Crystal Trim register can be obtained with the Qorvo call:

uint8_t value = dwt_getxtaltrim();

5.3. Antenna delay

Antenna delay is a parameter that allows for variations in the exact time of transmission and reception of UWB frames. When the value for transmission and reception "antenna delays" are correctly set the time stamps for transmission and reception correspond exactly to the time the signals leave and arrive at the antenna. This ensures accurate distance measurements.

For a given antenna design the antenna delay varies essentially due to small internal variations from UWB chip to chip. During production testing antenna delays for Tx and Rx are written into OTP memories 0x10 (channel 5) and 0x11 (Channel 9).

Since each antenna delay is a 16-bit word the Tx and Rx antenna delays (*tx_ant_dly and tx_ant_dly)* are stored in a single 32-bit OTP memory:

tx_ant_dly is stored in 16 LSBs
rx_ant_dly is stored in 16 MSBs

OTP memory value = *tx_ant_dly* + (*rx_ant_dly* << 16).

These antenna delays are given for signals that are in the linear range of the receiver. The calibration is carried out at signal levels equivalent to operating at 10m. For situations where better than 20cm precision is required at less than 1m range a series of measurements should be carried out in the project's physical environment and a small correction applied to the distance measurements as a function of distance or RSSI.

The values of antenna delay are valid when using the internal antenna. If an external antenna is used the offset between the calibrated delays and those for the external antenna can be obtained by distance measurements of a pair of devices. Once the offsets are known they can be applied to the OTP values without carrying out measurements for each device.

To use the OTP antenna delays the following lines of code are used during initialisation:

```
if (channel == 5) {
      rx_ant_dly = otp_memory[OTP_CH5_ANT_DLY_ADDR] != EMPTY_OTP_VAL ?
(otp_memory[OTP_CH5_ANT_DLY_ADDR] >> 16) & 0xFFFF : DEFAULT_RX_ANT_DLY;
      tx_ant_dly = otp_memory[OTP_CH5_ANT_DLY_ADDR] != EMPTY_OTP_VAL ?
otp_memory[OTP_CH5_ANT_DLY_ADDR] & 0xFFFF : DEFAULT_TX_ANT_DLY;
      dwt_setrxantennadelay(rx_ant_dly);
      dwt_settxantennadelay(tx_ant_dly);
   } else if (channel == 9) {
      rx_ant_dly = otp_memory[OTP_CH9_ANT_DLY_ADDR] != EMPTY_OTP_VAL ?
(otp_memory[OTP_CH9_ANT_DLY_ADDR] >> 16) & 0xFFFF : DEFAULT_RX_ANT_DLY;
      tx_ant_dly = otp_memory[OTP_CH9_ANT_DLY_ADDR] != EMPTY_OTP_VAL ?
otp_memory[OTP_CH9_ANT_DLY_ADDR] & 0xFFFF : DEFAULT_TX_ANT_DLY;
      dwt_setrxantennadelay(rx_ant_dly);
      dwt_settxantennadelay(tx_ant_dly);
   } else {
      NRF_LOG_ERROR("dwt_setrxantennadelay/dwt_settxantennadelay failed");
      return NRF_ERROR_INTERNAL;
   }
```

If, for some reasons, there is no calibration value in OTP memory, default values can be applied in order to get acceptable results:

```
#if defined(BOARD_ISP3080_UX_TG)
#define DEFAULT_TX_ANT_DLY 14445
#define DEFAULT_RX_ANT_DLY 18385
#elif defined(BOARD_ISP3080_UX_AN)
#define DEFAULT_TX_ANT_DLY 14453
#define DEFAULT_RX_ANT_DLY 18394
#endif
```

5.4. Output Power

In order to satisfy regulatory limits, the output power setting for maximum authorised power (-41.3dBm/MHz EIRP) is measured for each module and the value stored in OTP.
Values for both Channel 5 and Channel 9 are available using the standard 64 MHz PRF.

Power can be controlled for each portion of the UWB frame:

STS, SHR, PHR and DTA.

The power in each portion of the frame is controlled by 8 bits:

- Bits (0:1) provide coarse control. Values used are 0, 1, 2 corresponding to 0 3 6 dB gain respectively.
- Bits (2:7) provide fine control with a total of 25 dB range from 0 to 63 on a non-linear scale

Power in the PHR section is typically reduced by approximately 6dB to ensure regulatory conformity.

Further details concerning power control are given in QM33110 user manual, section:

8.2.2.20 Sub-register 0x01:04 – Transmit power control

To use the OTP Output power the following lines of code are used during initialisation:

```
if (channel == 5) {
      tx_config.power = otp_memory[OTP_CH5_PWR_ADDR] != EMPTY_OTP_VAL ?
otp_memory[OTP_CH5_PWR_ADDR] : DEFAULT_CH5_PWR;
      tx_config.PGdly = DEFAULT_CH5_PGDLY;
      dwt_configuretxrf(&tx_config);
      dwt_set_alternative_pulse_shape(0);
   } else if (channel == 9) {
      tx_config.power = otp_memory[OTP_CH9_PWR_ADDR] != EMPTY_OTP_VAL ?
otp_memory[OTP_CH9_PWR_ADDR] : DEFAULT_CH9_PWR;
      tx_config.PGdly = DEFAULT_CH9_PGDLY;
      dwt_configuretxrf(&tx_config);
      dwt_set_alternative_pulse_shape(1);
   } else {
      NRF_LOG_ERROR("dwt_configuretxrf failed");
      return NRF_ERROR_INTERNAL;
   }
```

If, for some reasons, there is no calibration value in OTP memory, default values can be applied in order to get acceptable results:

```
#define DEFAULT_CH5_PWR 0xf2f2f0f2 /**< Default TX power value for channel 5 */
#define DEFAULT_CH9_PWR 0xa2a2a0a2 /**< Default TX power value for channel 9 */
#define DEFAULT_CH5_PGDLY 0x34     /**< Default PG Delay value for channel 5 */
#define DEFAULT_CH9_PGDLY 0x27     /**< Default TX power value for channel 9 */
```