AN12846

K32W Power Consumption Analysis

Rev. 1 — 30 May 2022 Application Note

Contents

1 Introduction

This document provides information about the power consumption of K32W wireless MCU, how the hardware is designed and optimized for low-power operation, and how the software is configured to achieve the best low-power profile. Note that in this document only K32W is mentioned but the measurements apply for the complete family of products including K32W061 and K32W041. This document provides an overview and guidance on how to achieve the best low-power profile while still keeping the high performance of the system. The setup and the procedures to measure the current consumption of the K32W chip is also describes in this document.

The power consumption of wireless devices is a critical requirement for the fast-coming Internet of Things (IoT) world. As a result, the hardware has been gradually improved and optimized from the power consumption perspective and new communication standards have been developed. Bluetooth Smart also known as Bluetooth® Low Energy (Bluetooth LE) and IEEE 802.15.4 compliant, it is part of these new standards that have been developed for long-term battery operation, typically years.

K32W is a radio wireless MCU that supports Bluetooth LE v5.0 and 802.15.4 protocols.

The prerequisites for understanding this document are that the reader has a good knowledge about Bluetooth Smart and Zigbee protocols, as well as basic knowledge about ARM MCU architecture, and radio communication basics.

2 Bluetooth Low Energy application

2.1 Smart power metrics

2.1.1 Communication steps

During a Bluetooth Low Energy communication steps, both MCU and radio are in different state at different moment. They are either in active, sleep, or deep sleep mode. Compared to all other operation modes, the time spent in sleep/deep sleep is the longest time, as shown in Figure 1.

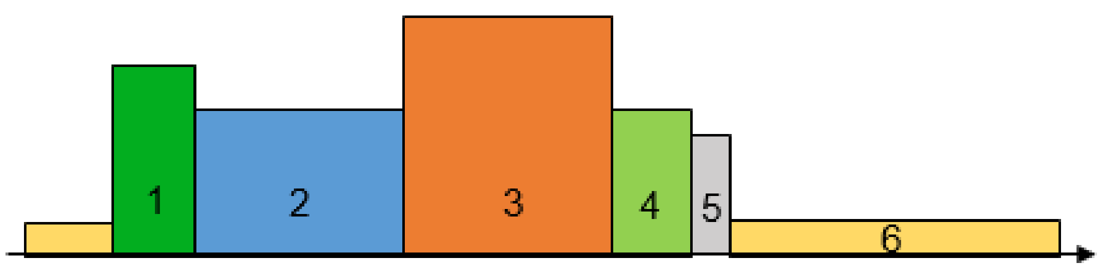


Figure 1. Bluetooth Low Energy communication steps

1. The MCU is woken up and performs system initialization and pre-processing.
2. The radio transceiver is woken up and ready to operate. The MCU may enter the STOP mode of the MCU if the software allows it.
3. The radio transceiver is performing one or more RX/TX sequences.
4. The MCU is processing the received or transmitted packets.
5. The radio transceiver is put back in sleep mode.
6. The MCU enters low-power (sleep/deep sleep mode).

2.1.2 Bluetooth Low Energy events

Two events are considered during a Bluetooth Low Energy (Bluetooth LE) communication. They are described in the following sections and their power consumption is described in Advertising mode and Connect mode.

2.1.2.1 Advertising mode

All communications between two devices start from advertising events.

Devices can advertise for various reasons:

- To broadcast promiscuously
- To transmit signed data to a previously bonded device
- To advertise their presence to a device wanting to connect
- To reconnect asynchronously due to a local event

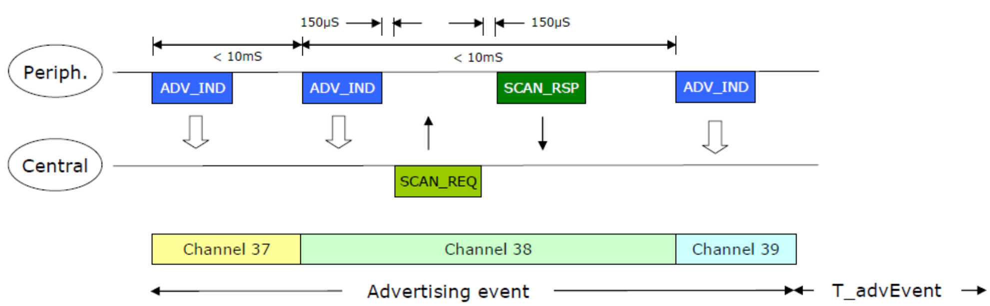


Figure 2. Communication start process between two devices

2.1.2.2 Connect mode

Once a connection is established:

- Master informs slave of hopping sequence and when to wake
- All subsequent transactions are performed in the 37 data channels (from channel 0 to channel 36)
- Transactions can be encrypted
- Both devices can go into deep sleep between transactions

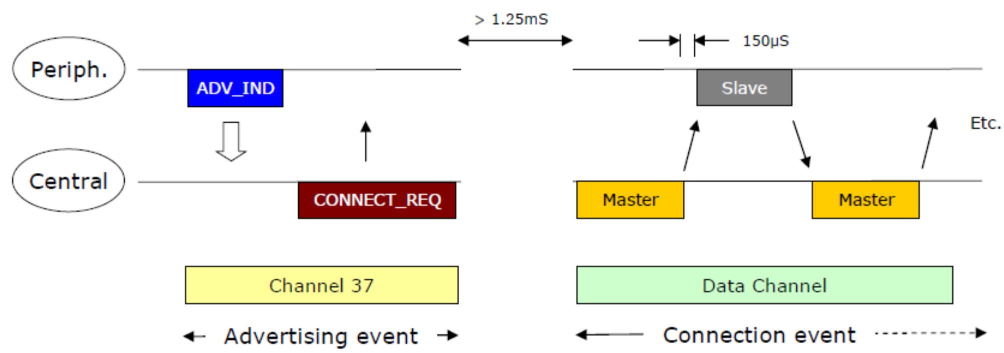


Figure 3. Connection mode process between two devices

- It occurs even when one (or both) sides have no data to send:
  - — Occurs periodically
  - — Slave devices can use slave latency, that is number of times it can ignore connection events from master when there is no data to transmit

2.2 Getting ready for low-power measurement

Specific applications software and hardware are used for the current measurements. The below sections describe how to set the hardware and the softwareto set the device in different modes to perform the low-powermeasurement. All the low-power measurements are done on the slave device.However, a master device will also be needed to perform measurement in connected mode.

2.2.1 Hardware configuration

The power consumption is measured on an optimized DK6 board for low-power tests with K32W module fitted as slave. For the needed hardware modification, see **Chapter 7** in *IOTZTB-DK006 Development Kit User Guide* (document UM11393).

The test set-up block diagram is shown in Figure 4.

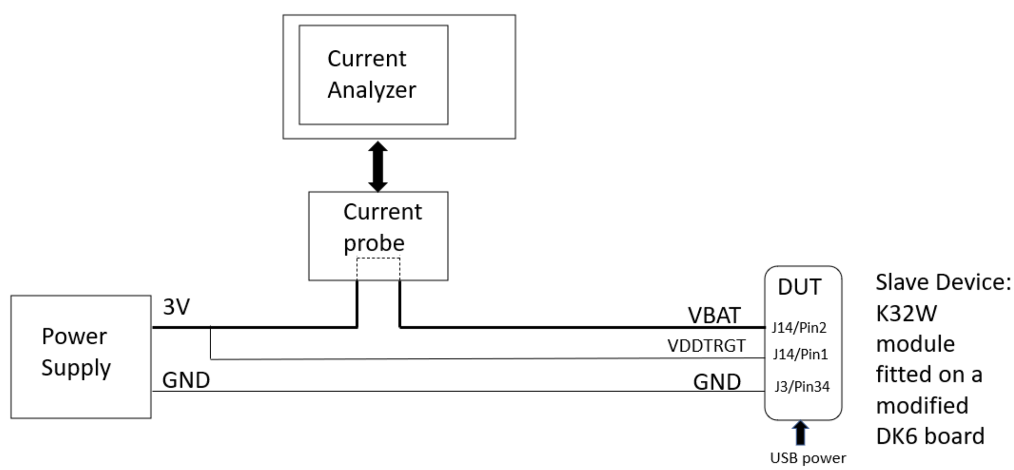


Figure 4. Test set-up block diagram for power profile measurement

The current probe of the current analyzer (Keysight CX3322A for instance), in series with external power supply, is connected to J14 on pin 2. From a supply standpoint VBAT = VDDTRGT.

Figure 5 shows the test connections.

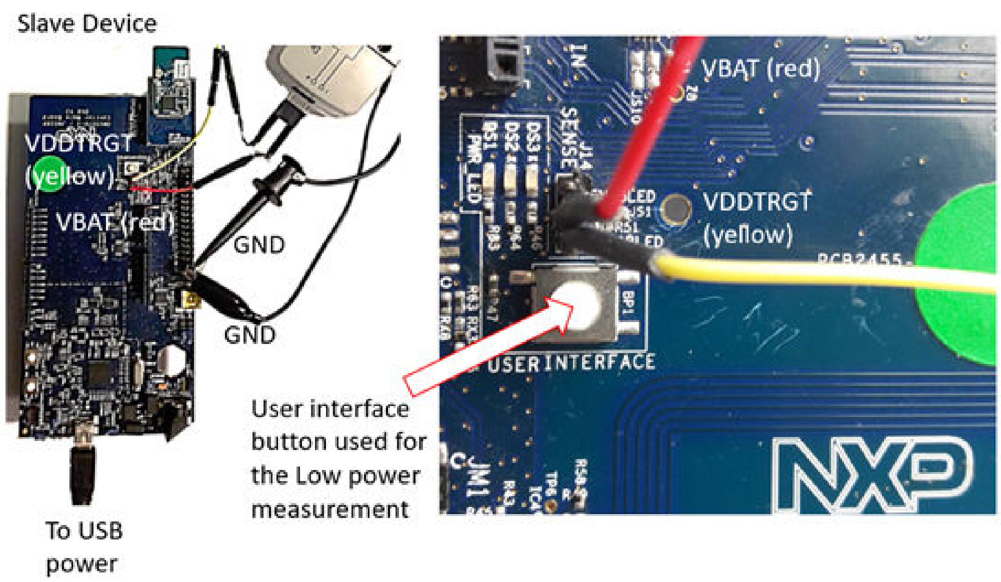


Figure 5. Hardware connection for Low-power measurement (external power supply)

NOTE

Especially for the deep power-down currents, a power supply capable of measuring low currents such as Keysight B2902A Source/Measure Unit, is preferred to the current waveform analyzer. Figure 6 shows the test set-up block diagram.

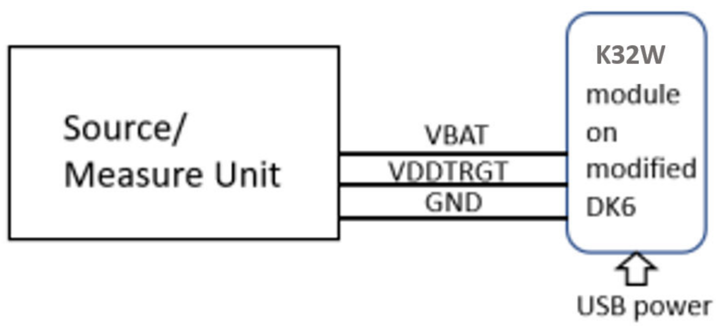


Figure 6. Test set-up block diagram for deep power-down currents with a Source/ Measure Unit

2.2.1.1 K32W DC-DC converter modes

On both the devices, only the Buck mode is enabled.

For more details, see *Using the DC-DC Feature* (document AN12893).

2.2.1.2 K32W Low-Power modes

Table 1 describes Power-Down currents from the datasheet.

Table 1. Deep Power-Down and Power-Down modes

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
I_{DD}	Supply current	Deep Power-Down (everything is powered off, wake-up on HW reset only)	—	250	—	nA
		Deep Power-Down-IO (everything is powered off, wake-up on HW reset only or an event on any of the 22 GPIOs and NTAG interrupt)	—	350	—	nA
		Power-Down (wake-up on HW reset or an IO event, wake-up timer ON, 32 kHz FRO on, no SRAM retention)	—	800	—	nA
		Power-Down-4K (wake-up on HW reset or an IO event, wake-up timer on, 32 kHz FRO on, with 4 KB SRAM retention)	—	1025	—	nA
		Power-Down-8K (wake-up on HW reset or an IO event, wake-up timer on, 32 kHz FRO on, with 8 KB SRAM retention)	—	1120	—	nA

NOTE

Each additional 4 KB consumes around 105 nA.

On the slave device, the modes are selected when User interface button (BP1) is pushed.

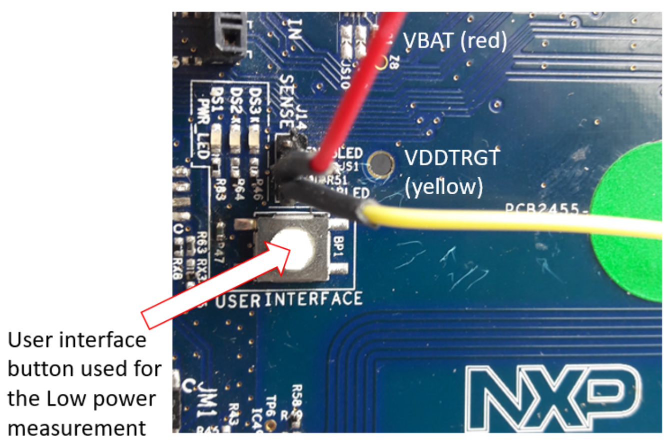


Figure 7. User interface button location

The user can evaluate the low-power currents in the configurations described below by pressing on the user interface button to navigate from one state to the next.

Table 2. Low-power application modes

Button press	Mode	Corresponding datasheet mode	Comments	Reference value @ 3 V
0/ power ON	Power-Down mode	Power-Down (wake-up on HW reset or an IO event, wake-up timer ON, 32 kHz FRO on, no SRAM retention).	Bluetooth LE device is in Power-Down mode where all blocks are de-activated except low-power timer and K32W is waiting for internal wake up event from its low-power timer or IO event (no memory retention).	800 nA
1	DPD-IO	Deep Power-Down-IO. Everything is powered off, wake-up on HW reset only or an event on any of the 22 GPIOs and NTAG interrupt) (for K32W061)	Bluetooth LE device is in Deep Power-Down mode where all block are de-activated and K32W is waiting for an IO event (no memory retention).	350 nA
2	Sleep	N/A	Sleep mode Bluetooth LE Device is in a Sleep mode where most of blocks are activated and SRAM content is maintained.	1 .9 mA
3	Rx Idle	-	Bluetooth LE device is fully operational and able to receive and transmit at any time.	2.4 mA (receiving or ready to receive, No TX)
4	Advertising mode The current measured is between two advertising events	Power-Down 36 KB (between Advertising events)	Between two advertising events, the Bluetooth LE device is in Power-Down mode where several blocks are off except the low-power timer, and the SRAM retention is at 36 KB to ensure fast wake-up. This mode can also be used between connection events. NOTE If time accuracy is required for connection event wake up, XTAL 32 kHz shall be used instead of FRO 32 kHz.	1.9 µA

2.2.2 Software configuration

WARNING

WARNING

The software is configured for two devices namely: Master and Slave.

- The slave is configured to advertise and then connected to master.
- The master is configured to scan and connect to the slave device.

As mentioned before, all the low-power measurements are done on the slave device. However, a master device will also be needed to achieve the measurements in connected mode. This can be achieved in two ways:

- By using a second DK6 boards mounted with its corresponding K32W modules (note that there are already three in each IOTZTB-DK006 kit content) connected to a computer.
- By using NXP IoT Toolbox mobile phone application.

For the different configuration for Master side, see K32W module acting as Master .

2.2.2.1 Preparing the Slave software

The software can be uploaded from the SDK. The slave application is named K32Wdk6_power_profiling_bm. To upload the bin files in the master and slave devices, a flash programmer is available. Instructions are described in the application note JN-SW-4407 which is contained within the Tools folder of the SDK. To perform the power measurements, K32Wdk6_power_profiling_bm.bin must be flashed on the slave device.

2.2.2.1.1 Software modification required to the power profiling project

To achieve the power setting, the following changes are required to the power profiling project in the SDK. An explanation on how to import the application is detailed in “Getting Started with MCUXpresso SDK for K32W.pdf” which is delivered with the K32W SDK. Once the application is imported, the following files and changes are required prior to compiling and testing.

2.2.2.1.1.1 App_preinclude.h

Because the app only requires one button to switch from one power state to another, the following lines inside app_preinclude.h of the power profile project need to be changed as follow:

```
/* Defines the number of available keys for the keyboard module: counting buttons and TSI keys  */
#define gKBD_KeysCount_c                1
```

The LEDs that are detailed in this document show the mode that the device is in. The LEDs need to be disabled to achieve the low-power values detailed in this document. Therefore, the following line of code needs to change as below:

```
/* Specifies the number of physical LEDs on the target board */
#define gLEDsOnTargetBoardCnt_c         0
```

The UART debug also needs to be disabled and this is done by adding the following line of code:

```
/* disable debug */
#define SDK_DEBUGCONSOLE DEBUGCONSOLE_DISABLE
```

To prevent any compilation issue, you will also need to remove the definition of SDK_DEBUGCONSOLE in the project settings by removing the below preprocessor symbol:

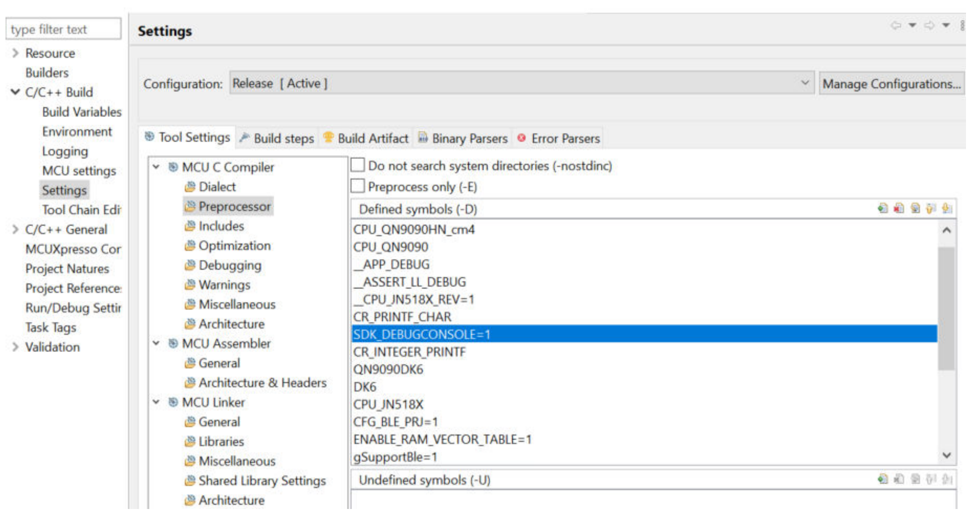


Figure 8. Power profiling preprocessor settings

2.2.2.1.1.2 powerprofile.h

Average power consumption varies with the value of interval in both advertising and connection event. The less the interval, the higher the power consumption. Interval tuning for specific application is crucial to leverage between power consumption and transaction time. Advertising interval can be adjusted by modifying macro definition in project file.

Advertising interval modification can be accomplished by changing the values of macro definitions in file power_profiling.h of the power_profiling project. The default values are:

```
#define gReducedPowerMinAdvInterval_c   1600 /* 1 s */
#define gReducedPowerMaxAdvInterval_c   4000 /* 2.5 s */
```

By default, during the advertising interval, two macros do not share the same values, so that the Bluetooth LE controller can select a proper one. To have a known and fixed interval, these two macros take the same value in the unit of 0.625 ms. In the case of a specified 1000 ms interval, a value 1600 = (1000 ms / 0.625 ms) is set for the two macros. Therefore, the values need to be changed to:

```
#define gReducedPowerMinAdvInterval_c   1600 /* 1 s */
#define gReducedPowerMaxAdvInterval_c   1600 /* 1 s */
```

2.2.2.1.1.3 powerprofile.c

In the file powerprofile.c of the power profile project the following 2 lines of code need to be added to confirm that the radio setting is set to 0 dBm, as defined in board.h. The lines of code must be added after the function call to Bas_Start (&basServiceConfig).

```
/* Set power level - default in board.h but can be overwritten in app_preinclude.h  */
Gap_SetTxPowerLevel(gAdvertisingPowerLeveldBm_c, gTxPowerAdvChannel_c);
Gap_SetTxPowerLevel(gConnectPowerLeveldBm_c, gTxPowerConnChannel_c);
```

2.2.2.1.1.4 Gpio_pins.h

Changes are only required within this file if the user is not using a DK6 board as the slave. In this example the slave device is a modified DK6 with a K32W module fitted which uses user interface (GPIO 1) to change functionality. Therefore the following needs to be change , to represent a GPIO that is a input/button/switch on their designed board:

```
#define BOARD_USER_BUTTON1_GPIO_PIN          1U
```

After all these needed modification are done, the slave application is ready to be compiled and flash into the K32W module. The next needed part is to configure the master device.

2.2.2.2 Preparing the Master software

As mentioned before, the master can be either a K32W module plugged on the DK6 board (JM1/JM2), properly configured and connected to a laptop or a smartphone with NXP IoT Toolbox application installed. In this chapter, we will explain how to setup and use both configuration.

2.2.2.2.1 K32W module acting as Master

To use as K32W module as a master, the easiest solution is to use the K32Wdk6_ble_fsci_black_box_bm application. This application can be found in the SDK and must be flashed on the master device. It must be used with the Connectivity Qtool application running on a laptop to control the master device. Connectivity Qtool can be downloaded from here.

1. Run Connectivity Qtool on the PC. The screen is displayed, as shown in Figure 9.

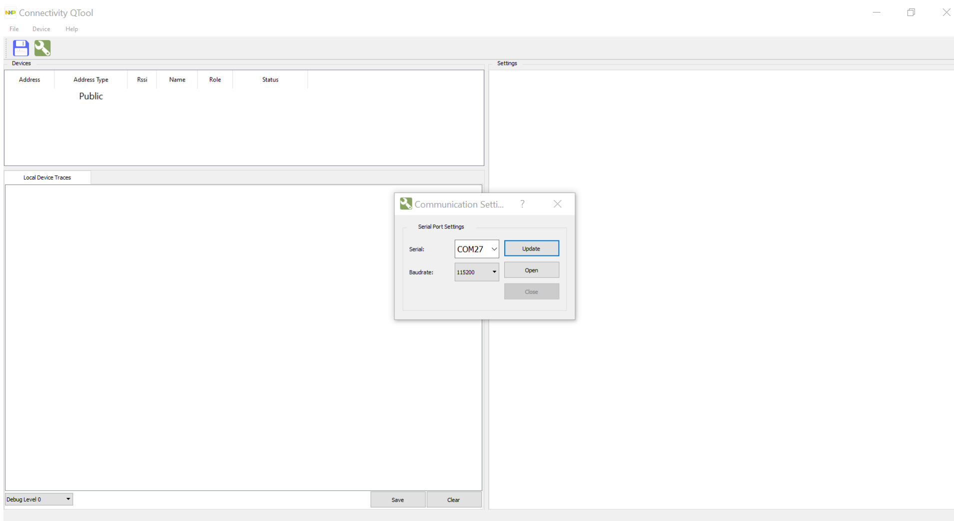


Figure 9. Master device communication port opening

2. Select the serial port of the master device and click **Open**.
3. A device address appears in the **Devices** window.

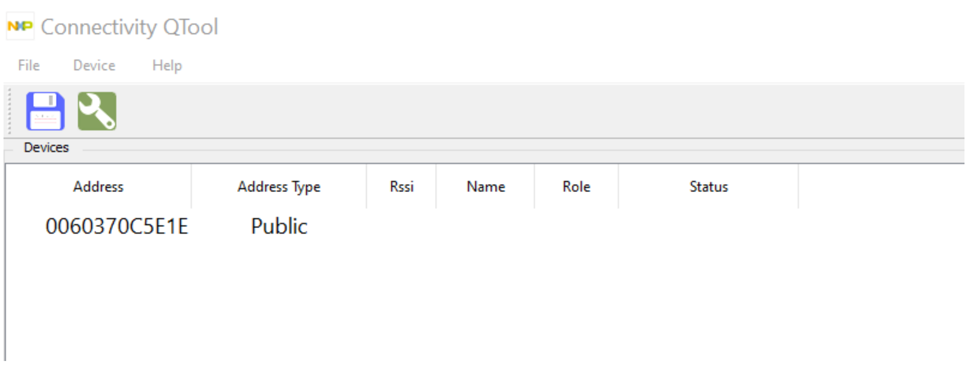


Figure 10. Master device address (example)

4. To start the scanning of the master device, select the device and click on **Start Scanning** button.

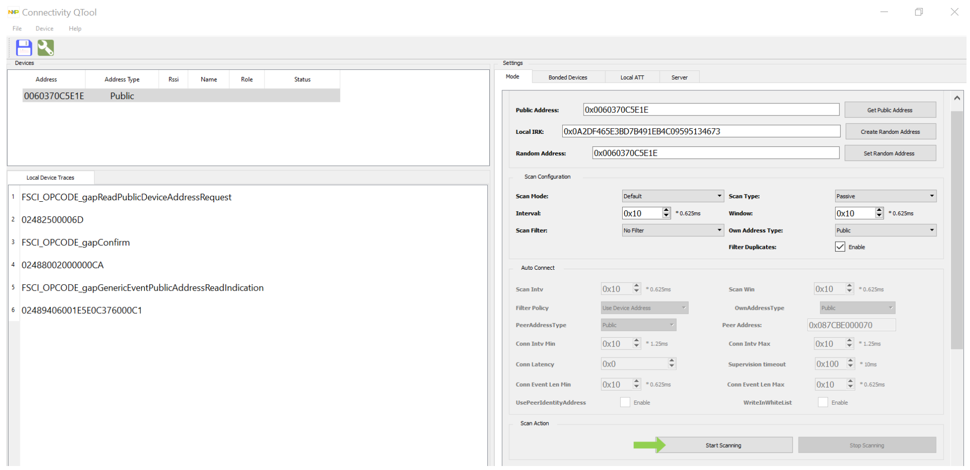


Figure 11. Master device scanning start

5. If the slave is advertising (if not, press the user interface button on the slave device to start it), it should appears in the **Devices** menu.

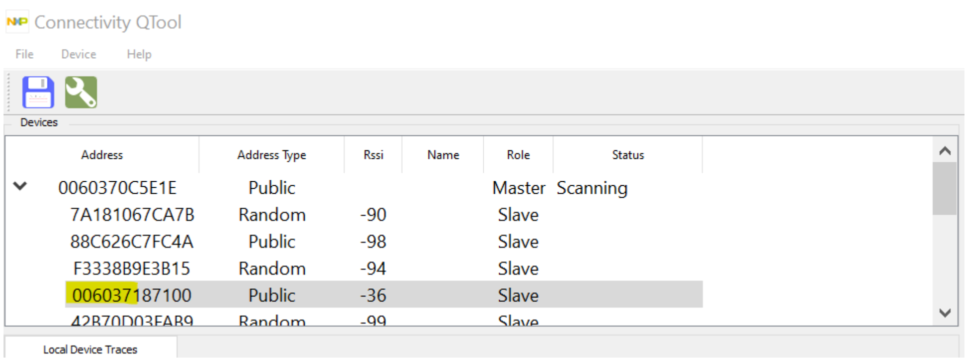


Figure 12. Slave device address (example)

Once the slave is scanned, scanning can be stopped with Stop Scanning button in Qtool.

6. Select the slave device (left click on it) and set the connection parameter in the right panel. In our test, the connection interval is set to 100ms (80 * 1.25 ms)

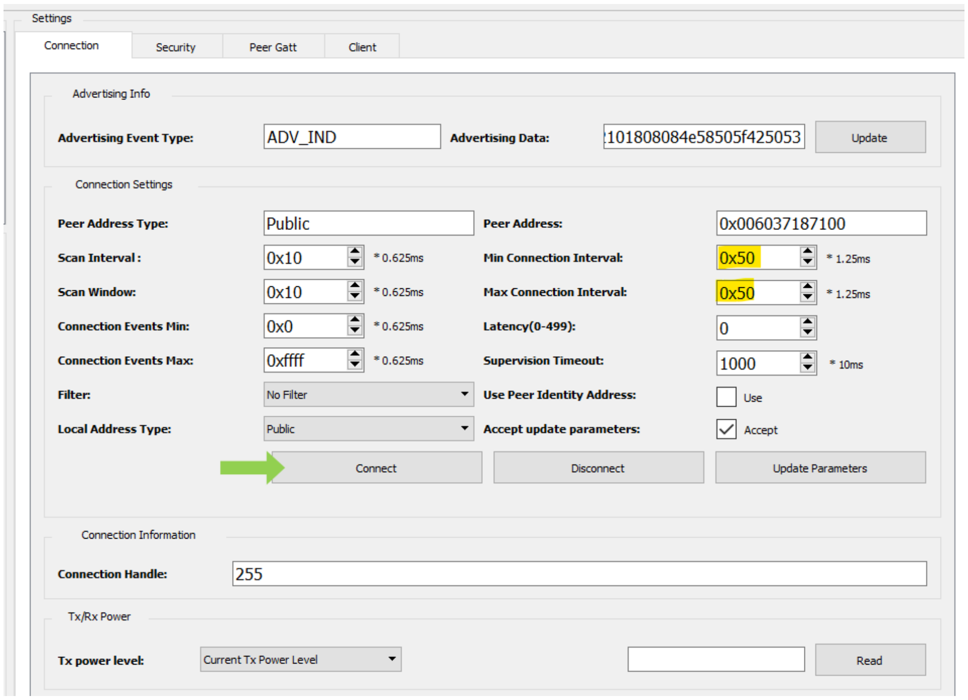


Figure 13. Connection to the slave device

2.2.2.2.2 Smartphone acting as Master

Test set-up: A smartphone with IoT Toolbox application (available on Google Play as well as on Apple App Store) and the K32W fitted on modified DK6 with the slave binary on the modified DK6 board as described in the above sections.

The procedure is as follows:

1. Open the IoT toolbox application on your cell phone.
2. Select the Blood Pressure icon.
3. The cell phone automatically starts scanning for available slave devices.
4. On the slave device, press sequentially four times on the user interface button to go in Advertising mode.
5. The cell phone detects the slave device. Figure 14 shows the cell phone application prior to connecting to the slave.

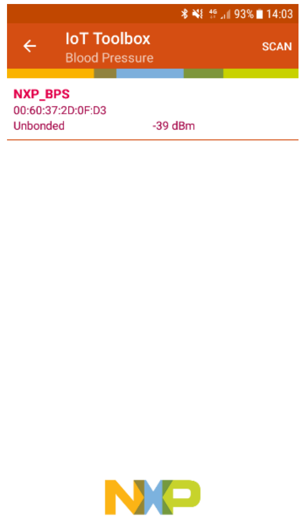


Figure 14. Screen capture of scanning step

The slave device is detected as NXP_BPS in this example.

6. Click on the NXP BPS device on your cell phone application to connect to it.

   When connected, the data is received and can be displayed as shown in Figure 15.

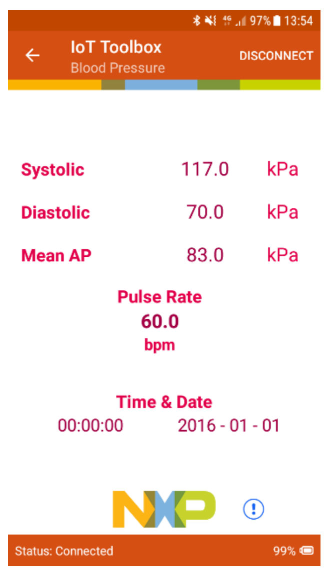


Figure 15. Screen capture of connect step

The measurements can be run on the slave device in the same way as with the K32W FSCI Black Box application as master.

2.3 Low-power results

2.3.1 Low-Power modes

Table 3. Low-Power modes

Button pressure #	Reference values	Measurement @ 3 V	Unit
0 / power ON	800	890	nA
1	350	330	nA
2	1.90	2.20[1]	mA
3	2.40	2.31	mA
4	1.90	1.80	µA

1. The current can be lowered a little without the Pull Up resistance (R47) and be closer to 1.90 mA

2.3.2 Advertising mode

Device configuration:

- TX output power: 0 dBm
- RAM retention: 36 KB
- Payload: 13 B
- Clock: 32 MHz

After pressing four times the user interface switch on the slave device, the advertising mode is launched as described in Table 2.

A sniffer is used to capture the over the air communication. Figure 16 is an example:

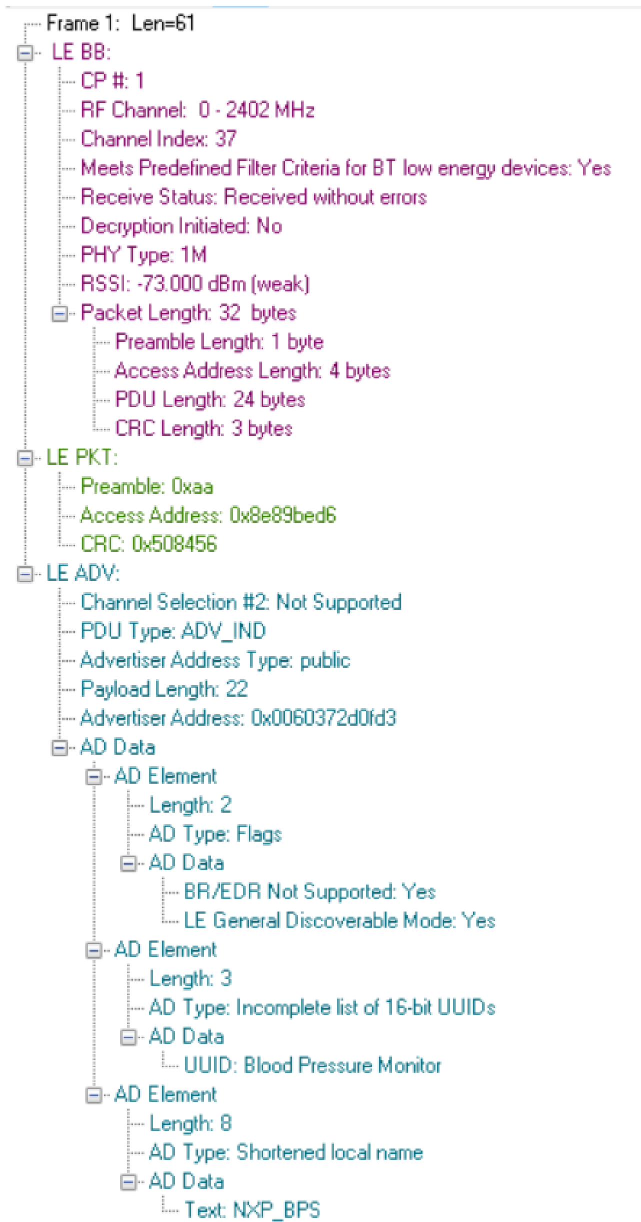


Figure 16. Advertising sniffer log

At this point, the current can be measured. Figure 17 shows the waveform of the advertising event.

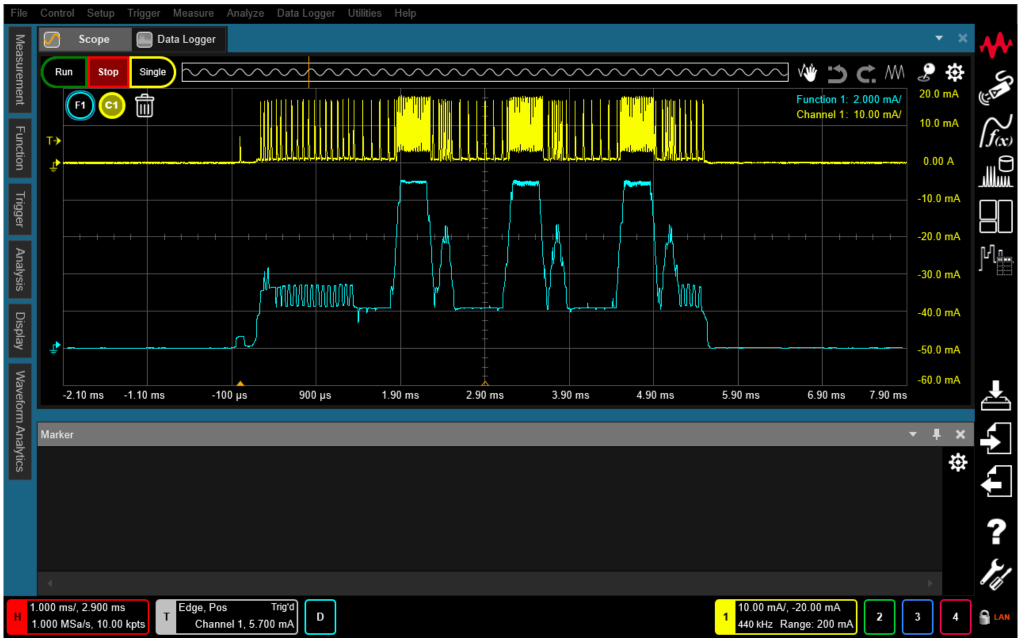


Figure 17. Advertising event

The power profile is analyzed and the energy consumption can be measured at 6.288 nAh under 3.0 V.

The total event duration 5.640 ms.

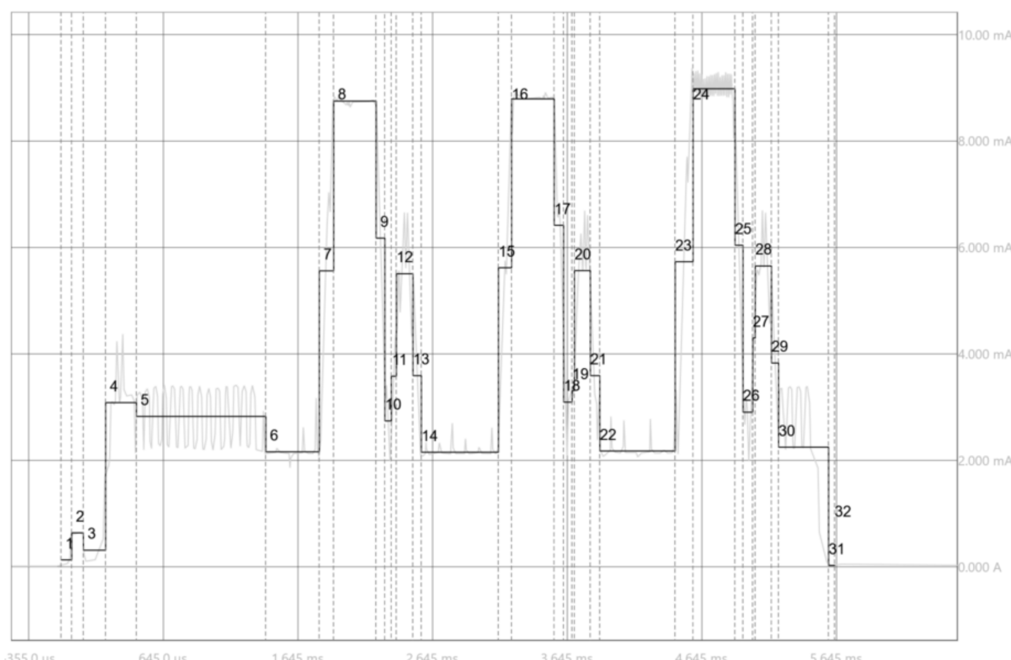


Figure 18. Advertising power profile

Table 4. Advertising break down table

Event	Duration	Current at 3 V
1. SoC in Power-Down mode	—	1.80 µA
2. SoC awakes from Power-Down mode	86.7 µs	638.7 µA
3 to 6. Pre-processing [including stack and Radio initialization]	1.752 ms	311.2 µA to 3.085 mA
7. TX warm up	106.8 µs	5.561 mA
8. Active TX	314.9 µs	8.750 mA
9 to 11. TX to RX	150.1 µs	4.439 mA
12. Active RX	123.0 µs	5.507 mA
13. Rx warm down	62.75 µs	3.592 mA
14. MCU STOP	572.3 µs	2.153 mA
15. TX Warm up	97.89 µs	5.620 mA
16. Active TX	316.3 µs	8.793 mA
17 to 19. TX to RX	150.6 µs	4.273 mA
20. Active RX	118.0 µs	5.565 mA
21. Rx warm down	70.28 µs	3.592 mA
22. MCU STOP	559.7 µs	2.176 mA
23. TX Warm up	133.0 µs	5.732 mA
24. Active TX	311.2 µs	8.982 mA
25 to 27. TX to RX	150.6 µs	4.415 mA
28. Active RX	120.5 µs	5.652 mA
29. Rx warm down	52.71 µs	3.830 mA
30. Post processing	416.68 µs	2.271 mA
31. Power Down mode	—	1.80 µA

NOTE

With a 48 MHz clock, the total energy is 5.603 nAh under 3.0 V.

2.3.3 Connect mode

Device configuration:

- TX output power: 0 dBm
- RAM retention: 36 KB
- Payload: 13 B
- Clock: 32 MHz

When the advertising is launched on the slave device, the scanning and connect on the master can be started as described in Preparing the Master software. In the next measurement results, a connection interval of 100 ms was used.

Figure 19 shows the connect waveform. C1 is the current waveform, and F1 is the smooth filtered one.

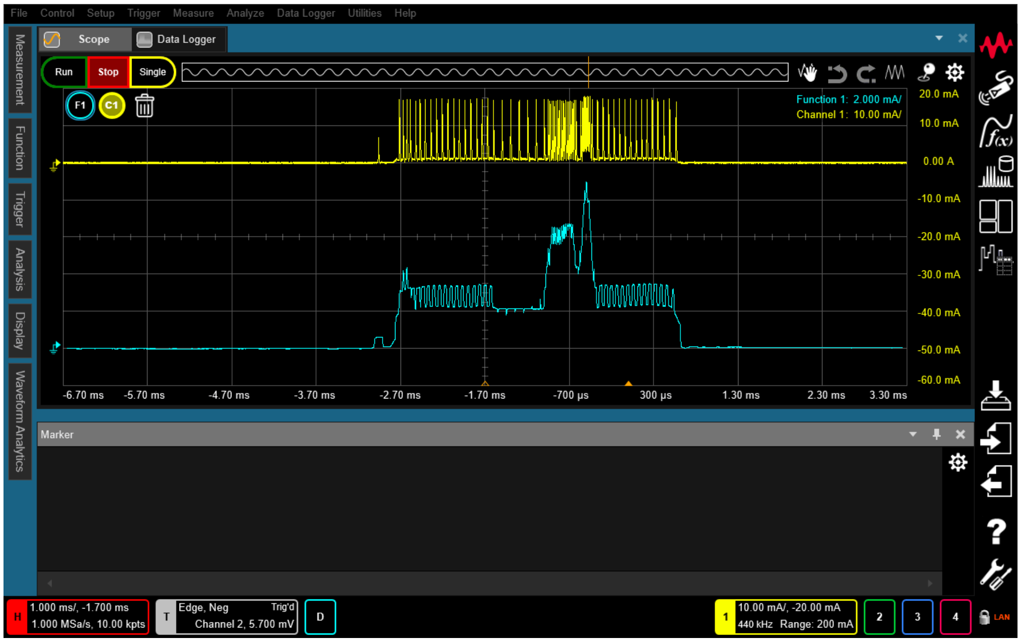


Figure 19. Connect event

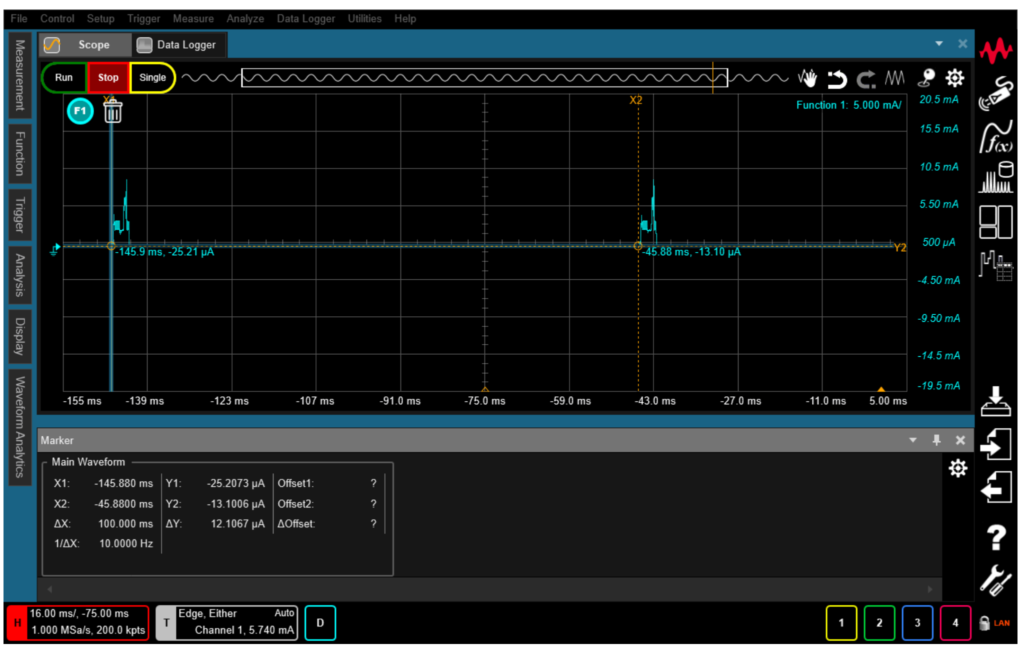


Figure 20. Connect period 100 ms

Figure 21 is an example of a sniffer capture of a Connect event.

The first frame is a Link Layer Feature Request from the master device.

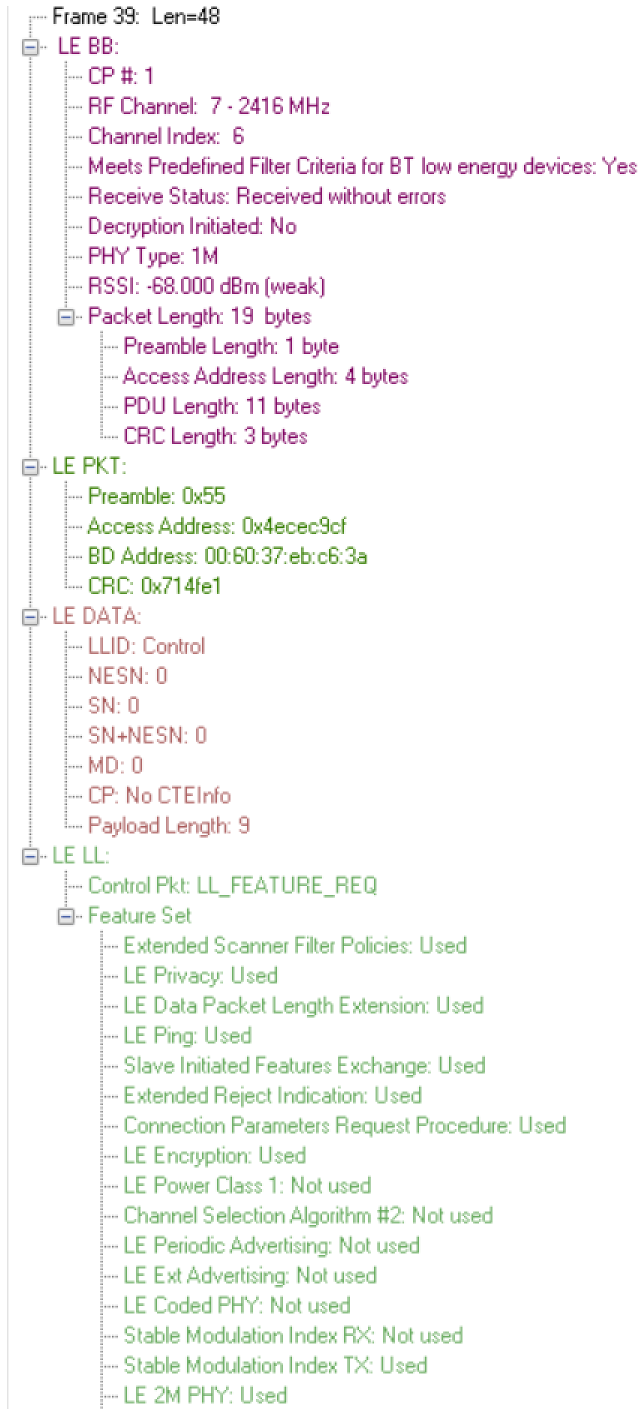


Figure 21. 1st Connect frame sniffer log

Figure 22 shows the slave corresponding packet response.

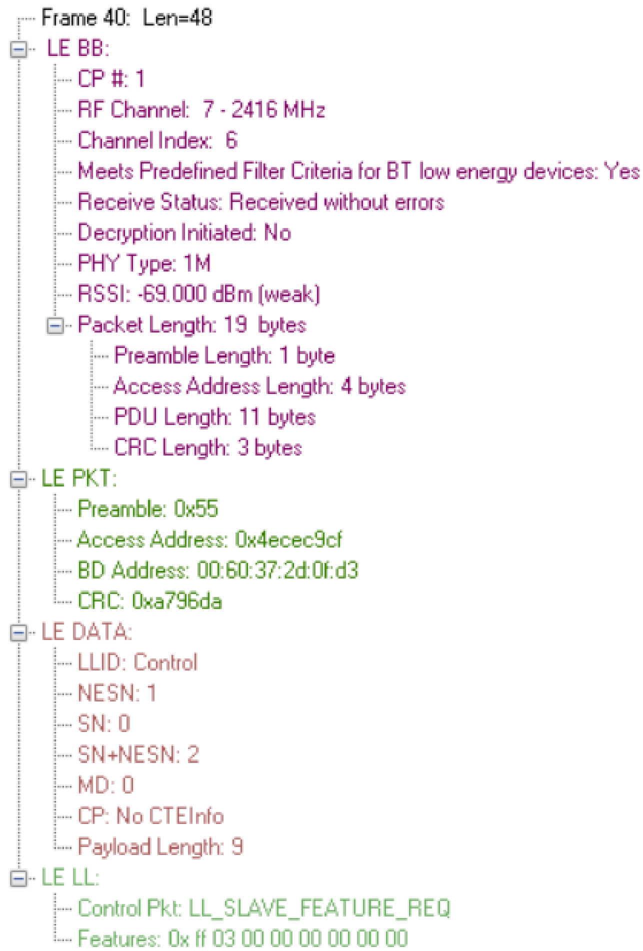


Figure 22. 2nd Connect frame sniffer log

The master then responds to the slave feature request.

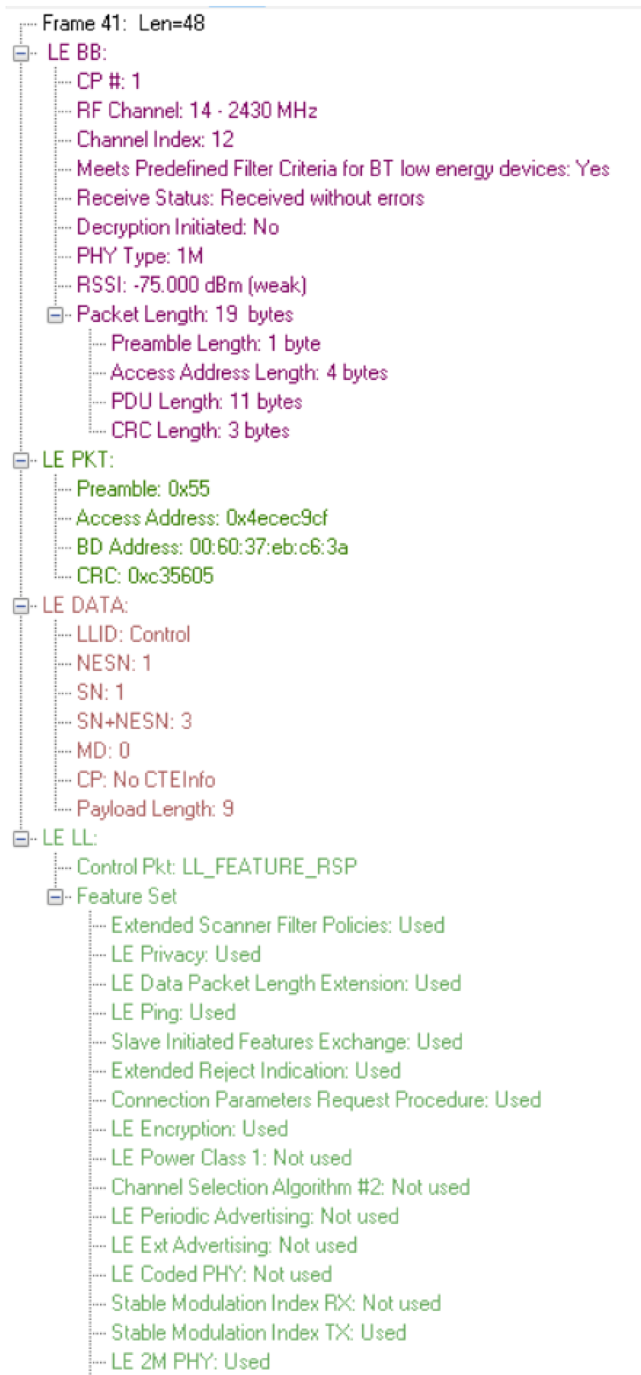


Figure 23. 3rd Connect frame sniffer log

And the slave responds to the master feature request.

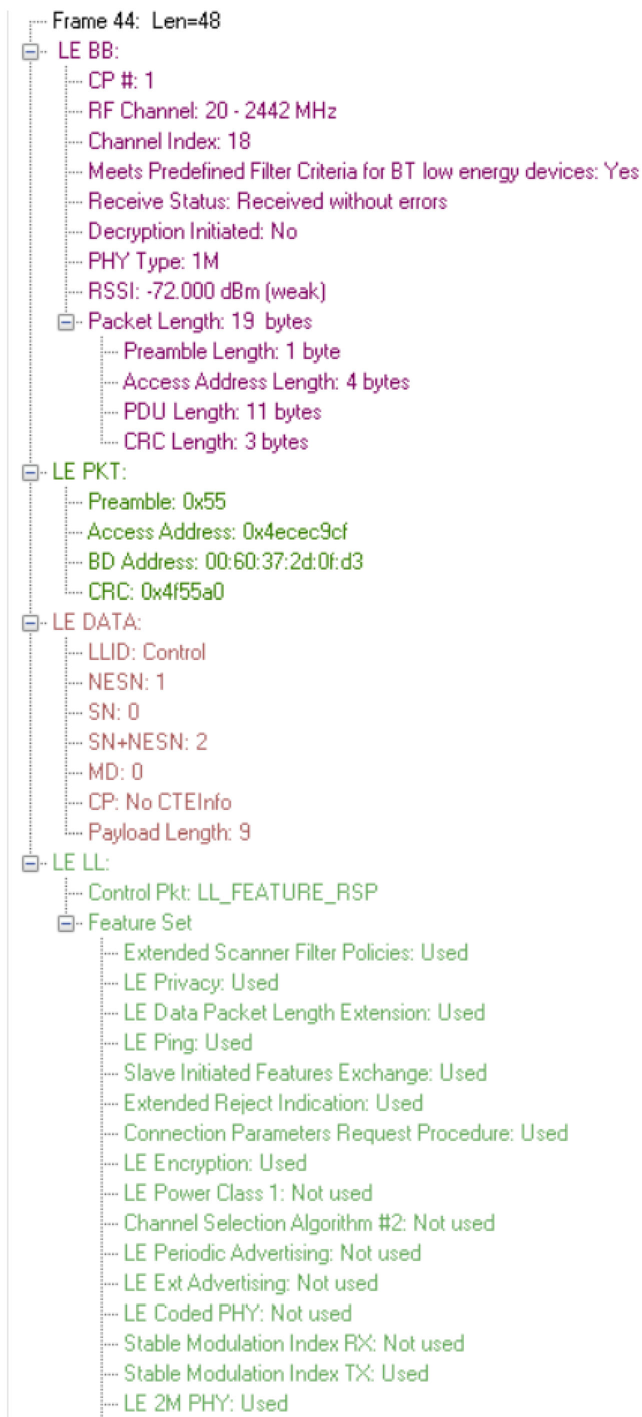


Figure 24. 4th Connect frame sniffer log

At this point, the devices are connected.

Figure 25 shows the power profile of the connect event. The total energy is 3.044 nAh under 3.0 V.

The total event duration is 3.299 ms.

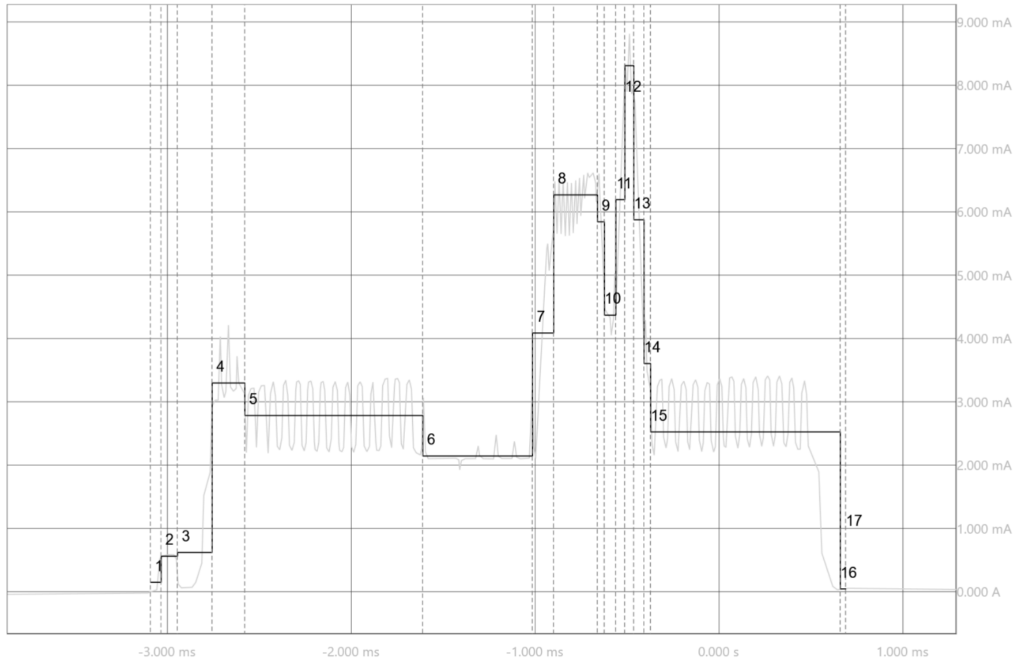


Figure 25. Connect event power profile

Table 5. Connect event break down table

Event	Duration	Current at 3 V
1. SoC in Power-Down mode	—	1.80 µA
2 to 6. Pre-processing [including stack and Radio initialization]	2.019 ms	561 µA to 3.298 mA
7. RX Warm up	115.1 µs	4.087 mA
8. Active RX	238.1 µs	6.269 mA
9 to 11. RX to TX transition	148.9 µs	5.468 mA
12. Active TX	48.81 µs	8.310 mA
13 to 14.TX warm down	88.97 µs	4.989 mA
15 to 16. MCU Post processing	1.061 ms	2.455 mA
17. SoC back to Power-Down mode	—	1.80 µA

2.4 Conclusion

This application note provides an overview of how to evaluate the low-power current consumption on a K32W during Bluetooth LE communication events. The measurements are based on an NXP IOTZTB-DK006 development kit and can be replicated at the customer side. All the measurements are in line with the specification.

3 IEEE 802.15.4 application

To perform low-power measurements on an IEEE 802.15.4 application, the DK6 board is modified. This minimizes leaking current and allow to measure very low currents. The modifications are described in the document IoT-ZTB-DK006 Development Kit user guide UM11393.

As reference for the measurements, power down and active currents are described in the data sheet. They are compared to the measurements results.

Firstly, the power down and RF static currents are measured with the Customer Module Evaluation Tool (CMET/AN1242).

Secondly, they are measured from a profile based on a Zigbee event.

The CMET version is 2038 whose radio driver version is 2085. Static measurement are based on this radio driver.

The Zigbee event currents are based on the radio driver 2088. The software is part of the SDK.

3.1 Power consumption measurement

3.1.1 Test set-up description

3.1.1.1 Hardware configuration

The test set-up is composed of:

- One K32W module on a mezzanine board.
- One modified DK6 board as described in IoT-ZTB-DK006 Development Kit user guide (document UM11393).

The test equipment chosen is a source/measure unit SMU (Keysight B2902A for instance). It is a power supply capable of measuring low currents.

Figure 26 shows test set-up block diagram.

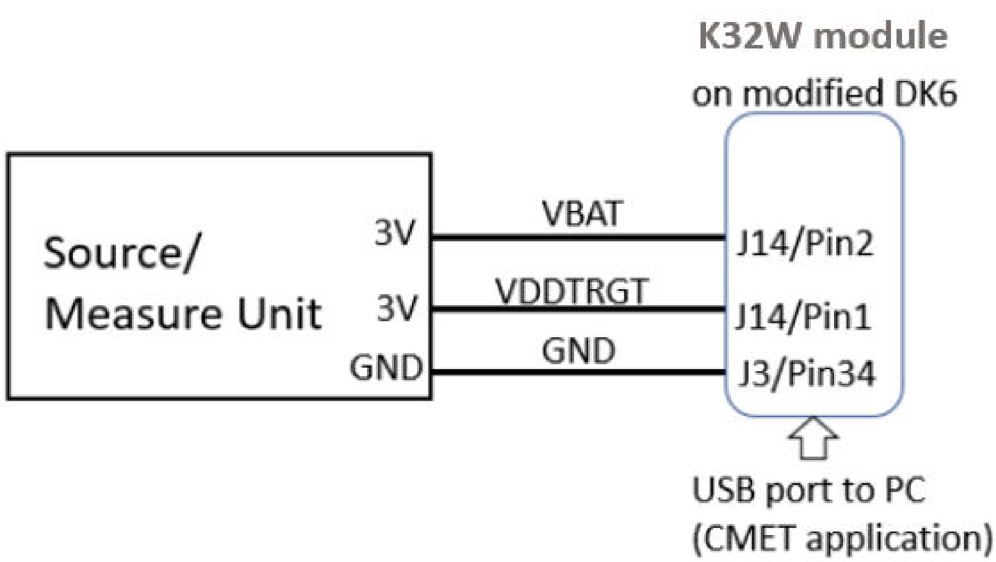


Figure 26. Test set-up block diagram for static measurements

The VBAT supplies the K32W device under test while the VDDTRGT is used to supply the rest of the board. The purpose is to be able to measure the current on the K32W independently of the consumption of the board. From a supply standpoint, VBAT = VDDTRGT.

Figure 27 shows the test connections.

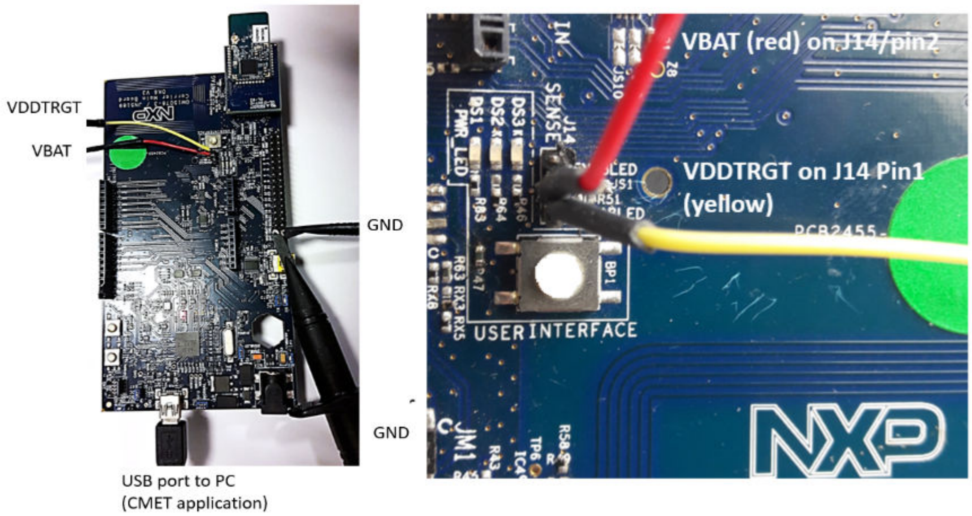


Figure 27. Test connections pictures for static measurements

3.1.1.2 Software configuration

CMET is the Software tool used for the power consumption measurement. It can be downloaded from the NXP website (CMET/AN1242).

As described in *High Performance M68HC11 System Design Using The WSI PSD4XX and PSD5XX Families* (document AN1242), the low power modes are as shown in Table 6.

Table 6. Power-down currents description

Power mode	CPU	CPU clock	RAM	Wake-up source
PM_DEEP_DOWN	OFF	OFF	OFF	HW reset, I/O event
PM_DOWN	OFF	OFF	Variable size Retention	HW reset, I/O event, wake-up timer
PM_SLEEP	ON	OFF	ON	Any interrupt

For this test, the CMET version used is as shown in Figure 28.

```
*********************************************
*    Customer Module Evaluation Tool        *
*    Version 2038                           *
*    Compiled Feb 28 2020 10:23:14          *
*    Radio Test version 2041                *
*    Radio Driver version 2085              *
*    Chip ID 000e2117                       *
*********************************************
```

Figure 28. CMET version

3.1.2 Power consumption in Low-Power modes

Power-down and deep power-down are covered by these measurements.

The currents measured with CMET are shown in Table 7.

Table 7. CMET current measurements

Symbol	Parameter	Conditions	Type(data sheet)	Measure with CMET @VBAT 3 V	Unit
IDD	Supply current	Deep power-down (everything is powered off, wake-up on HW reset only)	250	235	nA
		Deep power-down-IO (everything is powered off, wake-up on HW reset only or an event on any of the 22 GPIOs and NTAG interrupt)	350	360	nA
		Power-down (wake-up on HW reset or an IO event, wake-up timer ON, 32 kHz FRO on, no SRAM retention)	800	880	nA
		Power-down-4K (wake-up on HW reset or an IO event, wake-up timer on, 32 kHz FRO on, with 4 KB SRAM retention])	1025	1085	nA
		Power-down-8K (wake-up on HW reset or an IO event, wake-up timer on, 32 kHz FRO on, with 8 KB SRAM retention)	1120	1170	nA

3.1.3 Power consumption in Active mode

The RF currents are measured with the CMET and results are as shown in Table 8.

Table 8. Active current results with CMET

Parameter	Conditions	Requirement typical @ Vbat 3 V (CPU current not included)	CMET measurement @ Vbat 3 V (CPU current included)	Unit
Supply current	Radio in RX mode (IEEE 802.15.4)	4.30	6.84	mA
	Radio in TX mode (IEEE 802.15.4) output power 0 dBm	7.36	10.15	mA
	Radio in TX mode (IEEE 802.15.4) output power +3 dBm	9.44	12.21	mA
	Radio in TX mode (IEEE 802.15.4) output power +10 dBm	20.28	21.75	mA

NOTE

The gap compared to the data sheet is due to the CPU current that is already a part of the CMET measurements.

3.2 Power profile measurement

3.2.1 Hardware prerequisites

The set-up is composed of an IOTZTB-DK006 kit content: a control bridge, a light node and the switch device made of a K32W fitted on a DK6 board. Similarly to the previous sections, the DK6 of the switch device is modified for power measurement.

The K32W fitted on a modified DK6 board is called “the switch device” further on in this document.

Figure 29 shows the block diagram of the test set-up.

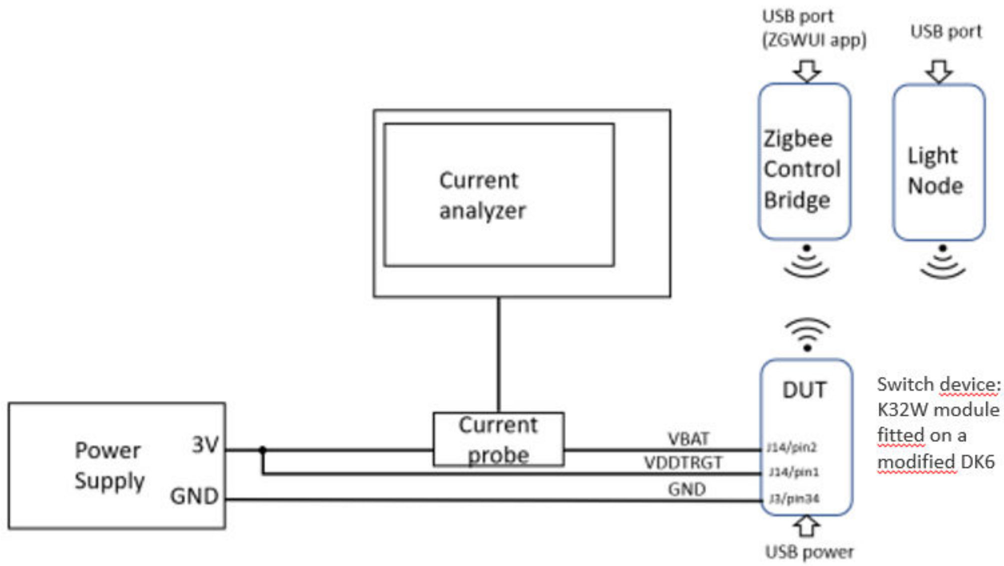


Figure 29. Test set-up block diagram

Figure 30 shows the modified DK6 with a K32W module fitted.

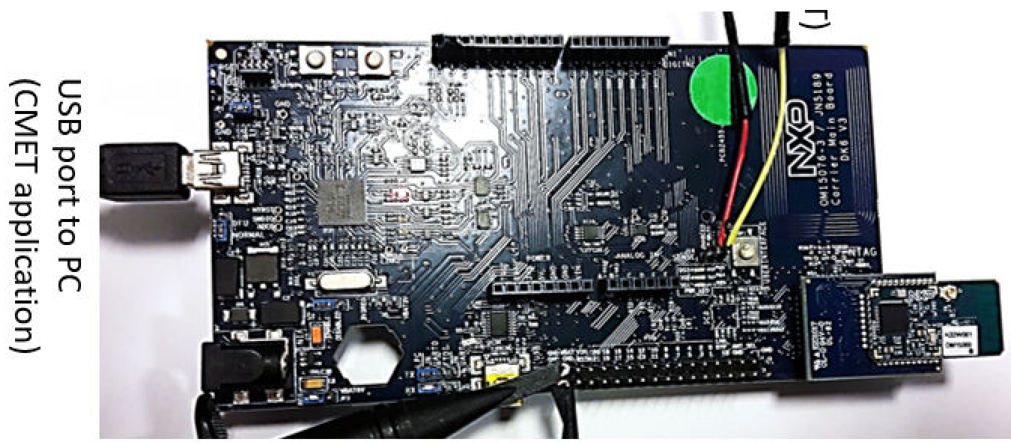


Figure 30. Modified DK6 and K32W module fitted

Figure 31 and Figure 32 show the Zigbee control bridge and the light node.

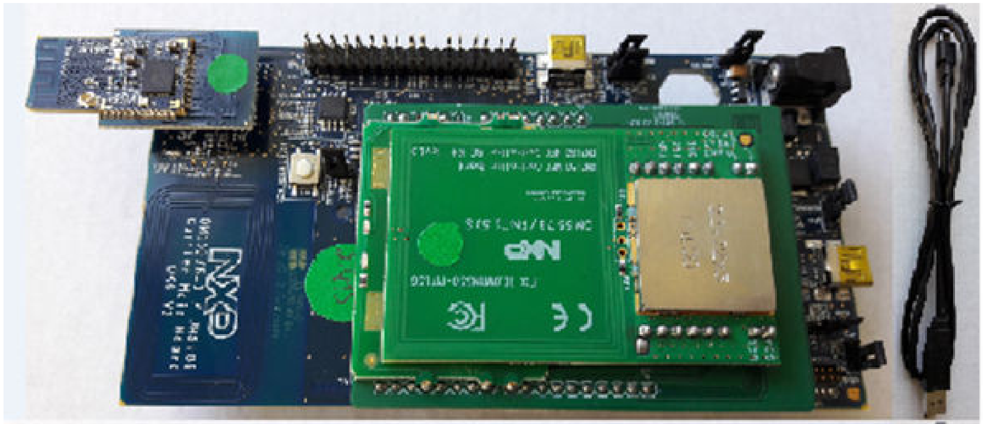


Figure 31. Zigbee control bridge picture

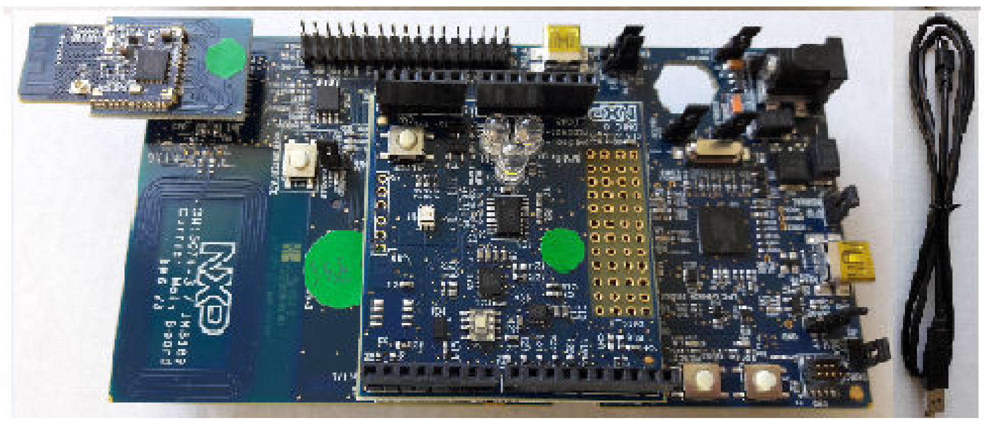


Figure 32. Light node picture

Figure 33 shows the test set-up.

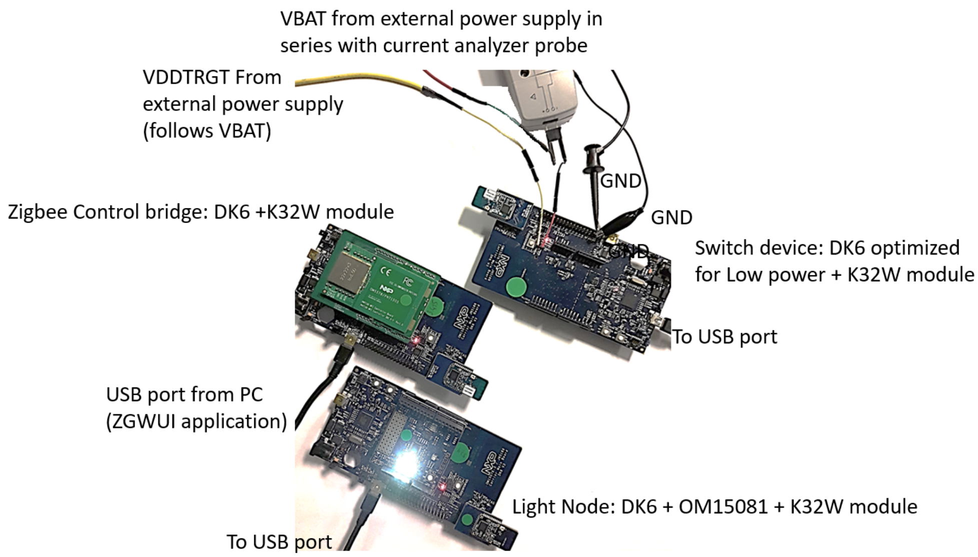


Figure 33. Test set-up

3.2.2 Software configuration

A Flash programmer is necessary to program the binary file into the flash memory of the device. The instructions are described in the application note JN-SW-4407 which is in the Tools folder of the SDK.

The control bridge is configured by using the instructions shown in document AN1247. The AN1223-Zigbee-IoT-Gateway-Control-Bridge (ZGWUI) must be installed on the PC to connect the control bridge.

The Light node is configured by using the instructions in document AN1244.

The switch device is configured by using the instructions in document JN-AN-1245.

The other settings for the next measurements are as follows::

- Payload: 37 Bytes
- RAM size: 4 KB
- TX output power: 10 dBm
- Radio driver version 2088

NOTE

After the binary files are programmed into the devices memory and prior to the procedure described in Measurement procedure, all the devices must be unplugged from their USB ports, or any external power supply.

NOTE

The DC-DC is always enabled in this measurement.

3.2.3 Use case description

A basic use case of a Zigbee network application is taken as an example.

A light node joins in a Zigbee network and is controlled by a switch device via a control bridge. The control bridge is logging the communication events because of the ZGWUI application on a PC.

3.2.4 Measurement procedure

3.2.4.1 Joining the network

The switch device must join the network to control the light node.

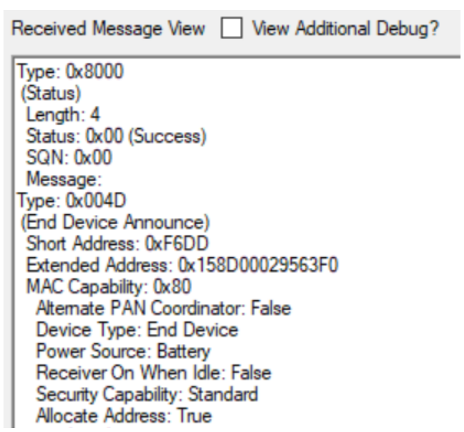
The ZGWUI application is used to start the network and it join the devices.

The joining procedure is as follows.

Table 9. The joining procedure

<table>
<tr>
<td>

1. Start the ZGWUI application on the PC.
2. In Settings menu, select the COM port that corresponds to the control bridge.
3. Then, select Open Port.
4. Erase PD.
5. Set the channel in CMSK field and select Set CMSK, enter 15.
6. Click Start NWK.
7. Connect the switch device on a USB port and to an external power supply (described in Power consumption measurement).
8. Power on the external power supply.
9. In ZGWUI, in Permit Join fields enter 0 in the first one and 20 in the second one. Then select Permit Join.
10. The switch device joins the network and can be verified in the log message on the ZGWUI as shown in the following figure.

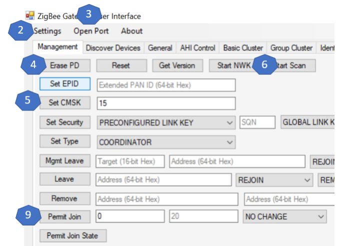


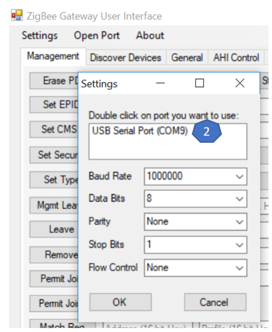
Received Message View ☐ View Additional Debug?

Type: 0x8000
(Status)
Length: 4
Status: 0x00 (Success)
SQN: 0x00
Message:
Type: 0x004D
(End Device Announce)
Short Address: 0xF6DD
Extended Address: 0x158D00029563F0
MAC Capability: 0x80
Alternate PAN Coordinator: False
Device Type: End Device
Power Source: Battery
Receiver On When Idle: False
Security Capability: Standard
Allocate Address: True

11. Power on the light node by connecting on a USB port.
12. The light node is flashing until it has joined the network, as logged below, then light is always ON.

</td>
<td>





</td>
</tr>
</table>

Table continues on the next page...

Table 9. The joining procedure

Received Message View ☐ View Additional Debug? Type: 0x8701 (Route Discovery Confirm) SQN: 0x00 Status: 0x00 Network Status: 0x00 Type: 0x004D (End Device Announce) Short Address: 0xAB57 Extended Address: 0x158D000295646B MAC Capability: 0x8E Alternate PAN Coordinator: False Device Type: Router Power Source: AC Receiver On When Idle: True Security Capability: Standard Allocate Address: True	

The power consumption can be observed when the switch device joins the network.

Figure 34. Joining power profile

The ZGWUI session must stay active for the next steps in the following sections.

3.2.4.2 Binding the switch device to the light node

When the switch device has joined the network, it is necessary to bind it to the light node. To do so, perform the following steps in the same ZGWUI session as in the previous section.

1. On the light node, push the reset button three times (SW4 on DK6 board).

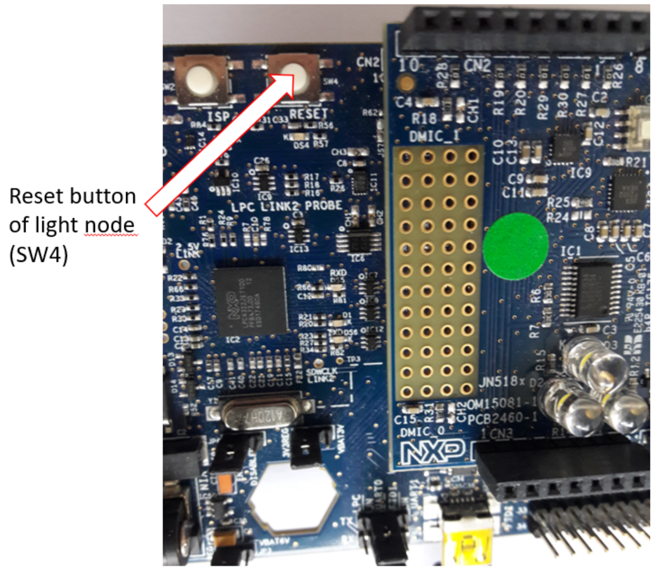


Figure 35. Light node: reset button location

2. The light node starts to flash.
3. On the switch device, press the user interface button (BP1) and release. The light node LEDs stop flashing and stays ON.

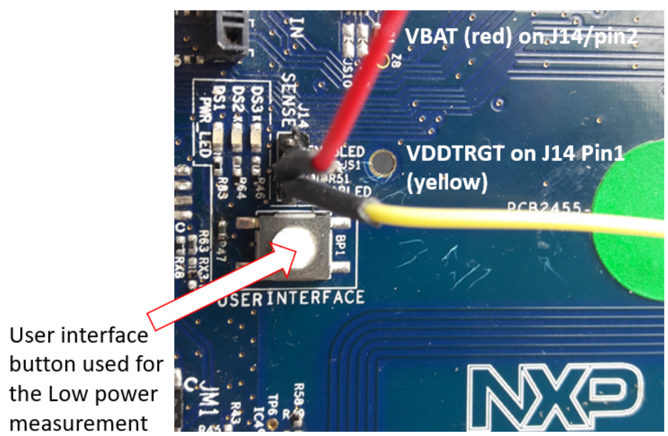


Figure 36. Switch device: user interface button location

4. The switch device and the light node are bound and the switch device can control the light node according to the following table.

On user interface button of switch device	Result on the light node
Push 2n+1, n = 0, 1, 2..	Light OFF
Push 2n, n = 0, 1, 2..	Light ON

The power profile is observed at the binding time as shown in the following figure.

Figure 37. Binding power profile

The sniffing trace of a binding event is as follows.

Packet Type	PAN Src	PAN Dst	MAC Src	MAC Dst	MAC Seq	NWK Src	NWK Dst
Bind Request		0xE1E9	0x0000	0x4EF4	176	0x0000	0x4EF4
Acknowledgement					176		
Acknowledgement		0xE1E9	0x4EF4	0x0000	153	0x4EF4	0x0000
Acknowledgement					153		
Bind Response		0xE1E9	0x4EF4	0x0000	154	0x4EF4	0x0000
Acknowledgement					154		
Data Request		0xE1E9	0x4EF4	0x0000	155		
Acknowledgement					155		
Transport Key		0xE1E9	0x0000	0x4EF4	177	0x0000	0x4EF4
Acknowledgement					177		
Verify Key		0xE1E9	0x4EF4	0x0000	156	0x4EF4	0x0000
Acknowledgement					156		
Link Status		0xE1E9	0x0000	0xFFFF	178	0x0000	0xFFFC
Data Request		0xE1E9	0x4EF4	0x0000	157		
Acknowledgement					157		
Acknowledgement		0xE1E9	0x0000	0x4EF4	179	0x0000	0x4EF4
Acknowledgement					179		
Data Request		0xE1E9	0x4EF4	0x0000	158		
Acknowledgement					158		
Bind Request		0xE1E9	0x0000	0x4EF4	180	0x0000	0x4EF4
Acknowledgement					180		
Acknowledgement		0xE1E9	0x4EF4	0x0000	159	0x4EF4	0x0000
Acknowledgement					159		
NWK Address Request		0xE1E9	0x4EF4	0x0000	160	0x4EF4	0xFFFD
Acknowledgement					160		
Bind Response		0xE1E9	0x4EF4	0x0000	161	0x4EF4	0x0000
Acknowledgement					161		

Figure 38. Binding sniffer trace

3.2.4.3 Switching on the light node with the switch device

When the user pushes the user interface button of the switch device, the device goes through the following three phases:

1. Wake up from Sleep mode
2. Transmit data
3. Go back to Sleep mode

In this case, the power profile can be measured as shown in Figure 39.

Figure 39. Light on event power profile

The shape of current profile is the same when pushing the user interface button again to switch off the light.

Figure 40 shows the sniffer trace of a light ON event.

Stack	Packet Type	PAN Src	PAN Dst	MAC Src	MAC Dst	MAC Seq	NWK Src	NWK Dst
ZigBee	On/Off: On		0xE1E9	0x4EF4	0x0000	200	0x4EF4	0x13BE
ZigBee	Acknowledgement					200		
ZigBee	On/Off: On		0xE1E9	0x0000	0x13BE	49	0x4EF4	0x13BE
ZigBee	Acknowledgement					49		

Figure 40. Light on event sniffer trace

The power profile is then processed as shown in Figure 41.

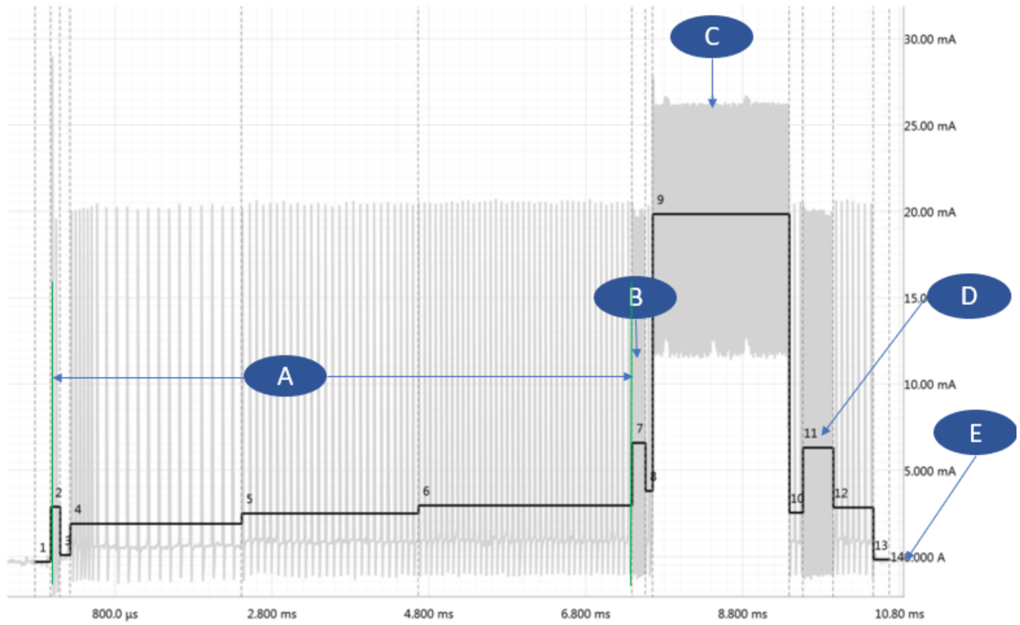


Figure 41. Current Profile of Zigbee event

The power consumption is analyzed for several Vbat voltages, as shown in Table 10.

Table 10. Current measurements from Zigbee profile

Step#	CPU	Radio	Mode	Current @ 2.6 V	Current @ 3.0 V	Current @ 3.6 V	Duration
A	Start	OFF	Initialization	2.5 mA	2.4 mA	2.4 mA	7.6 ms
B	ON	RX ON	RX Cal- CCA	6.6 mA	6.5 mA	5.9 mA	172 µs
C	ON	TX ON	10 dBm	22.6 mA	19.8 mA	16.9 mA	1.7 ms
D	ON	RX ON	Wait for Ack	6.7 mA	6.3 mA	5.7 mA	387 µs
E	ON	OFF	PD Mode0	2.70 µA	2.73 µA	2.85 µA	NA

3.3 Conclusion

This application note provides a step by step approach to measure low power performances of the K32W. The measurements are based on Zigbee events that can be replicated using the development kit (IOTZTB-DK006).

The total energy consumed is in line with the specifications which makes the K32W particularly suitable for low-power applications.

Legal information

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.

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.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

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.

Suitability for use in non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

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.

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.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μVision, Versatile — are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

Airfast — is a trademark of NXP B.V.

Bluetooth — the Bluetooth wordmark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by NXP Semiconductors is under license.

Cadence — the Cadence logo, and the other Cadence marks found at www.cadence.com/go/trademarks are trademarks or registered trademarks of Cadence Design Systems, Inc. All rights reserved worldwide.

CodeWarrior — is a trademark of NXP B.V.

ColdFire — is a trademark of NXP B.V.

ColdFire+ — is a trademark of NXP B.V.

EdgeLock — is a trademark of NXP B.V.

EdgeScale — is a trademark of NXP B.V.

EdgeVerse — is a trademark of NXP B.V.

eIQ — is a trademark of NXP B.V.

FeliCa — is a trademark of Sony Corporation.

Freescale — is a trademark of NXP B.V.

HITAG — is a trademark of NXP B.V.

ICODE and I-CODE — are trademarks of NXP B.V.

Immersiv3D — is a trademark of NXP B.V.

I2C-bus — logo is a trademark of NXP B.V.

Kinetis — is a trademark of NXP B.V.

Layerscape — is a trademark of NXP B.V.

Mantis — is a trademark of NXP B.V.

MIFARE — is a trademark of NXP B.V.

NTAG — is a trademark of NXP B.V.

Processor Expert — is a trademark of NXP B.V.

QorIQ — is a trademark of NXP B.V.

SafeAssure — is a trademark of NXP B.V.

SafeAssure — logo is a trademark of NXP B.V.

Synopsys — Portions Copyright © 2021 Synopsys, Inc. Used with permission. All rights reserved.

Tower — is a trademark of NXP B.V.

UCODE — is a trademark of NXP B.V.

VortiQa — is a trademark of NXP B.V.

Date of release: 30 May 2022

Document identifier: AN12846