MKW01 Simple Media Access Controller (SMAC) Demonstration Applications

User’s Guide

Document Number: MKW01SMACDAUG
Rev. 1
11/2015

How to Reach Us:

Home Page:
freescale.com

E-mail:
support@freescale.com

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Chapter 1
Introduction

Chapter 2
MKW01 Wireless UART application

Chapter 3
MKW01 Wireless Messenger application

Chapter 4
MKW01 Connectivity Test application

Chapter 5
MKW01 Low-Power Demo application

About this book

This guide provides a detailed description of the MKW01 demonstration applications.

For more details about the MKW01 MCU, see the appropriate reference manual and/or data sheet.

Audience

This document is intended for application developers using the MKW01 MCU, and these demonstration applications are a good starting point for developing proprietary applications.

Organization

This document is organized into five chapters, as follows:

- Chapter 1, "Introduction" – Describes the required software and hardware for correct demonstration application setup.
- Chapter 2, "MKW01 Wireless UART application" – Describes how to run the Wireless UART application, and explains the available application configurations.
- Chapter 3, "MKW01 Wireless Messenger application" – Describes how to run the Wireless Messenger application, and explains the available application configurations.
- Chapter 4, "MKW01 Connectivity Test application" – Describes how to run the Connectivity Test application, and explains the available menu options and application configurations.
- Chapter 5, "MKW01 Low-Power Demo application" – Describes how to run the Low-Power Demo application, and explains the available menu options and application configurations.

Revision history

The following table summarizes the substantive changes done to this document since the initial release.

Revision history

Rev. number	Date	Substantive changes
0	07/2015	Initial release.
1	11/2015	Added information on KW01 freedom board and USB dongle. Updated Screenshots.

Definitions, acronyms, and abbreviations

The following list defines the acronyms and abbreviations used in this document.

API	Application Program Interface
dBm	Decibels referred to one milliwatt
LQI	Link Quality Indicator
TWR	Tower System Modular Development Platform
IDE	Integrated Development Environment
MCU	Microcontroller Unit

OTA	Over-the-Air
PC	Personal Computer
RX	Receive(r)
TX	Transmit(ter)
TERM	Serial Port Terminal Application
XCVR	Transceiver
[ENTER]	The ENTER / RETURN key on the keyboard
[SPACE]	The SPACEBAR key on the keyboard
CCA	Clear Channel Assessment
ACK	Acknowledge
MRB	Modular Reference Board
FRDM	Freedom Development Platform
USB	USB Dongle Platform / USB connector
MKW01	FRDM-KW01, MRB-KW01 or USB-KW01
RTOS	Real-Time Operating System

References

The following sources were referenced to produce this book:

[1] *MKW01Z128 Reference Manual* (MKW01xxRM)

Chapter 1 Introduction

The Freescale MKW01 SMAC-based demonstration applications have to be used with a serial port terminal application.

This section describes the hardware and software requirements for using the MKW01.

1.1 Supported hardware

The MKW01 SMAC-based demonstration applications are designed to work with the MKW01 platforms (FRDM, USB or MRB). The MRB-KW01 platform can work standalone or plugged into the TWR-RF board.

For more information on the TWR-RF system, see the appropriate Freescale Tower System documentation at www.freescale.com/tower.

1.2 MKW01 software

The package contains the RTOS and KSDK-based framework, drivers, and connectivity stack, with SMAC as layer two, along with the three demo applications. The application projects are developed using IAR® Embedded Workbench®.

1.3 Hardware considerations

The FRDM-KW01 can be connected to the PC using the mini USB connector on J15. The following table contains the jumper configurations for the FRDM-KW01.

Table 1-1. Jumper settings for the FRDM-KW01 platforms

header	Install jumper on pins
J9	1-2
J8	1-2
J41	1-2
J42	1-2
J43	1-2
J44	1-2
J31	1-2

The MRB-KW01 can be connected to the PC in two ways:

- Connected directly from the PC to the mini USB connector (J16) of the MRB
- Mounted on a TWR-RF board (when using this setup, the cable from the PC must be connected to the mini USB connector (J2) of the TWR-RF board)

The jumper configurations for both setups are described in the following table.

Table 1-2. Jumper settings for USB connected directly to MRB (no TWR-RF)

header	Install jumper on pins
J3	1 - 2
J7	1 - 2
J8	1 - 2
J12	1 - 2
J12	3 - 4
J13	1 - 2
J13	3 - 4
J13	5 - 6
J13	7 - 8
J17	1 - 2
J17	4 - 5
J17	7 - 8
J17	10 - 11
J17	13 - 14

To connect the mini-USB cable to the TWR-RF platform, you must apply changes to J17. Instead of the (1 – 2), (4 – 5), (10 – 11), and (13 – 14) pairs, install jumpers on (2 – 3), (5 – 6), (11 – 12), and (14 – 15).

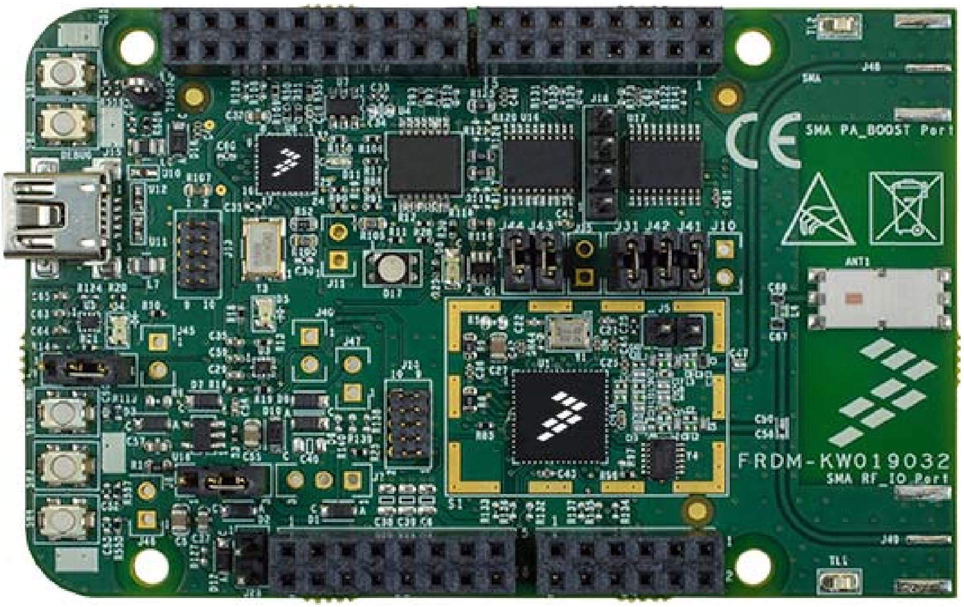


Figure 1-1. FRDM-KW01

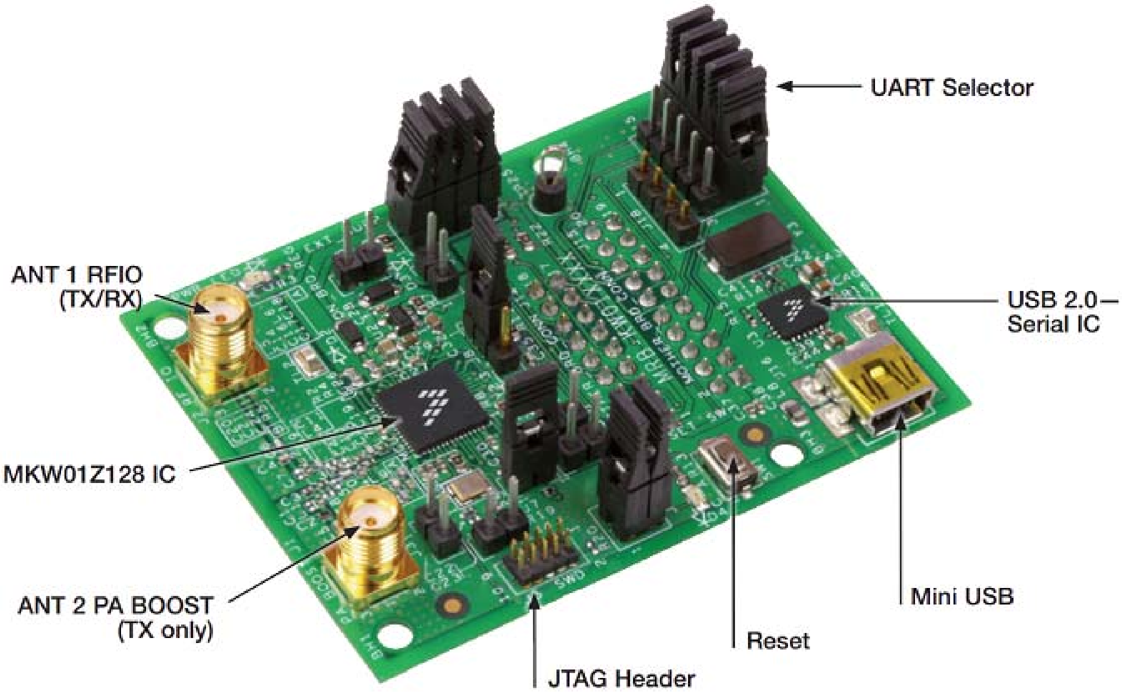


Figure 1-2. MRB-KW01

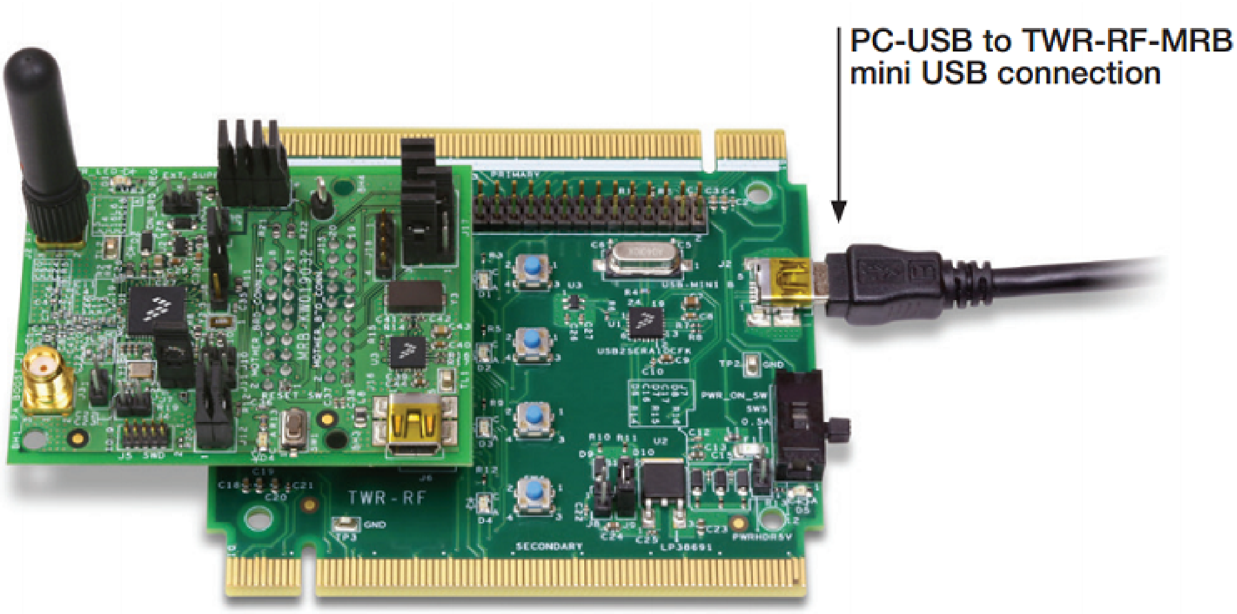


Figure 1-3. MRB-KW01 connection with the TWR-RF platform

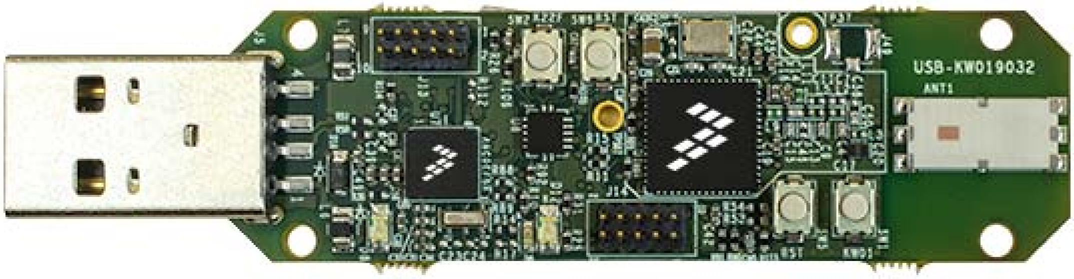


Figure 1-4. USB-KW01

Chapter 2
MKW01 Wireless UART application

2.1 Introduction

The MKW01 Wireless UART application functions as a wireless UART bridge between two (one-to-one) or several (one to many) boards. The application can be used with either a TERM, or with a software that is capable of opening a serial port and writing to it (or reading from it). The characters sent or received are not necessarily ASCII printable characters.

The entire application configuration must be done at compile time, because Wireless UART does not display any configuration menus.

2.2 Application configuration

The application can be configured using the following files:

- *SMAC_Config.h*

  This file contains configuration for the source / destination PAN id (*gDefaultPanID_c*), source node address (*gNodeAddress_c*), number of maximum retries and retransmissions (*gMaxRetriesAllowed_c*), and the backoff interval for the custom backoff algorithm.
- *Application_Interface.h*

  This file provides customization of the default destination address (*gDefaultAddress_c*), default channel number (*gDefaultChannelNumber_c*), and default output power, which has to range between minimum and maximum output power (*gDefaultOutputPower_c*). This file also contains the definition for two default PHY modes.
- *Wireless_UartApplication.c*

  The *InitProject* can be changed to enable the retry and retransmission mechanisms. By setting the boolean fiels to *TRUE*, and by setting the numerical fields to the desired number of retries or retransmissions, you can enable none, one, or both of these features.

  Here is an example:

  ```
  txConfigContext.autoAck           = FALSE; //TRUE for enabling automatic ACK
  txConfigContext.ccaBeforeTx      = FALSE; //TRUE for enabling automatic CCA before TX
  txConfigContext.retryCountAckFail = 0; // Number of retries in case no ACK is received
  txConfigContext.retryCountCCAFail = 0; // Number of retransmissions if channel is
  busy
  ```

  The *InitApp* function enables you to change the baudrate of the serial port. For other options, see the definition of the baudrate macro (*gUARTBaudRate115200_c*), and pick another member from the baud rates enumeration.

Other settings must be left as they are, because they can affect the application behavior. There are several settings that you can make in the IAR project compiler options to configure the number and size of the Serial Manager receive buffers and the buffers available for memory allocation, but it is recommended to keep their current values.

The application consists of three project files, each project is configured to run either on RTOS (FreeRTOS or MQX) configuration or bare-metal configuration (non-preemptive task scheduler). The project configurations available are Debug and Release. Before building and downloading the application, you can switch to either one by using the Workspace tab in IAR.

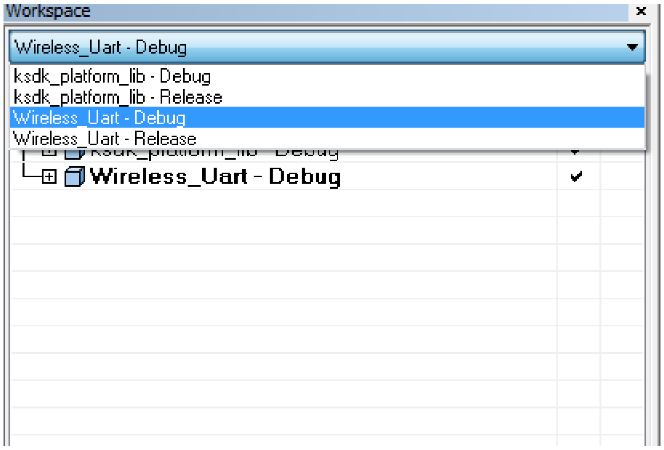


Figure 2-1. Change between Debug and Release for the bare-metal project in IAR

2.3 Running the Wireless UART application

By default, the application uses broadcast addresses for OTA communication. This way, the application can be directly downloaded and run without any user intervention. The following use case assumes that no changes have been made to the project.

1. Two (or more) MKW01 platforms must be connected to the PC using the mini USB cables.
2. The code must be downloaded to the platforms via J-Link (or other means).
3. Two (or more) TERM applications must be opened, and the serial ports must be configured to the same baudrate as the one in the project (default value is 115200). Other necessary serial configurations are 8-bit, no parity, and one stop bit.
4. To start the setup, each platform must be reset, and one of the (user) push-buttons found on the MKW01 platform must be pressed. This starts the application state machine, and you can start sending characters via UART to one of the platforms, which are broadcast OTA.

Chapter 3
MKW01 Wireless Messenger application

3.1 Introduction

The MKW01 Wireless Messenger application is an SMAC-based demo, which highlights the retry and retransmission mechanisms in case of *no ACK* and *Channel Busy* scenarios. It uses the basic SMAC primitives for data layer and management layer, and, optionally, it integrates a security module, which performs AES encryption and decryption in a cipher block chaining mode. The encryption and decryption processes are completely transparent to the application. The demo is presented in the form of a messenger-like application, and it requires a TERM.

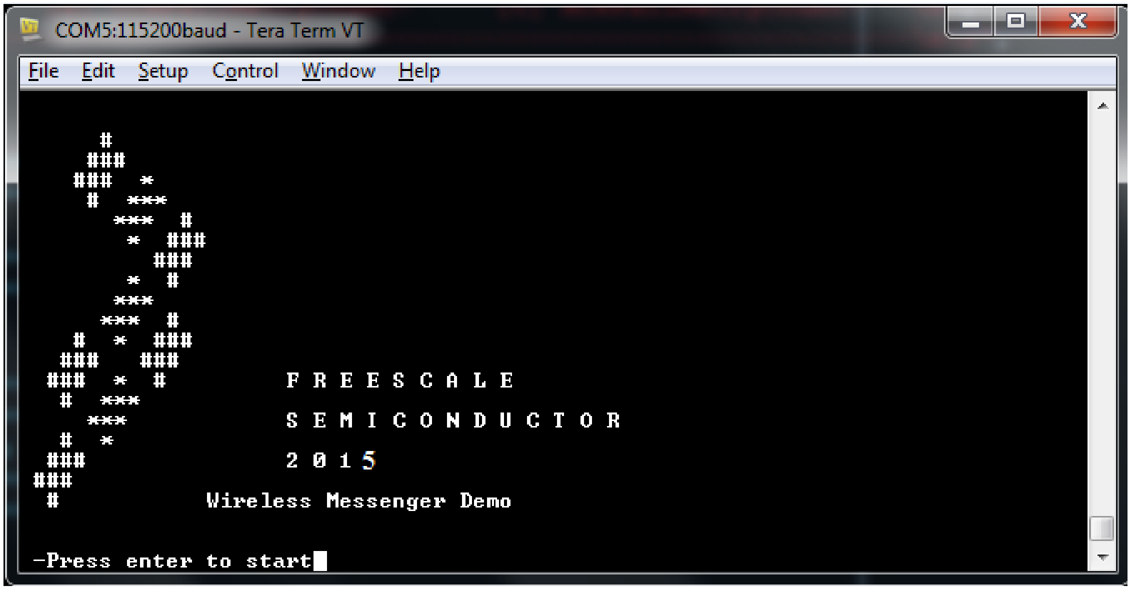
Most of the application configurations can be applied at runtime, except for changing the baudrate, and selecting between RTOS targets and a bare-metal target.

3.2 Wireless Messenger configuration menu

This section describes the configuration menu and the shortcuts menu, which help you in configuring the application. The options available in the shortcuts menu can be accessed both from the main menu, and from the configuration menu. The console menu has all the shortcuts disabled, so that they don’t interfere with message typing. With security enabled, the configuration menu is extended with another entry in the main menu, which helps you in configuring the initial vector and encryption key.

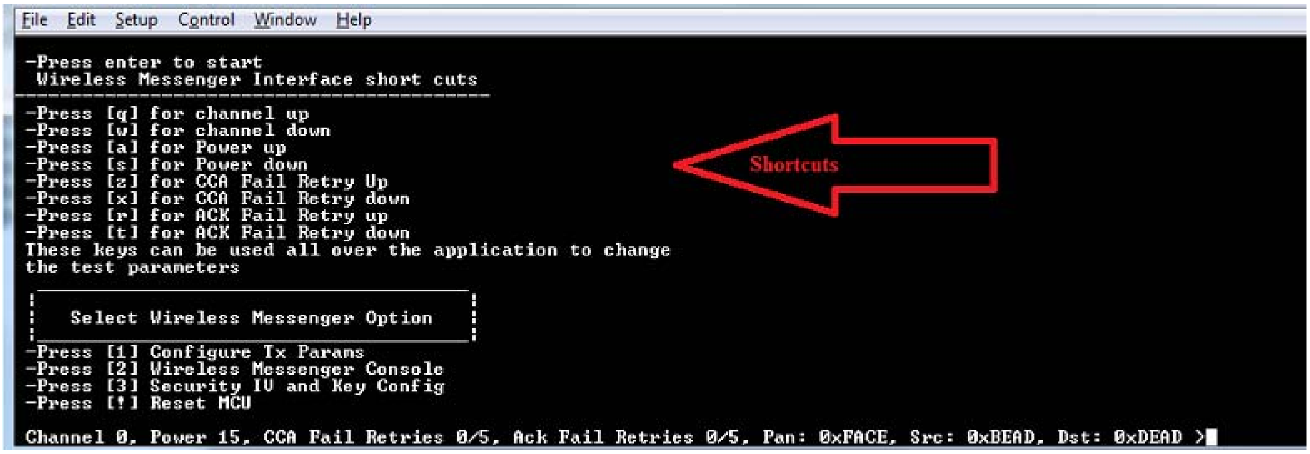
3.2.1 Shortcuts menu

After you press [ENTER] on the welcome screen, the main menu is displayed. In this menu, there is a shortcut entry displayed, which specifies what each shortcut key does.


COM5:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
     #
    ###
   ###  *
    #  ***
      ***  #
       *  ###
         ###
       *  #
      ***
     ***  #
   #  *  ###
  ###   ###
 ###  *  #         F R E E S C A L E
  #  ***           S E M I C O N D U C T O R
    ***
  #  *             2 0 1 5
 ###
###
 #           Wireless Messenger Demo

-Press enter to start
```

Figure 3-1. Wireless Messenger welcome screen



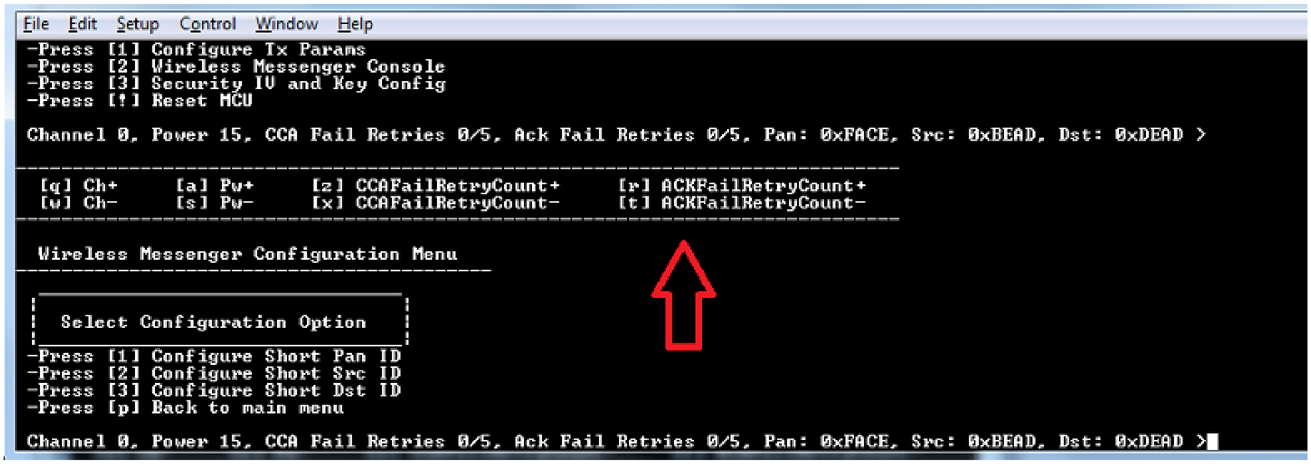
```
-Press enter to start
 Wireless Messenger Interface short cuts
-----------------------------------------
-Press [q] for channel up
-Press [w] for channel down
-Press [a] for Power up
-Press [s] for Power down
-Press [z] for CCA Fail Retry Up
-Press [x] for CCA Fail Retry down
-Press [r] for ACK Fail Retry up
-Press [t] for ACK Fail Retry down
These keys can be used all over the application to change
the test parameters

 ______________________________________
|                                      |
|   Select Wireless Messenger Option   |
|______________________________________|
-Press [1] Configure Tx Params
-Press [2] Wireless Messenger Console
-Press [3] Security IV and Key Config
-Press [!] Reset MCU

Channel 0, Power 15, CCA Fail Retries 0/5, Ack Fail Retries 0/5, Pan: 0xFACE, Src: 0xBEAD, Dst: 0xDEAD >
```

Figure 3-2. Shortcuts menu for the MKW01 Wireless Messenger demo application

Given the above figure:

- Pressing ‘q’ increments the channel number. When it overflows the maximum channel number (dependent on frequency band and PHY mode), the channel number is set to 0.
- Pressing ‘w’ decrements the channel number. If the previous channel number is 0, it switches to the last channel number.
- Pressing ‘a’ and ‘s’ is similar to ‘q’ and ‘w’, but it is used for setting the output power. The limits for this option are integers in the range [0x00 , 0x1F].
- Pressing ‘z’ increments the number of retransmissions in case of the *Channel Busy* scenario. Please keep in mind that a number higher than 0 for this option enables the automatic CCA before TX option.
- Pressing ‘x’ decrements the number of retransmissions in case of the *Channel Busy* scenario. If the number of retransmissions is decremented to 0, the feature is disabled.
- Pressing ‘r’ increments the number of retries in case of the *No ACK* situation. If the number of retries is higher than 0, the automatic ACK mechanism is enabled. This mechanism is bypassed in case of a broadcast transmission.
- Pressing ‘t’ decrements the number of retries in case of the *No ACK* situation. If the number of retries reaches 0, the automatic ACK mechanism is disabled.

A brief shortcuts menu description is also displayed in the configuration menu, as shown in Figure 3-3. This indicates that the shortcuts are still active.



```
-Press [1] Configure Tx Params
-Press [2] Wireless Messenger Console
-Press [3] Security IV and Key Config
-Press [!] Reset MCU

Channel 0, Power 15, CCA Fail Retries 0/5, Ack Fail Retries 0/5, Pan: 0xFACE, Src: 0xBEAD, Dst: 0xDEAD >

----------------------------------------------------------------------------------
 [q] Ch+      [a] Pw+      [z] CCAFailRetryCount+      [r] ACKFailRetryCount+
 [w] Ch-      [s] Pw-      [x] CCAFailRetryCount-      [t] ACKFailRetryCount-
----------------------------------------------------------------------------------

 Wireless Messenger Configuration Menu
--------------------------------------------

 ---------------------------------
 |  Select Configuration Option  |
 ---------------------------------
-Press [1] Configure Short Pan ID
-Press [2] Configure Short Src ID
-Press [3] Configure Short Dst ID
-Press [p] Back to main menu

Channel 0, Power 15, CCA Fail Retries 0/5, Ack Fail Retries 0/5, Pan: 0xFACE, Src: 0xBEAD, Dst: 0xDEAD >
```

Figure 3-3. Shortcuts menu description embedded in the configuration menu

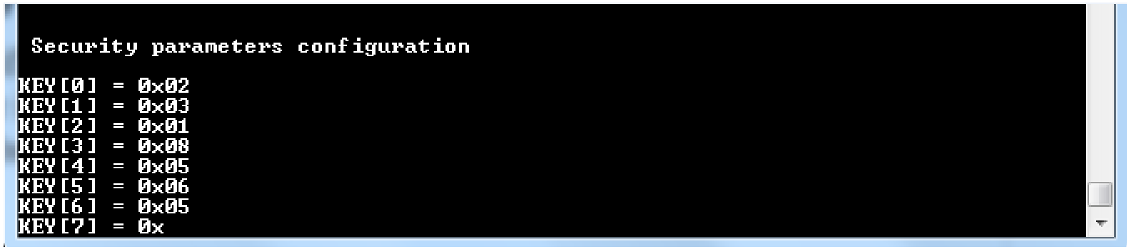
3.2.2 Configuration menu

The configuration menu is shown in Figure 3-3. There are three options available:

- *Configure Short Pan ID* – this option enables you to set the source / destination PAN address. You must enter four hexadecimal digits resembling a 16-bit address. The value entered is case-insensitive. For example, to set a broadcast address for the PAN ID, you can type both “0xFFFF” and “0xffff”.
- *Configure Short Src ID* – this option enables you to set the short address of the node. This option is similar to the one presented above.
- *Configure Short Dst ID* – this option enables you to set the destination address of the packets that are sent from the console menu. The feature is similar to the *Configure Short Pan ID* option.

3.2.3 Security IV and key configuration

This configuration menu appears only when the application is built with *gSmacUseSecurity_c* set to 1. This feature enables you to configure the initial vector and encryption key. For two devices to communicate properly, you must enable security on both of them, and the initial vector and the key must be identical.



```
 Security parameters configuration

KEY[0] = 0x02
KEY[1] = 0x03
KEY[2] = 0x01
KEY[3] = 0x08
KEY[4] = 0x05
KEY[5] = 0x06
KEY[6] = 0x05
KEY[7] = 0x
```

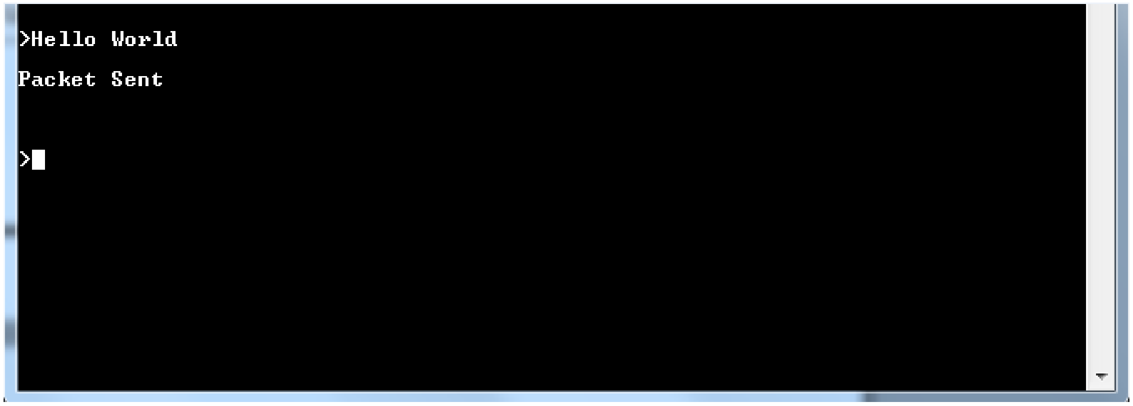
Figure 3-4. Security configuration menu (when security is enabled)

3.3 Wireless Messenger console menu

The console menu is used for sending and receiving messages. Outside this menu, the MKW01 does not receive any packets OTA. In this menu, all configuration options are disabled.

In the console menu, you can see packets intended for your MKW01 platform or the broadcast packets. When you type a message and hit [ENTER], the application includes the message into a packet payload, and sends it using the addressing information configured previously. The application notifies you about the status of the packet by writing a status message to the serial port. The status can be either the “Packet Sent” message, or an error message, in case the retry or retransmission mechanism is enabled and sending failed after the retry / retransmission counter exceeds the configured value. For example, if the *CCA Fail Retries* is set to 2 and the algorithm finds the channel idle after the second attempt, you receive the “Packet Sent” message in the TERM. If the channel is busy after the second retry, you receive the “Packet Sending Failed. Reason. Channel Busy!” notification.

You can return to the main menu by pressing the “escape” button ([Esc]).



```
>Hello World
Packet Sent


>
```

Figure 3-5. Console menu: “Packet Sent” notification after writing “Hello World” and pressing [ENTER]

Chapter 4 MKW01 Connectivity Test application

4.1 Introduction

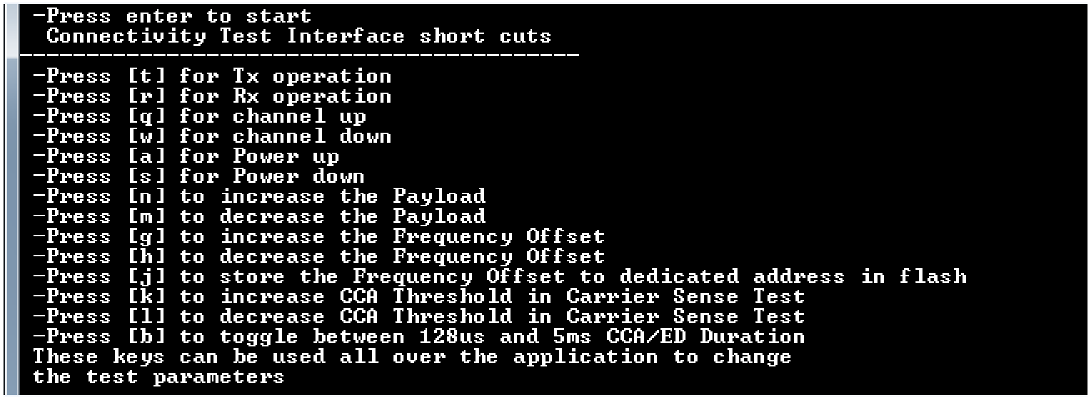
The MKW01 Connectivity Test application is a SMAC-based demo application, which provides you with means to test the basic transmission-reception functionalities, along with several advanced testing features based on the SMAC and ASP APIs.

Similar to the Wireless Messenger application, the Connectivity Test application can be configured at runtime (most of the settings), but several options must be set at compile time (for example addressing is set to broadcast by default, and it can be changed only from the source files).

4.2 MKW01 Connectivity Test application configuration

The compile-time settings of the application can be updated in the same way as for the MKW01 Wireless UART application. Most of the configuration settings can be found in the *Connectivity_Test_Platform.h* file. See Section 2.2, “Application configuration”.

The runtime configuration is performed using shortcut keys, which are available in most of the application menus. This is similar to the Wireless Messenger shortcuts menu, but it provides more options. Not all the options are necessary at a certain point, but different menus or tests change their behavior based on the settings applied.



```
-Press enter to start
 Connectivity Test Interface short cuts
------------------------------------------
-Press [t] for Tx operation
-Press [r] for Rx operation
-Press [q] for channel up
-Press [w] for channel down
-Press [a] for Power up
-Press [s] for Power down
-Press [n] to increase the Payload
-Press [m] to decrease the Payload
-Press [g] to increase the Frequency Offset
-Press [h] to decrease the Frequency Offset
-Press [j] to store the Frequency Offset to dedicated address in flash
-Press [k] to increase CCA Threshold in Carrier Sense Test
-Press [l] to decrease CCA Threshold in Carrier Sense Test
-Press [b] to toggle between 128us and 5ms CCA/ED Duration
These keys can be used all over the application to change
the test parameters
```

Figure 4-1. Connectivity Test shortcuts menu

Here is the description and effect of the keys depicted in Figure 4-1:

- ‘t’ – brings up the configuration menu for the transmitter in both the *PER* and *Range* tests.
- ‘r’ – brings up the configuration menu for the receiver in both the *PER* and *Range* tests.
- ‘q’ – increments the channel number. If pressed when the current channel is the last channel in the range, the channel number changes to 0.
- ‘w’ – decrements the channel number. If pressed when the current channel is 0, the channel number changes to the last channel in the channel range.

- ‘a’ – increments the output power value. If the output power is at the maximum and this key is pressed, the output power goes to the minimum (in this case 0x00).
- ‘s’ – decrements the output power value. If the output power is at the minimum and this key is pressed, the output power goes to the maximum (in this case 0x1F). These are not directly mapped to the *dBm* values, but the output power value is written to the appropriate register instead. Please see the reference manual to determine the relationship between the selected value and the power in *dBm*.
- ‘n’ – increments the length of the payload. This value is used in both PER TX and Transmission Control tests to build up the test payload.
- ‘m’ – decrements the length of the payload. The incrementation and decrementation are performed in the [17 , 243] interval. All the overflows at one end lead to setting the value at the other end.
- ‘k’ – increments the CCA threshold for the Carrier Sense test. In this test, a custom CCA before TX method is simulated at the application level, and the channel idle threshold is established using this parameter.
- ‘l’ – decrements the CCA threshold for the Carrier Sense test.
- ‘g’ – increases the channel frequency by a Fstep value (57 Hz or 61 Hz, depending on the frequency of the oscillator).
- ‘h’ – decreases the channel frequency by a Fstep value.
- ‘j’ – stores the number of Fstep values that need to be added or substracted from the channel frequency. This key, together with ‘g’ and ‘h’, helps you in calibrating the MKW01 channel frequency. For example, you can use the TX unmodulated feature to send a carrier signal OTA. You can then connect the MKW01 platform to a spectrum analyzer, and use ‘g’ and ‘h’ to fine-tune the frequency by adding or substracting an offset. Once the desired value is obtained, you can press ‘j’ to store the offset to flash, so that the MKW01 is still calibrated after the reset.
- ‘b’ – toggles between two CCA duration intervals (128 ms and 5 ms).

4.3 MKW01 Connectivity Test application usage

The MKW01 Connectivity Test application has the following main features:

a) *Continuous Test* – this menu option displays the following test suites:

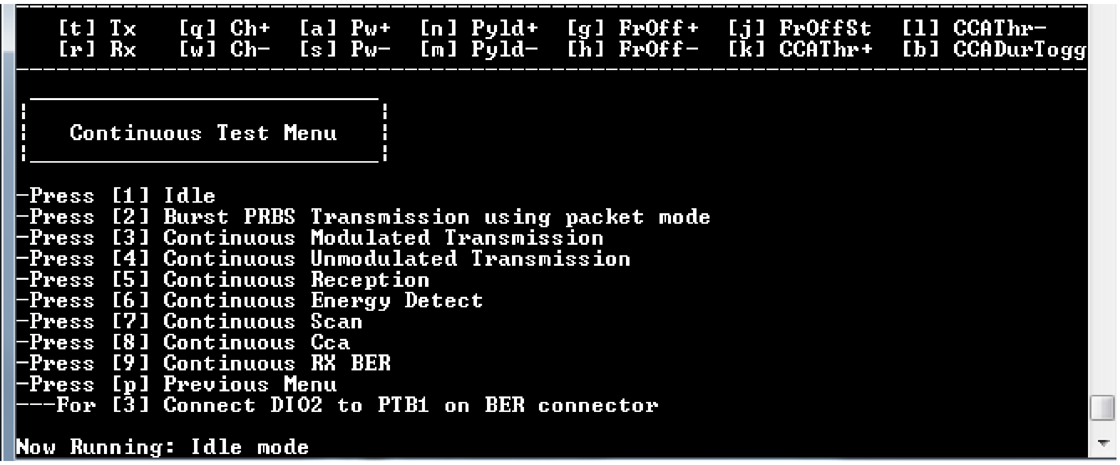
   – *IDLE* – this option sets the transceiver and all the state machines to idle.

   – *Burst PRBS Transmission using packet mode* – this option continuously sends packets containing a pseudo-random payload of fixed length.

   – *Continuous Modulated Transmission* – this option enables you to select between modulating 1’s, 0’s, or a pseudo-random sequence (PN), and sending them continuously OTA (in continuous mode). To use this mode, you must install a jumper on pins (3 – 4) of J18 for both FRDM-KW01 and MRB-KW01. USB-KW01 does not have a BER connector, but you can connect the TP24 test point with TP44.

   – *Continuous Unmodulated Transmission* – this option enables you to send unmodulated signal OTA, with frequency equal to the central frequency of the currently selected channel.

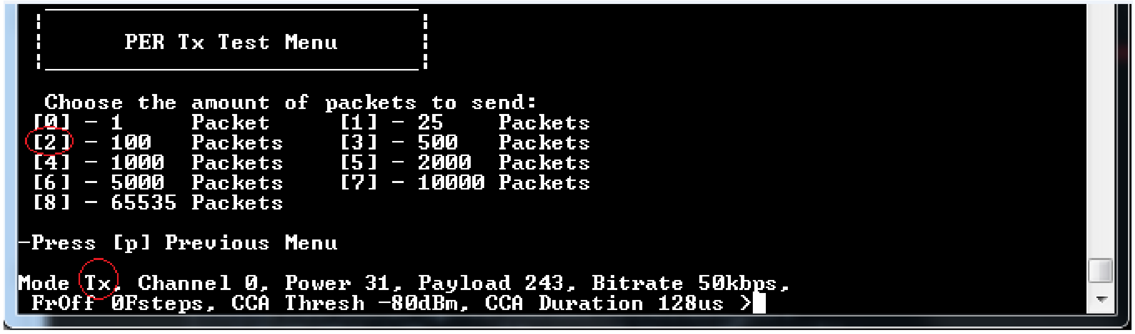
- *Continuous Reception* – this test places the transceiver to reception state, and dumps the payload bytes of the received packets to the TERM in ASCII-converted hexadecimal characters.
- *Continuous Energy Detect* – this option launches consecutive energy-detect requests in fixed hardcoded intervals for the current channel, and prints their values to the TERM.
- *Continuous Scan* – this option is similar to the previous one, except that it obtains the energy values on all channels at each iteration.
- *Continuous CCA* – this option launches consecutive CCA requests for the currently selected channel at a fixed hardcoded interval, and prints “Idle” or “Busy”, depending on the CCA result.
- *Continuous RX BER* – this option sets the transceiver into a continuous receive mode. Raw data is demodulated, and sent to pin 3 of J18 (DIO2) for both FRDM-KW01 and MRB-KW01 and to TP44 for USB-KW01. Synchronization is achieved by using the clock signal on pin 2 of J18 (DIO1) for FRDM-KW01 and MRB-KW01 and on TP47 for USB-KW01.



```
   [t] Tx    [q] Ch+  [a] Pw+  [n] Pyld+  [g] FrOff+  [j] FrOffSt  [l] CCAThr-
   [r] Rx    [w] Ch-  [s] Pw-  [m] Pyld-  [h] FrOff-  [k] CCAThr+  [b] CCADurTogg

  ___________________________
 |                           |
 |   Continuous Test Menu    |
 |___________________________|

-Press [1] Idle
-Press [2] Burst PRBS Transmission using packet mode
-Press [3] Continuous Modulated Transmission
-Press [4] Continuous Unmodulated Transmission
-Press [5] Continuous Reception
-Press [6] Continuous Energy Detect
-Press [7] Continuous Scan
-Press [8] Continuous Cca
-Press [9] Continuous RX BER
-Press [p] Previous Menu
---For [3] Connect DIO2 to PTB1 on BER connector

Now Running: Idle mode
```

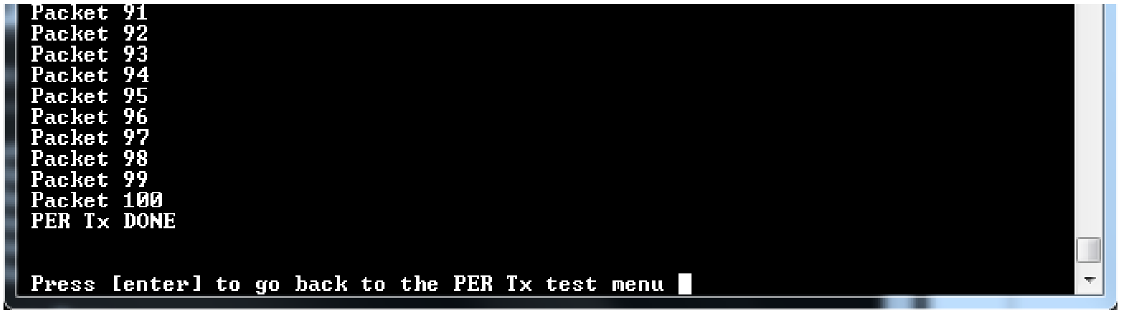
Figure 4-2. Continuous Test menu entries for FRDM-KW01

b) *Packet Error Rate* – this menu option displays a configuration menu for testing the packet error rate. The menu displayed also depends on the ‘r’ or ‘t’ shortcut keys. If ‘r’ is pressed, the following menu appears for *PER RX*, otherwise, a menu for *PER TX* appears. For example, if two MKW01 platforms have Connectivity Test loaded, one of the boards can be set to the RX, and the other to TX, as shown in the following figures.



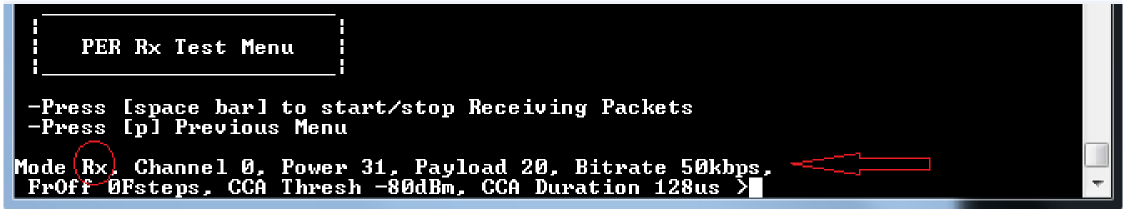
```
 ┌──────────────────────────┐
 │      PER Tx Test Menu     │
 └──────────────────────────┘

  Choose the amount of packets to send:
 [0] - 1      Packet     [1] - 25    Packets
 [2] - 100    Packets    [3] - 500   Packets
 [4] - 1000   Packets    [5] - 2000  Packets
 [6] - 5000   Packets    [7] - 10000 Packets
 [8] - 65535  Packets

-Press [p] Previous Menu

Mode Tx, Channel 0, Power 31, Payload 243, Bitrate 50kbps,
 FrOff 0Fsteps, CCA Thresh -80dBm, CCA Duration 128us >
```

Figure 4-3. Entering the PER menu with TX option selected, and pressing ‘2’ to send 100 packets



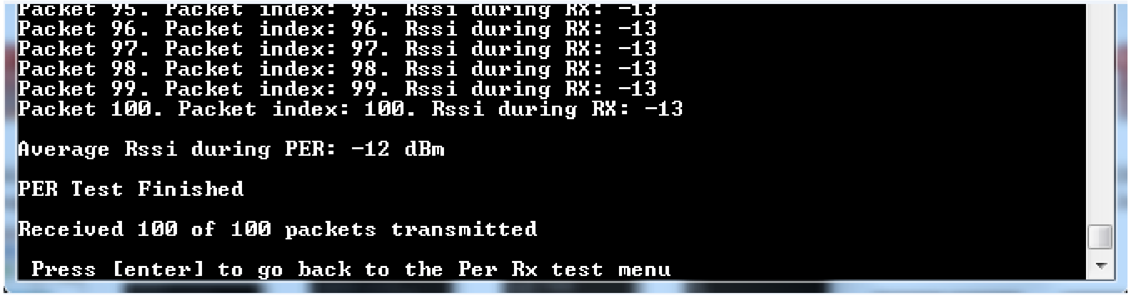
```
 Packet 91
 Packet 92
 Packet 93
 Packet 94
 Packet 95
 Packet 96
 Packet 97
 Packet 98
 Packet 99
 Packet 100
 PER Tx DONE


 Press [enter] to go back to the PER Tx test menu
```

Figure 4-4. PER TX menu after sending the 100 packets burst



```
 ┌──────────────────────┐
 │    PER Rx Test Menu   │
 └──────────────────────┘

 -Press [space bar] to start/stop Receiving Packets
 -Press [p] Previous Menu

Mode Rx, Channel 0, Power 31, Payload 20, Bitrate 50kbps,
 FrOff 0Fsteps, CCA Thresh -80dBm, CCA Duration 128us >
```

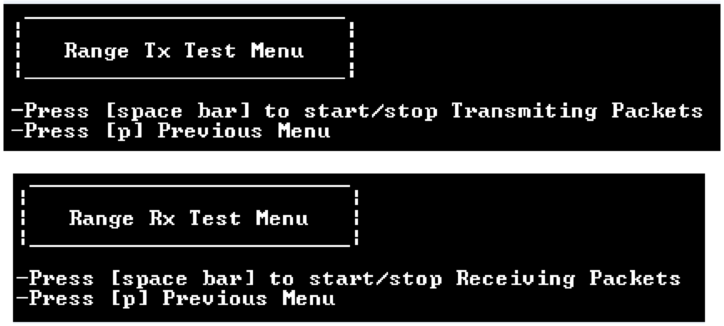
Figure 4-5. PER Test menu with RX option selected on the second platform



```
Packet 95. Packet index: 95. Rssi during RX: -13
Packet 96. Packet index: 96. Rssi during RX: -13
Packet 97. Packet index: 97. Rssi during RX: -13
Packet 98. Packet index: 98. Rssi during RX: -13
Packet 99. Packet index: 99. Rssi during RX: -13
Packet 100. Packet index: 100. Rssi during RX: -13

Average Rssi during PER: -12 dBm

PER Test Finished

Received 100 of 100 packets transmitted

 Press [enter] to go back to the Per Rx test menu
```

Figure 4-6. PER RX after pressing [SPACE] and starting the test

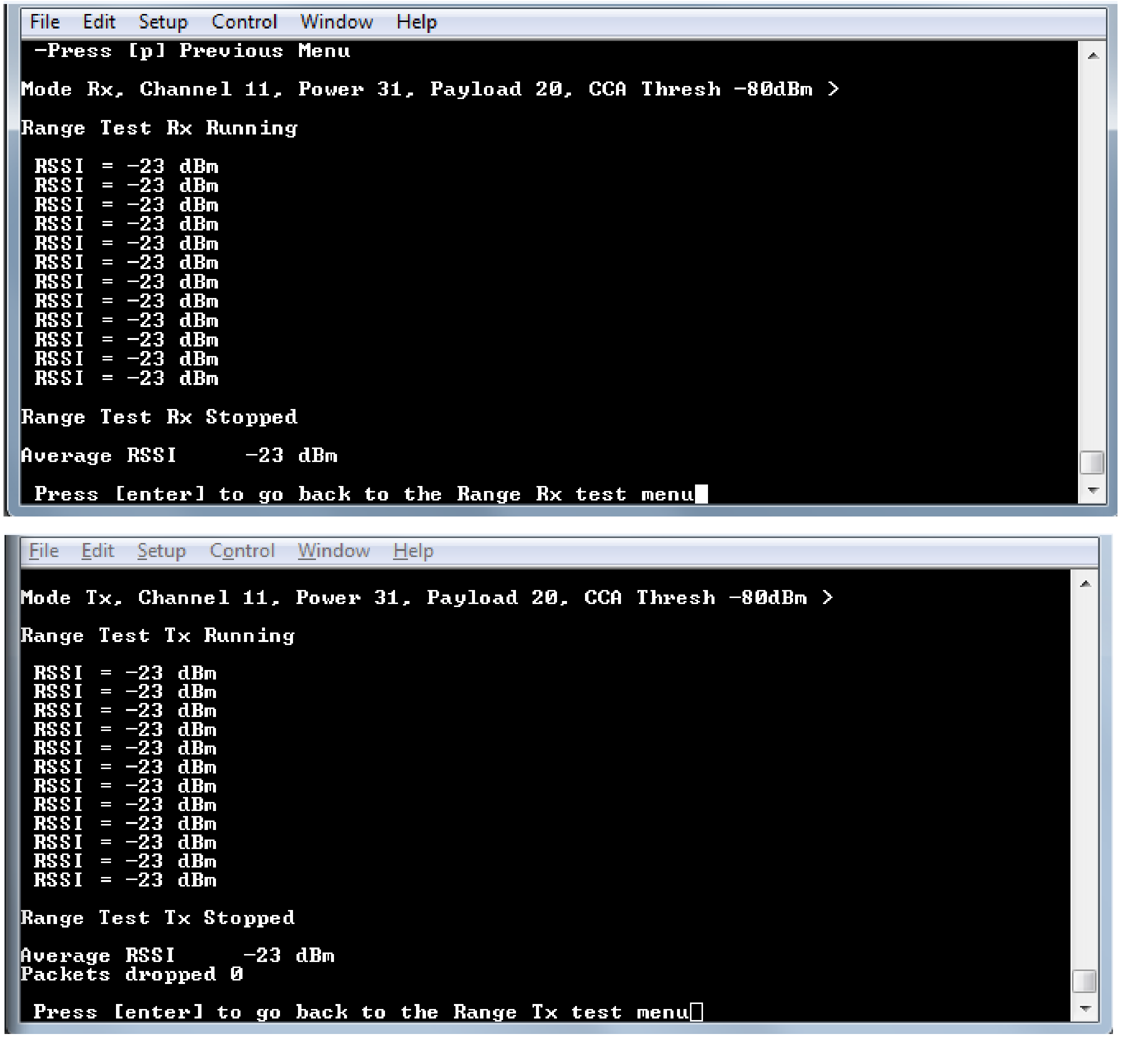
c) *Range Test* – this test displays a configuration menu that performs a ‘ping-pong’ test to help you in determining the range (distance between two platforms), in which the MKW01 platform can function properly. The submenu also depends on the ‘r’ and ‘t’ shortcuts, so that one of the platforms can be the initializer (first to start the TX), and the other can respond to requests. The test can be started and stopped by your intervention, and it displays the received signal strength indication for each received packet during its execution. At the end of the test, the platform configured as the initializer (TX) displays a summary of how many packets were lost, and what was the average RSSI.



```
 ______________________________
|                              |
|    Range Tx Test Menu        |
|______________________________|

-Press [space bar] to start/stop Transmiting Packets
-Press [p] Previous Menu
```

```
 ______________________________
|                              |
|    Range Rx Test Menu        |
|______________________________|

-Press [space bar] to start/stop Receiving Packets
-Press [p] Previous Menu
```

Figure 4-7. Range TX and RX submenus for two MKW01 platforms

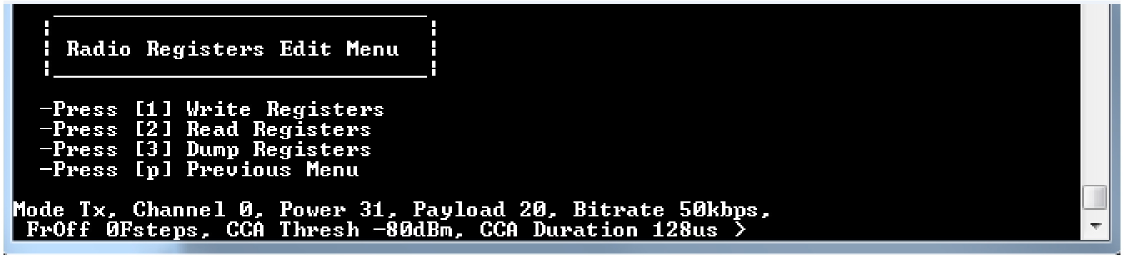


```
 -Press [p] Previous Menu

Mode Rx, Channel 11, Power 31, Payload 20, CCA Thresh -80dBm >

Range Test Rx Running

 RSSI = -23 dBm
 RSSI = -23 dBm
 RSSI = -23 dBm
 RSSI = -23 dBm
 RSSI = -23 dBm
 RSSI = -23 dBm
 RSSI = -23 dBm
 RSSI = -23 dBm
 RSSI = -23 dBm
 RSSI = -23 dBm
 RSSI = -23 dBm
 RSSI = -23 dBm

Range Test Rx Stopped

Average RSSI     -23 dBm

 Press [enter] to go back to the Range Rx test menu
```

```
Mode Tx, Channel 11, Power 31, Payload 20, CCA Thresh -80dBm >

Range Test Tx Running

 RSSI = -23 dBm
 RSSI = -23 dBm
 RSSI = -23 dBm
 RSSI = -23 dBm
 RSSI = -23 dBm
 RSSI = -23 dBm
 RSSI = -23 dBm
 RSSI = -23 dBm
 RSSI = -23 dBm
 RSSI = -23 dBm
 RSSI = -23 dBm
 RSSI = -23 dBm

Range Test Tx Stopped

Average RSSI     -23 dBm
Packets dropped 0

 Press [enter] to go back to the Range Tx test menu
```

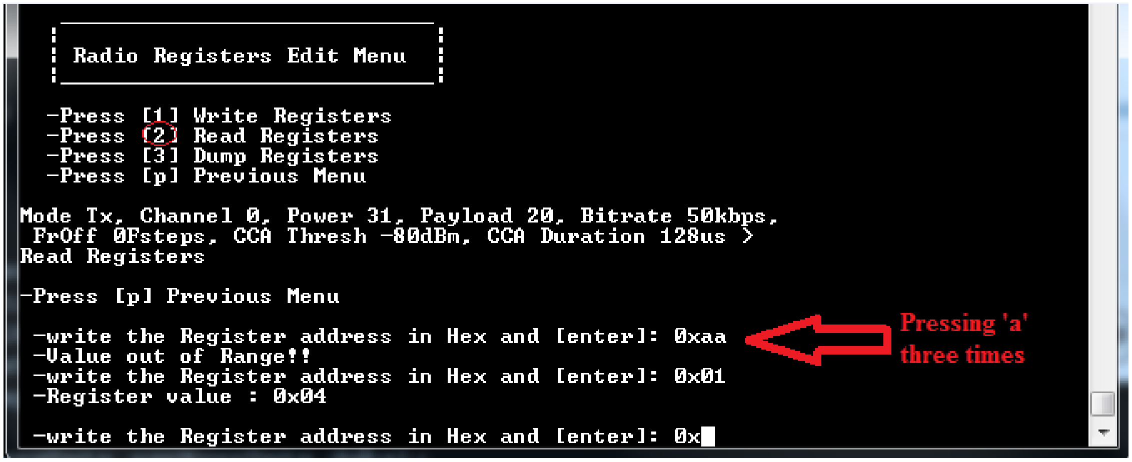
Figure 4-8. Range TX and RX after running the test for two MKW01 platforms

d) *Radio Registers Edit* – this menu enables you to read from and write to the transceiver registers, and to dump all address-value pairs from the transceiver registers to the TERM. The described features are accessible through the entries of this menu. For each access request (read or write) to a certain register, the register address is validated partially, and it is your task to access an existing register. For example, if the last accessible register is at 0xFD, the application only validates that the address is in the unsigned char range, but you must not request the register 0xFF. To ensure that a proper range is used, you can first use the dump register feature to see the valid address ranges.



```
 ________________________________
|                                |
| Radio Registers Edit Menu      |
|________________________________|

 -Press [1] Write Registers
 -Press [2] Read Registers
 -Press [3] Dump Registers
 -Press [p] Previous Menu

Mode Tx, Channel 0, Power 31, Payload 20, Bitrate 50kbps,
 FrOff 0Fsteps, CCA Thresh -80dBm, CCA Duration 128us >
```

Figure 4-9. Radio Registers Edit menu

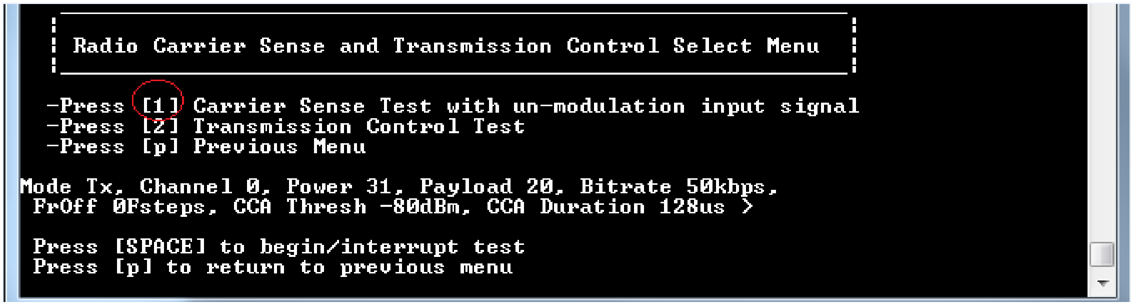


```
 ________________________________
|                                |
| Radio Registers Edit Menu      |
|________________________________|

 -Press [1] Write Registers
 -Press [2] Read Registers
 -Press [3] Dump Registers
 -Press [p] Previous Menu

Mode Tx, Channel 0, Power 31, Payload 20, Bitrate 50kbps,
 FrOff 0Fsteps, CCA Thresh -80dBm, CCA Duration 128us >
Read Registers

-Press [p] Previous Menu

 -write the Register address in Hex and [enter]: 0xaa
 -Value out of Range!!
 -write the Register address in Hex and [enter]: 0x01
 -Register value : 0x04

 -write the Register address in Hex and [enter]: 0x
```

Pressing 'a' three times

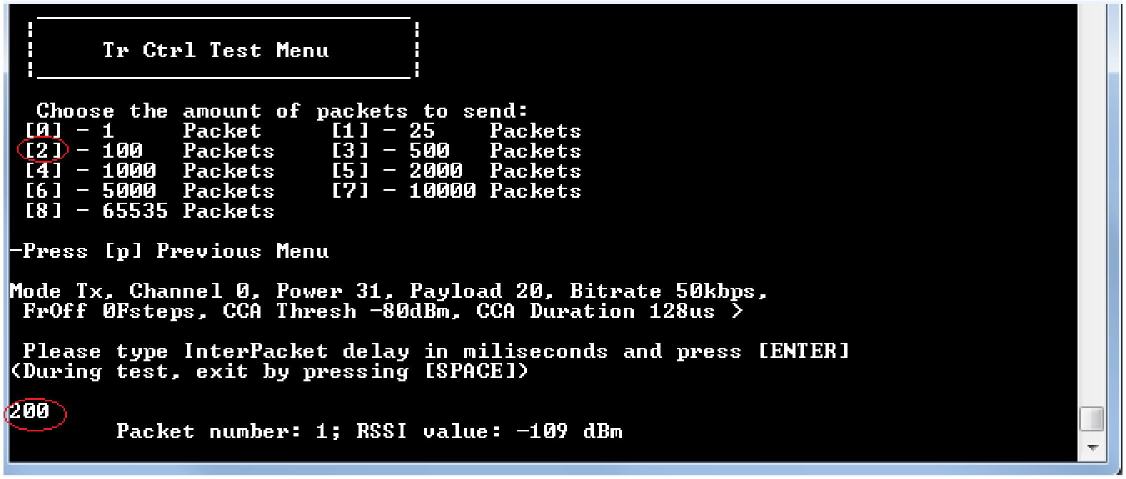
Figure 4-10. Read Registers example

e) *Carrier Sense and Transmission Control* – this menu enables you to choose between two tests. The first one is the Carrier Sense test, which performs ED continuously, until the ED value is above the CCA threshold (configured using ‘k’ and ‘l’ shortcuts), and then transmits a packet containing pseudo-random data with the payload size configured using ‘n’ and ‘m’ shortcuts. The other one is the Transmission Control test, which displays a selection menu for number of packets identical to the one in *PER TX* test, and then it prompts you to enter a decimal value resembling the inter-packet delay in miliseconds. After that, the application starts sending the selected number of packets with the selected inter-packet delay, using pseudo-random data for the payload, with the size configured using ‘n’ and ‘m’ shortcuts.



```
   ________________________________________________________
  |                                                        |
  | Radio Carrier Sense and Transmission Control Select Menu |
  |________________________________________________________|

  -Press [1] Carrier Sense Test with un-modulation input signal
  -Press [2] Transmission Control Test
  -Press [p] Previous Menu

Mode Tx, Channel 0, Power 31, Payload 20, Bitrate 50kbps,
 FrOff 0Fsteps, CCA Thresh -80dBm, CCA Duration 128us >

 Press [SPACE] to begin/interrupt test
 Press [p] to return to previous menu
```

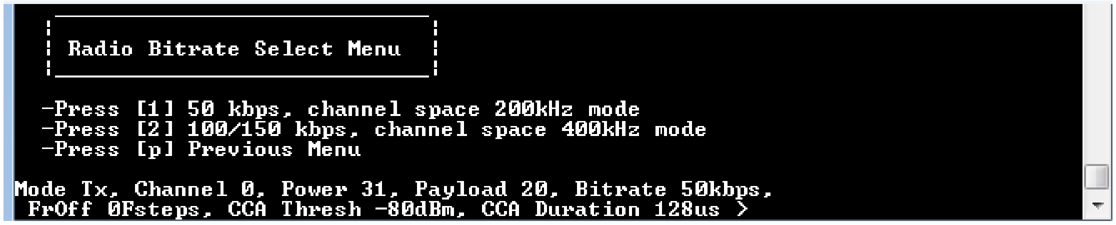
Figure 4-11. Carrier Sense submenu, pressing [SPACE] starts the ED loop



```
 ___________________________
|                           |
|      Tr Ctrl Test Menu    |
|___________________________|

  Choose the amount of packets to send:
 [0] - 1      Packet      [1] - 25    Packets
 [2] - 100    Packets     [3] - 500   Packets
 [4] - 1000   Packets     [5] - 2000  Packets
 [6] - 5000   Packets     [7] - 10000 Packets
 [8] - 65535 Packets

-Press [p] Previous Menu

Mode Tx, Channel 0, Power 31, Payload 20, Bitrate 50kbps,
 FrOff 0Fsteps, CCA Thresh -80dBm, CCA Duration 128us >

 Please type InterPacket delay in miliseconds and press [ENTER]
(During test, exit by pressing [SPACE])

200
        Packet number: 1; RSSI value: -109 dBm
```

Figure 4-12. Transmission Control submenu after selecting 100 packets and 200 ms delay

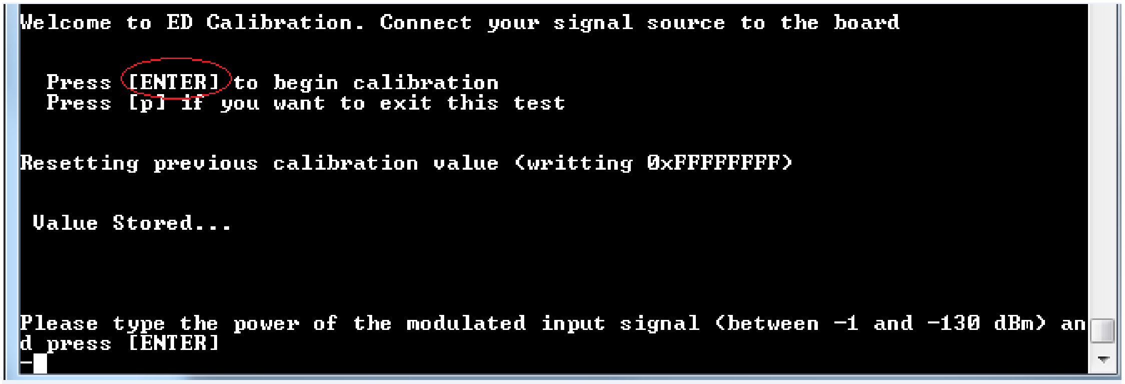
f) *Bitrate Select menu* – this option enables you to switch between two default PHY modes defined in *Connectivity_Test_Platform.h*. Currently, the default modes are the PHY modes 1 and 2.



```
   ___________________________
  |                           |
  | Radio Bitrate Select Menu |
  |___________________________|

  -Press [1] 50 kbps, channel space 200kHz mode
  -Press [2] 100/150 kbps, channel space 400kHz mode
  -Press [p] Previous Menu

Mode Tx, Channel 0, Power 31, Payload 20, Bitrate 50kbps,
 FrOff 0Fsteps, CCA Thresh -80dBm, CCA Duration 128us >
```

Figure 4-13. Bitrate Select menu

g) *Calibrate ED Measurement* – this is also a calibration feature. To use it, connect the signal generator to the MKW01 platform, and send a signal with a known power. Type in the power value in dBm and start the test. This test performs over 100 ED measurements, and computes the drift. This drift is stored to the flash, so that the calibration value can be used after reset.



```
Welcome to ED Calibration. Connect your signal source to the board

  Press [ENTER] to begin calibration
  Press [p] if you want to exit this test

Resetting previous calibration value (writting 0xFFFFFFFF)

 Value Stored...



Please type the power of the modulated input signal (between -1 and -130 dBm) an
d press [ENTER]
-
```

Figure 4-14. ED Measurement Calibration menu

Chapter 5
MKW01 Low-Power Demo application

5.1 Introduction

The MKW01 Low-Power Demo application helps you to learn how to handle the low-power use cases. There are two scenarios. The first scenario enables entering a desired low-power mode manually, by selecting the mode, the wake-up source, and the transceiver state during that mode. The second scenario is more dynamic, meaning that one platform (the receiver) runs on the internal clock with the transceiver in the listen mode, while the application configured as the transmitter periodically sends messages to wake up the receiver. The application is used with a TERM. The application configurations can be made both at compile time and at runtime (using the shortcut keys).

5.2 Application configuration

The application can be configured using three files:

- *SMAC_Config.h*

  This file contains the configuration of the source / destination PAN id (*gDefaultPanID_c*), source node address (*gNodeAddress_c*), number of maximum retries and retransmissions (*gMaxRetriesAllowed_c*), and the backoff interval for the custom backoff algorithm.

- *Application_Interface.h*

  This file provides the customization of the default channel number (*gDefaultChannelNumber_c*) and the default output power, which has to range between minimum and maximum output power (*gDefaultOutputPower_c*). This file also contains the definition for the two default PHY modes.

- *Low_Power_DemoApp.c*

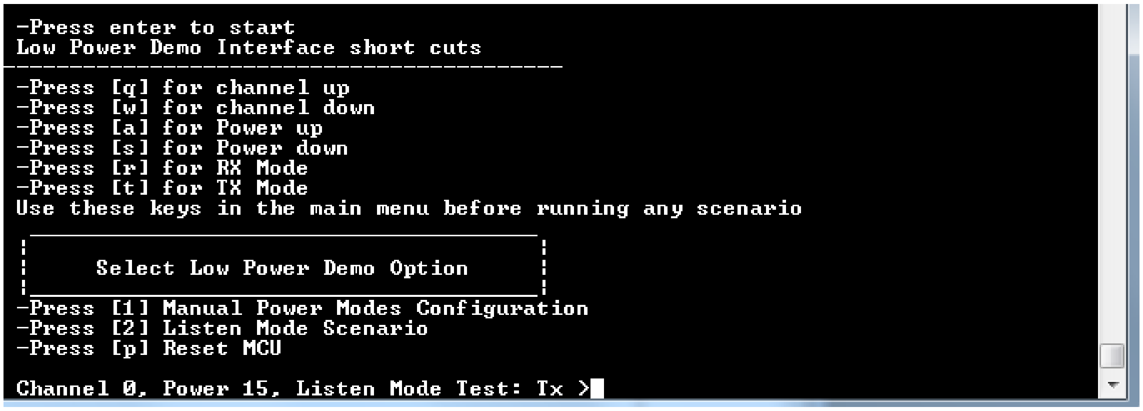
  The *InitApp* function enables you to change the baudrate of the serial port. For other options, see the definition of the baudrate macro (*gUARTBaudRate115200_c*), and pick another member from the baudrate enumeration. The listen interval is configured in this function, the wake-up message is prepared, and there is a switch to one of the default PHY modes defined in the *Application_Interface.h* header file.

The other settings must be left as they are, since they can affect the application behavior.

The application is developed for FRDM-KW01 and MRB-KW01 and it contains three project files configured to run either on RTOS (MQX or FreeRTOS) configuration, or bare-metal configuration (non-preemptive task scheduler). For each project, Debug and Release configurations are available.

5.3 Running the Low-Power Demo application

On startup, the application displays a menu for selecting one of the scenarios described above. In the main menu, you can use the shortcuts to change the active channel, output power, and the role for the second scenario (receiver or transmitter).



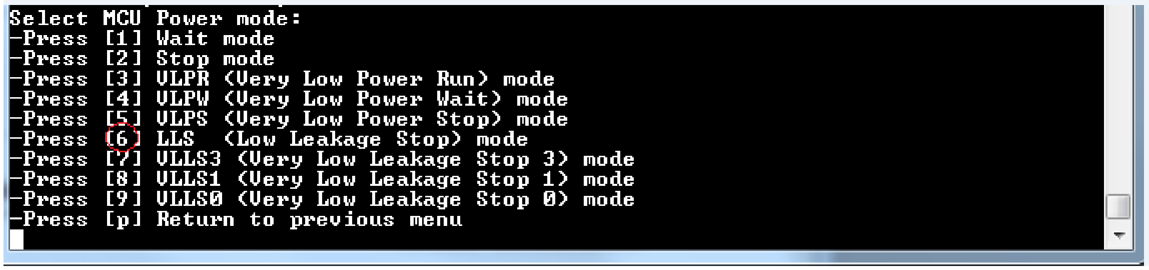
```
-Press enter to start
Low Power Demo Interface short cuts
------------------------------------------
-Press [q] for channel up
-Press [w] for channel down
-Press [a] for Power up
-Press [s] for Power down
-Press [r] for RX Mode
-Press [t] for TX Mode
Use these keys in the main menu before running any scenario
 _______________________________________
|                                       |
|     Select Low Power Demo Option      |
|_______________________________________|
-Press [1] Manual Power Modes Configuration
-Press [2] Listen Mode Scenario
-Press [p] Reset MCU

Channel 0, Power 15, Listen Mode Test: Tx >
```

Figure 5-1. Low-Power Demo main menu

5.3.1 Manual Power Modes Configuration

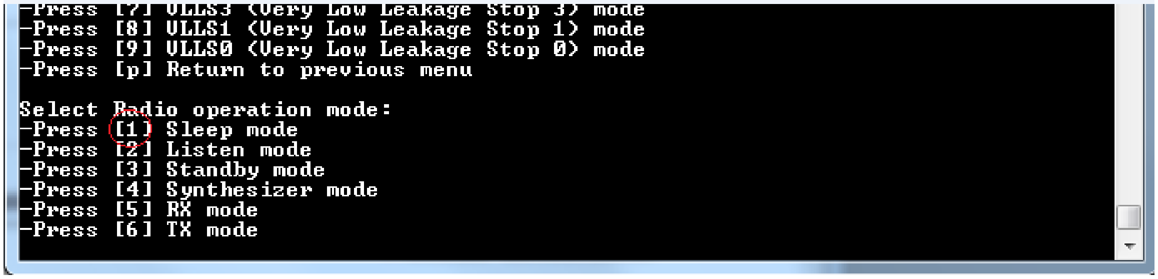
When entering this menu, a menu with available power modes appears. When you select one of the options, another menu with the available transceiver configurations pops up. After selecting a transceiver mode, the last menu prints the available wake-up sources with respect to the specified MCU mode. As soon as you select the wake-up source, all configurations are applied.

All VLLSx modes (0, 1, and 3) exit by reset, while other modes simply resume the application execution. The particular situation here is Very Low Power Run mode, which allows the MCU to continue its execution under special conditions. Although the MCU keeps running, the communication with the TERM is lost. To exit this mode, press and hold any push-button (except reset) for a longer time.



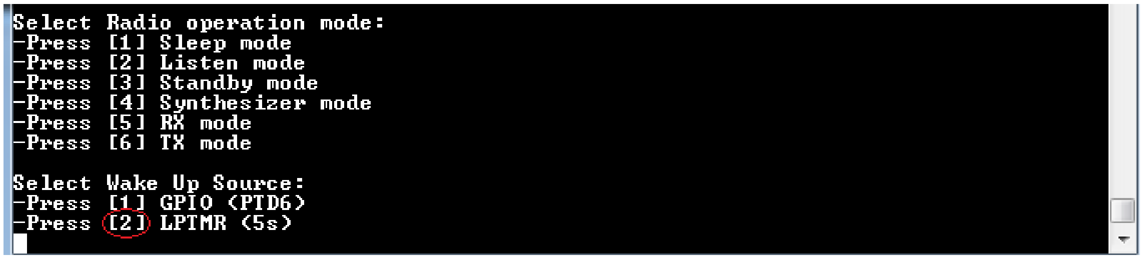
```
Select MCU Power mode:
-Press [1] Wait mode
-Press [2] Stop mode
-Press [3] VLPR (Very Low Power Run) mode
-Press [4] VLPW (Very Low Power Wait) mode
-Press [5] VLPS (Very Low Power Stop) mode
-Press [6] LLS  (Low Leakage Stop) mode
-Press [7] VLLS3 (Very Low Leakage Stop 3) mode
-Press [8] VLLS1 (Very Low Leakage Stop 1) mode
-Press [9] VLLS0 (Very Low Leakage Stop 0) mode
-Press [p] Return to previous menu
```

Figure 5-2. Selecting an available MCU mode



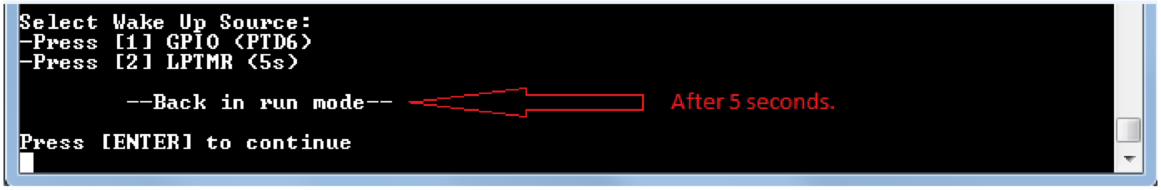
```
-Press [7] VLLS3 (Very Low Leakage Stop 3) mode
-Press [8] VLLS1 (Very Low Leakage Stop 1) mode
-Press [9] VLLS0 (Very Low Leakage Stop 0) mode
-Press [p] Return to previous menu

Select Radio operation mode:
-Press [1] Sleep mode
-Press [2] Listen mode
-Press [3] Standby mode
-Press [4] Synthesizer mode
-Press [5] RX mode
-Press [6] TX mode
```

Figure 5-3. Selecting an available transceiver configuration



```
Select Radio operation mode:
-Press [1] Sleep mode
-Press [2] Listen mode
-Press [3] Standby mode
-Press [4] Synthesizer mode
-Press [5] RX mode
-Press [6] TX mode

Select Wake Up Source:
-Press [1] GPIO (PTD6)
-Press [2] LPTMR (5s)
```

Figure 5-4. Selecting an available wake-up source



```
Select Wake Up Source:
-Press [1] GPIO (PTD6)
-Press [2] LPTMR (5s)

        --Back in run mode--

Press [ENTER] to continue
```

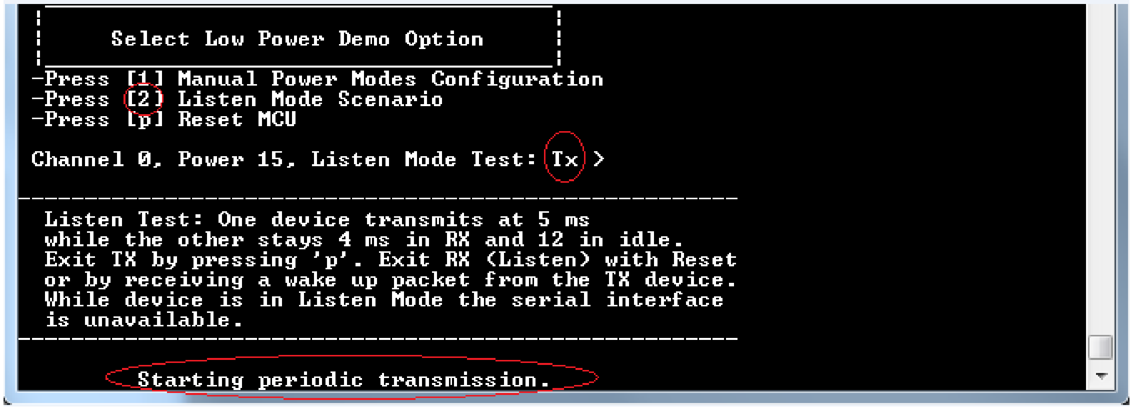
After 5 seconds.

Figure 5-5. Resuming after waking up from LLS

5.3.2 Listen mode scenario

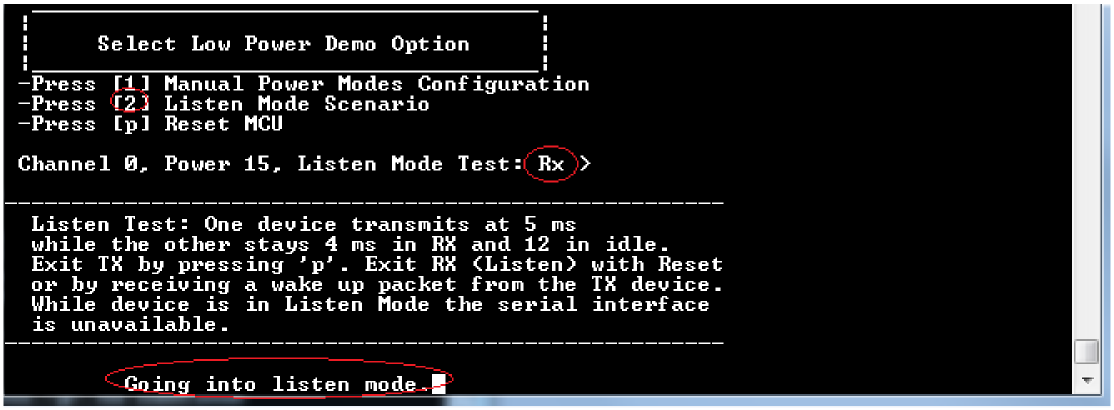
Before entering this menu, you must select the desired role. To enter the listen scenario as the transmitter, you must press ‘t’ (the configuration menu displays “Tx”). To assume the role of the receiver (switch to internal clock and switch the transceiver into listen mode, while PHY and SMAC go into RX), you must press ‘r’ and enter this menu.

The receiver exits the listen mode each time a packed is received, but if the packet does not contain the wake-up message, the MCU remains on the internal clock, and the transceiver goes back to the listen mode. The predefined settings for this scenario are: the transmitter sends a packet every 5 ms, while the receiver’s transceiver has the listen mode configured to sleep 12 ms, and to go into receive mode 4 ms.



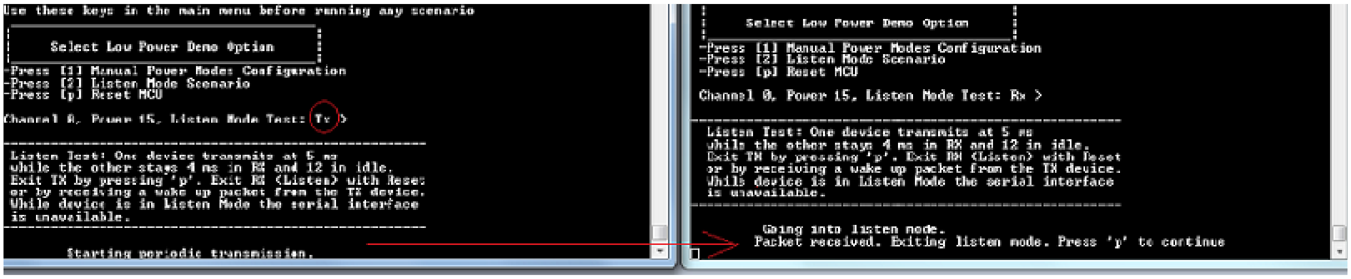
```
 |--------------------------------------|
 |                                      |
 |     Select Low Power Demo Option     |
 |                                      |
 |--------------------------------------|
-Press [1] Manual Power Modes Configuration
-Press [2] Listen Mode Scenario
-Press [p] Reset MCU

Channel 0, Power 15, Listen Mode Test: Tx >

------------------------------------------------------
 Listen Test: One device transmits at 5 ms
 while the other stays 4 ms in RX and 12 in idle.
 Exit TX by pressing 'p'. Exit RX (Listen) with Reset
 or by receiving a wake up packet from the TX device.
 While device is in Listen Mode the serial interface
 is unavailable.
------------------------------------------------------

        Starting periodic transmission.
```

Figure 5-6. Listen mode scenario, TX role



```
 |                                        |
 |    Select Low Power Demo Option        |
 |________________________________________|
-Press [1] Manual Power Modes Configuration
-Press [2] Listen Mode Scenario
-Press [p] Reset MCU

Channel 0, Power 15, Listen Mode Test: Rx >

-------------------------------------------------------
 Listen Test: One device transmits at 5 ms
 while the other stays 4 ms in RX and 12 in idle.
 Exit TX by pressing 'p'. Exit RX (Listen) with Reset
 or by receiving a wake up packet from the TX device.
 While device is in Listen Mode the serial interface
 is unavailable.
-------------------------------------------------------

        Going into listen mode.
```

Figure 5-7. Listen mode scenario, RX role



```
Use these keys in the main menu before running any scenario
 ________________________________________
|                                        |
|    Select Low Power Demo Option        |
|________________________________________|
-Press [1] Manual Power Modes Configuration
-Press [2] Listen Mode Scenario
-Press [p] Reset MCU

Channel 0, Power 15, Listen Mode Test: Tx >

-------------------------------------------------------
 Listen Test: One device transmits at 5 ms
 while the other stays 4 ms in RX and 12 in idle.
 Exit TX by pressing 'p'. Exit RX (Listen) with Reset
 or by receiving a wake up packet from the TX device.
 While device is in Listen Mode the serial interface
 is unavailable.
-------------------------------------------------------

        Starting periodic transmission.
```

```
 |                                        |
 |    Select Low Power Demo Option        |
 |________________________________________|
-Press [1] Manual Power Modes Configuration
-Press [2] Listen Mode Scenario
-Press [p] Reset MCU

Channel 0, Power 15, Listen Mode Test: Rx >

-------------------------------------------------------
 Listen Test: One device transmits at 5 ms
 while the other stays 4 ms in RX and 12 in idle.
 Exit TX by pressing 'p'. Exit RX (Listen) with Reset
 or by receiving a wake up packet from the TX device.
 While device is in Listen Mode the serial interface
 is unavailable.
-------------------------------------------------------

        Going into listen mode.
       Packet received. Exiting listen mode. Press 'p' to continue
```

Figure 5-8. Listen mode scenario, waking up from listen