MKW01 Simple Media Access Controller (SMAC) Demonstration Applications

User's Guide

Document Number: MKW01SMACDAUG
Rev. 0.0
5/2015

freescale™ semiconductor

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

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 device, see the appropriate Reference Manual and/or Data Sheet.

Audience

This document is intended for application developers using the MKW01 device and these demonstration applications as a starting point for developing proprietary applications.

Organization

This document is organized into three chapters as follows.

Chapter 1 **Introduction** — Describes the required software and hardware for correct demonstration application setup.

Chapter 2 **Wireless Uart**— Details how to run the Wireless Uart application and explains available application configurations.

Chapter 3 **Wireless Messenger—** Details how to run the Wireless Messenger application and explains available application configurations.

Chapter 4 **Connectivity Test —** Details how to run the Connectivity Test application and explains available menu options and application configurations.

Chapter 5 **Low Power Demo —** Details how to run the Low Power Demo application and explains available menu options and application configurations.

Revision History

This is the first revision of this document(Rev. 0.0).

Revision History

Location	Revision
Entire document	This is the first document containing information on the KSDK and RTOS enabled SMAC Demonstration Applications.

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 Linq Quality Indicator

FRDM Freedom Board

USB USB Dongle

TWR Tower Card Platform

IDE	Integrated Development Environment.
MCU	MicroController Unit.
OTA	Over-the-air.
PC	Personal Computer.
RX	Receive.
TX	Transmit.
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
MKW01	FRDM-MKW01 or MRB-MKW01 or USB-MKW01

References

The following sources were referenced to produce this book:

[1] Freescale MKW01 Reference Manual (MKW01xxRM.pdf)

Chapter 1
Introduction

The Freescale MKW01 SMAC based Demonstration Applications have to be used with a serial port terminal application.

This chapter details hardware and software requirements to initially use the MKW01.

1.1 Supported Hardware

The MKW01 SMAC based Demo Applications are designed to work with the MKW01 platforms. The MRB-KW01 platform can be free-standing or plugged into the TWR-RF board.

For more information on the TWR-RF system, see the appropriate Freescale Tower, Modular Development Platform 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 7.40.1*.

1.3 Hardware Considerations

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

- Directly connecting from the PC to the mini USB connector (J16) of the MRB, or
- Mounting the MRB on a TWR-RF board, (if using this setup the cable from the PC should be conected to the mini USB connector (J2) of the TWR-RF board).

The jumper configurations for each of the above set-ups as well as the default jumper configuration for the KW01 Freedom Board are described in the tables below.

Table 1-1. 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	3 - 4
J12	5 - 6
J13	1 - 2
J13	3 - 4
J13	5 - 6
J13	7 - 8

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

header	Install jumper on pins
J17	2 - 3
J17	5 - 6
J17	8 - 9

To connect the USB to the TWR-RF platform, the only changes that must be applied are to J17. Instead of the (5 - 6) and (8 - 9) pairs install jumpers on (4 - 5) and (7 - 8).

Table 1-2. Default jumper configuration for FRDM-KW01

header	Install jumper on pins
J8	2 - 3
J9	1 - 2
J31	1 - 2

Figure 1-1. MRB-KW01

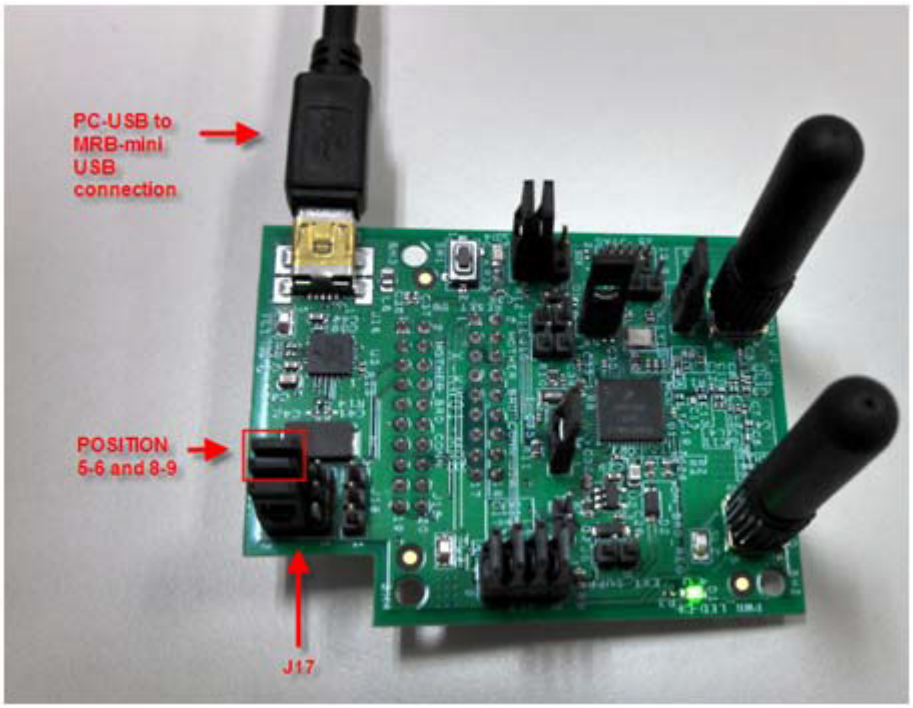


Figure 1-2. MRB-KW01 connection with the TWR-RF platform

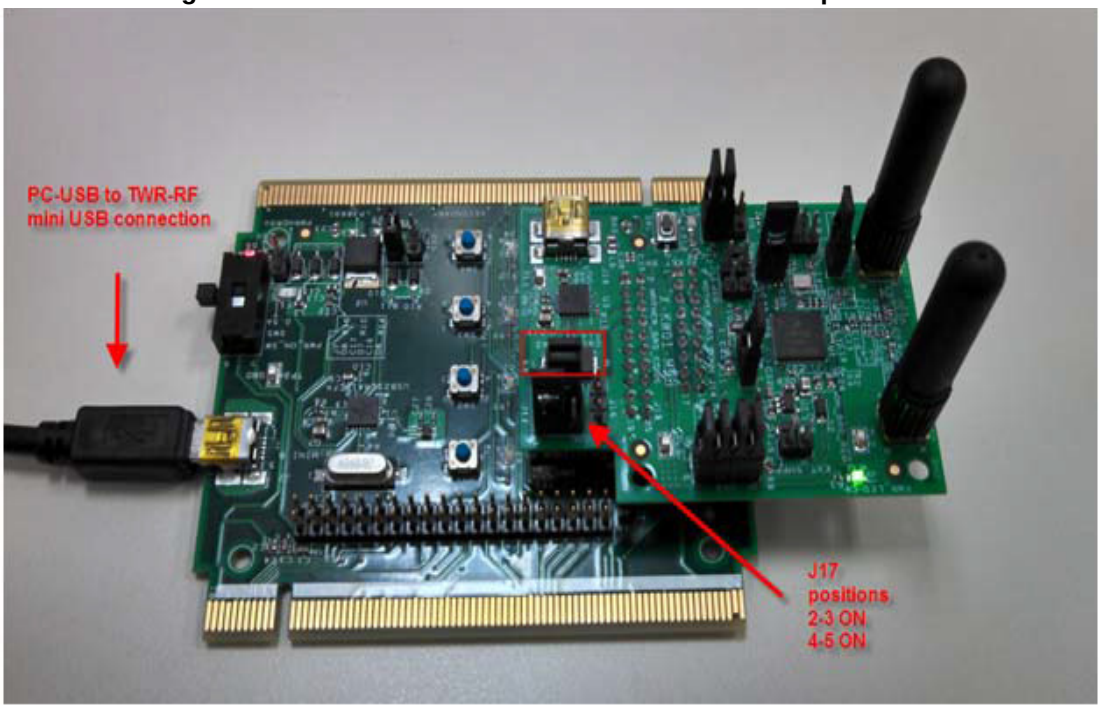


Figure 1-3. FRDM-KW01

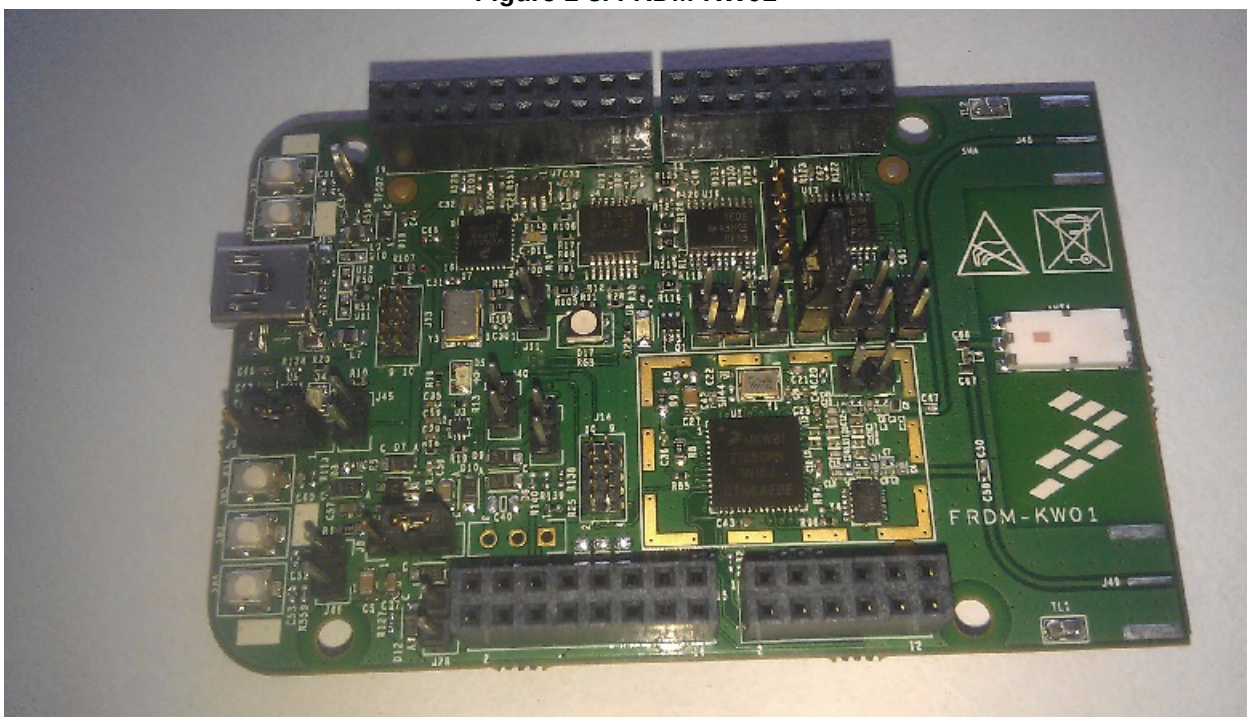
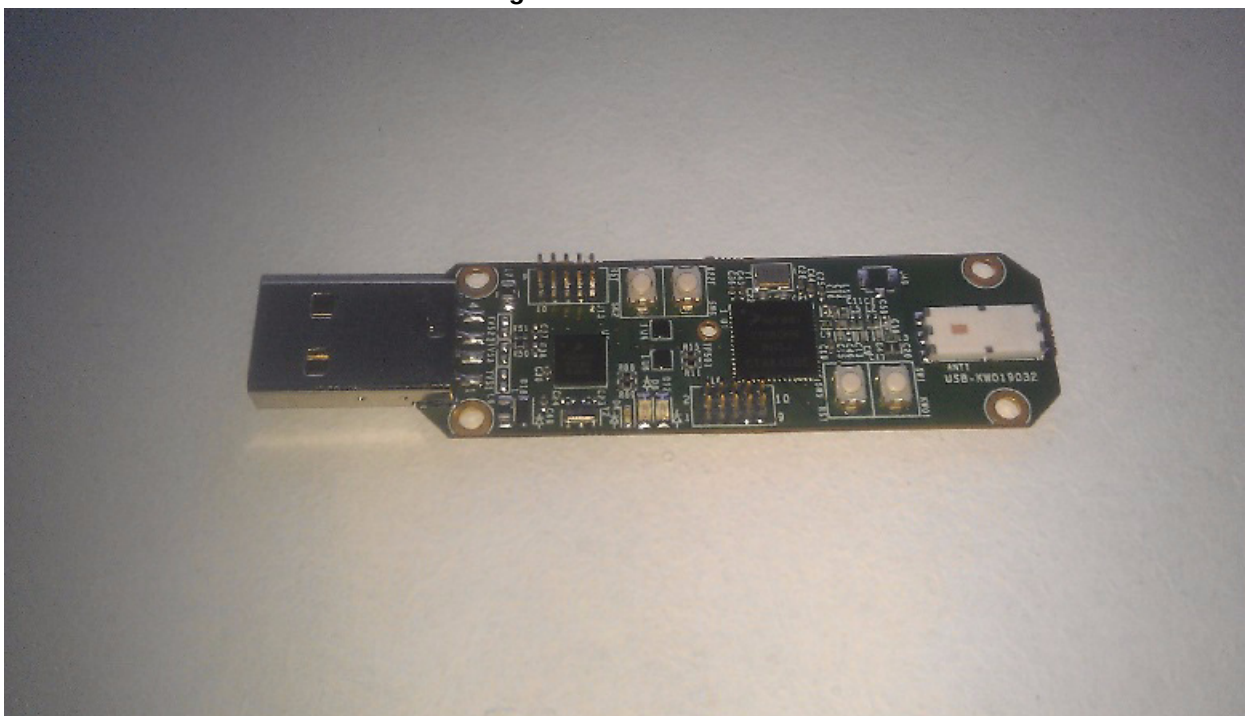


Figure 1-4. USB-KW01



Chapter 2
MKW01 Wireless Uart Application

2.1 Introduction

The MKW01 Wireless Uart application functions as an wireless UART bridge between two (one to one) or several (one to many) boards. The application can be used with both 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.

All of the application configurations must be done at compile time since Wireless Uart does not display any configuration menus.

2.2 Application configuration

The application can be configured from three files:

- SMAC_Config.h

  It contains the 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*), 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 re-transmission mechanisms. By setting the boolean fiels to *TRUE* and the numerical fields to the desired number of retries, respectively re-transmissions the user 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

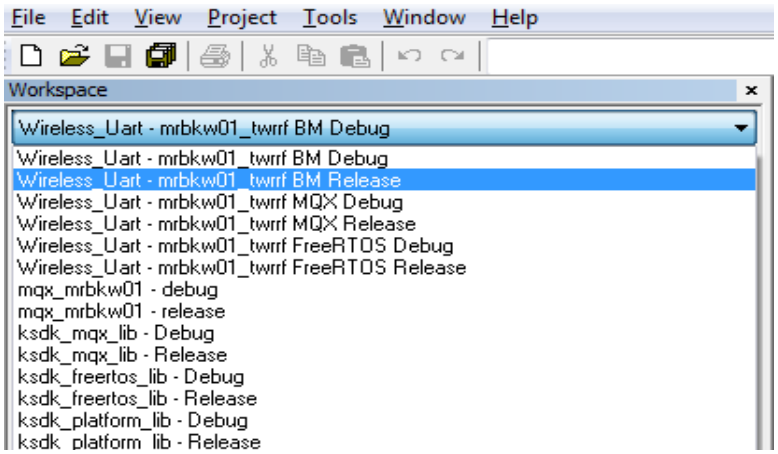
  Also, the *InitApp* function allows the user to change the baudrate of the serial port. For other options look at the definition of the baudrate macro (*gUARTBaudRate115200_c*) and pick another member from the baud rates enumeration. Moreover, this function also switches to one of the default PHY modes defined in the Application_Interface header file.

Other settings should be kept “as-is” since they could affect the application’s behavior. There are several settings that can be made in the IAR project compiler options to configure 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 project itself can be configured to run on RTOS (MQX or FreeRTOS) configuration or bare-metal configuration (non pre-emptive task scheduler). All three configurations are available both as Debug and Release configurations. Before building and downloading the application the user can switch to either of them from the Workspace tab in IAR.

Figure 2-1. Change between bare-metal and MQX in IAR



2.3 Running the Wireless Uart Application

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

- Two (or more) MKW01 platforms have to be connected to the PC using the mini-USB cables.
- Then, the code must be downloaded on the platforms via J-Link (or other means).
- After that, two (or more) TERM applications must be opened and the serial ports must be configured with the same baud rate as the one in the project (default is 115200). Other serial configurations necessary are 8 bit, no parity, 1 stop bit.
- To start the setup, each platform must be resett and one of the (user) push buttons found on the MKW01 platform must be pressed. This will start the state machine of the application and the user can start sending characters via UART to one of the platforms, which will be 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 re-transmission 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 cipher block chaining mode and is 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 changing the baud-rate and selecting between MQX target and bare metal target.

3.2 Wireless Messenger configuration menu

This paragraph describes the configuration menu and the shortcuts menu which aid the user in configuring the application. The options available in the shortcuts menu can be accessed both in the main menu and in the configuration menu. The console menu has all the shortcuts disabled so that they will not interfere with message typing. Also with security enabled the configuration menu is extended with another entry in the main menu which aids the user in configuring the initial vector and encryption key.

3.2.1 Shortcuts Menu

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

Figure 3-1. Wireless Messenger Welcome Screen

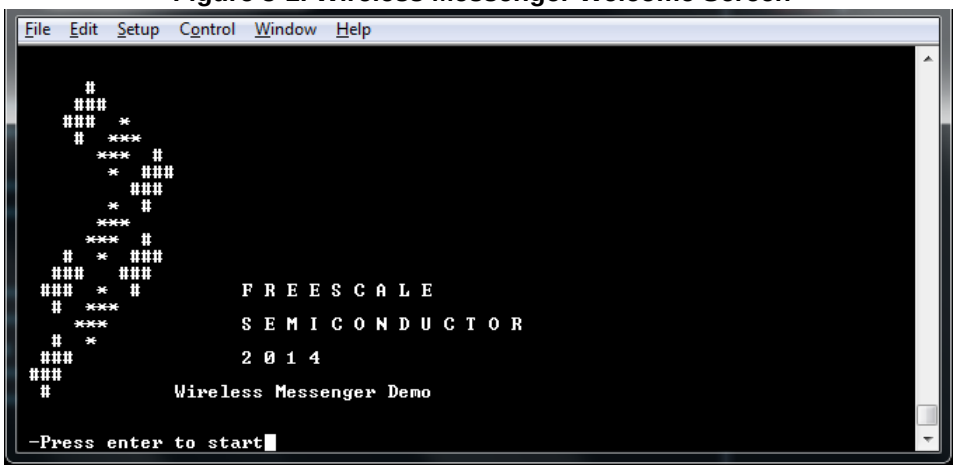
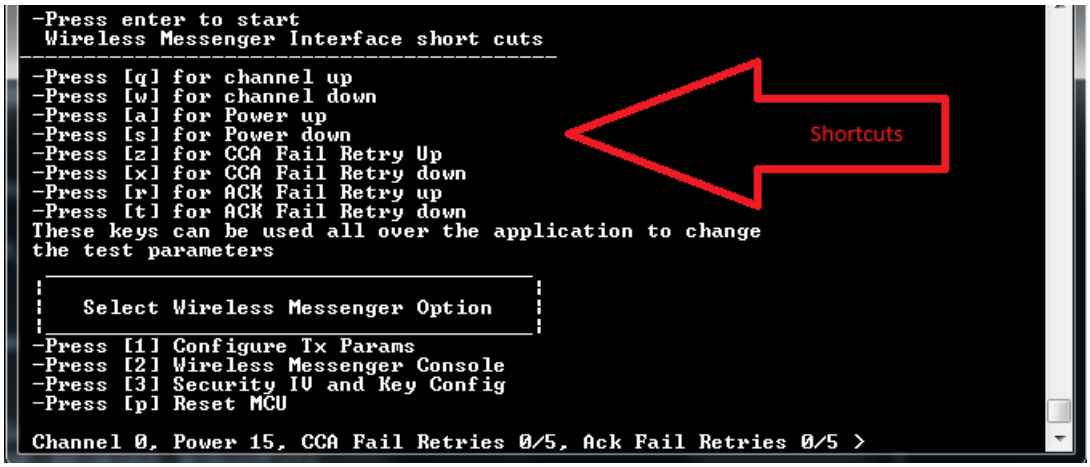


Figure 3-2. Shortcuts Menu for the MKW01 Wireless Messenger Demo App

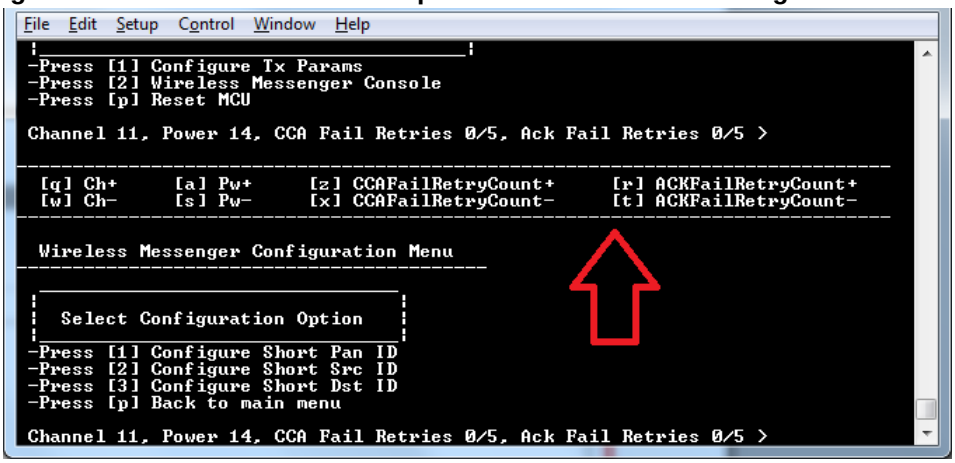


Given the above figure:

- pressing ‘q’ will increase the channel number. When it overflows the maximum channel number (dependent on frequency band and phy mode) the channel number will be set to 0.
- pressing ‘w’ will decrease the channel number. If previous channel number is 0, the next channel will be the last channel number.
- pressing ‘a’ and ‘s’ is similar to ‘q’ and ‘w’ but for setting the output power. The limits for this option are [0x01, 0x1F].
- pressing ‘z’ will increase the number of retransmissions in case of a channel busy scenario. Keep in mind that a number greater than 0 for this option will enable automatic CCA before TX option.
- pressing ‘x’ will decrease the number of retransmission in case of a channel busy status. If the number of retransmissions is decremented to 0, the feature will be disabled.
- pressing ‘r’ will increment the number of retries in case of a No Ack situation. If the number of retries is greater than 0, the automatic ACK mechanism is enabled. This mechanism is bypassed in case of a broadcast transmission.
- pressing ‘t’ will decrement the number of retries in case of a 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 Figure 3-3 ilustrates it. This indicates that the shortcuts are still active.

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



3.2.2 Configuration Menu

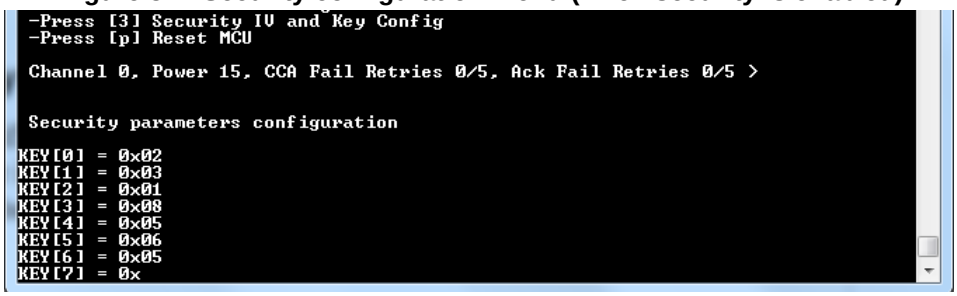
The configuration menu is presented in Figure 3-3. There are three options available:

- *Configure Short Pan ID*: this option will allow the user to set the source/destination PAN address. The user 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, the user can type both “0xFFFF” or “0xffff”.
- *Configure Short Src ID*: this option will allow the user to set the short address of the node. The option is similar to the one presented above.
- *Configure Short Dst ID*: this option will allow the user to set the destination address of the packets that will be sent from the console menu. The feature is similar to the *Configure Short Pan ID* option.

3.2.3 Security IV and Key Config

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

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 will not receive any packets OTA. Also, in this menu all configuration options are disabled.

In the console menu, the user can see packets intended for his MKW01 platform or broadcast packets. Also when the user types a message and hits [ENTER] the application will include the message into a packet payload and send it using the addressing information previously configured. The application will notify the user about the status of the packet by writing a status message to the serial port. The status can be either a “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 two and the algorithm finds the channel idle after a second attempt, the user will receive a “Packet Sent” message in the TERM. If the channel is busy after a second retry the user will receive a “Packet Sending Failed. Reason. Channel Busy!” notification.

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

Chapter 4
MKW01 Connectivity Test Application

4.1 Introduction

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

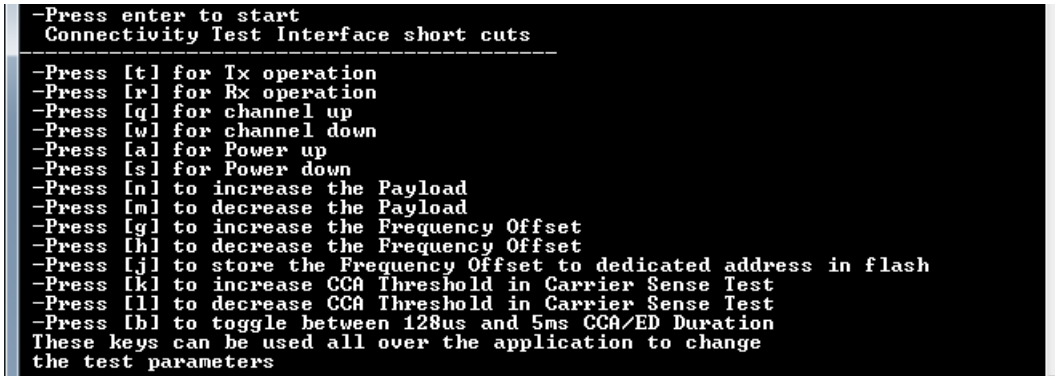
Similar to the *Wireless Messenger Application*, Connectivity Test can be configured both at run-time (most of the settings) but several options must be set at compile time (for example addressing is set default to broadcast and can be changed only from the project files).

4.2 MKW01 Connectivity Test Application Configuration

The compile time settings of the application can be updated in the same manner as for the MKW01 Wireless UART Application. Most of the configuration settings can be found in Connectivity_Test_Platform.h. Please read **2.2 Application configuration.**

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

Figure 4-1. Connectivity Test Shortcuts Menu



According to the above figure, the following keys have the effect described below:

- 't' : brings up the configuration menu for the transmitter in both *PER* and *Range* tests.
- 'r' : brings up the configuration menu for the receiver in both *PER* and *Range* tests.
- 'q' : increments channel number. If pressed when current channel is the last channel in the range, the channel number will change to 0.
- 'w' : decrements channel number. If pressed when current channel is 0, the channel number will change to the last channel in the channel range.
- 'a' : increments output power value. If output power is at maximum and this key is pressed, the output power will go to minimum (in this case 0x01).

- ‘s’ : decrements output power value. If output power is at minimum and this key is pressed, the output power will go to maximum (in this case 0x1F). These are not directly mapped to *dBm* values. Instead the output power value is written to the appropriate register. The user should consult the reference manual to determine the relationship between selected value and power in *dBm*.
- ‘n’ : increments the length of the payload. This value is used in both *PER TX* test to build-up the payload and in *Transmission Control* test for the same reason.
- ‘m’ : decrements the length of the payload. Incrementation and decrementation is performed in the [17, 243] interval. All overflows at one end lead to setting the other end’s value.
- ‘k’ : increments the CCA threshold for the *Carrier Sense* test. In this test the *CCA before TX* algorithm is implemented at 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 with a Fstep value (57 Hz or 61Hz depending on clock out value from the transceiver).
- ‘h’: decreases the channel frequency with 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’ aid the user in calibrating the MKW01 channel frequency. For example the user can use the TX unmodulated feature to send a carrier signal OTA. Then he can 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 the user can press ‘j’ to store the offset in flash so that after reset the MKW01 is still calibrated.
- ‘b’: toggles between two CCA durations intervals (128 us and 5ms).

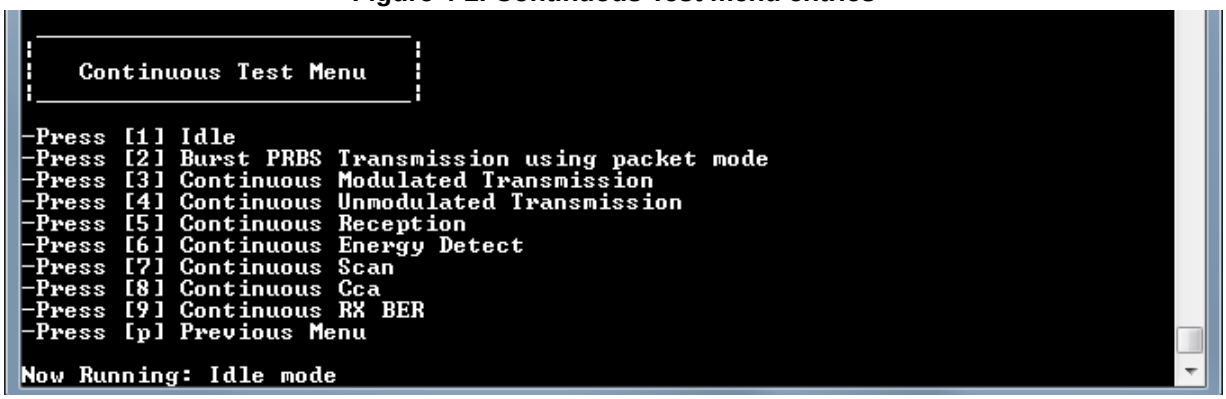
4.3 MKW01 Connectivity Test Application Usage

The MKW01 Connectivity Test Application has four main features:

a) *Continuous Tests*: This menu option will display several test suites
   - *IDLE*: this option will set the transceiver and all the state machines to idle.
   - *Burst PRBS Transmission using packet mode*: this option will continuosly send packets which contain a pseudo-random payload of fixed length.
   - *Continuous Modulated Transmission*: this option will allow the user to select between modulating 1’s, 0’s or a pseudo-random sequence (PN) and sending them OTA continuously (in continuous mode). To use this mode, a jumper must be installed on pins (3 - 4) of J18.
   - *Continuous Unmodulated Transmission*: this option will allow the user to send an unmodulated signal OTA having the frequency equal to the central frequency of the currently selected channel.
   - *Continuous Reception*: this test will place the transceiver in reception and will dump the payload bytes of the received packets to the TERM in ASCII converted hexadecimal characters.
   - *Continuous Energy Detect*: this option will launch consecutive energy detect requests at fixed hard-coded intervals for the current channel and will print their values to the TERM.

– *Continuous Scan*: this option is similar to the previous one, except that at each iteration it will obtain the energy values on all channels.
– *Continuous CCA*: this option will launch consecutive CCA requests for the currently selected channel, at a fixed hard-coded interval and will print “Idle” or “Busy” depending on the CCA result.
– *Continuous RX BER:* this option will set the transceiver into continuous receive mode. Raw data will be demodulated and sent to pin 3 of J18 (DIO2). Synchronization is achieved by using the clock signal on pin 2 of J18 (DIO1).

Figure 4-2. Continuous Test menu entries



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

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

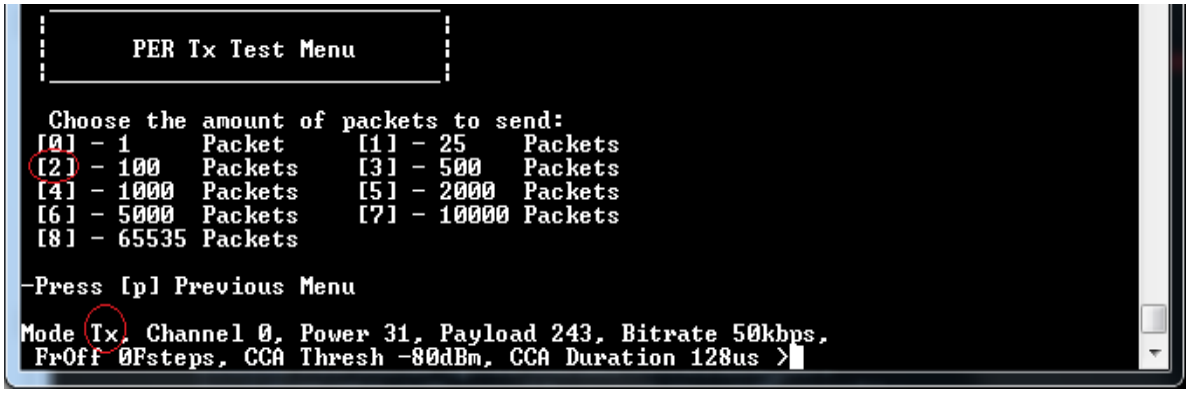


Figure 4-4. PER TX menu after sending the 100 packets burst

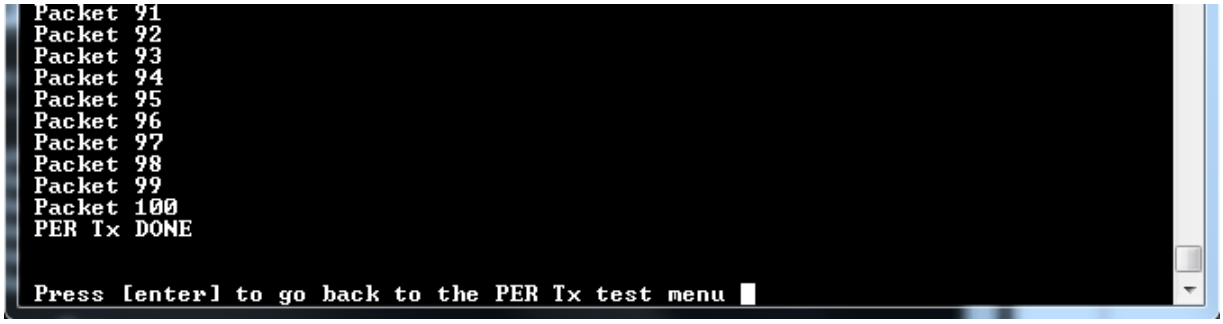


Figure 4-5. PER Test Menu with RX option selected on the second platform

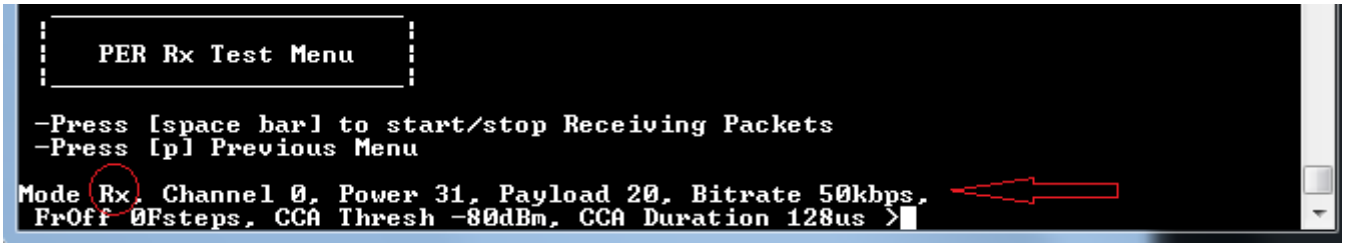
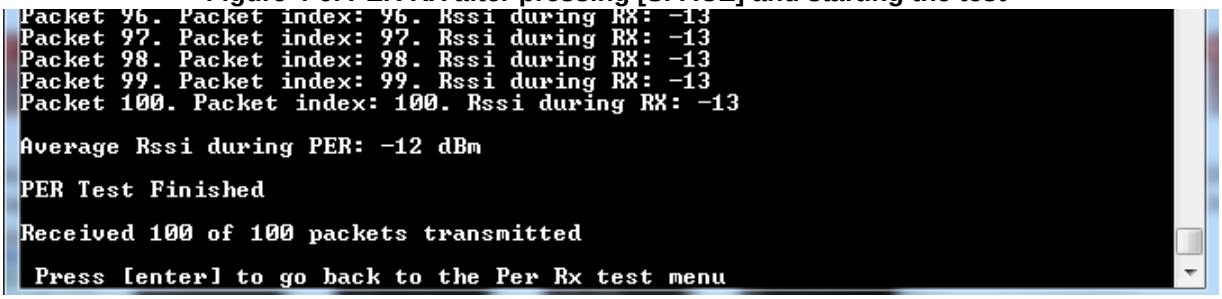


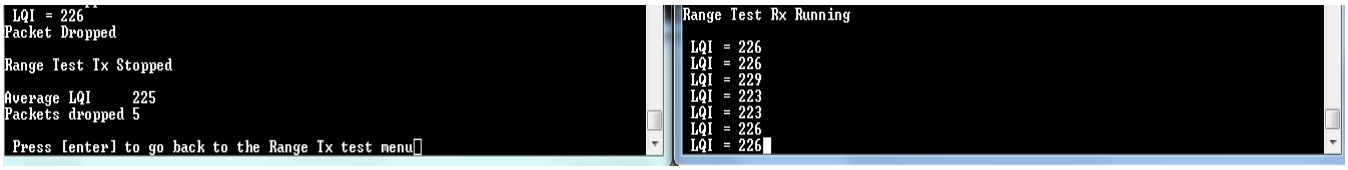
Figure 4-6. PER RX after pressing [SPACE] and starting the test



c) *Range Test*: This test will display a configuration menu that will perform a ‘ping-pong’ test to aid the user in determining the range (as distance between two platforms) in which the MKW01 platform can function properly. The sub-menu also depends on the ‘r’ and ‘t’ shortcuts so that one of the platforms can be the initializer (first to start a TX) and the other can respond to requests. The test is started and stopped only by user intervention and during its execution it will display the linq quality for each received packet. Also, at the end of the test, the platform configured as the initializer (TX) will display a summary of how many packets were lost and what was the average LQI.

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

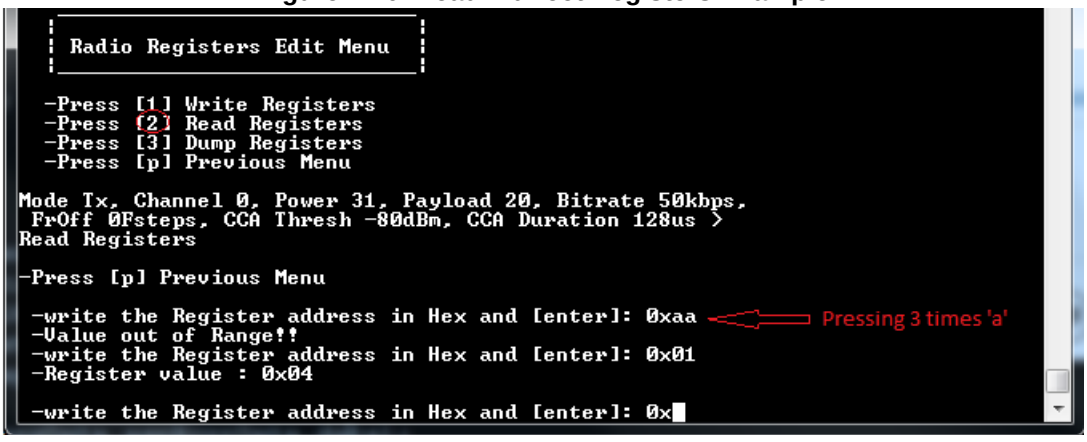
Figure 4-8. Range TX and RX after running the test for two MKW01 platforms



d) *Radio Registers Edit*: this menu will allow the user to read-write 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 the responsability of the user to access an existing register. For example if the last accessible register is at 0xFD, the application will only validate that the address is in the unsigned char range but user has the possibility to request register 0xFF. To ensure that a proper range is used, user can first use the dump register feature to see the valid address ranges.

Figure 4-9. Radio Registers Edit menu

Figure 4-10. Read Indirect Registers Example



e) *Carrier Sense and Transmission Control*: this menu will allow the user to choose between two tests. The former is the *Carrier Sense* test which will perform ED continuosly until the ED value is above the CCA threshold (configured using ‘k’ and ‘l’ shortcuts) and then transmit a packet which contains pseudo-random data with the payload size configured using ‘n’ and ‘m’ shortcuts. The latter is the *Transmission Control* test which displays a selection menu for number of packets identical with the one in *PER TX* test and then it prompts the user to enter a decimal value resembling the inter-packet delay in miliseconds. After that, the application will start sending the selected number of packets with the selected inter-packet delay, using pseudo-random data for the payload with the size configured with ‘n’ and ‘m’ shortcuts.

Figure 4-11. Carrier Sense sub-menu. Pressing [SPACE] will start the ED loop

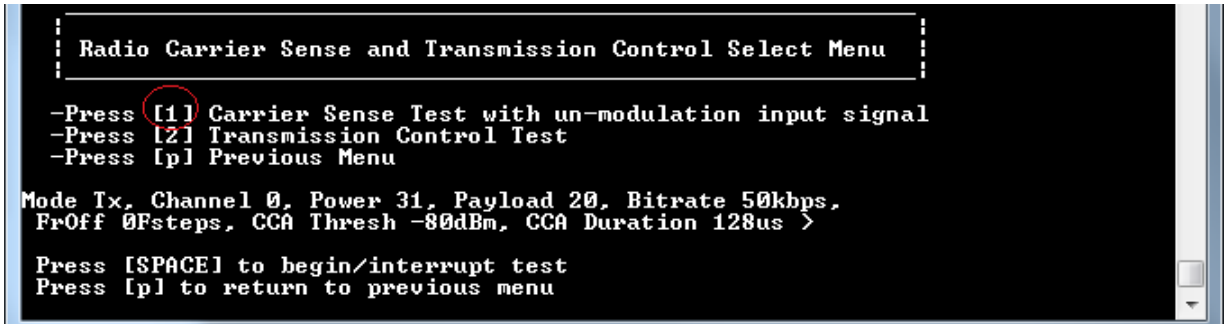
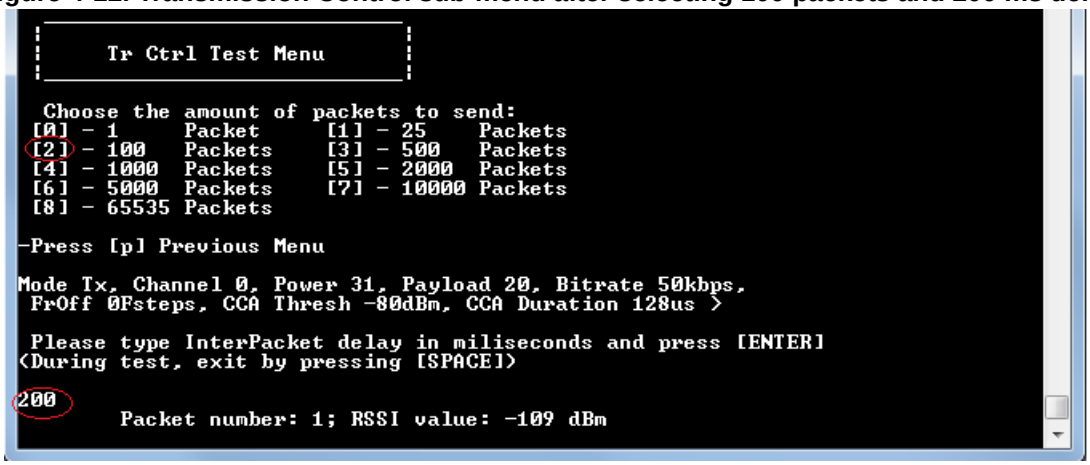
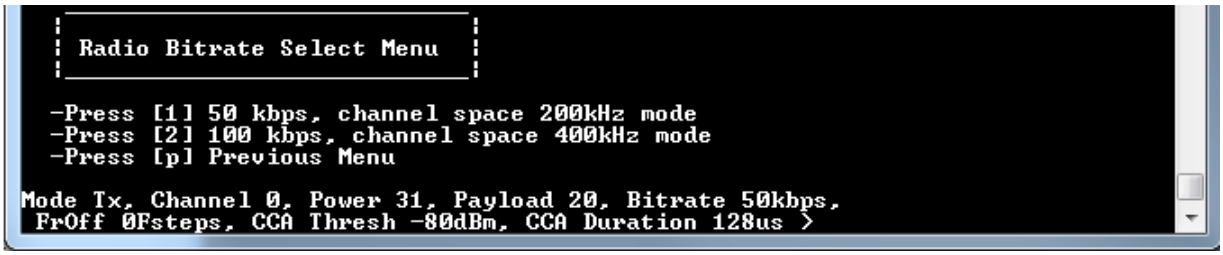


Figure 4-12. Transmission Control sub-menu after selecting 100 packets and 200 ms delayk



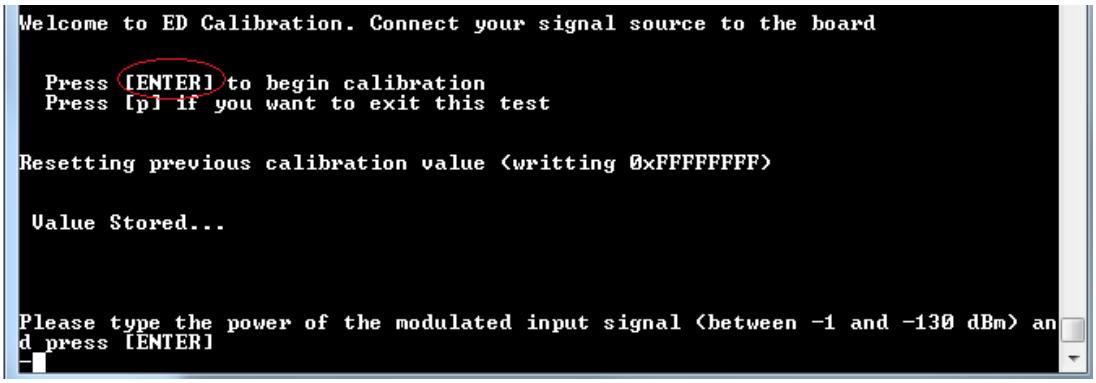
f) *Bitrate Select menu*: this option allows the user to switch between two default PHY modes defined in Connectivity_Test_Platform.h. Currently, the default modes are PHY modes 1 and 2.

Figure 4-13. Bitrate Select menu



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

Figure 4-14. Calibrate ED Measurement Menu



Chapter 5
MKW01 Low Power Demo Application

5.1 Introduction

The MKW01 Low Power Demo application aids the user in learning how to handle the low power use cases. There are two scenarios. The first scenario allows 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 listen mode, while the application configured as transmitter sends a message periodically to wake up the receiver. The application is used with a TERM. The application configurations can be done both at compile time and at runtime (using the shortcuts keys).

5.2 Application configuration

The application can be configured from three files:

- SMAC_Config.h

  It contains the 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 default channel number (*gDefaultChannelNumber_c*), 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.

- Low_Power_DemoApp.c

  The *InitApp* function allows the user to change the baudrate of the serial port. For other options look at the definition of the baudrate macro (*gUARTBaudRate115200_c*) and pick another member from the baud rates enumeration. Also, in this function the listen interval can be configured, the wake up message is prepared and there is a switch to one of the default PHY modes defined in the Application_Interface header file.

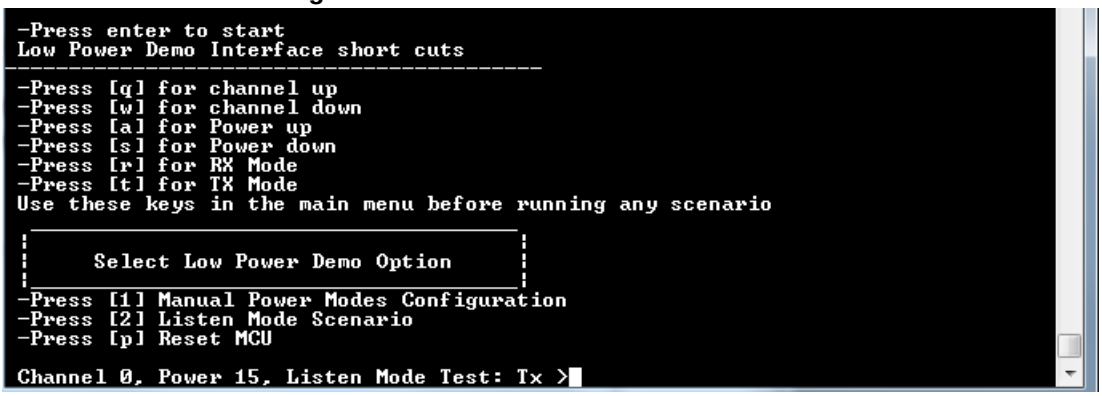
Other settings should be kept “as-is” since they could affect the application’s behavior.

The project itself can be configured to run on RTOS (MQX or FreeRTOS) configuration or bare-metal configuration (non pre-emptive task scheduler). The above options are available both as Debug and Release configurations.

5.3 Running the Low Power Demo Application

At startup, the application displays a menu for selecting one of the scenarios described above. Also, in the main menu, the shortcuts can be used to change the active channel, output power and the role for the second scenario (receiver or transmitter).

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 is listed. If one of the options is selected, another menu pops up with the available transceiver configurations. After selecting a transceiver mode, the last menu will print the available wake up sources in respect to the specified MCU mode. As soon as the user selects the wake up source, all configurations are applied.

All VLLSx (0, 1 and 3) modes exit by reset, while other modes will simply resume the application execution. A particular situation here is the very low power run which allows the MCU to continue its execution under special conditions. Although the MCU still runs, the communication with the TERM is lost. To exit this mode, the user should long press any switch button (except reset) .

Figure 5-2. Selecting an available MCU mode

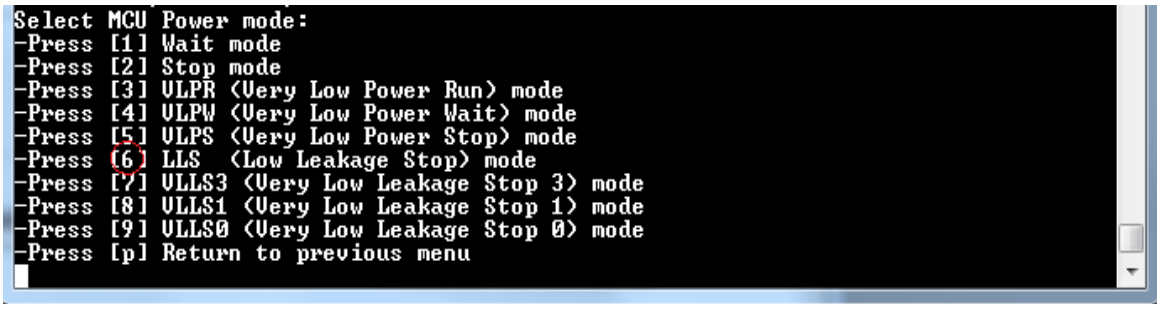


Figure 5-3. Selecting an available transceiver configuration

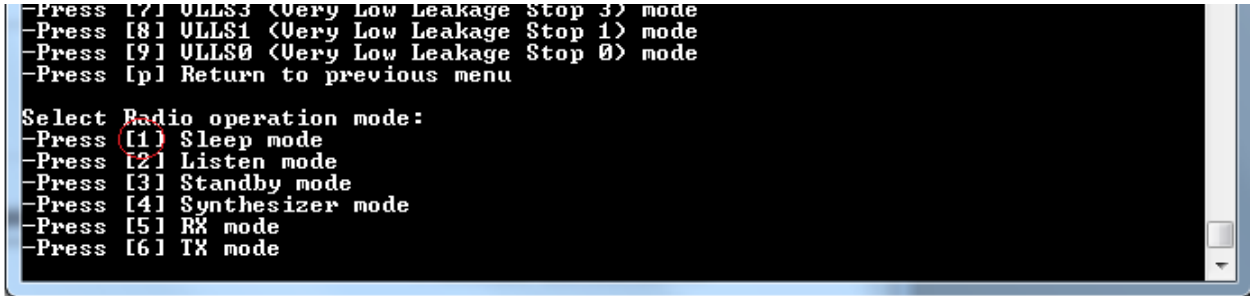
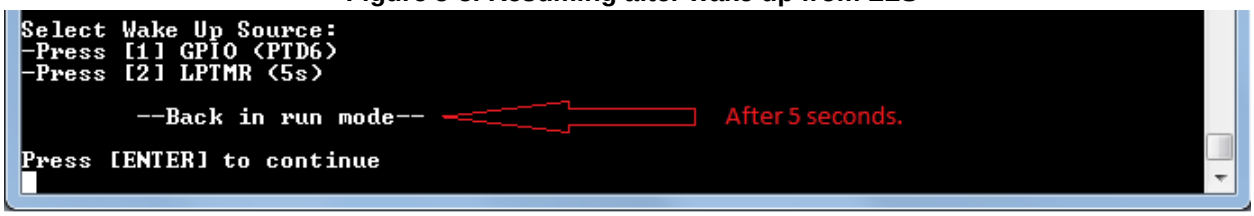


Figure 5-4. Selecting an available wake up source

Figure 5-5. Resuming after wake up from LLS



5.3.2 Listen Mode Scenario

Prior to entering this menu, the user should select the desired role. To enter listen scenario as the transmitter, user should press ‘t’ (the configuration menu will display “Tx”). To assume the role of the receiver, which means switch to internal clock and the transceiver into listen mode, while PHY and SMAC go into RX, the user should press ‘r’ and enter this menu.

The receiver exits 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 in listen mode. The predefined settings for this scenario are: the transmitter sends a packet every 5 ms, while the receiver’s transceiver has listen mode configured to sleep 12ms and go into receive mode 4 ms.

Figure 5-6. Listen Mode Scenario, Tx role

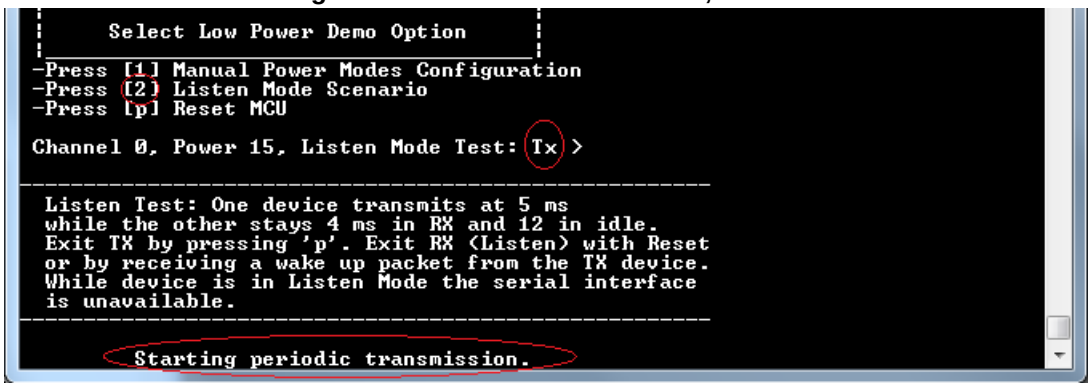


Figure 5-7. Listen Mode Scenario, Rx role

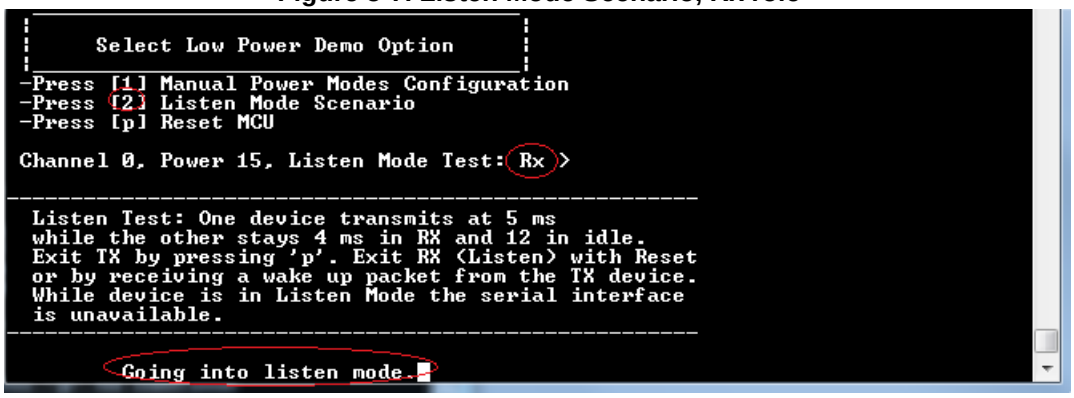


Figure 5-8. Listen Mode Scenario, wake up from listen.