
MKW2xD Simple Media Access Controller (SMAC) Demonstration Applications

User's Guide

Document Number: MKW2xDSMACDAUG

Rev. 0

05/2015

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

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

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor 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 Semiconductor 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 Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

IAR and IAR Workbench are registered trademarks of IAR (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

Freescale, Freescale logo, Freescale Tower System, and Freescale Freedom development platform are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2015. All rights reserved.

Chapter 1

Introduction

1.1. Supported hardware	5
1.2. MKW2xD software	5
1.3. Hardware considerations	5

Chapter 2

MKW2xD Wireless UART Application

2.1. Introduction	9
2.2. Application configuration	9
2.3. Running the Wireless UART Application	10

Chapter 3

MKW2xD Wireless Messenger Application

3.1. Introduction	11
3.2. Wireless Messenger configuration menu	11
3.3. Wireless Messenger console menu	14

Chapter 4

MKW2xD Connectivity Test Application

4.1. Introduction	15
4.2. MKW2xD Connectivity Test Application configuration	15
4.3. MKW2xD Connectivity Test Application usage	16

About this book

This guide provides a detailed description of the MKW2xD demonstration applications.

For more details about the MKW2xD device, see the appropriate Reference Manual and/or Data Sheet.

Audience

This document is intended for application developers using the MKW2xD 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.

Revision History

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

Revision History

Rev. number	Date	Substantial changes
0	05/2015	Initial release.

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
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
MKW2xD	FRDM-MKW2xD or TWR-MKW2xD or USB-MKW2xD
RTOS	Real-Time Operating System

References

The following sources were referenced to produce this book:

[1] *Freescale MKW2xD Reference Manual* (MKW2xDDxxxRM)

Chapter 1

Introduction

The Freescale MKW2xD 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 MKW2xD.

1.1 Supported hardware

The MKW2xD SMAC based Demo Applications are designed to work with the MKW2xD platforms. For a more detailed view about the boards see *FRDM-KW2x Hardware Reference Manual* (FRDMKW2XHWRM), *TWR-KW2x Hardware Reference Manual* (TWRKW2XHWRM), and *USB-KW2x Hardware Reference Manual* (USBKW2XHWRM).

1.2 MKW2xD 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 MKW2xD (all platforms except USB-KW24D512) can be connected to the computer via the mini USB connector using an emulated serial port driver. Also, each platform except TWR-KW21D256 is enabled by an USB stack capable of running a virtual communication port via the CDC class.



Figure 1-1. TWR-KW24D512

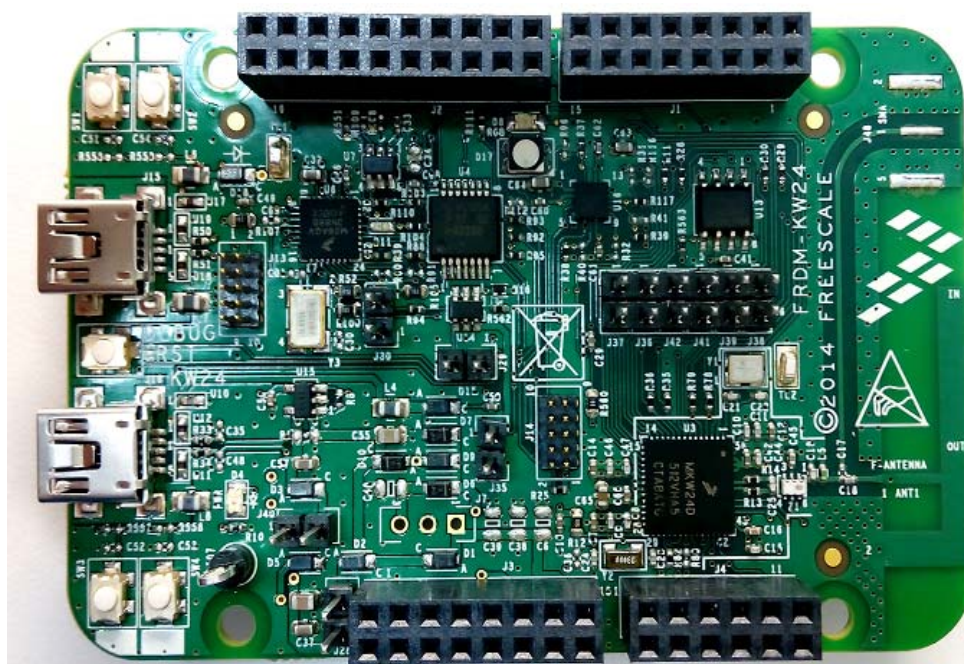


Figure 1-2. FRDM-KW24

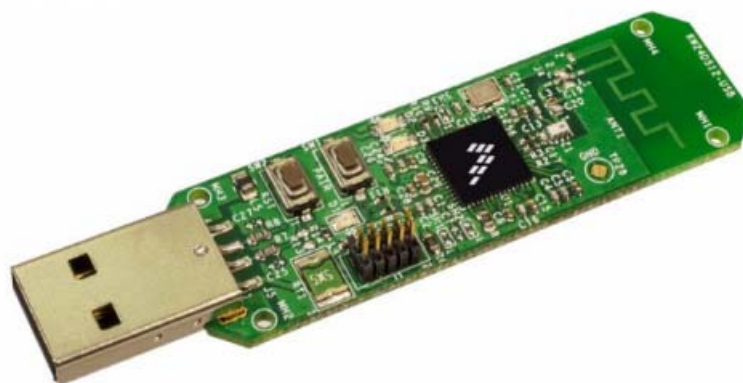


Figure 1-3. USB-KW24D512

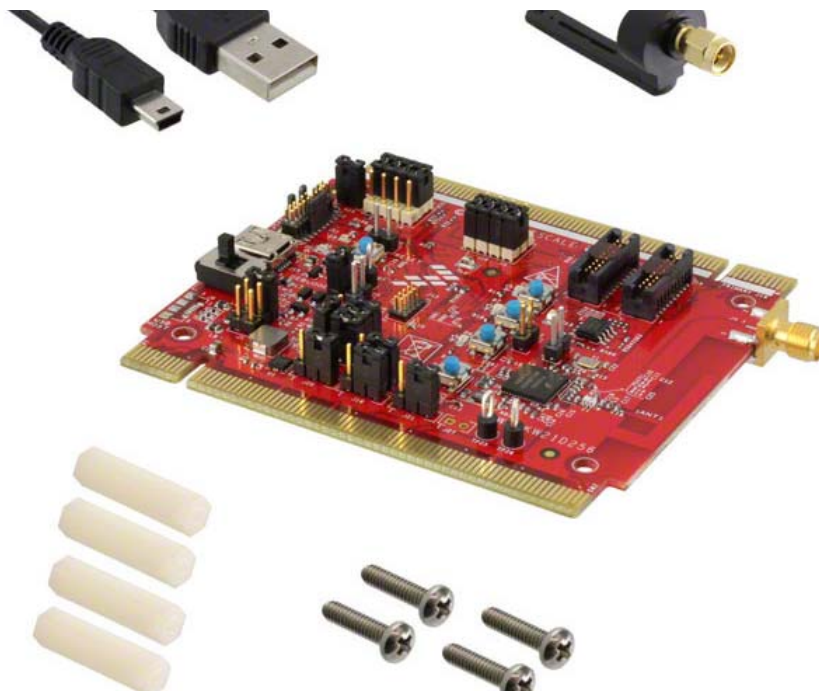


Figure 1-4. TWR-KW21D256

NOTE

From a software point of view, the user can switch between the serial interfaces (UART and USB) by modifying the corresponding macro definitions in the “*board.h*” file, which is board specific and can be found in the “*boards\<board_name>*” directory. By default, the associated macros for the serial communication interface are defined as:

```
#define APP_SERIAL_INTERFACE_TYPE gSerialMgrUart_c
#define APP_SERIAL_INTERFACE_INSTANCE 1
//#define APP_SERIAL_INTERFACE_TYPE gSerialMgrUSB_c
//#define APP_SERIAL_INTERFACE_INSTANCE 0
```

This means that the SMAC based applications will use UART. To switch from the UART interface to the USB interface (does not apply to the TWR-KW21D256 platform) the user should comment the first two lines and uncomment the last two.

```
//#define APP_SERIAL_INTERFACE_TYPE gSerialMgrUart_c
//#define APP_SERIAL_INTERFACE_INSTANCE 1
#define APP_SERIAL_INTERFACE_TYPE gSerialMgrUSB_c
#define APP_SERIAL_INTERFACE_INSTANCE 0
```


Chapter 2

MKW2xD Wireless UART Application

2.1 Introduction

The MKW2xD 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*).
- **Wireless_UartApplication.c**
The *InitProject* can be changed to enable the retry and re-transmission mechanisms. By setting the boolean fields 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.

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 (FreeRTOS or MQX) configuration or bare-metal configuration (non pre-emptive task scheduler). Before building and downloading the application the user can switch to either of them from the Workspace tab in IAR.

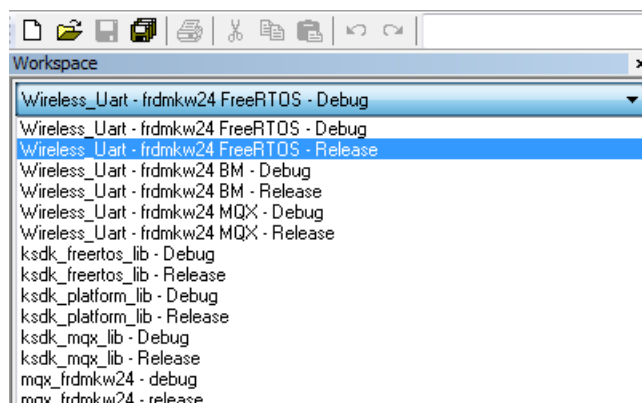


Figure 2-1. Change between targets in IAR

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) MKW2xD 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 reset and one of the (user) push buttons found on the MKW2xD 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

MKW2xD Wireless Messenger Application

3.1 Introduction

The MKW2xD 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 RTOS targets 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.

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.

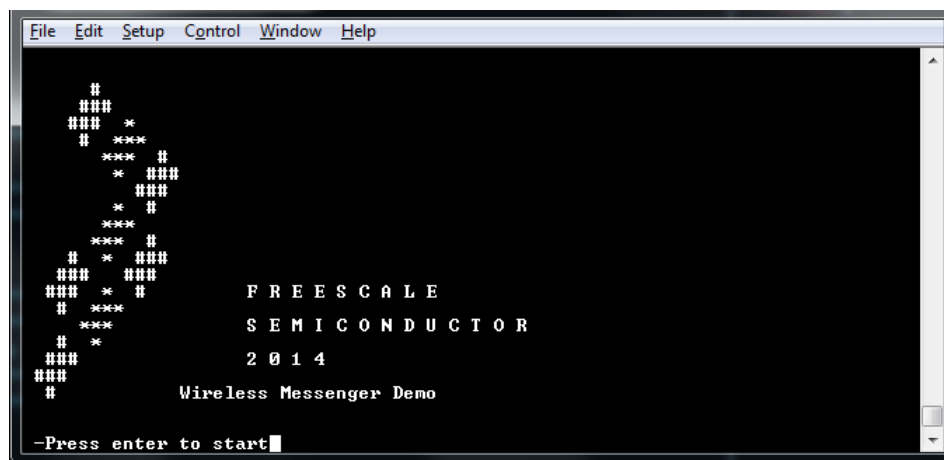


Figure 3-1. Wireless Messenger welcome screen

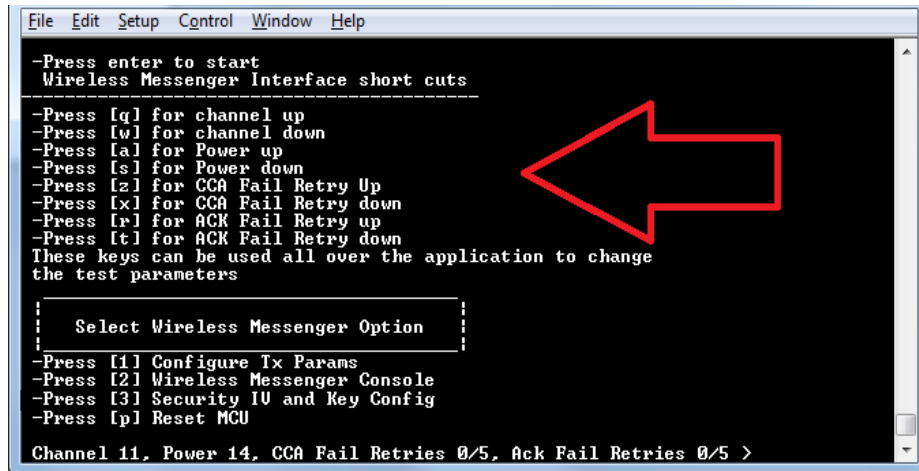


Figure 3-2. Shortcuts menu for the MKW2xD Wireless Messenger demo application

Given the above figure:

- pressing 'q' will increase the channel number. When it reaches 26 the channel number will be set to 11.
- pressing 'w' will decrease the channel number. If previous channel number is 11, the next channel will be 26
- pressing 'a' and 's' is similar to 'q' and 'w' but for setting the output power. The limits for this option are [0x03, 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](#) illustrates it. This indicates that the shortcuts are still active.

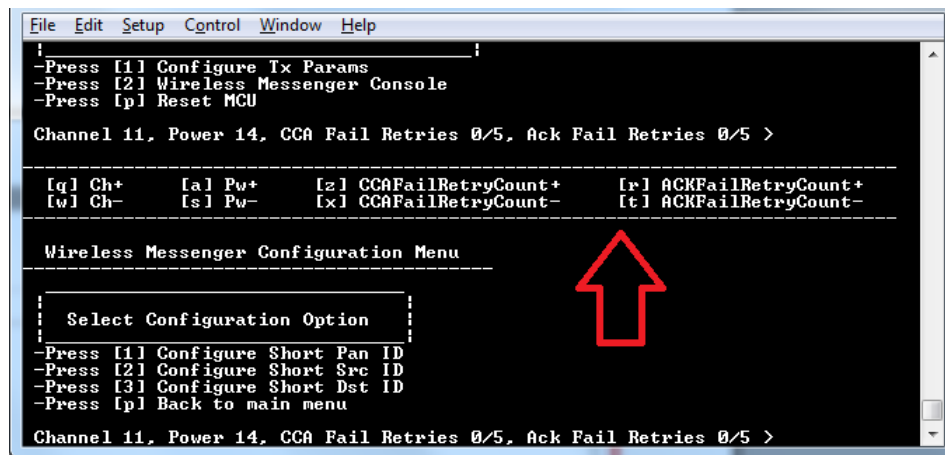


Figure 3-3. Shortcuts menu description embedded in the configuration menu

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 configuration

- 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.

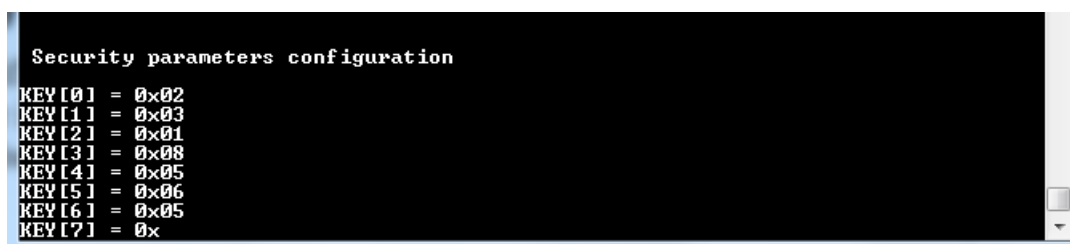


Figure 3-4. Security configuration menu (when security is enabled)

3.3 Wireless Messenger console menu

The console menu is used for sending and receiving messages. Outside this menu, the MKW2xD 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 MKW2xD 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.

To return to the main menu the user must press the escape key ([Esc]).



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

Chapter 4

MKW2xD Connectivity Test Application

4.1 Introduction

The MKW2xD 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 MKW2xD Connectivity Test Application configuration

The compile time settings of the application can be updated in the same manner as for the MKW2xD Wireless UART Application, but most of the configurations can be found in the `Connectivity_Test_Platform.h` header file. 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.

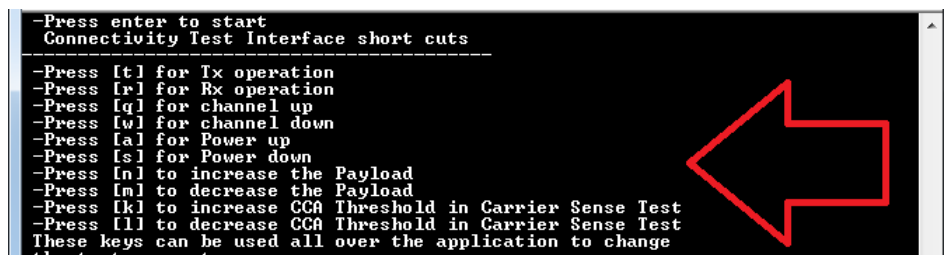


Figure 4-1. Connectivity Test Shortcuts Menu

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 26, the channel number will change to 11.
- ‘w’ : decrements channel number. If pressed when current channel is 11, the channel number will change to 26.
- ‘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 0x03).
- ‘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, 116] 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.

4.3 MKW2xD Connectivity Test Application usage

The MKW2xD 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 continuously 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).
 - *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.

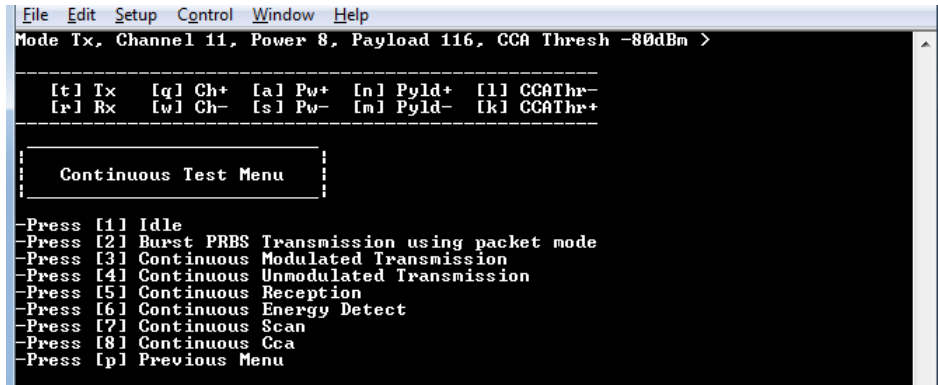


Figure 4-2. Continuous Test menu entries

- 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 MKW2xD platforms have Connectivity Test loaded, one of the boards can be set in RX and the other in TX as in the following figures.

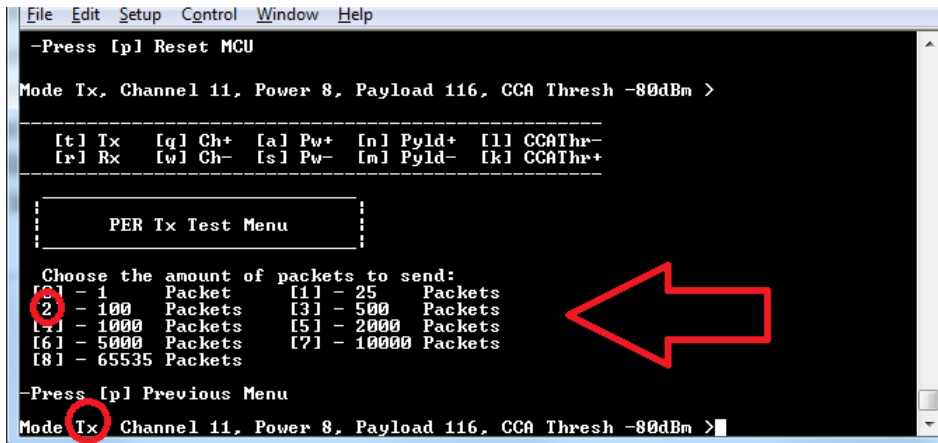


Figure 4-3. Entering PER menu with TX option selected, and pressing '2' to send 100 packets

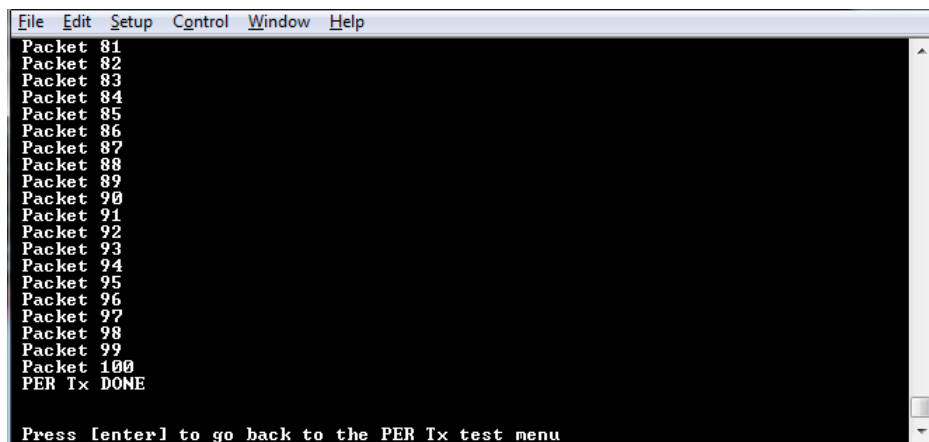


Figure 4-4. PER TX menu after sending the 100 packets burst

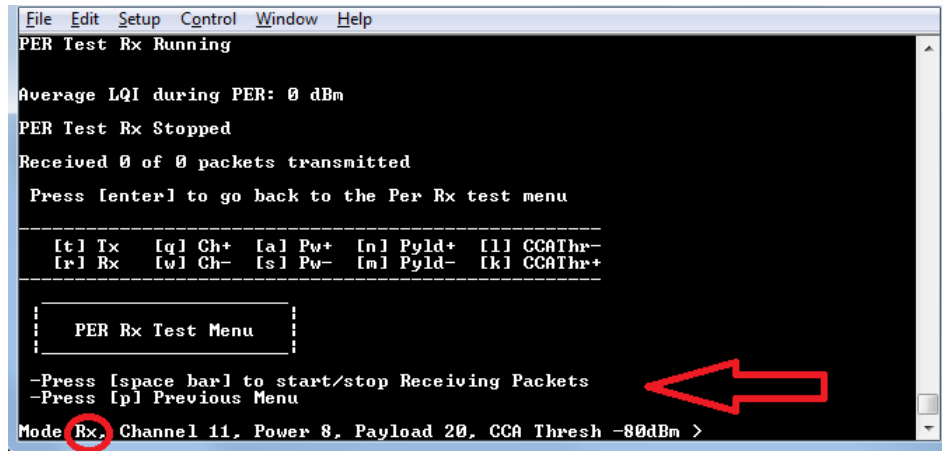


Figure 4-5. PER Test menu with RX option selected on the second platform

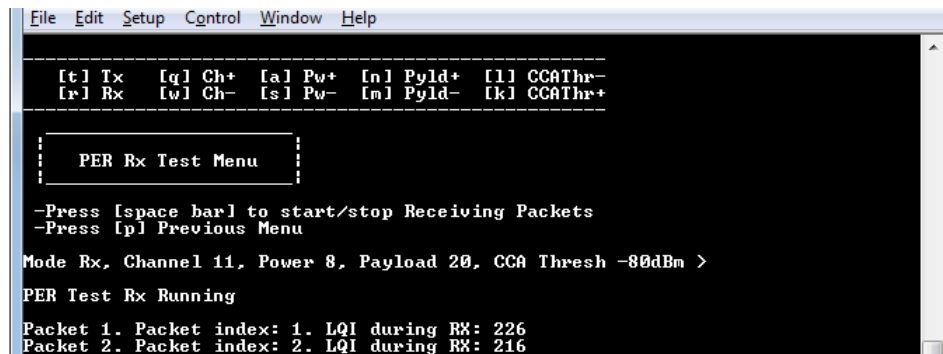


Figure 4-6. PER RX after pressing [SPACE] and starting the test

- 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 MKW2xD 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 link 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.

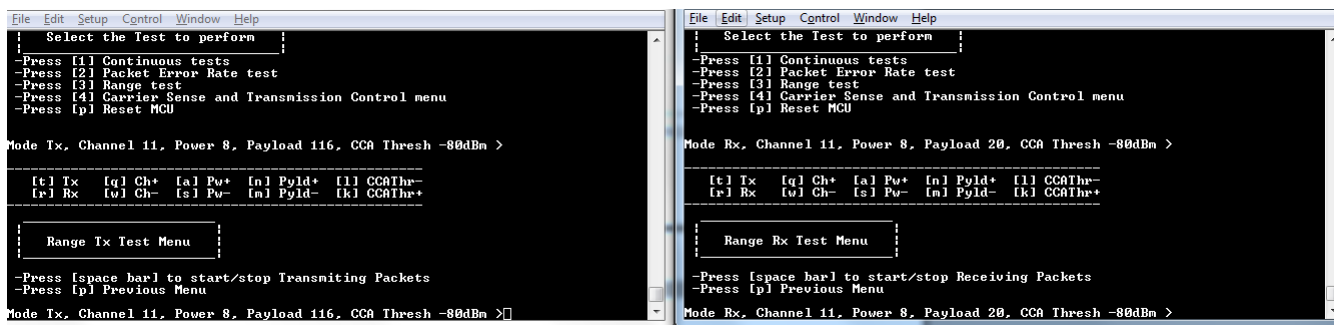


Figure 4-7. Range TX and RX submenus for two MKW2xD platforms

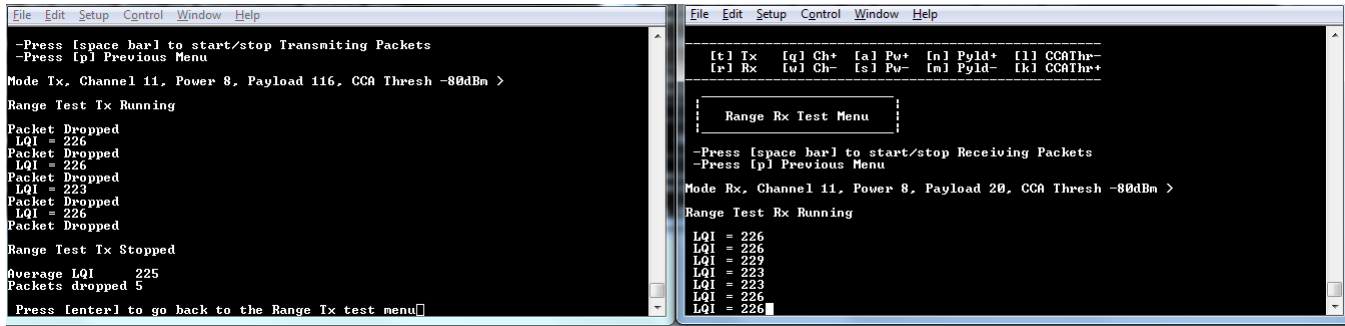


Figure 4-8. Range TX and RX after running the test for two MKW2xD platforms

- 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 responsibility 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.

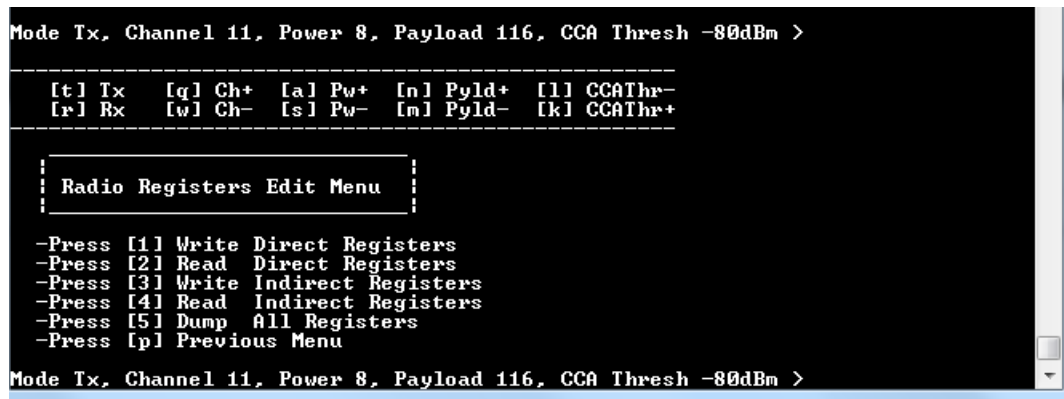


Figure 4-9. Radio Registers Edit menu

```

-----
Radio Registers Edit Menu
-----
-Press [1] Write Direct Registers
-Press [2] Read Direct Registers
-Press [3] Write Indirect Registers
-Press [4] Read Indirect Registers
-Press [5] Dump All Registers
-Press [6] Previous Menu

Mode Tx, Channel 11, Power 15, Payload 20, CCA Thresh -80dBm >
Read Registers

-Press [6] Previous Menu

-write the Register address in Hex and [Enter]: 0xaa ← Pressing 3 times 'a'
-Value out of Range!!
-write the Register address in Hex and [Enter]: 0x

```

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 continuously 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 milliseconds. 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.

```

File Edit Setup Control Window Help

-Press [4] Carrier Sense and Transmission Control menu
-Press [6] Reset MCU

Mode Tx, Channel 11, Power 8, Payload 116, CCA Thresh -80dBm >

-----
Radio Carrier Sense and Transmission Control Select Menu
-----

-Press [1] Carrier Sense Test with un-modulation input signal
-Press [2] Transmission Control Test
-Press [6] Previous Menu

Mode Tx, Channel 11, Power 8, Payload 116, CCA Thresh -80dBm >

Press [SPACE] to begin/interrupt test
Press [6] to return to previous menu

```

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

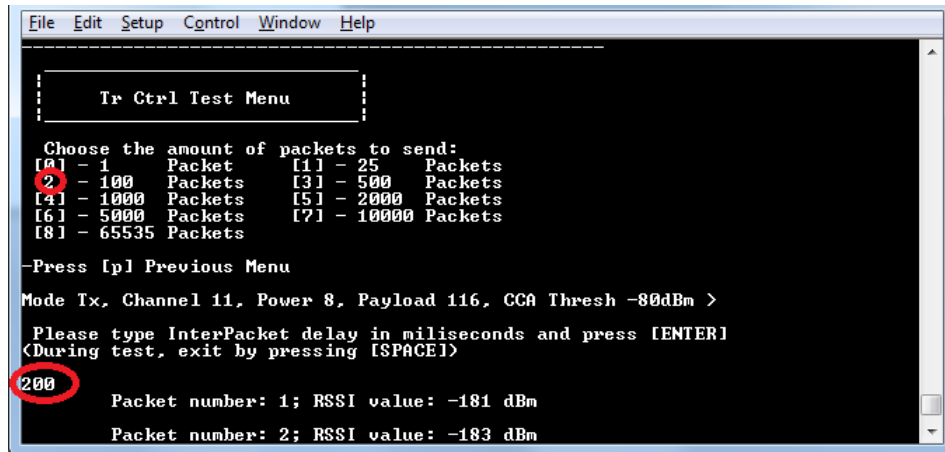


Figure 4-12. Transmission Control submenu after selecting 100 packets and 200 ms delay