
MCR20A Simple Media Access Controller (SMAC) Demonstration Applications

User's Guide

Document Number: MCR20ASMACDAUG
Rev. 0
07/2015



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.

Freescale and the Freescale Logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm Off. Tower is a trademark of Freescale Semiconductor, Inc. ARM, IAR, and IAR Workbench are registered trademarks of IAR (or its subsidiaries) in the EU and/or elsewhere. All other product or service names are the property of their respective owners. All rights reserved.

© 2015 Freescale Semiconductor, Inc.

Chapter 1

Introduction

1.1. Supported hardware	5
1.2. MCR20A software	5
1.3. Hardware considerations	5

Chapter 2

MCR20A Wireless UART application

2.1. Introduction	7
2.2. Application configuration	7
2.3. Running the Wireless UART application	8

Chapter 3

MCR20A Connectivity Test application

3.1. Introduction	9
3.2. MCR20A Connectivity Test application configuration	9
3.3. MCR20A Connectivity Test application usage	10

Chapter 4

MCR20A wireless messenger application

4.1. Introduction	17
4.2. Wireless messenger configuration menu	17
4.3. Wireless messenger console menu	20

Chapter 5

MCR20A Low Power Demo application

5.1. Introduction	21
5.2. Application configuration	21
5.3. Running the Low Power Demo application	22

About this book

This guide provides a detailed description of the MCR20A demonstration applications.

For more details about the MCR20A device, see the appropriate Reference Manual and/or Data Sheet.

Audience

This document is intended for application developers using the MCR20A device and these demonstration applications as a 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, “MCR20A Wireless UART application”](#) — Details how to run the Wireless UART application and explains available application configurations.
- [Chapter 4, “MCR20A Wireless Messenger application”](#) — Details how to run the Wireless Messenger application and explains available application configurations.
- [Chapter 3, “MCR20A Connectivity Test application”](#) — Details how to run the Connectivity Test application and explains available menu options and application configurations.
- [Chapter 5, “MCR20A Low Power Demo application”](#) — Details how to run the MCR20A Lower Power Demo application and explains how to handle low power use cases.

Revision history

The table summarizes revisions made to this document.

Revision history

Rev. number	Date	Substantive changes
0	07/2015	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	Linq Quality Indicator
FRDM	Freescale Freedom Development 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
RTOS	Real Time Operating System

References

The following sources were referenced to produce this book:

[1] *Freescale MCR20A Reference Manual* (document MCR20RM)

Chapter 1

Introduction

The Freescale MCR20A SMAC based Demonstration Applications have to be used with a serial port terminal application. This chapter details hardware and software requirements to begin using the MCR20A.

1.1 Supported hardware

The MCR20A SMAC based Demo Applications are designed to work with the FRDM-CR20A Freescale Freedom Development platform. The current release supports the FRDM-K64F and FRDM-KL46Z Freescale Freedom Development platforms as masters for the FRDM-CR20A Freescale Freedom Development platform.

1.2 MCR20A software

The package contains the RTOS and Kinetis Software Development Kit (KSDK) based framework, drivers, and connectivity stack with SMAC as a second layer along with the four demo applications. The application projects are developed using IAR Embedded[®] Workbench[®].

1.3 Hardware considerations

The MCR20A can be connected to any platform through the Arduino[®] headers. For example, the SMAC demonstration application suite is provided for the FRDM-CR20A Freescale Freedom Development platform plugged into (as shield) the FRDM-K64F or FRDM-KL46Z Freescale Freedom Development platforms. Both platforms can be connected to the PC, using mini (or micro) USB cables from two interfaces. One interface is through UART as a serial interface with OpenSDA as a UART-to-USB converter, while the other is through the native USB by enabling the USB virtual communication port capabilities from software. The OpenSDA USB and platform USB connectors are different, so the user should become familiar with these platforms before using the SMAC Demonstration Applications.



Figure 1-1. FRDM-CR20A Freescale Freedom Development platform

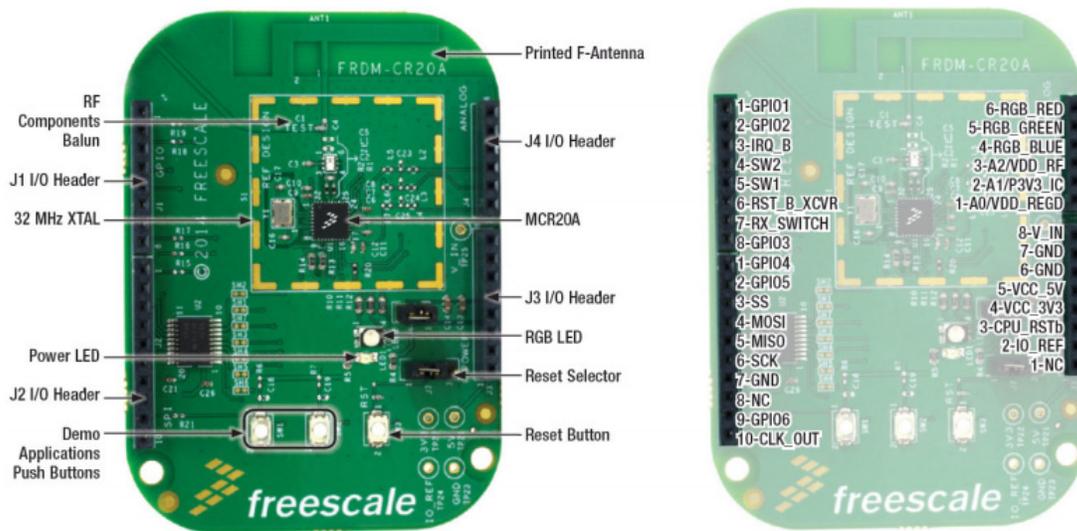


Figure 1-2. FRDM-CR20A callouts and I/O headers

NOTE

From a software point of view, the user can switch between the serial interfaces (UART and USB) by modifying or redefining the corresponding macro definitions in the `board.h` file, which are 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:

```
#ifndef APP_SERIAL_INTERFACE_TYPE
#define APP_SERIAL_INTERFACE_TYPE (gSerialMgrUart_c)
#endif

#ifndef APP_SERIAL_INTERFACE_INSTANCE
#define APP_SERIAL_INTERFACE_INSTANCE (0)
#endif
```

This means that the SMAC based applications use UART. To switch from the UART interface to the USB interface, the user should redefine or change the macros, as shown below:

```
#ifndef APP_SERIAL_INTERFACE_TYPE
#define APP_SERIAL_INTERFACE_TYPE (gSerialMgrUSB_c)
#endif

#ifndef APP_SERIAL_INTERFACE_INSTANCE
#define APP_SERIAL_INTERFACE_INSTANCE (0)
#endif
```

Chapter 2

MCR20A Wireless UART application

2.1 Introduction

The MCR20A 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 software that is capable of opening a serial port and writing to or reading from it. The characters sent or received are not necessarily ASCII printable characters.

All 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`), and 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, the re-transmissions the user can enable are none, one, or both of these features.

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 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 buffers available for memory allocation, but it is recommended to keep their current values.

The application is divided into three projects. Each project is RTOS dependent. The available RTOSes are FreeRTOS, MQX™ RTOS, and bare-metal scheduler (non pre-emptive task scheduler). All projects have

both Debug and Release configurations. Before building and downloading the application, the user can switch to either of them from the Workspace tab in IAR.

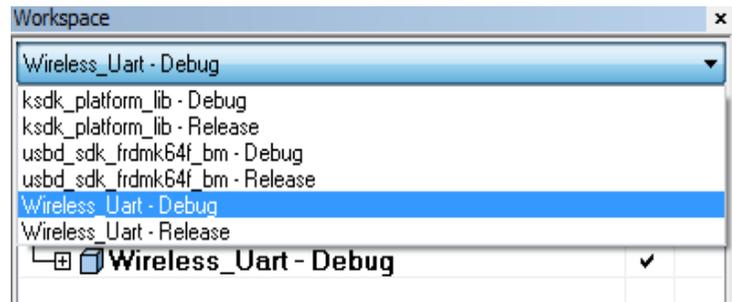


Figure 2-1. Change between targets in IAR

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 no changes have been done to the project.

- Two (or more) MCR20A platforms (plugged into the FRDM-K64F or FRDM-KL46Z Freescale Freedom Development platforms) have to be connected to the PC using the mini/micro-USB cables.
- The code must be downloaded on the platforms via CMSIS-DAP (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 baud rate is 115200). Other necessary serial configurations are 8 bit, no parity, and 1 stop bit.
- To start the setup, each platform must be reset, and one of the (user) push buttons found on the MCR20A platform must be pressed. The user can press any of the non-reset buttons on the FRDM-K64F or FRDM-KL46Z Freescale Freedom Development platforms as well. This initiates the state machine of the application so user can start sending characters via UART (or USB) to one of the platforms, which are broadcast OTA.

Chapter 3

MCR20A Connectivity Test application

3.1 Introduction

The MCR20A Connectivity Test application is an 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.

3.2 Connectivity Test application configuration

The compile time settings of the application can be updated in the same manner as for the MCR20A Wireless UART application, but most of the configurations can be found in the `Connectivity_Test_Platform.h` header file. See [Section 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.

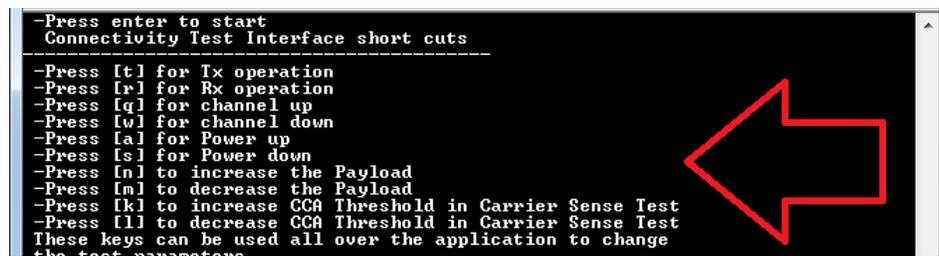


Figure 3-1. Connectivity Test shortcuts menu

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 the current channel is 26, the channel number changes to 11.
- ‘w’ : Decrements channel number. If pressed when the 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 goes to the minimum (in this case 0x03).
- ‘s’ : Decrements output power value. If output power is at minimum and this key is pressed, the output power goes to the 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 are 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.

3.3 Connectivity Test application usage

The MCR20A Connectivity Test application has five main features:

- a) *Continuous Tests*: This menu option displays several 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 which contain a pseudo-random payload of fixed length.
 - *Continuous Modulated Transmission*: This option allows 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 allows 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 places the transceiver in reception 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 at fixed hard-coded intervals for the current channel, and prints their values to the TERM.
 - *Continuous Scan*: This option is similar to the previous one, except that at each iteration it obtains the energy values on all channels.
 - *Continuous CCA*: This option launches consecutive CCA requests for the currently selected channel at a fixed, hard-coded interval, and prints “Idle” or “Busy” depending on the CCA result.

```

File Edit Setup Control Window Help
Mode Tx, Channel 11, Power 8, Payload 116, CCA Thresh -80dBm >

-----
[t] Tx   [q] Ch+ [a] Pw+ [n] Pyl+ [l] CCAThr-
[r] Rx   [w] Ch- [s] Pw- [m] Pyl- [k] CCAThr+
-----

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 [p] Previous Menu

```

Figure 3-2. Continuous Test menu entries

- 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 key. If 'r' is pressed, the following menu is for *PER RX*, otherwise it is for *PER TX*. For example, if two MCR20A platforms have Connectivity Test loaded, one of the boards can be set in RX and the other in TX as in the following figures.

```

File Edit Setup Control Window Help
-Press [!] Reset MCU
Mode Tx, Channel 11, Power 8, Payload 116, CCA Thresh -80dBm >

-----
[t] Tx   [q] Ch+ [a] Pw+ [n] Pyl+ [l] CCAThr-
[r] Rx   [w] Ch- [s] Pw- [m] Pyl- [k] CCAThr+
-----

PER Tx Test Menu

Choose the amount of packets to send:
[1] - 1 Packet           [11] - 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 11, Power 8, Payload 116, CCA Thresh -80dBm >

```

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

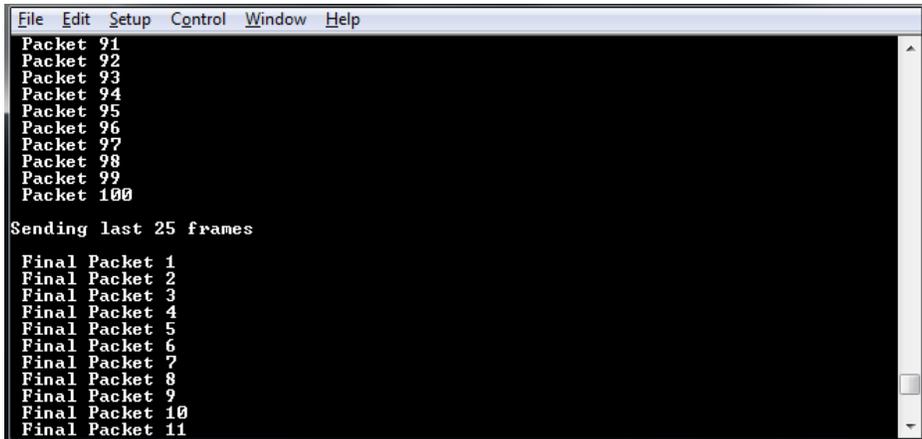


Figure 3-4. PER TX menu after sending the 100 packets burst

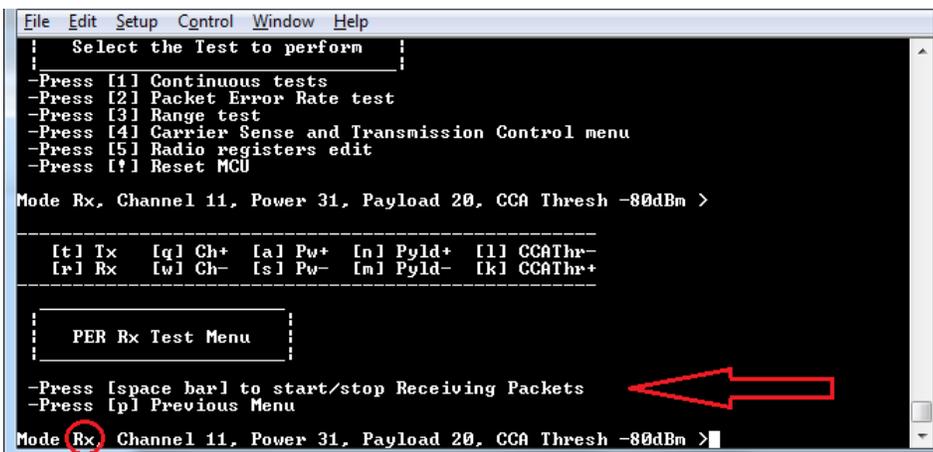


Figure 3-5. PER Test Menu with RX option selected on the second platform

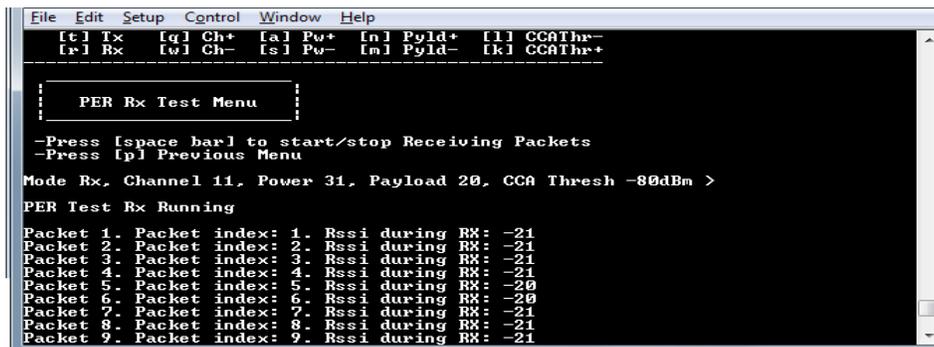


Figure 3-6. PER RX after pressing [SPACE] and starting the test

- c) *Range Test*: This test displays a configuration menu that performs a ‘ping-pong’ test to aid the user in determining the range (as distance between two platforms) in which the MCR20A 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 signal strength for each received packet. 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.

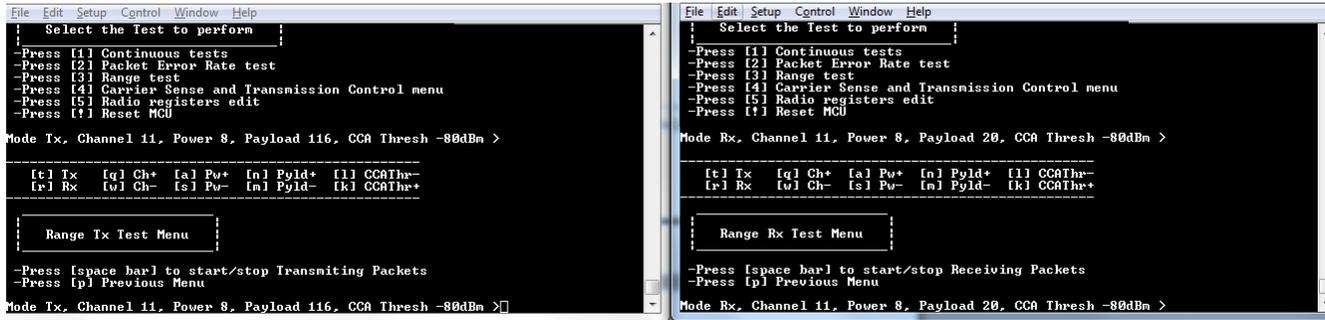


Figure 3-7. Range TX and RX sub-menus for two MCR20A platforms

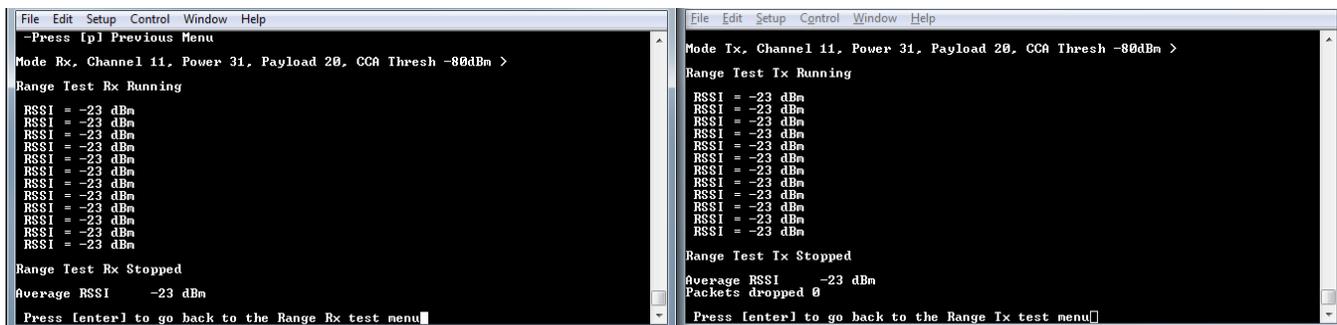


Figure 3-8. Range TX and RX after running the test for two MCR20A platforms

- d) *Radio Registers Edit*: This menu allows 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 only validates that the address is in the unsigned char range, but the user has the possibility to request register 0xFF. To ensure that a proper range is used, the user should first use the dump register feature to see the valid address ranges.

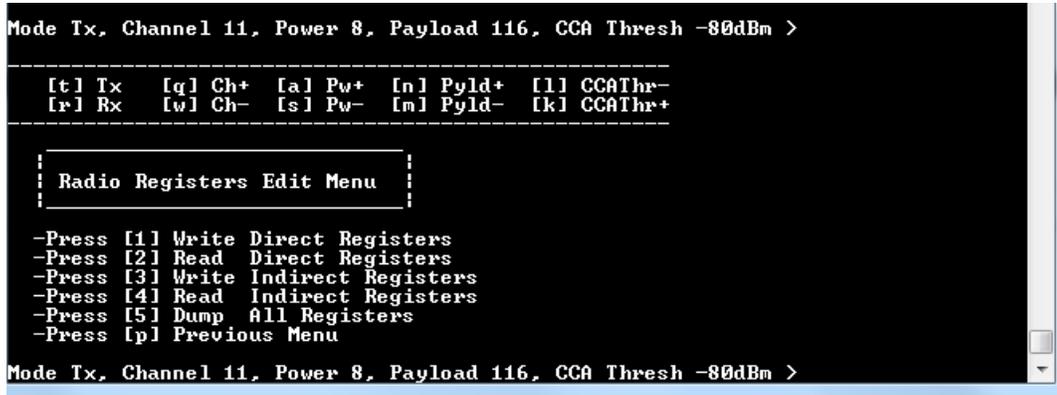


Figure 3-9. Radio registers edit menu

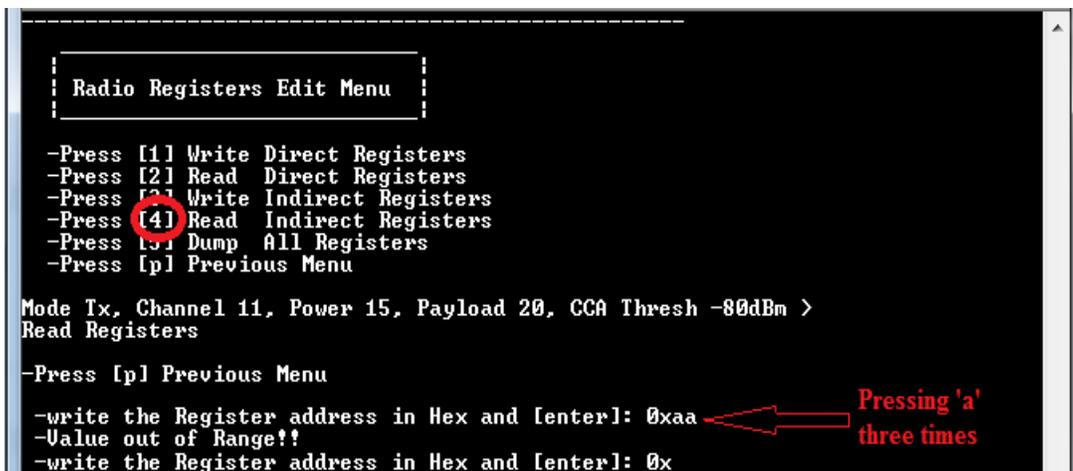


Figure 3-10. Read indirect registers example

- e) *Carrier Sense and Transmission Control*: This menu allows the user to choose between two tests. The former 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 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, then prompts the user to enter a decimal value resembling the inter-packet delay in milliseconds. 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 with ‘n’ and ‘m’ shortcuts.

```

File Edit Setup Control Window Help
-Press [4] Carrier Sense and Transmission Control menu
-Press [5] Radio registers edit
-Press [!] Reset MCU

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

-----
[!] Tx [q] Ch+ [a] Pw+ [n] Pyl+ [l] CCATh-
[r] Rx [w] Ch- [s] Pw- [m] Pyl- [k] CCATh+
-----

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

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

```

Figure 3-11. Carrier Sense sub-menu

```

File Edit Setup Control Window Help

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

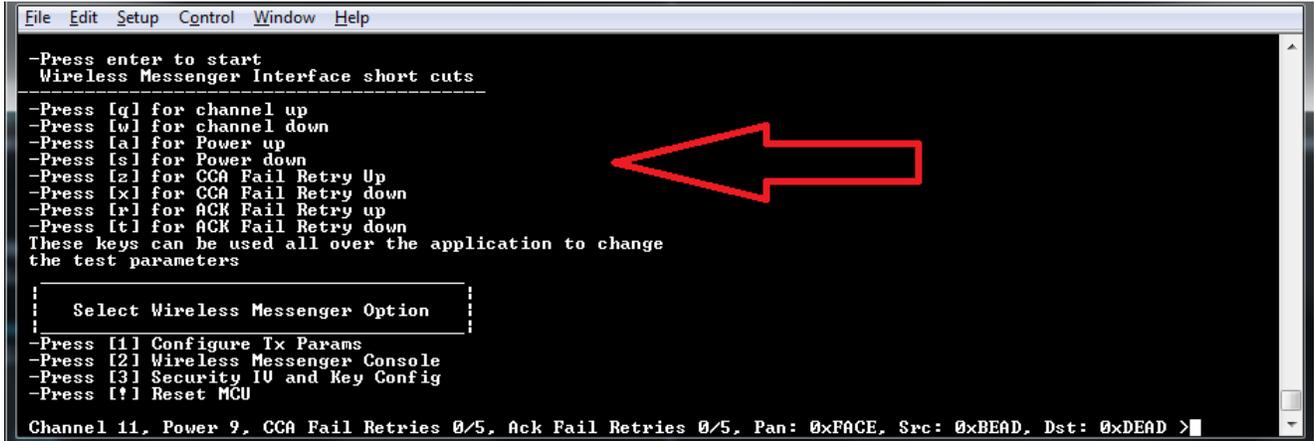
Please type InterPacket delay in milliseconds and press [ENTER]
(During test, exit by pressing [SPACE])

200

Packet number: 1; RSSI value: -81 dBm
Packet number: 2; RSSI value: -83 dBm

```

Figure 3-12. Transmission Control sub-menu after selecting 100 packets and 200 ms delay



```

File Edit Setup Control Window Help
-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 IU and Key Config
-Press [!] Reset MCU

Channel 11, Power 9, CCA Fail Retries 0/5, Ack Fail Retries 0/5, Pan: 0xFACE, Src: 0xBEAD, Dst: 0xDEAD >

```

Figure 4-2. Shortcuts menu for the MCR20A wireless messenger demo application

Given the above figure:

- Pressing ‘q’ increases the channel number. When it reaches 26, the channel number will be set to 11.
- Pressing ‘w’ decreases the channel number. If a 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’ increases the number of retransmissions in case of a channel busy scenario. Keep in mind that a number greater than 0 for this option enables automatic CCA before TX option.
- Pressing ‘x’ decreases the number of retransmissions in case of a channel busy status. If the number of retransmissions is decremented to 0, the feature is disabled.
- Pressing ‘r’ increments 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’ decrements 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 illustrated in [Figure 4-3](#). This indicates that the shortcuts are still active.

```

File Edit Setup Control Window Help
-Press [1] Configure Tx Params
-Press [2] Wireless Messenger Console
-Press [3] Security IV and Key Config
-Press [!] Reset MCU

Channel 11, Power 9, 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 12, Power 9, CCA Fail Retries 0/5, Ack Fail Retries 0/5, Pan: 0xFACE, Src: 0xBEAD, Dst: 0xDEAD >

```

Figure 4-3. Shortcuts menu description embedded in the configuration menu

4.2.2 Configuration menu

The configuration menu is presented in Figure 4-3. There are three options available:

- *Configure Short Pan ID*: This option allows 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 allows 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 allows the user 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.

4.2.3 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 devices, and the initial vector and 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 4-4. Security configuration menu (security enabled)

4.3 Wireless Messenger console menu

The console menu is used for sending and receiving messages. Outside this menu, the MCR20A does not receive any packets OTA. In this menu, all configuration options are disabled.

In the console menu, the user can see packets intended for the MCR20A platform or broadcast packets. When the user types a message and hits [ENTER], the application includes the message into a packet payload and sends it using the previously configured addressing information. The application notifies 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 the case the retry or retransmission mechanism is enabled, and sending failed after the retry/retransmission counter exceeded 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 receives a “Packet Sending Failed. Reason. Channel Busy!” notification.

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

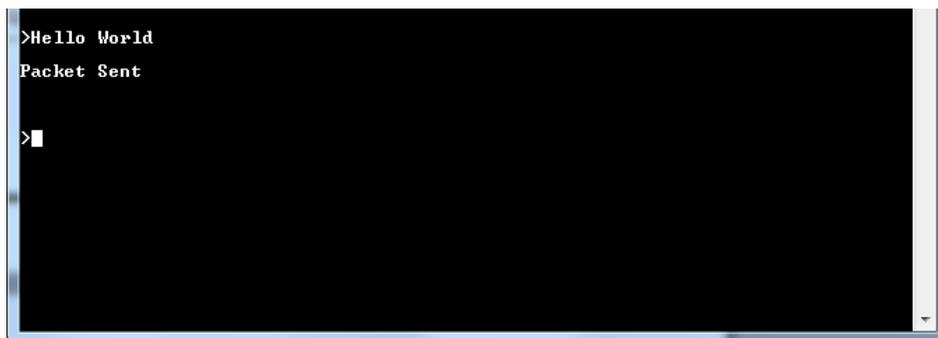
A screenshot of a terminal window with a black background and white text. The text displayed is: >Hello World, Packet Sent, and > followed by a cursor. The terminal window has a light blue border and a scrollbar on the right side.

Figure 4-5. Console menu: “Packet Sent” notification after writing “Hello World” and hitting [ENTER]

Chapter 5

MCR20A Low Power Demo application

5.1 Introduction

The MCR20A 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 low power mode, what transceiver configuration is active during the low power mode and the wake up source. The second scenario is divided in two roles: a TX device, which sleeps in Very Low Power Stop mode for two seconds then wakes up and sends a “wake up” message, and an RX device configured with the MCU in Very Low Power Stop, and the transceiver in receive mode with Low Power Preamble Search feature enabled. The TX device ends the test after sending 5 consecutive packets, while the RX device can be interrupted on any key or upon receiving the “wake up message”. The application is used with a TERM. The application configurations can be applied both at compile time and at runtime (using the shortcuts keys).

NOTE

The user should never use USB as the default interface type for the serial manager. The application functions properly only if UART/LPSCI is used as a transport layer to the PC.

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 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 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. 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 application project has three configurations depending on the scheduler used: RTOS (FreeRTOS or MQX), 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 (TX or RX) for the Low Power Preamble Search scenario.

```

File Edit Setup Control Window Help
-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] Low Power Preamble Search Scenario
-Press [!] Reset MCU

Channel 11, Power 7, LPPS Test: Tx >

```

Figure 5-1. Low Power Demo main menu

5.3.1 Manual power modes configuration

When entering this menu, a sub-menu with available power modes is listed. After one of the options is selected, another menu pops up with available transceiver configurations, followed by a menu with available wake up sources in respect to the selected low power mode. As soon as the user selects a wake up source, all configurations are applied.

All VLLSx (0, 1, 2, and 3) modes exit by reset, while other modes 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).

```

File Edit Setup Control Window Help
-----
-Press [1] Manual Power Modes Configuration
-Press [2] Low Power Preamble Search Scenario
-Press [!] Reset MCU

Channel 11, Power 7, LPPS Test: Tx >
Select MCU Power mode:
-Press [0] Wait mode
-Press [1] Stop mode
-Press [2] ULPR <Very Low Power Run> mode
-Press [3] ULPW <Very Low Power Wait> mode
-Press [4] ULPS <Very Low Power Stop> mode
-Press [5] LLS <Low Leakage Stop> mode
-Press [6] ULLS3 <Very Low Leakage Stop 3> mode
-Press [7] ULLS2 <Very Low Leakage Stop 2> mode
-Press [8] ULLS1 <Very Low Leakage Stop 1> mode
-Press [9] ULLS0 <Very Low Leakage Stop 0> mode
-Press [p] Return to previous menu

```

Figure 5-2. Selecting an available MCU mode

```

File Edit Setup Control Window Help
-Press [1] Stop mode
-Press [2] ULPR (Very Low Power Run) mode
-Press [3] ULPW (Very Low Power Wait) mode
-Press [4] ULPS (Very Low Power Stop) mode
-Press [5] LLS (Low Leakage Stop) mode
-Press [6] ULLS3 (Very Low Leakage Stop 3) mode
-Press [7] ULLS2 (Very Low Leakage Stop 2) mode
-Press [8] ULLS1 (Very Low Leakage Stop 1) mode
-Press [9] ULLS0 (Very Low Leakage Stop 0) mode
-Press [p] Return to previous menu

Select Radio operation mode:
-Press [1] Idle mode
-Press [2] Hibernate mode
-Press [3] Doze mode
-Press [4] Auto Doze mode
-Press [5] RX mode
-Press [6] TX mode

```

Figure 5-3. Selecting an available transceiver configuration

```

File Edit Setup Control Window Help
-Press [0] Wait mode
-Press [1] Stop mode
-Press [2] ULPR (Very Low Power Run) mode
-Press [3] ULPW (Very Low Power Wait) mode
-Press [4] ULPS (Very Low Power Stop) mode
-Press [5] LLS (Low Leakage Stop) mode
-Press [6] ULLS3 (Very Low Leakage Stop 3) mode
-Press [7] ULLS2 (Very Low Leakage Stop 2) mode
-Press [8] ULLS1 (Very Low Leakage Stop 1) mode
-Press [9] ULLS0 (Very Low Leakage Stop 0) mode
-Press [p] Return to previous menu

Select Radio operation mode:
-Press [1] Idle mode
-Press [2] Hibernate mode
-Press [3] Doze mode
-Press [4] Auto Doze mode
-Press [5] RX mode
-Press [6] TX mode

Select Wake Up Source:
-Press [1] GPIO (SW2 - K64F)
-Press [2] LPTMR (5s)

```

Figure 5-4. Selecting an available wake up source

```

File Edit Setup Control Window Help
-Press [4] ULPS (Very Low Power Stop) mode
-Press [5] LLS (Low Leakage Stop) mode
-Press [6] ULLS3 (Very Low Leakage Stop 3) mode
-Press [7] ULLS2 (Very Low Leakage Stop 2) mode
-Press [8] ULLS1 (Very Low Leakage Stop 1) mode
-Press [9] ULLS0 (Very Low Leakage Stop 0) mode
-Press [p] Return to previous menu

Select Radio operation mode:
-Press [1] Idle mode
-Press [2] Hibernate mode
-Press [3] Doze mode
-Press [4] Auto Doze mode
-Press [5] RX mode
-Press [6] TX mode

Select Wake Up Source:
-Press [1] GPIO (SW2 - K64F)
-Press [2] LPTMR (5s)

--Back in run mode--
Press [ENTER] to continue

```

Figure 5-5. Resuming after wake up from LLS

5.3.2 Low Power Preamble Search (LPPS) Scenario

Prior to entering this menu option the user has to select the role of the platform. To do this, in the main menu the ‘r’ and ‘t’ keys are used to switch between TX device and RX device. If TX is active and the LPPS Scenario is selected, the platform behaves as a TX device with the functionality described in the introduction. If the RX option is active, the platform behaves as an RX device. As previously mentioned, this means that the device enters Very Low Power Stop with the transceiver in reception, and LPPS feature activated, waiting for a “wake up message” or a key stroke.

```
File Edit Setup Control Window Help
-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] Low Power Preamble Search Scenario
-Press [!] Reset MCU

Channel 11, Power 17, LPPS Test: Tx >

-----
*LPPS: TX device goes in ULPS for 2s and wakes up
to send wake up packet. RX device goes in ULPS
and xcvr in reception with lpps on.
* TX exits after 5 transmissions.
* Exit RX (LPPS) by pressing any key
or by receiving a wake up packet from the TX device.
-----

-Press [ENTER] to start
-Press [p] to return to previous menu
```

Figure 5-6. LPPS menu

```
File Edit Setup Control Window Help

-----
Select Low Power Demo Option
-----
-Press [1] Manual Power Modes Configuration
-Press [2] Low Power Preamble Search Scenario
-Press [!] Reset MCU

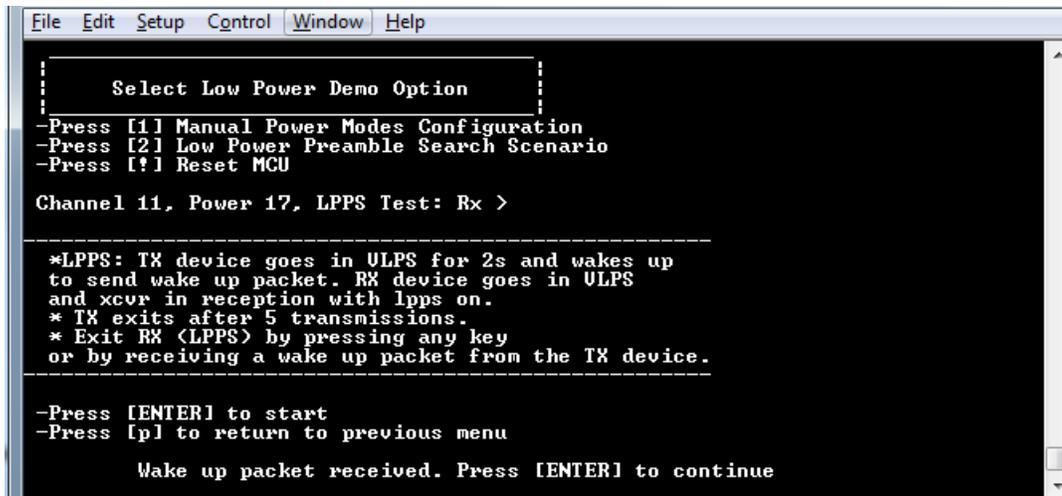
Channel 11, Power 17, LPPS Test: Tx >

-----
*LPPS: TX device goes in ULPS for 2s and wakes up
to send wake up packet. RX device goes in ULPS
and xcvr in reception with lpps on.
* TX exits after 5 transmissions.
* Exit RX (LPPS) by pressing any key
or by receiving a wake up packet from the TX device.
-----

-Press [ENTER] to start
-Press [p] to return to previous menu

- Sent packet number 1
- Sent packet number 2
```

Figure 5-7. LPPS TX role start



```
File Edit Setup Control Window Help
-----
Select Low Power Demo Option
-----
-Press [1] Manual Power Modes Configuration
-Press [2] Low Power Preamble Search Scenario
-Press [?] Reset MCU

Channel 11, Power 17, LPPS Test: Rx >
-----
*LPPS: TX device goes in ULPS for 2s and wakes up
to send wake up packet. RX device goes in ULPS
and xcvr in reception with lpps on.
* TX exits after 5 transmissions.
* Exit RX (LPPS) by pressing any key
or by receiving a wake up packet from the TX device.
-----
-Press [ENTER] to start
-Press [p] to return to previous menu

Wake up packet received. Press [ENTER] to continue
```

Figure 5-8. LPPS RX receiving wake up packet