Application Report
SWRA493–October 2015

CC112x-CC1190 Boost Software Examples

Thomas Almholt

ABSTRACT

A common issue with developing a new system from start based on transceivers is how to test the various parts of the system either by themselves or as part of the entire system.

Texas Instruments provides a very powerful and comprehensive tool called SmartRF™ Studio 7, which enables testing of all our devices; however, in some cases, the hardware requirements for the system to operate may not be available or using a PC may not be the preferred way to performing testing.

The software described in this application report provides a simple alternative in these use cases. The software is based on basic drivers already developed for MSP430's to implement a simple UART-based test system.

The software discussed in this document can be downloaded from the following URL: http://www.ti.com/lit/zip/swra493.

Contents

SmartRF is a trademark of Texas Instruments.
All other trademarks are the property of their respective owners.

1 Features and Benefits of UART-Based Test Suite

This software is based on a set of standard drivers that have already been released for various transceivers and microcontrollers. The new feature is the ability to run a test suite over a simple two-wire universal asynchronous receiver/transmitter (UART) connection to a simple terminal program.

The most common UART speed is 115200 (8N1), this setting is used for all projects except those involving the MSP430G2 Launch PAD, which only supports 9600 (8N1).

```
Long Range Sub1Ghz demonstration
Select from items below:
***********************************************************************
 1) Continuous Unmodulated TX   (line, chan) eg.(1 25)
 2) Continuous Modulated TX     (line, chan) eg.(2 25)
 3) Static RX (RX leakage test) (line, chan) eg.(3 25)
 4) TX Single Channel BER       (line, chan, pkt) eg.(4 25 1000)
 5) RX Single Channel BER       (line, chan, pkt) eg.(5 25 1000)
 6) TX freq. hopping BER        (line, pkt) eg.(6 1000)
 7) RX freq. hopping BER        (line, pkt) eg.(7 1000)
 8) RX sniff mode        (line, chan, len) eg.(8 25 20) len=0 > variable
 9) SmartRF Selector     (line, 1=Low, 2=Med, 3=High) eg.(9 2)
10) Radio Setup          (line, freq, chan spc) eg. (10 902 750 500)
***********************************************************************
Current configuration: (Device = CC1101)
   Current configuration:
Config:  9 Low data rate => 1200bps, 4kHz dev, 25kHz RX BW
Freq  : 10 902 750 500
*************************************************************
CMD >
```

After the software has been successfully downloaded to the chosen development kit and a correctly configured UART connection has been established, the text-based menu system seen above should appear. If it does not appear on the first try, pressing hard reset on the development kits will start the process from the beginning and the UART menu will appear.

The following subsections provide a brief description of each menu items.

1.1 Continuous Unmodulated TX

This command starts a special test mode that produces an unmodulated carrier at a specified frequency; choose the frequency by selecting a channel starting from 0. The value 0 produces a frequency at the base frequency of the current configuration; the current configuration can be altered. For more details, see Section 1.10.

```
Continuous Unmodulated TX   (line, chan), for example,(1 25)
```

This command is enabled by entering 1 followed by the desired channel number for this test. For details on base frequency and channel spacing configuration, see Section 1.10.

```
***********************************************************************
Current configuration: (Device = CC1101)
   Current configuration:
Config:  9 Low data rate => 1200bps, 4kHz dev, 25kHz RX BW
Freq  : 10 902 750 500
*************************************************************
CMD >
1 25
Started unmodulated TX, Press RETURN key twice to stop
```

To exit the test mode, send two consecutive “carriage-returns”. This is done by hitting the return key on the terminal program twice, which resets the chip into idle mode ready for the next command.

1.2 Continuous Modulated TX

This command starts a second special test mode that produces a modulated carrier at a specified frequency; choose the frequency by selecting a channel starting from 0. The value 0 produces a frequency at the base frequency currently configured in Section 1.10.

```
Continuous Modulated TX   (line, chan), for example,(2 25)
```

This command is enabled by entering 2 followed by the desired channel number for this test. For details on base frequency and channel spacing configuration, see Section 1.10. To exit test mode, see Section 1.1.

1.3 Static RX

This command starts a third special test mode that forces the device into RX at a specified frequency; choose the frequency by selecting a channel starting from 0. The value 0 produces a frequency at the base frequency currently configured in Section 1.10.

```
Static RX (RX leakage test) (line, chan), for example,(3 25)
```

This command is enabled by entering 3 followed by the desired channel number for this test. For details on base frequency and channel spacing configuration, see Section 1.10. To exit test mode, see Section 1.1.

1.4 TX Single Channel BER

This command starts the transmit side of a bit error rate test at a specific frequency and based on the configuration specified in Section 1.10.

```
TX Single Channel BER       (line, chan, pkt), for example, (4 25 1000)
```

This command is enabled by entering 4 followed by the desired channel number and the number of packets wanted in the test. The maximum number of packets accepted by the tool is 65535.

- The TX single channel test mode is automatically entered if no buttons are touched within the first 60 seconds after power up, which enables range testing without the need of a computer.

The details of data link can be found on the UART output during the test. These details include packet counter, measured RSSI and byte errors found during the transmission of reception of each individual packet.

```
***********************************************************************
Current configuration: (Device = CC1101)
   Current configuration:
Config:  9 Low data rate => 1200bps, 4kHz dev, 25kHz RX BW
Freq  : 10 902 750 500
*************************************************************
CMD > 4 25 1000

Perform RF burst transmit function with fixed channel
 PKT FW-RSSI/ERR RW-RSSI/ERR DELAY
   0 -74   0     -76   0     -11109
   1 -73   0     -76   0     -11109
   2 -73   0     -76   0     -11109
   3 -72   0     -76   0     -9522
   4 -72   0     -76   0     -11109
```

1.5 RX Single Channel BER

This command starts the receiver side of a bit error rate test at a specific frequency and based on the configuration specified in Section 1.10.

```
RX Single Channel BER       (line, chan, pkt), for example, (5 25 1000)
```

This command is enabled by entering 5 followed by the desired channel number and the number of packets wanted in the test. The maximum number of packets accepted by the tool is 65535.

```
***********************************************************************
Current configuration: (Device = CC1101)
   Current configuration:
Config:  9 Low data rate => 1200bps, 4kHz dev, 25kHz RX BW
Freq  : 10 902 750 500
*************************************************************
CMD > 5 25 1000

Perform RF burst receive function with fixed channel
 PKT FW-RSSI/ERR FREQ_ERR
   0 -74   0     11109
   1 -73   0     11109
   2 -73   0     11109
```

This mode can also be entered by just pressing a button (varies for each hardware configuration) on the development kit within the first 60 seconds after power up, which enables range testing without the need of a computer.

1.6 TX Frequency Hopping BER

This command starts the transmit side of a frequency hopping bit error rate test based on the configuration specified in Section 1.10.

```
TX freq. hopping BER        (line, pkt), for example, (6 1000)
```

```
***********************************************************************
Current configuration: (Device = CC1101)
   Current configuration:
Config:  9 Low data rate => 1200bps, 4kHz dev, 25kHz RX BW
Freq  : 10 902 750 500
*************************************************************
CMD > 6 1000

Perform RF burst receive function with fixed channel
 PKT FW-RSSI/ERR FREQ_ERR
   0 -74   0     11109
   1 -73   0     11109
   2 -73   0     11109
```

This command is enabled by entering 6 followed by the desired number of packets in the test. The maximum number of packets accepted by the tool is 65535.

1.7 RX Frequency Hopping BER

This command starts the receive side of a frequency hopping bit error rate test based on the configuration specified in Section 1.10.

```
***********************************************************************
Current configuration: (Device = CC1101)
   Current configuration:
Config:  9 Low data rate => 1200bps, 4kHz dev, 25kHz RX BW
Freq  : 10 902 750 500
*************************************************************
CMD > 7 1000

Perform RF burst receive function with fixed channel
 PKT FW-RSSI/ERR FREQ_ERR
   0 -74   0     11109
   1 -73   0     11109
   2 -73   0     11109
```

This command is enabled by entering 7 followed by the desired number of packets in the test. The maximum number of packets accepted by the tool is 65535.

Due to the frequency hopping operation, the time it takes for the initial link to be made will vary. This results in the first set of packet being missed by the transmitter, which is expected behavior.

1.8 RX Sniff Mode

This command enables a special conspicuous listening mode or sniff mode. The receiver tries to demodulate any packet on the air at the specified channel and print the content of the packet on the UART port. The configuration specified in Section 1.10 will be used for packet sniffing.

```
*****************************************************************************
Current configuration: (Device = CC1101)
   Current configuration:
Config:  9 Low data rate => 1200bps, 4kHz dev, 25kHz RX BW
Freq  : 10 902 750 500
******************************************************************
CMD > 8

Perform RF burst receive function with fixed channel
 PKT FW-RSSI/ERR FREQ_ERR
   0 -74   0     11109
   1 -73   0     11109
   2 -73   0     11109
```

1.9 SmartRF Selector

This command selects between three default settings: low, medium and high data rate. The settings are taken from SmartRF Studio version 7.0 (2.1.0). Simple modifications to the SYNC words have been done to enable interoperability between CC112x and CC110x series devices.

```
*****************************************************************************
Current configuration: (Device = CC1101)
   Current configuration:
Config:  9 Low data rate => 1200bps, 4kHz dev, 25kHz RX BW
Freq  : 10 902 750 500
******************************************************************
CMD > 9 0
```

When selecting the low data and the medium data rate, they are interoperable between the two generations of devices. However, the CC112x does not support data rates above 200kbps so the high-speed mode is not compatible across device families.

1.10 Radio Setup

The only setting that is available to change from the UART interface system is the base frequency in steps of 1kHz. This enables you to change the base frequency of the test and allows you to dial-in any frequency error that there might be between to boards operating in narrow band but running of standard XTALs.

```
*****************************************************************************
Current configuration: (Device = CC1101)
   Current configuration:
Config:  9 Low data rate => 1200bps, 4kHz dev, 25kHz RX BW
Freq  : 10 902 750 500
******************************************************************
CMD > 10 903 000 501
```

The interpreter used to decode the commands is very simple and cannot interpret the entire 9-digit frequency value. Therefore, you have to write the wanted frequency in three sets of 3 digits as shown above.

1.11 Automatic Mode

Each of the development kits are able to run in autonomous mode. To enable the use case of least available hardware, it was chosen to have the software automatically start in TX mode after a 60 second timeout; however, if a button has been pressed within the first 60 seconds the board immediately enters RX mode. In this mode, it is not possible to change the configuration of the radios in run-time; the lowest data rate option is automatically selected.

Use the LEDs to get feedback on the quality of the data link (green LED is generally good and red LED is for missed or failed packets).

2 Hardware Requirements to Run the Software

The software has been written in an effort to make it portable and very small. The required flash size is less than 16KB when compiled on an MSP430 and the required RAM is less than 512 bytes. This makes it possible to run the software on the lowest capacity value line MSP430 LaunchPAD, the MSP430G2553. Port exists for the MSP430F5529 LaunchPAD and the MSP430F5438A-based development kit called the “TRXEB”.

3 Configuring Hardware Options

The main configuration file for the various development kits is called *hal_spi_rf_xxx.c*. For this release, three such board support package files were generated:

- hal_spi_rf_exp5529.h
- hal_spi_rf_exp430g2.h
- hal_spi_rf_trxeb

Inside each of these files are three statements that define the specific devices used on the hardware. Here, the selections are the CC1101 or the CC112x series of transceivers and then, optionally, the range extender with each of the transceivers is used.

```
#define USE_CC1101
#define USE_CC112X
#define ENABLE_RANGE_EXTENDER
```

This next statement defines the crystal frequency used on the hardware. For each device, there is a range of applicable crystal frequencies to choose between (this statement is used to configure the software for the specific values chosen).

```
#define RF_XTAL 26000
```

4 Port to Test Software to Custom Hardware

This section describes how to port the code to a different piece of hardware either customer specific or a TI development kit not currently supported.

4.1 SPI Configuration for the Transceiver

The serial peripheral interface (SPI) port on the MSP430 should be configured as a 3-pin device where the chip select pin is controlled manually. The detailed pin configuration is found in *hal_spi_rf_xx.h*, where xx denotes which specific hardware is being compiled.

```
/* Transceiver SPI signal */
#define     RF_PORT_SEL           P3SEL
#define     RF_PORT_OUT           P3OUT
#define     RF_PORT_DIR           P3DIR
#define     RF_PORT_IN            P3IN

#define     RF_MOSI_PIN           BIT0
#define     RF_MISO_PIN           BIT1
#define     RF_SCLK_PIN           BIT2
```

4.2 Chip Select for the Transceiver

The next set of definitions help to configure the location of the chip select pin to use to communicate with the transceiver.

```
#define     RF_CS_N_PORT_SEL      P2SEL
#define     RF_CS_N_PORT_DIR      P2DIR
#define     RF_CS_N_PORT_OUT      P2OUT
#define     RF_CS_N_PIN           BIT2
```

4.3 Reset Pin for the Transceiver

For the CC112x series of devices there is a reset pin, which can be optionally used. On all of the TI development kits the reset pin is used; therefore, it needs to be configured correctly.

```
/* Transceiver optional reset signal */
#define     RF_RESET_N_PORT_SEL      P2SEL
#define     RF_RESET_N_PORT_DIR      P2DIR
#define     RF_RESET_N_PORT_OUT      P2OUT
#define     RF_RESET_N_PIN           BIT6
```

4.4 Interrupt Configuration for the Transceiver

These next sets of definitions configure the location of the interrupt enabled signals between the transceiver and the MCU.

```
/* Transceiver interrupt configuration */
#define     RF_PORT_VECTOR         PORT2_VECTOR
#define     RF_GDO_OUT             P2OUT
#define     RF_GDO_DIR             P2DIR
#define     RF_GDO_IN              P2IN
#define     RF_GDO_SEL             P2SEL
#define     RF_GDO_PxIES           P2IES
#define     RF_GDO_PxIFG           P2IFG
#define     RF_GDO_PxIE            P2IE
#define     RF_GDO_PIN             BIT0

/* Transceiver interrupt configuration */
#define     RF_PORT_VECTOR         PORT2_VECTOR
#define     RF_GDO_OUT             P2OUT
#define     RF_GDO_DIR             P2DIR
#define     RF_GDO_IN              P2IN
#define     RF_GDO_SEL             P2SEL
#define     RF_GDO_PxIES           P2IES
#define     RF_GDO_PxIFG           P2IFG
#define     RF_GDO_PxIE            P2IE
#define     RF_GDO_PIN             BIT0
```

4.5 Optional Control Pins for the Range Extender

To enable the use of the optional front-end module, specifically the CC1190, the specific pins are configured as shown below:

```
/* CC1190 Control signals */
#define    RF_LNA_EN_PxOUT         P1OUT
#define    RF_LNA_EN_PxDIR         P1DIR
#define    RF_LNA_EN_PIN           BIT6

#define    RF_PA_EN_PxOUT          P2OUT
#define    RF_PA_EN_PxDIR          P2DIR
#define    RF_PA_EN_PIN            BIT7
```

4.6 Optional Pins for Creating a Button User Interface

To enable the use of the optional button-based user interface, the specific pins are configured as shown below:

```
/* Optional button interrupt configuration */
#define     BUTTON_VECTOR          PORT1_VECTOR
#define     BUTTON_OUT             P1OUT
#define     BUTTON_DIR             P1DIR
#define     BUTTON_IN              P1IN
#define     BUTTON_SEL             P1SEL
#define     BUTTON_PxIES           P1IES
#define     BUTTON_PxIFG           P1IFG
#define     BUTTON_PxIE            P1IE
#define     BUTTON_PIN             BIT1
#define     BUTTON_REN             P1REN
```

4.7 Optional LED Pin Configuration

It is strongly recommended to use LED indicators that are configured as shown below:

```
/* Macro to enable LEDs */
#define     LED1_PxOUT      P1OUT
#define     LED1_PxDIR      P1DIR
#define     LED1_PIN        BIT0
#define     LED2_PxOUT      P4OUT
#define     LED2_PxDIR      P4DIR
#define     LED2_PIN        BIT7
#define     LED3_PxOUT      P1OUT
#define     LED3_PxDIR      P1DIR
#define     LED3_PIN        BIT0
#define     LED4_PxOUT      P4OUT
#define     LED4_PxDIR      P4DIR
#define     LED4_PIN        BIT7
```

Many of the hardware configurations only have two physical LEDs on them. For those cases, an acceptable experience is provided by just reusing those two LEDs as shown in the example above.

5 References

1. *CC1120-CC1190 BoosterPack* (SWRA492)
2. *CC112X/CC1175 Low-Power High Performance Sub-1 GHz RF Transceivers/Transmitter* (SWRU295)
3. *CC120X Low-Power High Performance Sub-1 GHz RF Transceivers User's Guide* (SWRU346)
4. *CC1120 High-Performance RF Transceiver for Narrowband Systems Data Sheet* (SWRS112)
5. *CC1121 High-Performance Low-Power RF Transceiver Data Sheet* (SWRS111)
6. *CC1125 Ultra-High Performance RF Narrowband Transceiver Data Sheet* (SWRS120)
7. *CC1200 Low-Power, High-Performance RF Transceiver Data Sheet* (SWRS123)
8. *CC112x, CC1175 Silicon Errata* (SWRZ039)
9. *SmartRF Studio 7* (SWRC176)