EZRadioPRO® RF Testing Quick-Start Guide

1. Introduction

This user's guide allow the user to quickly verify basic TX and RX performance of RF Test Cards (such as the DKDBx series of RF Test Cards available from Silicon Labs) when installed in a Load Board and controlled by Wireless Development Suite (WDS) Ver 3.0 software. These TX and RX functional tests may be performed by creating (or loading) and then running an appropriate script from within WDS.

This user's guide assumes that the user has already installed WDS 3.0 software, and understands its basic usage and functionality. This includes such operations as connection of the Load Board and RF Test Card(s), selection of the Radio Control Panel or Register Settings Panel applications, and loading, saving, or running scripts. If the user is not yet familiar with these operations, we recommend that the user first review the WDS Chip Configurator User's Guide prior to performing the procedures outlined in this user's guide. The WDS Chip Configurator user's guide may be downloaded from the Silicon Labs website.

1.1. Hardware Requirements

This user's guide covers testing one (or more) of the following types of EZRadioPRO® RF Test Cards:

- 443x-DKDB0 Antenna Diversity TRX Test Card
- 443x-DKDB1 TX/RX Split TRX Test Card (High Band)
- 443x-DKDB2 Single Antenna TRX Test Card (High Band)
- 443x-DKDB5 TX/RX Split TRX Test Card (Low Band)
- 443x-DKDB6 Single Antenna TRX Test Card (Low Band)

The RF Test Cards listed above are designed to plug into, and be controlled through the Load Board v2.2 (part of Silicon Labs ISM Development Kit ISM-DK3).

1.2. Hardware Limitations

Not every functional test described may be performed with ***all*** of the RF Test Cards listed above. Any RF Test Card that contains either a TX/RX Switch (e.g., DKDB2 or DKDB6) or an Antenna Diversity Switch (e.g., DKDB0) requires the mapping of two GPIO pins (GPIO1 and GPIO2) on the RFIC as control lines for the switch. As a result, these types of RF Test Cards have only one remaining "free" GPIO pin (GPIO0) for configuration for user-defined functionality (such as RXDATA output, TXDATA input, etc.).

Several of the scripts and functional tests described later in this document require ***two*** available GPIO pins in order to function properly. As an example, the script and functional tests in "3.3.1. TX Synchronous Direct Mode" on page 18 that demonstrate TX Synchronous Direct modulation require one GPIO pin for the TXCLK output signal and another GPIO pin for the TXDATA input signal. This script ***should not*** be used to control a DKB0/DKDB2/DKDB6-type of RF Test Card, as erroneous results will occur.

In a similar fashion, several of the RX-functional scripts shown later also require two GPIO pins, one for the RXDATA output signal and one for the RXCLK output signal. These scripts also should not be used to control a DKDB0/DKDB2/DKDB6-type of RF Test Card, as erroneous results will again occur.

When possible, Silicon Labs has taken care to construct example scripts that are not specific to one single type of RF Test Card. However, the user must keep in mind that such universal board compatibility is not possible for all scripts and test functions.

1.3. General Overview of RFIC Settings and Script Commands

All TX and RX functional tests share certain common required steps or register settings. That is to say, all scripts will contain some amount of commonality, as it is always necessary to specify such things as the desired frequency, the desired V_{DD} supply voltage, etc. The following is a list of the SPI register settings that are common to nearly all TX or RX functional tests and scripts. Some are applicable to TX functions (e.g., TX output power) while others are applicable to RX functions (e.g., RX modem settings).

- Setting the V_{DD} supply voltage (from the Load Board)
- Selecting the channel center frequency
- Adjusting the crystal frequency (to minimize frequency error)
- Selecting the modulation type (e.g., FSK/GFSK/OOK/CW)
- Selecting the data rate and deviation
- Selecting the output power level (TX only)
- Configuring the RX modem settings (RX only)
- Configuring the GPIO pins (for RF Switch control, TX/RX Data functions, etc)
- Selecting operational mode (e.g., Direct or Packet Mode)

The scripts shown in this document deliberately show the minimum number of commands or register settings required to demonstrate a given TX or RX function. To enable more complex functionality (such as enabling AFC, configuring Antenna Diversity, enabling Manchester encoding, etc.), additional commands may be added to these example scripts. However, the main purpose of this guide is to quickly get the user's test hardware "up and running" by easy-to-understand demonstration of the most basic register settings and SPI commands.

In addition to understanding the effect of the various SPI commands shown in these example scripts, the user should be aware of the source of these SPI commands. Silicon Labs has created an EZRadioPRO® Register Calculator worksheet (in Microsoft Excel, available for download from the Silicon Labs website. The desired operational parameters (such as center channel frequency, data rate, deviation etc.) may be entered into this worksheet, and the corresponding required SPI register settings are clearly displayed. In all cases, the SPI register settings contained within these example scripts were obtained directly from this EZRadioPRO® Register Calculator worksheet. Silicon Labs has also created "AN453: EZRadioPRO® Register Calculator User's Guide" to demonstrate how to make use of the Register Calculator worksheet.

The most basic rules and commands governing the script file language are summarized here (see the WDS Chip Configurator user's guide for more detailed information). A batch file (also called a script) can consist of comments and hardware commands:

- If a line starts with the # character, then the line is considered a comment and is used for user reference only.
- If a line starts with #BATCHNAME, then the text in this line will appear as the title of the script, in the line located just above the batch command textbox window.
- If a line starts with a valid hardware command (such as those used to change the value of an SPI Register or to set the V_{DD} power supply voltage), then the given command will be sent to the hardware through the USB interface.

A partial summary of the most commonly used hardware commands is shown in Table 1.

Table 1. Load Board Hardware Commands

Load Board Command	Description
Vxx	Set V_{DD} supply voltage from Load Board to RF Test Card. (64h = 2.0 V to C1h = 5.0 V)
Ln	Set Load Board output bit “n” to a logic LOW. (The SDN pin of the EZRadioPRO® device is connected to Load Board output bit 6.)
Hn	Set Load Board output bit “n” to a logic HIGH. (The SDN pin of the EZRadioPRO® device is connected to Load Board output bit 6.)
X0	Select TXDATA source from SMA connector on Load Board.
S2 xxxx	Send xxxx (hex) 2-byte command to EZRadioPRO® device via the SPI interface.

1.4. General Overview of Hardware Functional Tests

The following sections will demonstrate how to perform basic functional tests on the RF Test Cards including such tasks as the following:

- Setting the center channel frequency and TX output power
- Selecting the modulation type (e.g., FSK/GFSK/OOK)
- Selecting the data rate and transmit deviation
- Selecting the operational mode (e.g., Direct Mode, Packet FIFO Mode)
- Setting the RX modem parameters
- Performing RX BER measurements
- Performing a basic one-way link test

All of the scripts referenced in this guide may be downloaded from the Silicon Labs website. The user will need to download the Chip_Configurator_Scripts.zip file from within the Software Downloads section of the website. The list of scripts contained within this zip file, along with a brief description of their functionality, is shown below:

- **TX_913.0MHz_CW.txt**
  Creates an unmodulated transmit carrier signal at 913.0 MHz, maximum output power.
- **TX_913.0MHz_IntPN9_GFSK_40kbps_20kDev.txt**
  Creates a transmit signal at 913.0 MHz, maximum output power level, GFSK-modulated by an internally-generated PN9 sequence, data rate of 40 kbps, 20 kHz peak deviation.
- **TX_913.0MHz_IntPN9_OOK_10kbps.txt**
  Creates a transmit signal at 913.0 MHz, maximum output power level, OOK-modulated by an internally-generated PN9 sequence, data rate of 10 kbps.
- **TX_913.0MHz_DirectModSync_GFSK_40kbps_20kDev.txt**
  Creates a transmit signal at 913.0 MHz, maximum output power level, GFSK-modulated by a TXDATA stream obtained synchronously from an external data source, data rate of 40 kbps, 20 kHz peak deviation.
- **TX_913.0MHz_DirectModAsync_FSK_20kDev.txt**
  Creates a transmit signal at 913.0 MHz, maximum output power level, FSK-modulated by a TXDATA stream obtained asynchronously from an external data source, 20 kHz peak deviation. Data rate is determined by external data source.
- **TX_913.0MHz_PacketFIFO_GFSK_40kbps_20kDev.txt**
  Creates a transmit signal at 913.0 MHz, maximum output power level, GFSK-modulated in a Packet structure at a data rate of 40 kbps, 20 kHz peak deviation. The contents of the packet (e.g., Preamble, Sync, Payload fields) are programmed via the SPI interface.

- **RX_913.0MHz_BER_ContPream_GFSK_40kbps_20kDev.txt**
  Configures the RF Test Card for Continuous Receive Mode at 913.0 MHz. RX modem parameters are optimized to receive a GFSK-modulated signal at a data rate of 40 kbps, 20 kHz peak deviation, with a data pattern equivalent to a continuous Preamble (i.e., infinitely-long “1010” pattern).
- **RX_913.0MHz_BER_PN9/11/15_GFSK_40kbps_20kDev.txt**
  Configures the RF Test Card for Continuous Receive Mode at 913.0 MHz. RX modem parameters are optimized to receive a GFSK-modulated signal at a data rate of 40 kbps, 20 kHz peak deviation, with a data pattern equivalent to a continuously-looped PN sequence. Three example scripts are provided: one for a PN9 sequence, one for a PN11 sequence, and one for a PN15 sequence.
- **RX_913.0MHz_BER_PN9_OOK_10kbps.txt**
  Configures the RF Test Card for Continuous Receive Mode at 913.0 MHz. RX modem parameters are optimized to receive an OOK-modulated signal at a data rate of 10 kbps, with a data pattern equivalent to a continuously-looped PN9 sequence.
- **RX_913.0MHz_PacketDirect_GFSK_40kbps_20kDev.txt**
  Configures the RF Test Card for Direct Receive Mode at 913.0 MHz. RX modem parameters are optimized to receive a GFSK-modulated signal at a data rate of 40 kbps, 20 kHz peak deviation, with a Packet structure comprised of a Preamble, Sync word, and a fixed-length Payload. RXDATA bits are output directly to a GPIO pin.
- **RX_913.0MHz_PacketFIFO_GFSK_40kbps_20kDev.txt**
  Configures the RF Test Card for Packet FIFO Receive Mode at 913.0 MHz. RX modem parameters are optimized to receive a GFSK-modulated signal at a data rate of 40 kbps, 20 kHz peak deviation, with a Packet structure comprised of a Preamble, Sync word, and a fixed-length Payload. The received data bits are stored in internal FIFO memory for retrieval over the SPI interface.

2. Current Consumption Measurements

Continual current consumption measurements may be taken by connecting a standard multimeter to either JP3 or JP4 on the Load Board v2.2, as shown in Figure 1.

Both of the jumpers, JP3 and JP4, are wired directly in parallel on the Load Board. Installing a jumper (or connecting a multimeter) between *either* pair of pins will result in connection of the programmed V_{DD} supply voltage to the RF Test Card. At first glance, providing two sets of jumpers may seem like unnecessary duplication of circuitry. However, this board configuration allows for easy on-the-fly or "hot" connection of a current meter, in the following fashion:

- Install a jumper on only one pair of pins (e.g., JP3)
- Command the RF Test Card to a desired state by running an appropriate script
- Connect a current meter between the other pair of pins (e.g., JP4)
- Remove the first jumper (JP3, in this example)
- Current meter now displays the current consumption, without interruption of the operational state of the RF Test Card

The user should be aware that most current meters typically exhibit some voltage drop across their terminals, when placed in series with the device under test (DUT). This amount of voltage drop is usually a function of the selected display range on the current meter, with a larger voltage drop corresponding to an increase in sensitivity of the current display range. As a result, the V_{DD} supply voltage delivered to the RF Test Card may be somewhat lower than the programmed V_{DD} supply voltage, when current consumption measurements are performed. The user may need to adjust the programmed V_{DD} supply voltage to compensate for this voltage drop, if any.

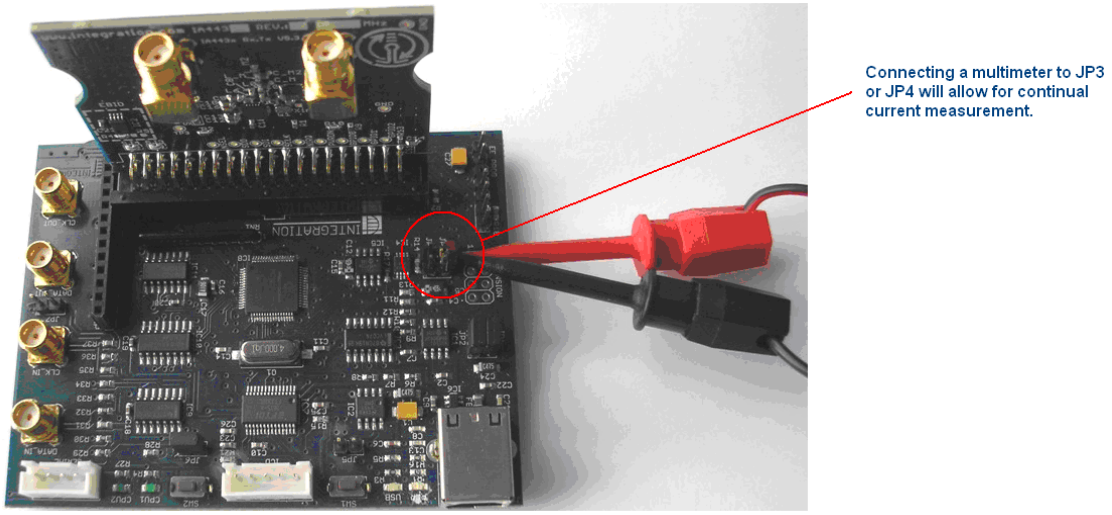

Figure 1. Current Consumption Measurement

3. Transmitter Functional Tests

3.1. TX CW Mode (Output Spectrum and Power Measurements)

The most basic TX functional test that may be performed with an EZRadioPRO RF Test Card is to simply command the RF Test Card to output a constant, unmodulated transmit signal. This allows easy measurement of the TX output power level, as well as the spectrum of the TX output signal. To perform this test, plug an appropriate EZRadioPRO RF Test Card into the Load Board. The Load Board and RF Test Card should then be connected to standard laboratory test equipment as shown in Figure 2.

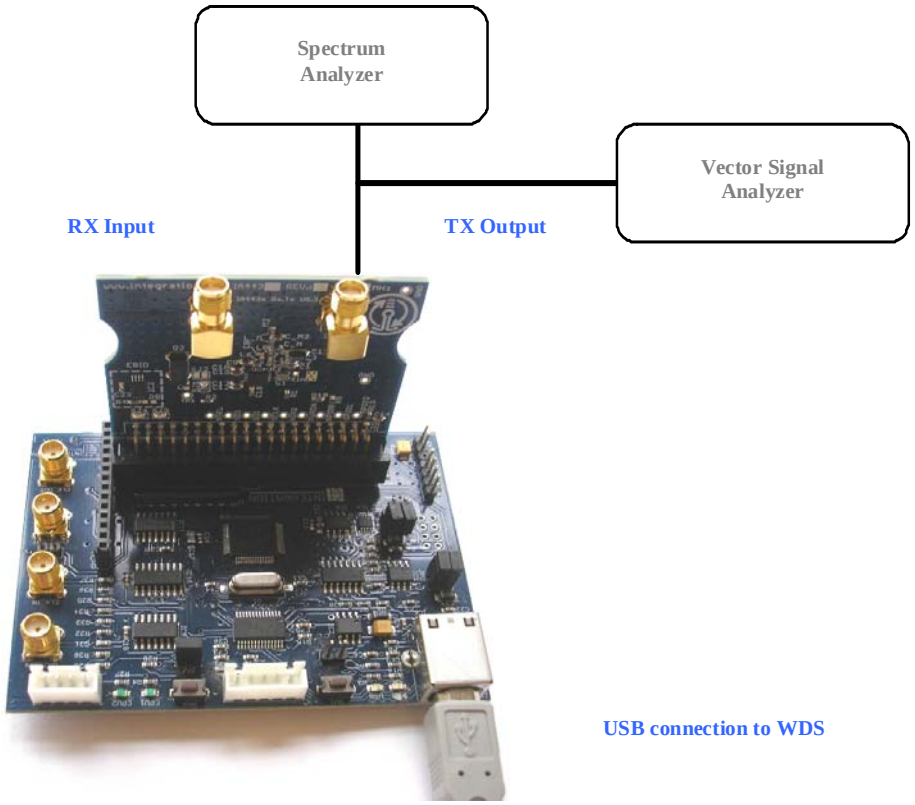

Figure 2. Test Setup for Output Spectrum and Power Measurements

The following script may be run from within WDS to place the RF Test Card into such a transmit state, at a center channel frequency of 913.0 MHz. This script is appropriate for use with any EZRadioPRO® RF Test Card except the DKDB0 Antenna Diversity TRX Test Card.

Script Name: TX_913.0MHz_CW.txt

```
#BATCHNAME TX CW 913.0MHz
# This script is appropriate for use with EZRadioPRO:
#  DKDB1 Split TX/RX or DKDB2 Single Antenna RF Test Card(s)
#  Do NOT use with DKDB0 AntDiv RF Test Card
#    (GPIO pins are not programmed for control of AntDiv Switch)
# Set SDN Pin 20 = LOW
L6
# Set VDD:  1.8V=V54, 2.4V=V7E, 3.0V=V98, 3.3V=VA1, 3.6V=VA9
VA1
# Apply Software Reset
S2 8780
# Adjust Crystal for zero freq error (User may need to modify this value)
S2 8971
### Set Desired Transmit Frequency = 913.0 MHz ###
S2 F575
S2 F6A2
S2 F780
# Set TX output power (ED07 = max, ED00 = min)
S2 ED07
### Configure GPIOs ###
# Configure GPIO0 = OFF (not used)
S2 8B1F
# Configure GPIO1 = TX State (to control RF Switch, if any)
S2 8C12
# Configure GPIO2 = RX State (to control RF Switch, if any)
S2 8D15
### Turn Transmitter ON ###
S2 8708
```

The spectrum plot in Figure 3 shows the expected TX output spectrum after running the script shown above. This output power measurement was taken at the TX SMA connector on a DKDB1 TX/RX Split TRX Test Card; the output power would be approximately 1 dB higher if measured directly at the TX pin of the RFIC (per the data sheet specification).

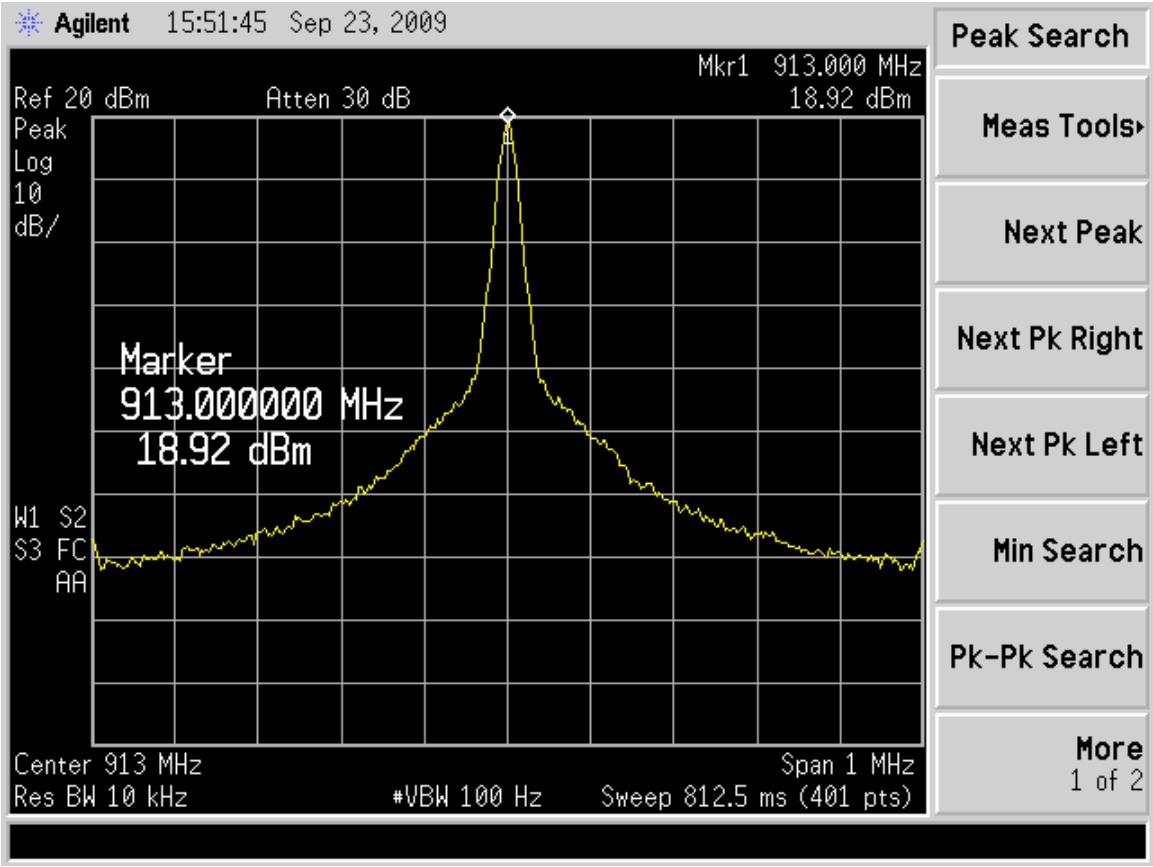

Figure 3. TX Output in CW Mode

3.1.1. Adjusting the TX Output Power

The TX output power may be easily adjusted by setting the 'txpow[2:0]' field in SPI Register 6Dh. Although the absolute value of output power observed will depend upon the specific type of RFIC within the EZRadioPRO family of chips, the relative change in output power is approximately 3 dB per step. The traces in Figure 4 were taken with the following three different script variations:

```
# Set TX output power (ED07 = max)
S2 ED07
OR
# Set TX output power (ED04 = mid)
S2 ED04
OR
# Set TX output power (ED00 = min)
S2 ED00
```

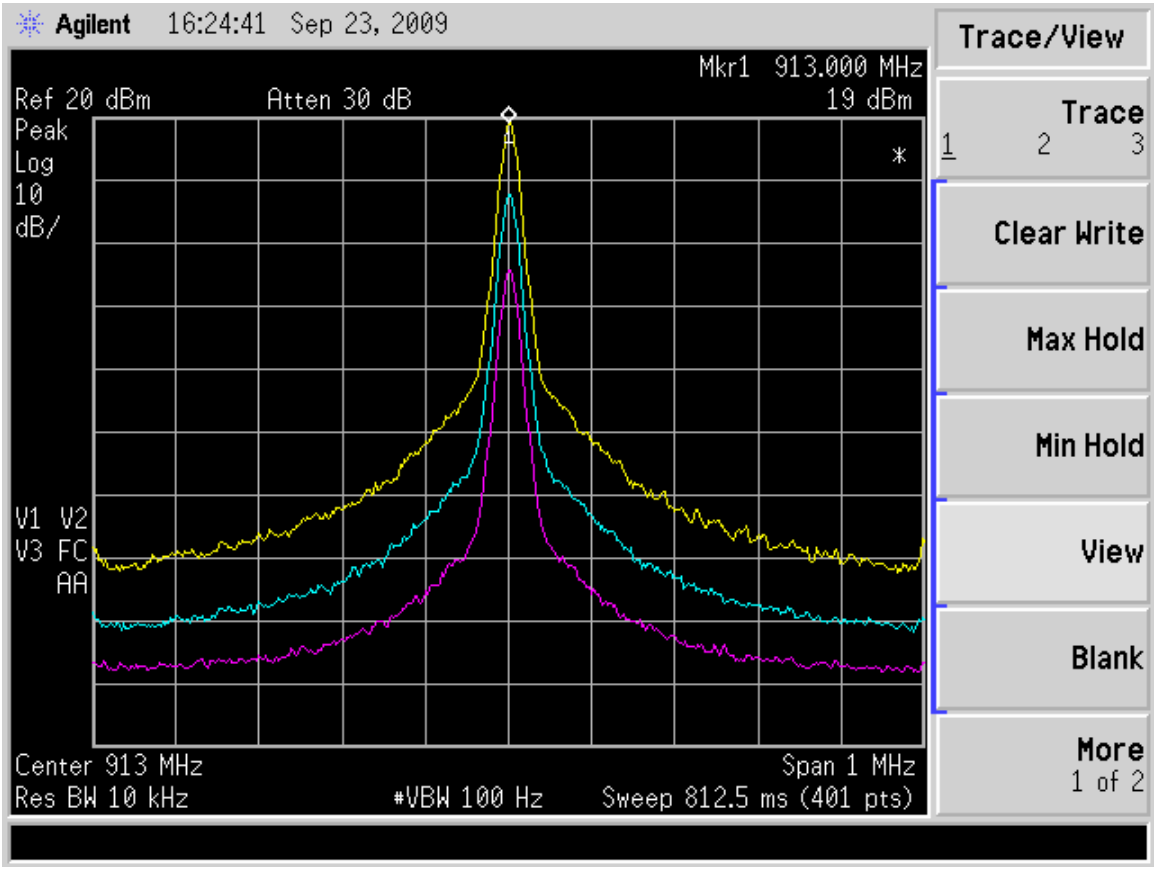

Figure 4. Adjustment of TX Output Power

The measurements shown in Figure 4 were taken on a Si4432 chip, with the yellow trace corresponding to SPI Register 6Dh = 07h, the blue trace = 04h, and the purple trace = 00h.

3.1.2. Changing the Output Frequency

The center channel frequency may easily be changed by setting three different SPI registers that control the divide ratios within the PLL Frequency Synthesizer. The exact mathematical formula for deriving the values to program in these registers is somewhat complex, and is not shown here (refer to the Si443x data sheet for a detailed discussion). It is far easier to instead use the EZRadioPRO Register Calculator worksheet to perform the calculations; simply enter the desired center channel frequency into the worksheet, and read off the corresponding values to be programmed into SPI Registers F5h–F7h. The traces in Figure 5 were taken with the following three different script variations:

```
### Set Desired Transmit Frequency = 905.0 MHz ###
S2 F575
S2 F63E
S2 F780
OR
### Set Desired Transmit Frequency = 915.0 MHz ###
S2 F575
S2 F6BB
S2 F780
OR
### Set Desired Transmit Frequency = 925.0 MHz ###
S2 F576
S2 F63E
S2 F780
```

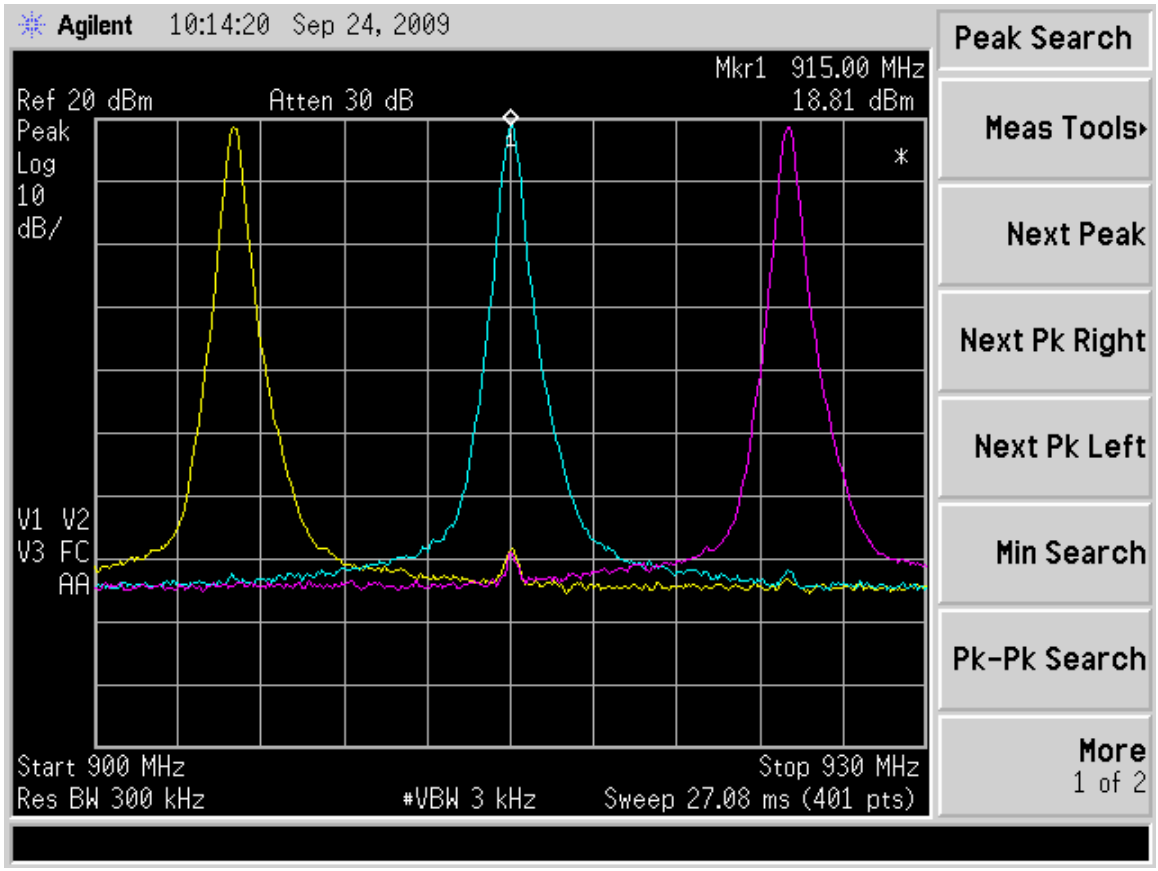

Figure 5. Changing the Center Channel Frequency

3.1.3. Compensating for Crystal Error

All crystal blanks will exhibit some small frequency error or tolerance in their resonant frequency of oscillation. This frequency tolerance is usually expressed as a percentage of the nominal frequency of oscillation, in parts-per-million or ppm. Typical crystal blank frequency tolerances may range from 2 to 5 ppm for very high quality crystals, to 20 to 50 ppm for lower quality crystals.

As the accuracy of the PLL frequency synthesizer in the RFIC depends directly upon the accuracy of the crystal oscillator frequency, it is desirable to tune the frequency of the crystal oscillator to exactly 30.000 MHz. If this frequency error is not trimmed out, the result may be a significant amount of end-to-end frequency error in the link, to the point where it may not be possible to establish the link. Fortunately, the RFIC contains a bank of programmable internal capacitors which are connected from the XIN and XOUT pins to GND, and which may be used to adjust the frequency of oscillation of the crystal.

Theoretically, the crystal frequency could be adjusted by measuring the 30.000 MHz crystal oscillator signal directly. However, it is much easier (and more accurate) to measure the frequency of the RF output signal, and to simply adjust the crystal capacitor bank until the output signal is exactly at the desired channel center frequency (913.000 MHz, in this example). The crystal capacitor bank is adjusted by setting SPI Register 09h.

```
# Adjust Crystal for zero freq error (User may need to modify this value)
S2 8971
```

The resulting output spectrum for three significantly-different values of the Crystal Cap Bank setting is shown in Figure 6.

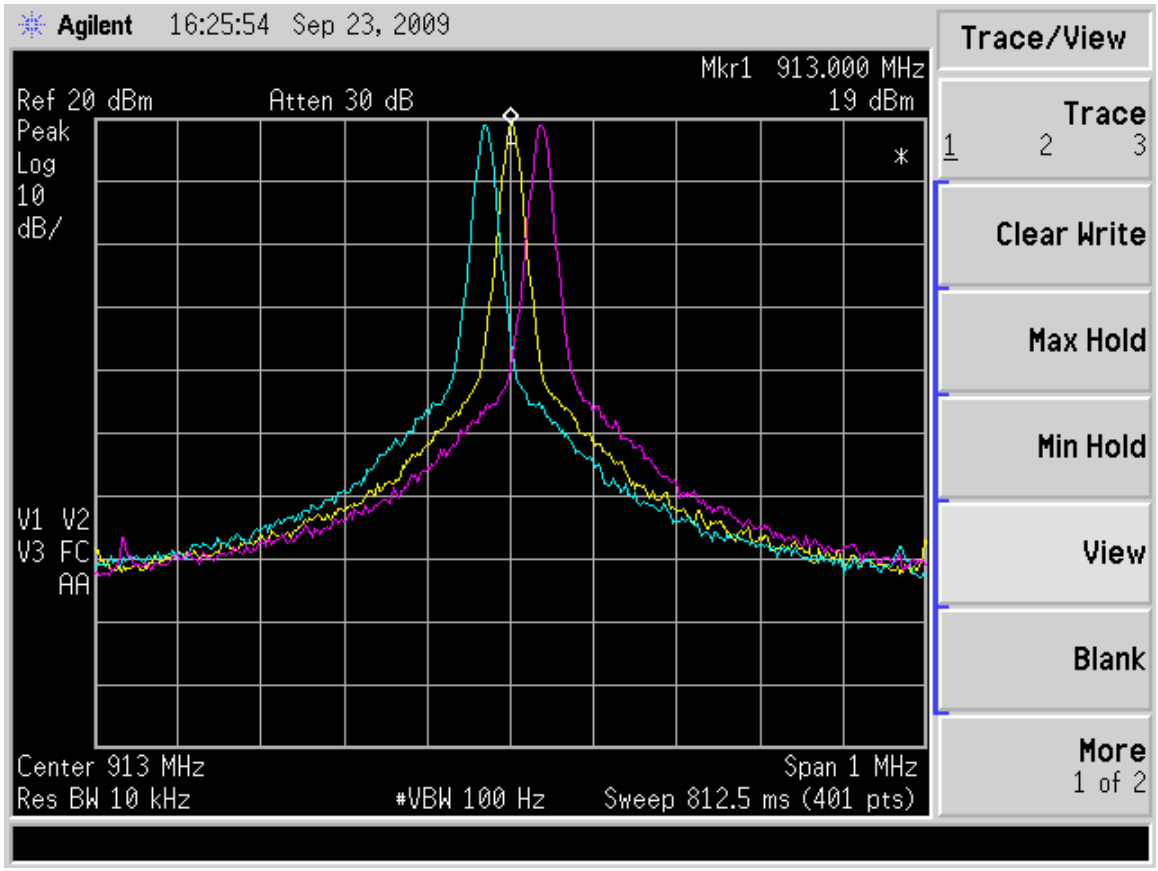

Figure 6. Adjusting the Crystal Frequency

The Crystal Cap Bank value shown in this example script (e.g., SPI Register 09 = 71h) is nothing more than a "placeholder value"; each user will need to modify this value to trim out the frequency error of their particular crystal blank.

3.2. TX Mode with Internal PN9 Modulation Source

So far, all of the measurements have been performed on an unmodulated CW carrier signal. We now demonstrate how to add modulation to the transmit signal.

In general, there are four additional parameters that must be specified in order to obtain the desired modulation:

- Select the *type* of modulation (e.g., FSK / GFSK / OOK)
- Select the *source* of modulation (e.g., internal FIFO, internal PN9 generator, external pin)
- Select the data rate
- Select the deviation (in case of FSK or GFSK)

The example script shown below will place the RF Test Card into Transmit Mode at a center channel frequency of 913.0 MHz, with GFSK modulation at a data rate of 40 kbps and a deviation of 20 kHz (i.e., mod index h=1). The source of modulation is provided by an internal PN9 sequence generator; there is no need for the user to supply a TXDATA pattern from an external source (such as an MCU or pattern generator). This script is appropriate for use with any EZRadioPRO RF Test Card *except* the DKDB0 Antenna Diversity TRX Test Card.

Script Name: TX_913.0MHz_IntPN9_GFSK_40kbps_20kDev.txt

```
#BATCHNAME TX 913.0MHz IntPN9 GFSK
# This script is appropriate for use with EZRadioPRO:
#  DKDB1 Split TX/RX or DKDB2 Single Antenna RF Test Card(s)
#  Do NOT use with DKDB0 AntDiv RF Test Card
#    (GPIO pins are not programmed for control of AntDiv Switch)
# Set SDN Pin 20 = LOW
L6
# Set VDD:  1.8V=V54, 2.4V=V7E, 3.0V=V98, 3.3V=VA1, 3.6V=VA9
VA1
# Apply Software Reset
S2 8780
# Adjust Crystal for zero freq error (User may need to modify this value)
S2 8971
### Set Desired Transmit Frequency = 913.0 MHz ###
S2 F575
S2 F6A2
S2 F780
# Select the modulation mode for GFSK, internal PN9 sequence.
#   Send 'S2 F132' for FSK, internal PN9.
S2 F133
### Transmitter Specific Parameters ###
# Set Data Rate = 40 kbps
S2 EE0A
S2 EF3D
S2 F00C
# Set Deviation to 20 kHz
S2 F220
# Set Tx output power (ED07 = max, ED00 = min)
S2 ED07
### Configure GPIOs ###
# Configure GPIO0 = OFF (not used)
S2 8B1F
# Configure GPIO1 = TX State (to control RF Switch, if any)
S2 8C12
# Configure GPIO2 = RX State (to control RF Switch, if any)
S2 8D15
### Turn Transmitter ON ###
S2 8708
```

3.2.1. Changing the Modulation Type

The type of modulation is selected by programming the “modtyp[1:0]” field in SPI Register 71h.

```
# Select the modulation mode for GFSK, internal PN9 sequence.
S2 F133

OR

# Select the modulation mode for FSK, internal PN9 sequence.
S2 F132
```

The resulting output spectrum for both FSK (blue trace) and GFSK (yellow trace) is shown in Figure 7 (40 kbps data rate, with 20 kHz peak deviation). The GFSK trace has modulation side lobes that are clearly lower than the FSK trace; this reduction in occupied spectral bandwidth is an important consideration in choosing the type of modulation.

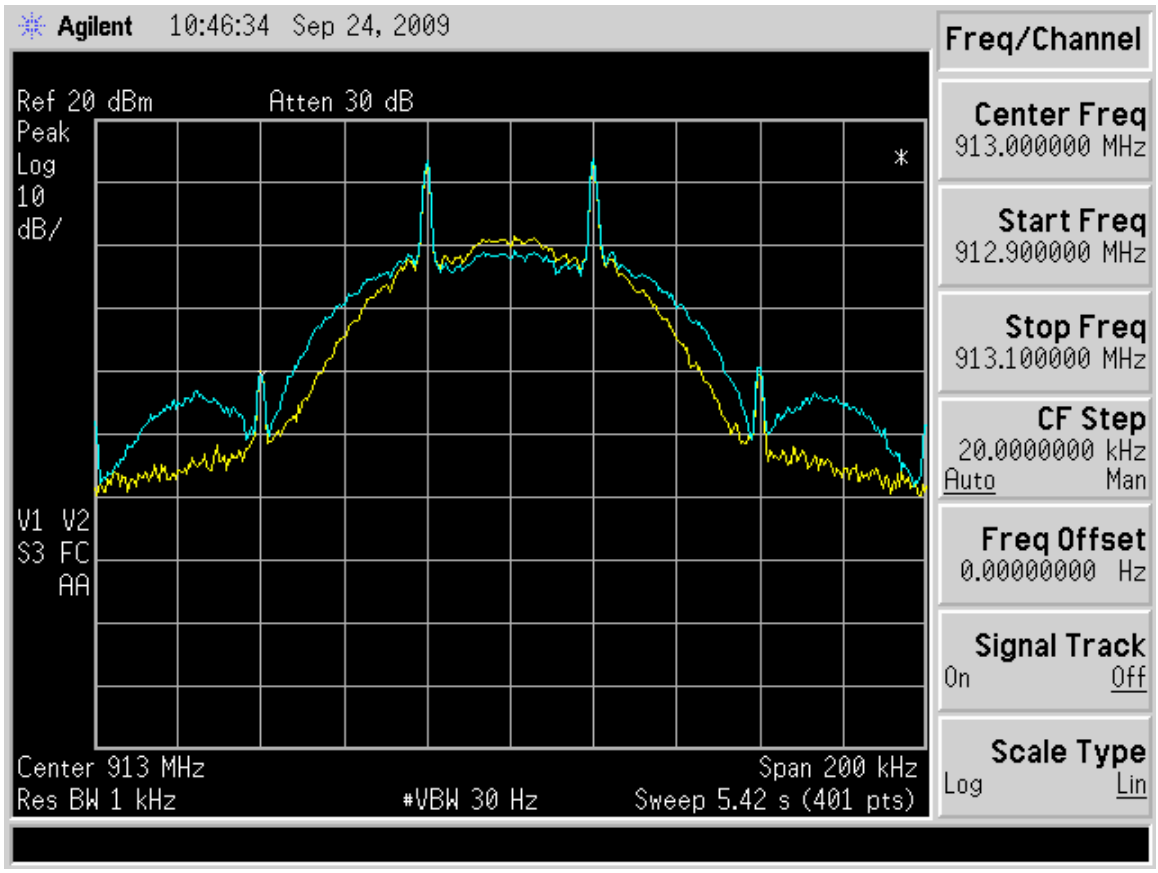

Figure 7. TX Spectrum for FSK and GFSK Modulation

If the user has access to a Vector Signal Analyzer (or other test equipment that can perform FM demodulation), it is also possible to view the modulation signal in the time domain. This is shown in Figure 8 for both FSK and GFSK.

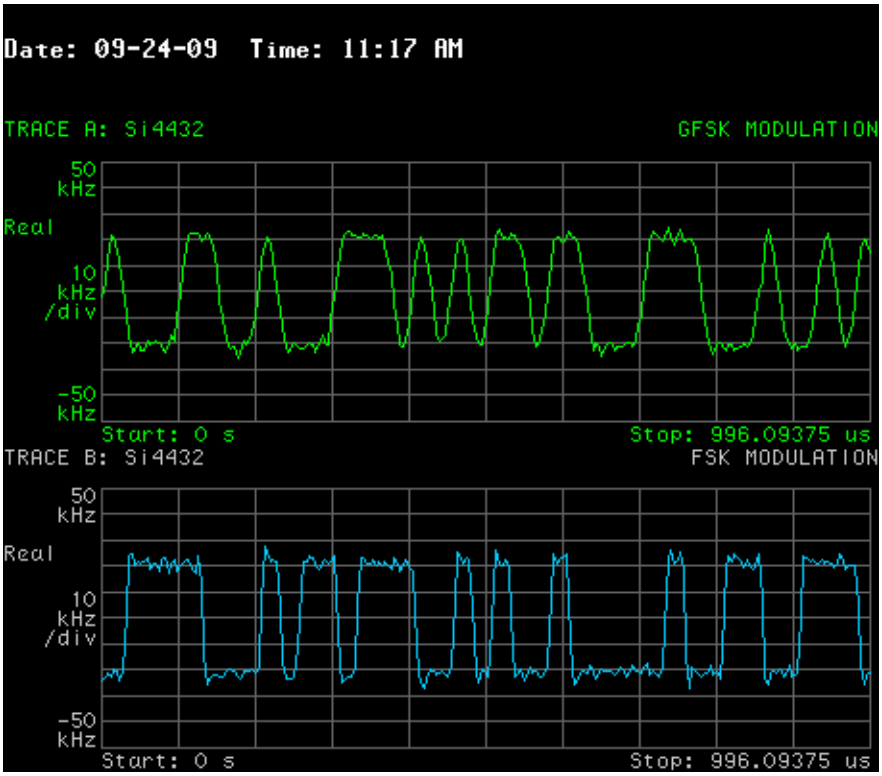

Figure 8. TX Modulation Waveforms for FSK and GFSK

It is also possible to configure the RFIC for OOK modulation.

```
# Select the modulation mode for OOK, internal PN9 sequence.
S2 F131
```

The resulting modulation waveform is shown in Figure 9 (again for 40 kbps data rate). This plot was taken with the spectrum analyzer placed in Zero-Span Mode, resulting in displaying the signal power vs. time. This clearly demonstrates the OOK modulation capability of the EZRadioPRO family of RFICs.

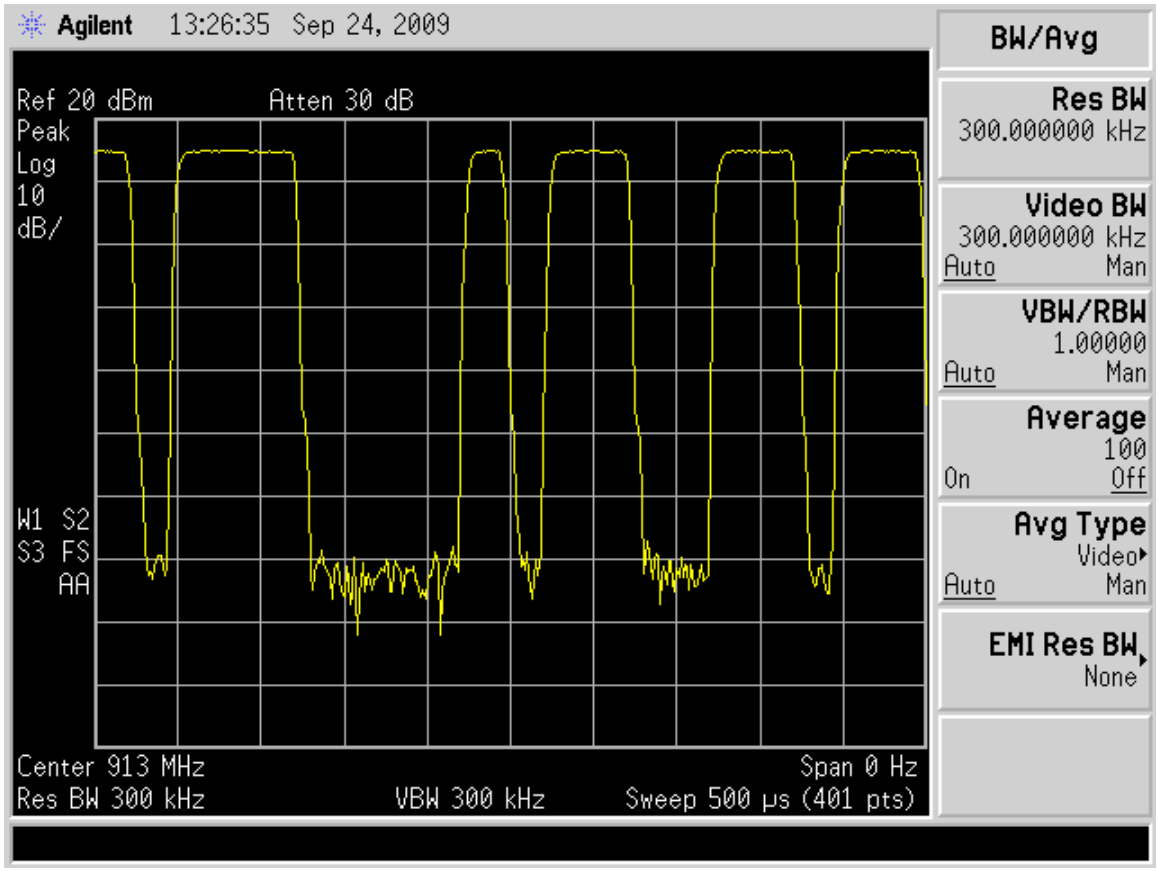

Figure 9. TX Modulation Waveform for OOK

3.2.2. Changing the Modulation Data Rate

The previous example script demonstrated FSK/GFSK/OOK modulation with a data rate of 40 kbps, and a peak deviation of 20 kHz (in the case of FSK and GFSK). It is a simple matter to also change these modulation parameters to obtain a wide variety of data rates and/or deviations.

The data rate is selected by programming the “txdr[15:0]” field in SPI Registers 6Eh and 6Fh. The exact mathematical formula for deriving the value of the “txdr[15:0]” field is shown in the Si443x data sheet. Once again, it is far simpler to use the EZRadioPRO Register Calculator worksheet to perform the calculations; simply enter the desired data rate into the worksheet, and read off the corresponding values to be programmed into SPI Registers 6Eh–6Fh.

The user should be aware that there is one additional bit that affects the programming of the desired data rate. This is the “txdtrtscale” bit (TX Data Rate Scale Factor) bit in SPI Register 70h, bit D5. The value programmed into the “txdr[15:0]” field takes on a different meaning (or scale factor), depending upon the “txdtrtscale” bit. When the “txdtrtscale” bit is set, the maximum data rate that can be programmed by the “txdr[15:0]” field is much less, but can be specified with much finer frequency resolution. This setting is recommended for data rates below 30 kbps and is automatically handled by the EZRadioPRO Register Calculator worksheet.

The traces in Figure 10 were taken for GFSK modulation, 20 kHz peak deviation, for three different data rates obtained with the following script variations:

```
# Set Data Rate = 40 kbps
S2 EE0A
S2 EF3D
S2 F00C
OR
# Set Data Rate = 20 kbps
S2 EEA3
S2 EFD7
S2 F02C
OR
# Set Data Rate = 10 kbps
S2 EE51
S2 EFEC
S2 F02C
```

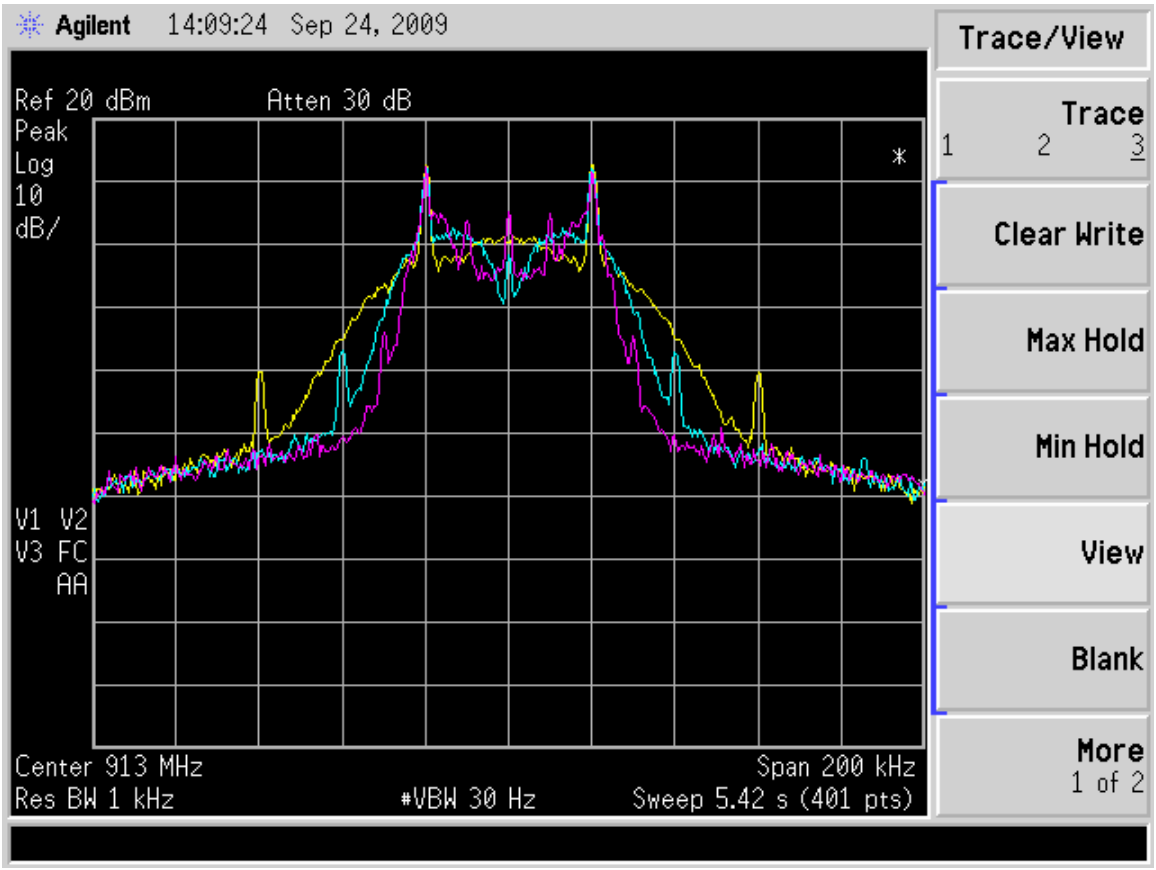

Figure 10. TX Output Spectrum for Different Data Rates

In Figure 10, the yellow trace is for 40 kbps data rate, the blue trace is for 20 kbps, and the purple trace is for 10 kbps. Because the peak deviation remained the same, the modulation index is considerably different for these three scenarios.

3.2.3. Changing the Modulation Deviation (for FSK/GFSK)

The deviation is selected by programming the “fd[8:0]” field in SPI Registers 71h and 72h. As the frequency deviation field is 9-bits in length, it cannot be contained within one single 8-bit register; one bit (the MSB) is located in SPI Register 71h. The exact mathematical formula for deriving the value of the “fd[8:0]” field is shown in the Si443x data sheet. Once again, it is far simpler to use the EZRadioPRO Register Calculator worksheet to perform the calculations; simply enter the desired deviation into the worksheet, and read off the corresponding values to be programmed into SPI Registers 71h–72h. The user should note that the maximum allowed value of modulation index is h=32. Thus for any given data rate, there is a maximum deviation value which should not be exceeded.

The traces in Figure 11 were taken for GFSK modulation, 40 kbps data rate, for three different deviations obtained with the following script variations:

```
# Set Deviation to 20 kHz
S2 F220
OR
# Set Deviation to 40 kHz
S2 F220
OR
# Set Deviation to 80 kHz
S2 F280
```

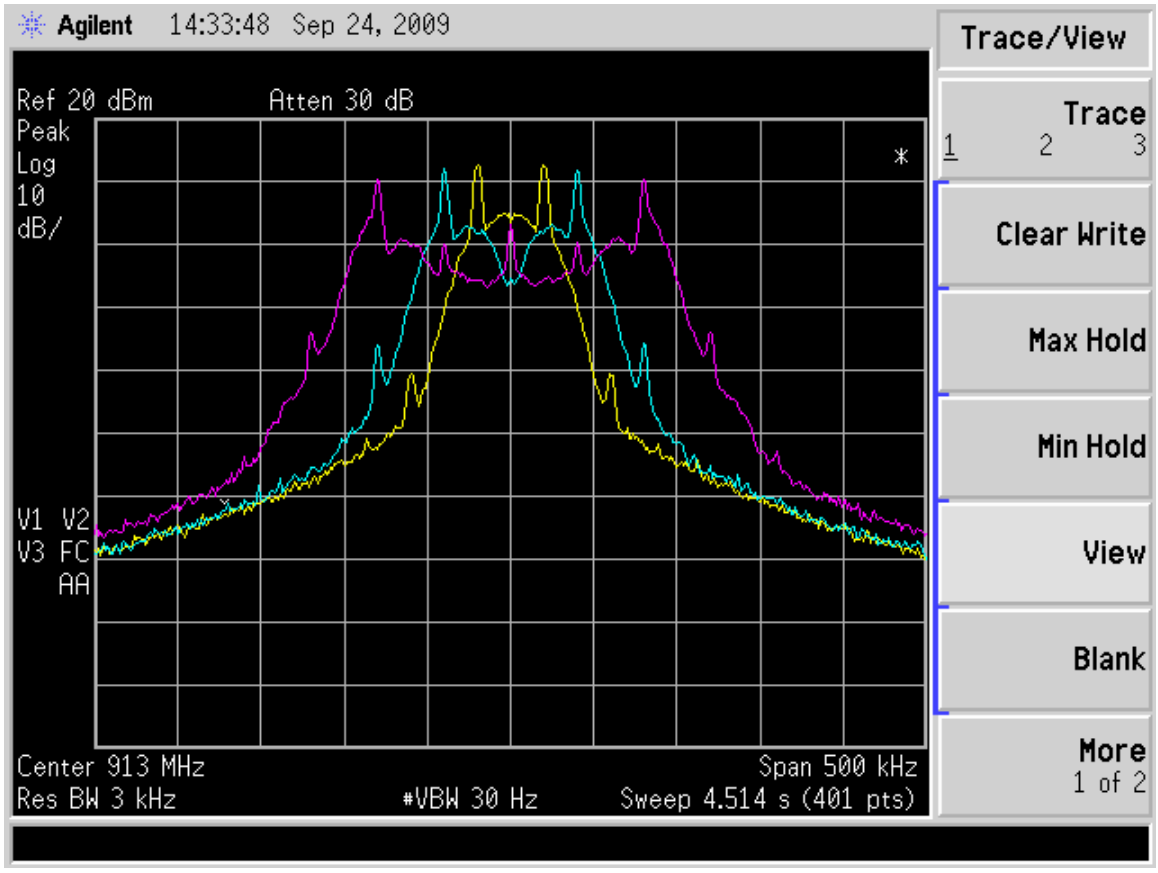

Figure 11. TX Output Spectrum for Different Deviations

In Figure 11, the yellow trace is for 20 kHz peak deviation (h=1), the blue trace is for 40 kHz peak deviation (h=2), and the purple trace is for 80 kHz peak deviation (h=4).

3.3. TX Mode with External Modulation Source

The previous example script selected the Internal PN9 Generator as the source of modulation. This modulation Mode has limited usefulness other than for diagnostic purposes. In this section, we demonstrate how the TXDATA to be transmitted may be obtained from an external source (such as an MCU, Bit Error Rate Analyzer, or laboratory Pattern Generator).

There are three basic methods by which the TXDATA to be transmitted may be obtained from an external source:

- Synchronous Direct Mode
- Asynchronous Direct Mode
- FIFO Mode

The term “direct” refers to a mode in which the signal on a specified TXDATA input pin is sent in real-time (i.e., directly). In FIFO Mode, the data from an external source is stored inside the RFIC to be transmitted at a later point in time.

These modes of external modulation should not be confused with the physical pin(s) by which the TXDATA is supplied to the RFIC. The EZRadioPRO family of chips does not provide one dedicated input pin for the TXDATA input function, but instead allows flexibility in selecting any one of the three GPIO pins or the SDI pin for the TXDATA input function.

3.3.1. TX Synchronous Direct Mode

In TX Synchronous Direct Mode, the RFIC is configured to provide a TXCLK signal as an *output* to the external device that is providing the TXDATA stream. This TXCLK signal is a square wave with a frequency equal to the programmed data rate. The external modulation source (e.g., MCU) must accept this TXCLK signal as an input, and respond by providing one bit of TXDATA back to the RFIC, synchronous with one edge of the TXCLK signal. In this fashion, the rate of the TXDATA input stream from the external source is controlled by the programmed data rate of the RFIC; no TXDATA bits are made available at the input of the RFIC until requested by another cycle of the TXCLK signal. The TXDATA bits supplied by the external source are transmitted directly in real-time (i.e., not stored internally for later transmission).

All modulation types (FSK/GFSK/OOK) are valid in TX Synchronous Direct Mode. (As will be discussed later, there are limits on modulation types in TX Asynchronous Direct Mode.)

The test setup for TX Synchronous Direct Mode is shown in Figure 12. Note that connections are now made to the SMA connectors provided on the Load Board. As discussed above, the EZRadioPRO family of chips provides a wide range of flexibility in selecting which pin of the RFIC is configured for use as the TXDATA input function. However, one of the SMA connectors on the Load Board is conveniently labeled for the DATA_IN function. This particular SMA connector is wired to connect to the GPIO0 pin on the RFIC, and for this reason we deliberately construct the example script to configure GPIO0 as the TXDATA input function. For the same reason, the example script configures GPIO2 for the TXCLK output function, as it is wired to connect to the convenient CLK_OUT SMA connector on the Load Board. (However, the user should understand that this functional mapping may be changed, if desired, in their final application.)

The Load Board contains a MUX between the SMA connected labeled “DATA_IN’” and the GPIO0 pin on the 40-pin connector. This MUX must be commanded to route the signal from the DATA_IN SMA connector to GPIO0 by inclusion of the following hardware command in the script:

```
# Configure Load Board to route DATA_IN SMA connector to GPIO0.
X0
```

Because TX Synchronous Direct Mode requires the use of two GPIO pins (for the TXCLK and TXDATA functions), only one GPIO pin remains available for other purposes. The DKDB2 Single Antenna and the DKDB0 Antenna Diversity RF Test Cards from Silicon Labs require the use of both GPIO1 and GPIO2 for control of their RF switches. As a result, these types of RF Test Cards ***should not*** be used to test TX Synchronous Direct Mode; use of a DKDB1 TX/RX Split RF Test Card is recommended instead.

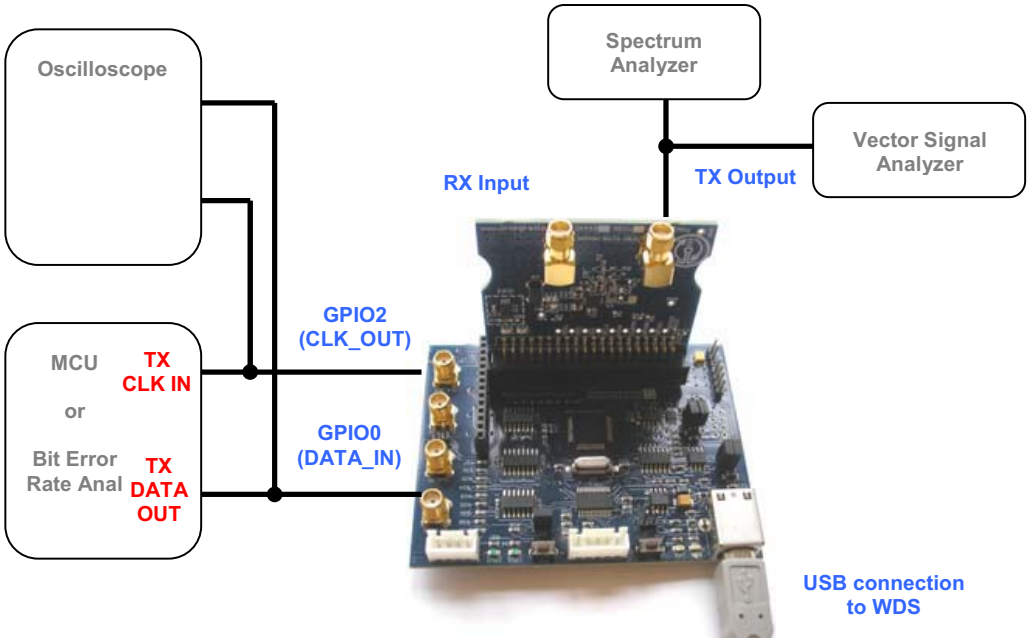

Figure 12. Test Setup for TX Synchronous Direct Mode

The example script shown below will again place the RF Test Card into Transmit Mode at a center channel frequency of 913.0 MHz, with GFSK modulation at a data rate of 40 kbps and a deviation of 20 kHz. However, the source of modulation is now provided by an external source, synchronously clocked by the TXCLK signal from the RFIC. This script is appropriate for use with *only* DKDB1 TX/RX Split RF Test Cards; all other EZRadioPRO RF Test Cards are not recommended for use with this script.

Script Name: TX_913.0MHz_DirectModSync_GFSK_40kbps_20kDev.txt

```
#BATCHNAME TX 913.0MHz SyncDirMod GFSK
# This script is appropriate for use with EZRadioPRO:
#  DKDB1 Split TX/RX RF Test Card
#  Do NOT use with DKDB0 AntDiv or DKDB2 Single Antenna RF Test Card(s)
#    (GPIO pins are not available for control of AntDiv or RF Switch)
# Set SDN pin = LOW
L6
# Set VDD:  1.8V=V54, 2.4V=V7E, 3.0V=V98, 3.3V=VA1, 3.6V=VA9
VA1
# Configure Load Board to route DATA_IN SMA connector to GPIO0.
X0
# Apply Software Reset
S2 8780
# Adjust Crystal for zero freq error (User may need to modify this value)
S2 8971
### Set Desired Transmit Frequency = 913.0 MHz ###
S2 F575
S2 F6A2
S2 F780
# Select the modulation mode for GFSK, Sync Direct Mod via a GPIO pin.
#   Send 'S2 F142' for FSK, Sync Direct Mod via GPIO pin.
S2 F143
### Transmitter Specific Parameters ###
# Set Data Rate = 40kbps
S2 EE0A
S2 EF3D
S2 F00C
# Set Deviation to 20kHz
S2 F220
# Set Tx output power (ED07 = max, ED00 = min)
S2 ED07
### Configure GPIOs ###
# Configure GPIO0 = TXDATA in
#   (GPIO0 is conveniently hard-wired to SMA connector on Load Board)
S2 8B10
# Configure GPIO1 = OFF (not used)
S2 8C1F
# Configure GPIO2 = TXCLK out
#   (GPIO2 is conveniently hard-wired to SMA connector on Load Board)
#   This TXCLK signal should be used to trigger the MCU or data
#   source to synchronously provide another TXDATA bit on the
#   TXDATA input pin (GPIO0 in this example)
S2 8D0F
### Turn Transmitter ON ###
S2 8708
```

The traces in Figure 13 illustrate the TXCLK signal available as an output from the RFIC, and the corresponding TXDATA input stream supplied synchronously by the external source in response to the TXCLK signal.

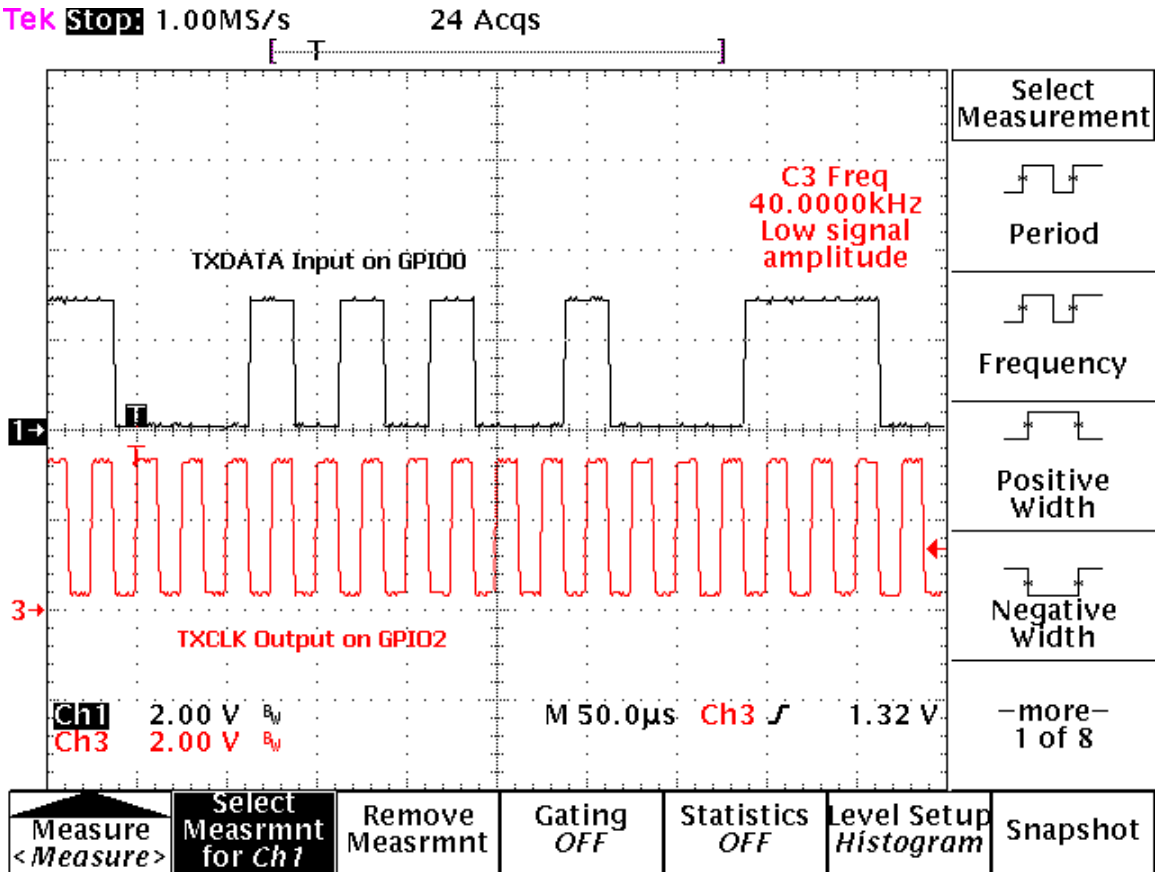

Figure 13. TXDATA and TXCLK in Synchronous Direct Mode

In this example we have connected a bit error rate (BER) analyzer as the external source of the TXDATA stream, and have configured it to provide a PN9 sequence. As the TXDATA pattern remains the same as before (PN9), along with the same data rate (40 kbps) and deviation (20 kHz), the resulting output spectrum also remains the same as previously shown in Figure 8.

3.3.2. TX Asynchronous Direct Mode

In TX Asynchronous Direct Mode, the RFIC no longer controls the data rate of the TXDATA input stream. Instead, the data rate is controlled ***only*** by the external TXDATA source; the RFIC simply accepts the data applied to its TXDATA input pin, at whatever rate it is supplied. This means that there is no longer a need for a TXCLK output signal from the RFIC, as there is no synchronous “handshaking” between the RFIC and the external data source. The TXDATA bits supplied by the external source are transmitted directly in real-time (i.e., not stored internally for later transmission).

It is not necessary to program the data rate parameter when operating in TX Asynchronous Direct Mode. The chip still internally samples the incoming TXDATA stream to determine when edge transitions occur; however, rather than sampling the data at a pre-programmed data rate, the chip now internally samples the incoming TXDATA stream at its maximum possible oversampling rate. This allows the chip to accurately determine the timing of the bit edge transitions without prior knowledge of the data rate. (Of course, it is still necessary to program the desired peak frequency deviation.)

Only FSK and OOK modulation types are valid in TX Asynchronous Direct Mode; GFSK modulation is not available in Asynchronous Mode. This is because the RFIC does not have knowledge of the supplied data rate, and thus cannot determine the appropriate Gaussian lowpass filter function to apply to the incoming data.

The test setup for TX Asynchronous Direct Mode is shown in Figure 14. Again we supply the TXDATA input stream (from the external source) to the RFIC by means of the DATA_IN connector on the Load Board, which is connected

to GPIO0 on the RFIC. GPIO2 is not used in this test setup, as there is no need for a TXCLK signal in this mode. Because neither GPIO1 nor GPIO2 are used for modulation functions in this mode, they are available for control of an RF switch, and thus TX Asynchronous Direct Mode may be tested with either a DKDB1 TX/RX Split or DKDB2 Single Antenna RF Test Card.

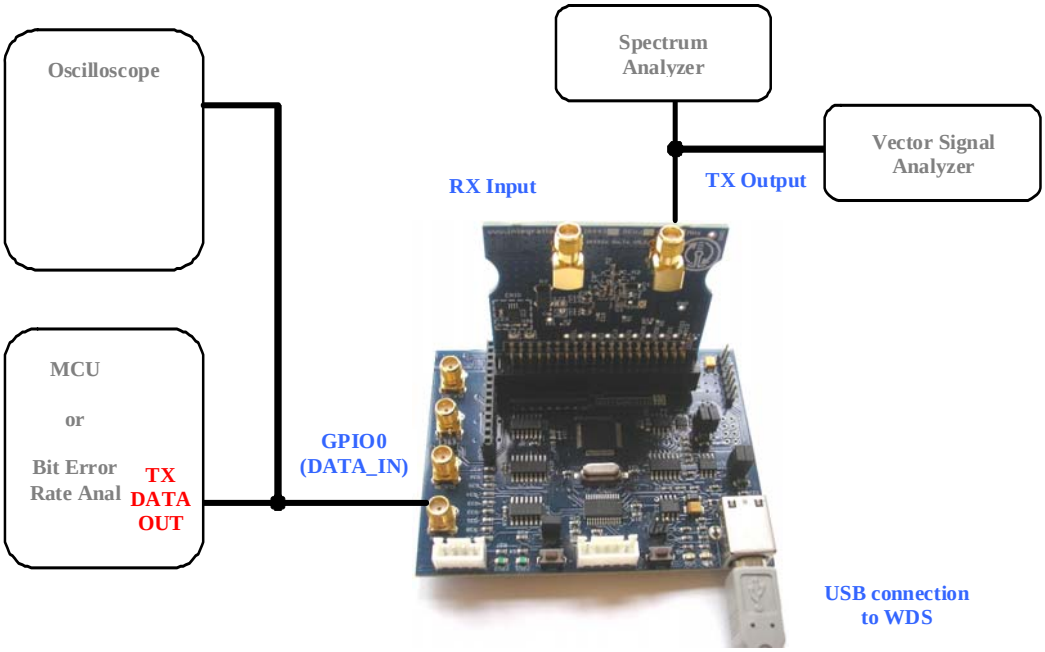

Figure 14. Test Setup for TX Asynchronous Direct Mode

The example script shown below will again place the RF Test Card into Transmit Mode at a center channel frequency of 913.0 MHz, with FSK modulation at a data rate of 40 kbps and a deviation of 20 kHz. The source of modulation is provided by an external source, with the data rate determined solely by the external source. This script is appropriate for use with any EZRadioPRO RF Test Card *except* the DKDB0 Antenna Diversity TRX Test Card.

Script Name: TX_913.0MHz_DirectModAsync_FSK_20kDev.txt

```
#BATCHNAME TX 913.0MHz AsyncDirMod FSK
# This script is appropriate for use with EZRadioPRO:
#  DKDB1 Split TX/RX or DKDB2 Single Antenna RF Test Card(s)
#  Do NOT use with DKDB0 AntDiv RF Test Card
#    (GPIO pins are not programmed for control of AntDiv Switch)
# Set SDN pin = LOW
L6
# Set VDD:  1.8V=V54, 2.4V=V7E, 3.0V=V98, 3.3V=VA1, 3.6V=VA9
VA1
# Configure Load Board to route DATA_IN SMA connector to GPIO0.
X0
# Apply Software Reset
S2 8780
# Adjust Crystal for zero freq error (User may need to modify this value)
S2 8971
### Set Desired Transmit Frequency = 913.0 MHz ###
S2 F575
S2 F6A2
S2 F780
# Select the modulation mode for FSK, Async Direct Mode via a GPIO pin.
#   (Note that GFSK is not available in Async Direct Mode.)
S2 F102
### Transmitter Specific Parameters ###
# Data Rate is not programmed in Async Direct Mode, but
#   is determined by rate of data applied to TXDATA input pin.
# Set Deviation to 20kHz
S2 F220
# Set Tx output power (ED07 = max, ED00 = min)
S2 ED07
### Configure GPIOs ###
# Configure GPIO0 = TXDATA in
#   (GPIO0 is conveniently hard-wired to SMA connector on Load Board)
S2 8B10
# Configure GPIO1 = TX State (to control RF Switch, if any)
S2 8C12
# Configure GPIO2 = RX State (to control RF Switch, if any)
S2 8D15
### Turn Transmitter ON ###
S2 8708
```

If the external data source (e.g., BER Analyzer or MCU) is configured to provide the same TXDATA pattern as before (PN9), at the same data rate (40 kbps), the resulting output spectrum will also remain the same as previously shown in Figure 8 (for FSK).

3.3.3. TX Packet FIFO Mode

In TX Packet FIFO Mode, the TXDATA to be transmitted is not applied to a physical input pin, but instead is programmed (byte by byte) into the chip's FIFO registers over the SPI bus. If Automatic Packet Handling is enabled (SPI Register 30h bit D3), these stored bytes of data are retrieved from internal FIFO memory and "packaged" together with Preamble bytes, Sync word(s), Header bytes, and CRC Checksums to create a complete packet structure. If Automatic Packet Handling is disabled, the packet structure to be transmitted consists ***only*** of what the user has programmed into TX FIFO memory. The entire packet is then transmitted at the programmed data rate and deviation.

Because the data stored in the FIFO must be programmed from an external source, the TX Packet FIFO Mode of operation may still be considered as "external modulation", in a certain sense. However, this mode differs from Direct Mode in that the programmed data is not sent immediately, but is instead stored internally for transmission at a later point in time. This is a primary advantage of TX Packet FIFO Mode; the data to be transmitted may be programmed at any time (including SLEEP Mode) without the need for synchronous or real-time coordination with the transmit burst.

The example script shown below will again place the RF Test Card into Transmit Mode at a center channel frequency of 913.0 MHz, with GFSK modulation at a data rate of 40 kbps and a deviation of 20 kHz. In this mode, however, the TXDATA is programmed into the FIFO (SPI Register FFh). Because Automatic Packet Handling has been enabled (SPI Register 30h bit D3), there a number of additional script commands now required to configure such Packet Handling parameters as Preamble length, Sync word, Header bytes, etc. This script is appropriate for use with any EZRadioPRO RF Test Card *except* the DKDB0 Antenna Diversity TRX Test Card. The test setup of Figure 3 may be used for this test; as the TXDATA is programmed into the RFIC over the SPI interface, there is no need for additional connections (i.e., GPIO lines) to an external data source.

Script Name: TX_913.0MHz_PacketFIFO_GFSK_40kbps_20kDev.txt

```
#BATCHNAME TX 913.0MHz FIFO GFSK
# This script is appropriate for use with EZRadioPRO:
#  DKDB1 Split TX/RX or DKDB2 Single Antenna RF Test Card(s)
#  Do NOT use with DKDB0 AntDiv RF Test Card
#    (GPIO pins are not programmed for control of AntDiv Switch)
# Set SDN Pin 20 = LOW
L6
# Set VDD:  1.8V=V54, 2.4V=V7E, 3.0V=V98, 3.3V=VA1, 3.6V=VA9
VA1
# Apply Software Reset
S2 8780
# Adjust Crystal for zero freq error (User may need to modify this value)
S2 8971
### Set Desired Transmit Frequency = 913.0 MHz ###
S2 F575
S2 F6A2
S2 F780
# Select the modulation mode for GFSK, FIFO-mode.
#   Send 'S2 F122' for FSK FIFO-mode.
S2 F123
### Transmitter Specific Parameters ###
# Set Data Rate = 40kbps
S2 EE0A
S2 EF3D
S2 F00C
# Set Deviation to 20kHz
S2 F220
# Enable automatic TX Packet Handling, disable CRC.
S2 B008
```

```
# Set Fixed Packet Length, No Header, SYNC word = 2 Bytes:  Sync Word 3 & 2
S2 B30A
# Set Preamble Length = 16 nibbles (64 bits) of '1010' pattern.
S2 B410
# Set Sync Word 2-bytes value (optional)
S2 B62D
S2 B7D4
# Set Packet Length = 4 bytes
S2 BE04
# Write data to the FIFO (4 example byes shown here)
S2 FF0F
S2 FFAA
S2 FFCC
S2 FFAA
# Set Tx output power (ED07 = max, ED00 = min)
S2 ED07
### Configure GPIOs ###
# Configure GPIO0 = OFF (not used)
S2 8B1F
# Configure GPIO1 = TX State (to control RF Switch, if any)
S2 8C12
# Configure GPIO2 = RX State (to control RF Switch, if any)
S2 8D15
### Turn Transmitter ON ###
S2 8708
# Note that running this script will result in sending only ONE TX packet,
#   after which the RFIC automatically turns off.  Subsequent TX packets
#   may be transmitted by sending only a 'S2 8708' command; there is no
#   need to re-send this entire script.
```

The user should be aware that when the RFIC is placed into automatic TX Packet Handling Mode, only one packet is transmitted, after which time the RFIC turns itself off. Attempting to capture and observe one such brief packet transmission on a spectrum analyzer may be a non-trivial task. For this reason, it is often more effective to run the above example script once (to configure the RFIC for TX FIFO Packet Mode), but to then repeatedly re-transmit this same packet by looping the following one-line script:

```
### Turn Transmitter ON ###
S2 8708
```

The resulting TX output spectrum may best be observed with the spectrum analyzer placed in MAX HOLD Mode. This is shown in Figure 15. Because the transmitted data is no longer pseudo-random, but instead now contains repetitive patterns, the spectrum contains more discrete tones.

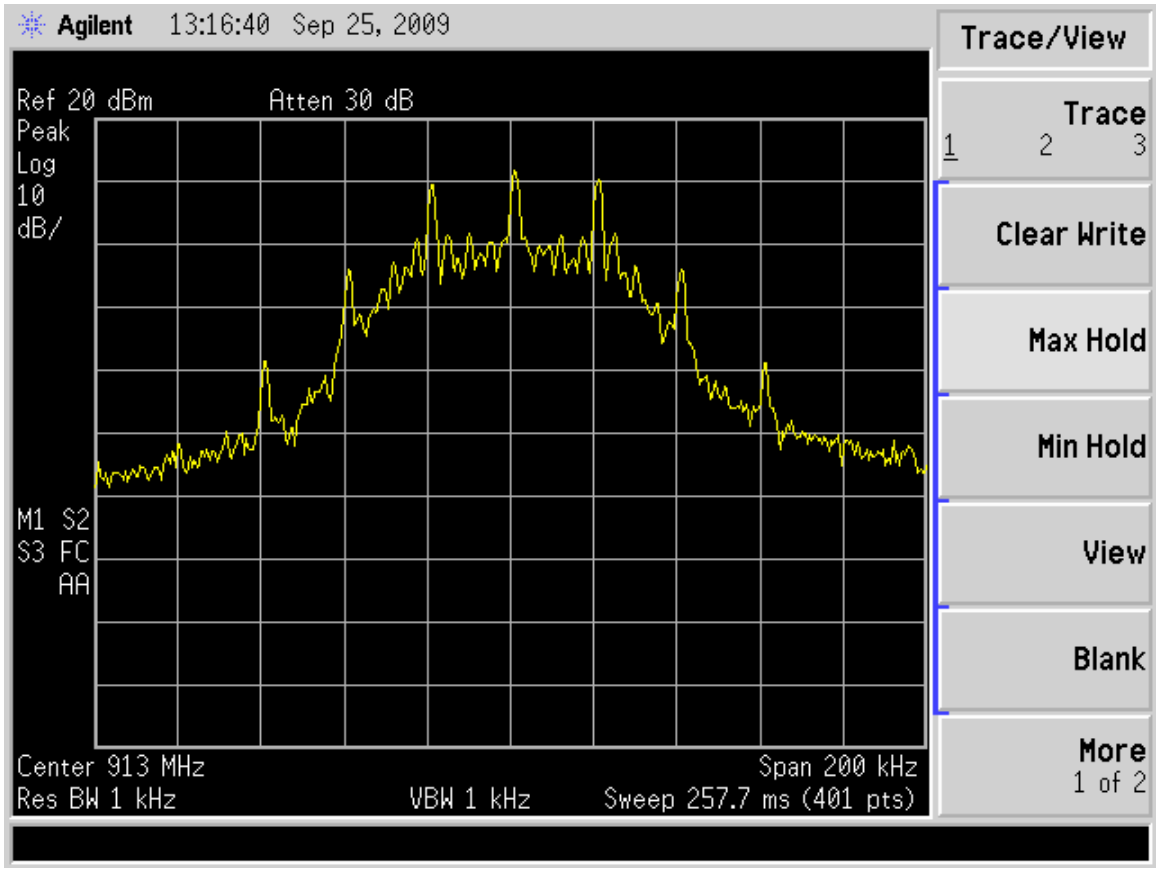

Figure 15. TX Output Spectrum in Packet FIFO Mode (MAX HOLD)

If the user has access to a vector signal analyzer (or other test equipment that can perform FM demodulation), it is also possible to view the modulation signal in the time domain. This is shown in Figure 16. The “packet” nature of this mode may be clearly seen. Upon closer inspection, the individual bits corresponding to fields in the packet may also be identified (such as the Preamble, the Sync word, Payload bytes, etc.) These are conveniently labeled in Figure 16.

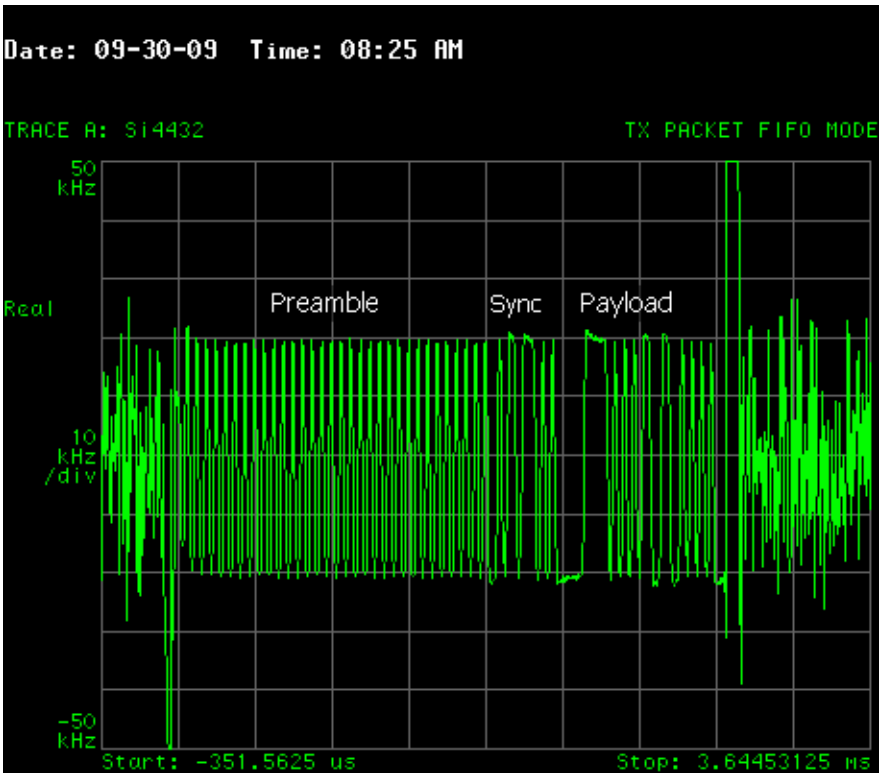

Figure 16. TX Modulation Waveform in Packet FIFO Mode

If the user is testing with a DKDB1 TX/RX Split RF Test Card (which does not contain an RF switch and thus has all three of the GPIO pins available for user-defined purposes), it is possible to reconfigure the GPIO pins to output a variety of interesting internal digital signals related to TX Packet timing.

```
### Configure GPIOs for Digital Test Bus signals ###
S2 8B0B
S2 8C0B
S2 8D0B
# Set Digital Test Bus = 29h = 41d = ‘data_start’, ‘tx_out’, ‘pk_sent’
S2 D129
```

With the above script modification, the user may observe the internal Packet Handler signals shown in Figure 17. GPIO0 as an output signal is not available at an SMA connector on the Load Board; it is only available at the SMA connector as an input signal. Therefore, it will be necessary for the user to observe GPIO0 by connecting a scope probe directly to the GPIO0 test point on the RF Test Card.

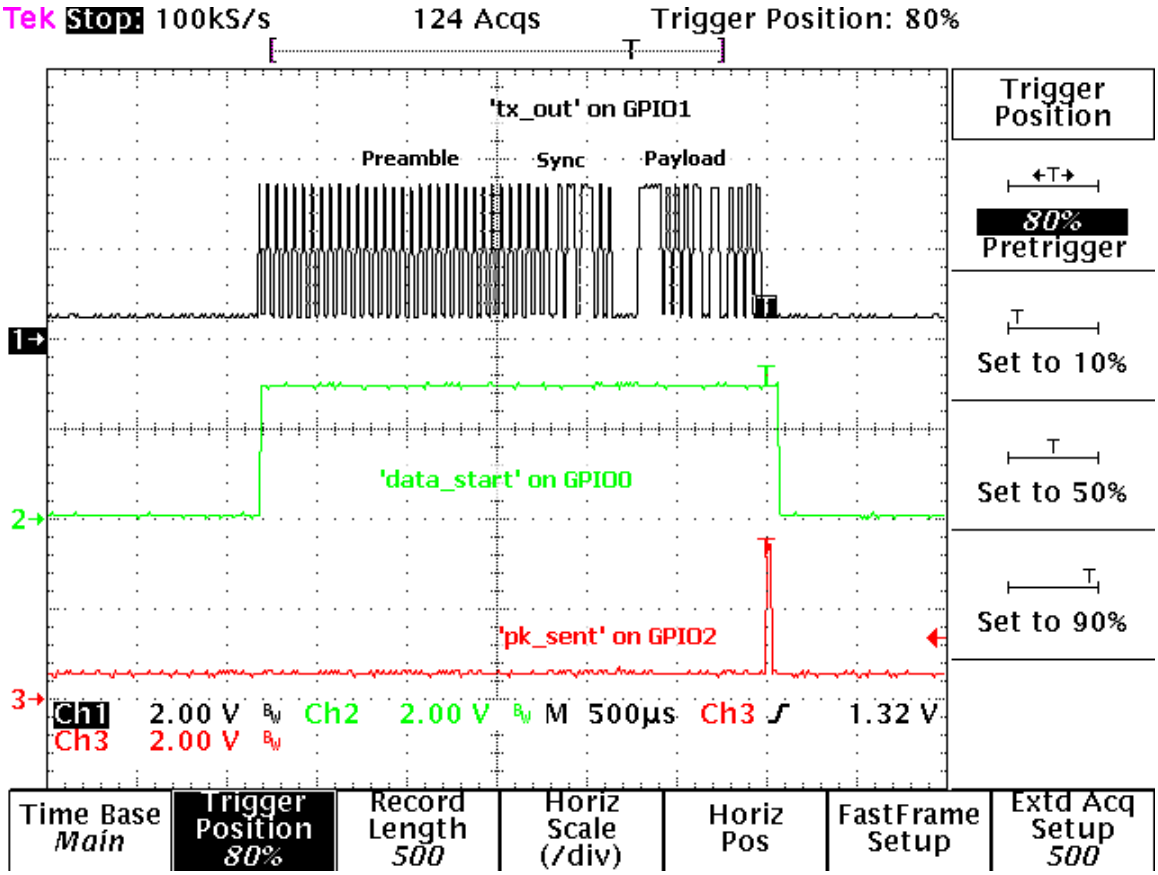

Figure 17. Internal Packet Handler Signals in TX Packet FIFO Mode

4. Receiver Functional Tests

4.1. RX CW Direct Mode (BER Sensitivity Measurements)

The most basic RX functional test that may be performed with an EZRadioPRO RF Test Card is to measure the bit error rate (BER) for a given RX input power level. Although BER and packet error rate (PER) tests may be performed in Packet Mode, it is far simpler to start with BER measurements in Continuous RX Mode, using only a sinewave or a repetitive PN sequence as the test data pattern. Silicon Labs has created an application note "AN432: Si443x RX BER Measurement with Looped PN Sequence" that provides further details regarding measurement of BER using a looped PN sequence. This application note is available for download from the Silicon Labs website.

The test setup for RX CW Direct Mode BER Measurements is shown in Figure 18. As the GPIO1 and GPIO2 pins are used for the RXDATA and RXCLK output functions (respectively), they are unavailable for other purposes. The DKDB2 Single Antenna and the DKDB0 Antenna Diversity RF Test Cards from Silicon Labs require the use of both GPIO1 and GPIO2 for control of their RF switches. As a result, these types of RF Test Cards ***should not*** be used to test RX Direct Mode BER Sensitivity; use of a DKDB1 TX/RX Split RF Test Card is recommended instead.

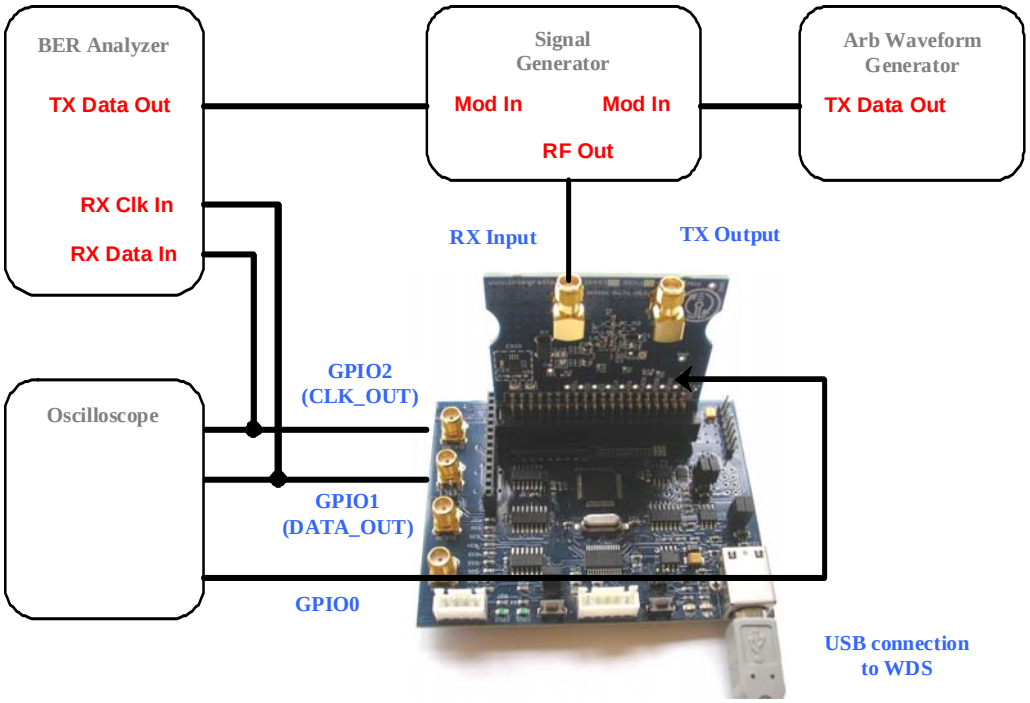

Figure 18. Test Setup for RX BER Sensitivity Measurements

4.1.1. RX CW Direct Mode BER w/Sinewave Modulation

In RX CW Direct Mode, the RFIC is operated in Continuous Receive Mode and configured to provide the RXDATA directly (i.e., in real-time) on a specified output pin. The RFIC attempts to synchronize an internal bit clock to the data rate of the received signal. This RXCLK signal is a square wave with a frequency equal to the received data rate, and may also be output on a pin and used to clock the RXDATA into an external device (e.g., MCU or BER Analyzer). In this fashion, the received data bits may be transferred in a synchronous fashion. The RXDATA bits received by the chip are output directly in real-time (i.e., not stored internally for later retrieval).

The EZRadioPRO family of chips is optimized for reception of a data structure that contains both a Preamble field and a Sync word field. The RX modem inside the RFIC uses the repetitive "1010" pattern of the Preamble field to acquire bit timing, thus essentially locking the RXCLK signal to the frequency and phase of the RXDATA transitions. It is therefore important that any test data pattern that is used for RX BER measurements contain a preamble-like field.

Fortunately, the simplest “test pattern” that can be constructed is an infinitely-long Preamble. This is easily obtained by modulating a lab RF signal generator with a constant sinewave. Because one cycle of a sinewave is equivalent to two bits (both a 1 and a 0), the frequency of the sinewave used for modulation should be one-half of the programmed data rate of the RFIC (e.g., for the data rate of 40 kbps shown below, the signal generator should be configured for modulation by a sinewave with a frequency of 20 kHz). This test setup has the virtue of simplicity; as most lab RF signal generators have internal FM modulation capability, no other test equipment is absolutely required (other than an oscilloscope to observe the RXDATA signal).

The example script shown below will place the RF Test Card into Continuous Receive Mode at a center channel frequency of 913.0 MHz. The RX modem parameters are optimized to receive a GFSK-modulated signal at a data rate of 40 kbps, 20 kHz peak deviation, with a data pattern equivalent to a continuous preamble (i.e., infinitely-long 1010 pattern). The RX modem parameters used in this script are obtained directly from the EZRadioPRO Register Calculator worksheet. This script is appropriate for use with only DKDB1 TX/RX Split RF Test Cards; all other EZRadioPRO RF Test Cards are not recommended for use with this script.

Script Name: RX_913.0MHz_BER_ContPream_GFSK_40kbps_20kDev.txt

```
#BATCHNAME RX 913.0 MHz BER Continuous Preamble 40kbps
# This script is appropriate for use with EZRadioPRO:
#  DKDB1 Split TX/RX RF Test Card
#  Do NOT use with DKDB0 AntDiv or DKDB2 Single Antenna RF Test Card(s)
#    (GPIO pins are not available for control of AntDiv or RF Switch)
# Set SDN Pin 20 = LOW
L6
# Set VDD:  1.8V=V54, 2.4V=V7E, 3.0V=V98, 3.3V=VA1, 3.6V=VA9
VA1
# Apply Software Reset
S2 8780
# Adjust Crystal for zero freq error (User may need to modify this value)
S2 8971
### Set Desired Receive Frequency = 913.0 MHz ###
S2 F575
S2 F6A2
S2 F780
# Select TR Data Clock out via GPIO, Direct Mode GFSK.
S2 F143
### Set RX Packet Parameters ###
# Disable automatic RX Packet Handling
S2 B000
# Set Preamble Length = 4 nibbles (16 bits)
S2 B404
# Set Preamble Detection Threshold = 3 nibbles (12 bits)
S2 B518
### Set RX Modem Parameters (from Excel Register Calculator) ###
# 40KBPS data rate, 20kHz dev, Mod Index=1, BW=83.2kHz
S2 9C02
S2 A064
S2 A101
S2 A247
S2 A3AE
S2 A402
S2 A591
S2 9D40
### Configure GPIOs ###
# Set GPIO0 = OFF (not used)
S2 8B1F
```

```
# Set GPIO1 = RXDATA output (with increased GPIO drive)
#   (GPIO1 is conveniently hard-wired to SMA connector on Load Board)
S2 8C94
# Configure GPIO2 = RXCLK out (with increased GPIO drive)
#   (GPIO2 is conveniently hard-wired to SMA connector on Load Board)
#   This RXCLK signal may be used to clock the RXDATA into a BER Analyzer.
S2 8D8F
### Turn Receiver ON ###
S2 8704
```

With the above example script, the user may observe the RXDATA and RXCLK signals as shown in Figure 19. The RX Sensitivity performance of a typical DKDB1 TX/RX Split RF Test Card is about –108 dBm for BER = 1E-3 (0.1%) for data rate = 40 kbps and peak deviation = 20 kHz. If the user does not have a BER Analyzer available, this level of BER corresponds to visually observing ***occasional*** bit errors in the displayed pattern on the oscilloscope.

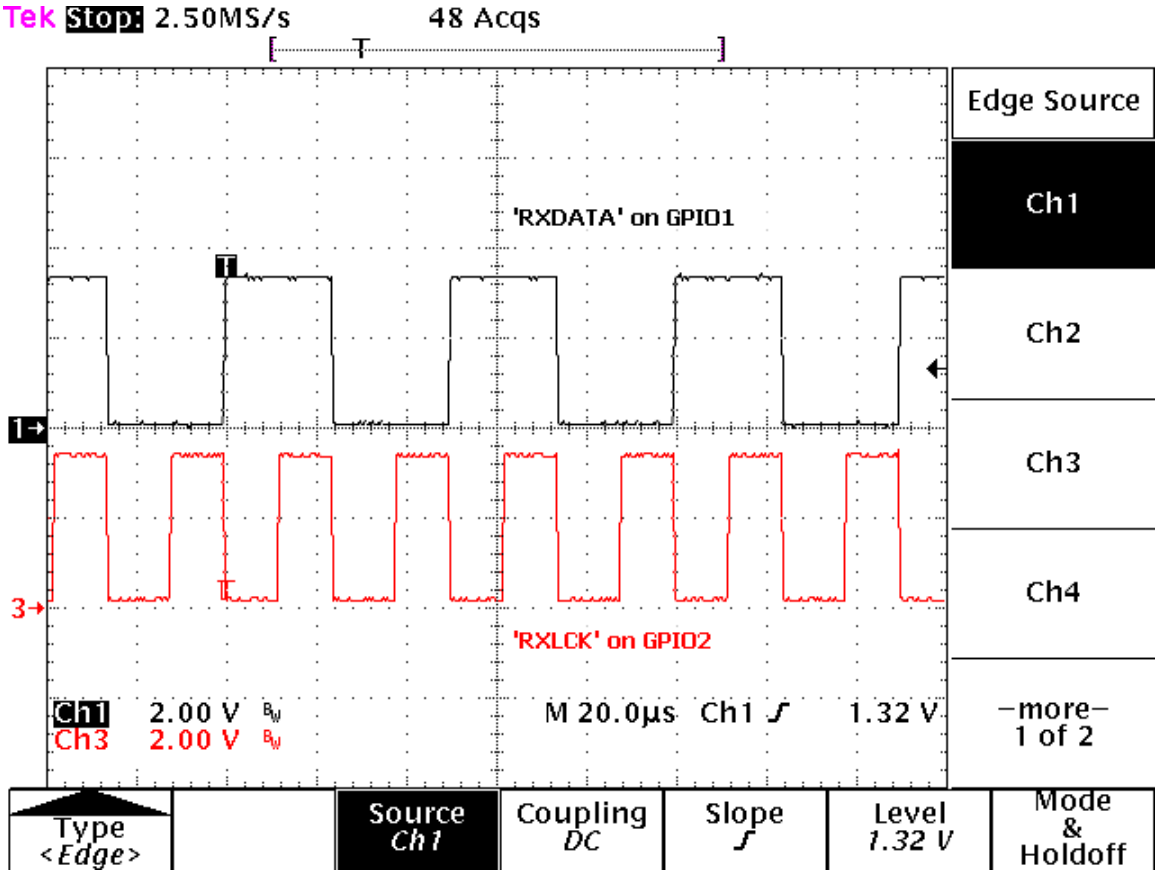

Figure 19. RXDATA/RXCLK Signals for Continuous Preamble

4.1.2. RX CW Direct Mode BER with PN9 Sequence Modulation

A test pattern that is more commonly used for BER measurements is a pseudo-random or PN sequence (e.g., PN9, PN11, PN15, etc.). It is possible to configure the RF Test Card for BER measurement using only a looped (i.e., repetitive) PN sequence.

The test setup of Figure 18 is also appropriate for this measurement. If the BER Analyzer is not capable of providing a Gaussian-filtered PN sequence suitable for modulating the RF signal generator, it may be necessary to create the modulating signal using a separate Arbitrary Waveform Generator (ArbGen). Alternatively, some newer RF signal generators may have built-in options allowing for the internal generation of customized modulation

signals such as FSK/GFSK PN-sequences. One such example is the Agilent ESG-D E443x family of Digital Signal Generators with Option UN8 I/Q Baseband Generator. Some RF signal generators provide not only generation of the FSK/GFSK modulated PN sequence, but also provide BER analysis of the RXDATA stream as well (e.g., Agilent's Option UN7 Internal Bit Error Rate Analyzer).

The example script shown below will place the RF Test Card into Continuous Receive Mode at a center channel frequency of 913.0 MHz. The RX modem parameters are optimized to receive a GFSK-modulated signal at a data rate of 40 kbps, 20 kHz peak deviation, with a looped PN9 sequence as the data pattern. The RX modem parameters used in this script are again obtained directly from the EZRadioPRO Register Calculator worksheet. This script is appropriate for use with *only* DKDB1 TX/RX Split RF Test Cards; all other EZRadioPRO RF Test Cards are not recommended for use with this script.

Script Name: RX_913.0MHz_BER_PN9_GFSK_40kbps_20kDev.txt

```
#BATCHNAME RX 913.0 MHz PN9 BER 40kbps
# This script is appropriate for use with EZRadioPRO:
#  DKDB1 Split TX/RX RF Test Card
#  Do NOT use with DKDB0 AntDiv or DKDB2 Single Antenna RF Test Card(s)
#    (GPIO pins are not available for control of AntDiv or RF Switch)
# Set SDN Pin 20 = LOW
L6
# Set VDD:  1.8V=V54, 2.4V=V7E, 3.0V=V98, 3.3V=VA1, 3.6V=VA9
VA1
# Apply Software Reset
S2 8780
# Adjust Crystal for zero freq error (User may need to modify this value)
S2 8971
### Set Desired Receive Frequency = 913.0 MHz ###
S2 F575
S2 F6A2
S2 F780
# Select TR Data Clock out via GPIO, Direct Mode GFSK.
S2 F143
### Set RX Packet Parameters ###
# Disable automatic RX Packet Handling
S2 B000
# Set Preamble Length = 4 nibbles (16 bits)
S2 B404
# Set Preamble Detection Threshold = 2 nibbles (8 bits)
#   (This may need to change depending upon PN sequence)
S2 B510
# Set SYNC word length = 3 Bytes:  Sync Word 3 & 2 & 1
S2 B304
# The SYNC word values below may need to be modified for each PN sequence.
S2 B602
S2 B795
S2 B8E5
### Set RX Modem Parameters (from Excel Register Calculator) ###
# 40KBPS data rate, 20kHz dev, Mod Index=1, BW=83.2kHz
S2 9C02
S2 A064
S2 A101
S2 A247
S2 A3AE
S2 A402
S2 A591
```

```
# Disable AFC (the length of "preamble" pattern found within a
#   PN sequence is not sufficient in length to settle AFC.)
S2 9D00
### Configure GPIOs ###
# Set GPIO0 = OFF (not used)
S2 8B1F
# Set GPIO1 = RXDATA output (with increased GPIO drive)
#   (GPIO1 is conveniently hard-wired to SMA connector on Load Board)
S2 8C94
# Configure GPIO2 = RXCLK out (with increased GPIO drive)
#   (GPIO2 is conveniently hard-wired to SMA connector on Load Board)
#   This RXCLK signal may be used to clock the RXDATA into a BER Analyzer.
S2 8D8F
### Turn Receiver ON ###
S2 8704
```

The EZRadioPRO family of chips is optimized for reception of a data structure that contains both a Preamble field and a Sync word field. The RX modem inside the RFIC uses the repetitive “1010” pattern of the Preamble field to acquire bit timing, thus essentially locking the RXCLK signal to the frequency and phase of the RXDATA transitions. This initial bit clock recovery (BCR) process is performed during the Preamble with a very fast time constant in the bit clock tracking loop, with a corresponding increase in clock jitter. Once the Preamble is acquired, the RX modem switches to a much slower time constant (with reduced clock jitter) for tracking the remainder of the data packet. However, the RFIC will ***only*** remain in Slow Time Constant Mode if it also successfully detects the Sync word; if the Sync word is not found, the chip will return to Fast Time Constant Mode and attempt to reacquire Preamble timing.

This means that for optimum RX BER sensitivity performance, it is necessary for the chip to successfully recognize both a Preamble as well as a Sync word. If the RX modem does not successfully detect both Preamble and Sync, the chip will spend an excessive amount of time in Fast Time Constant Tracking Mode, with a corresponding increase in RXCLK jitter and thus degradation in BER performance.

Within any given PN sequence, there is at least some short sequence of a repetitive “1010” pattern. For a PN9 sequence, we can find at least eight bits of a “1010” pattern in a row. Within a PN15 sequence, this increases to at least sixteen bits of a “1010” pattern in a row. In the example script above (created for use with a PN9 sequence), we set the Preamble Detection Threshold field preath[4:0] in SPI Register 35h for a value of preath[4:0] = 2 nibbles = 8 bits.

```
# Set Preamble Detection Threshold = 2 nibbles (8 bits)
S2 B510
```

After successful detection of this very-short Preamble, the RX modem will begin to search for a Sync word. This script defines the Sync word as the exact same sequence of 24 bits (3 bytes) that immediately follows the 8 bits of “preamble” found within the PN9 sequence. (This bit pattern for the Sync word may be found by simply visually inspecting the PN sequence on the oscilloscope.) Thus the RX modem will also detect the Sync word and will remain in Slow Time Constant Tracking Mode, resulting in minimum RXCLK jitter and optimum BER performance.

```
# The SYNC word values below may need to be modified for each PN sequence.
S2 B602
S2 B795
S2 B8E5
```

With the above example script, the user may observe the RXDATA and RXCLK signals as shown in Figure 20. The RX Sensitivity performance of a typical DKDB1 TX/RX Split RF Test Card is again about –108 dBm for BER = 1E-3 (0.1%) for data rate = 40 kbps and peak deviation = 20 kHz. If the user does not have a BER Analyzer available, this level of BER corresponds to visually observing occasional bit errors in the displayed pattern on the oscilloscope. Automatic Gain Control (AGC) remains enabled in this script, allowing the user to vary the RX signal level up and down during the test with the RFIC automatically adjusting its gain accordingly.

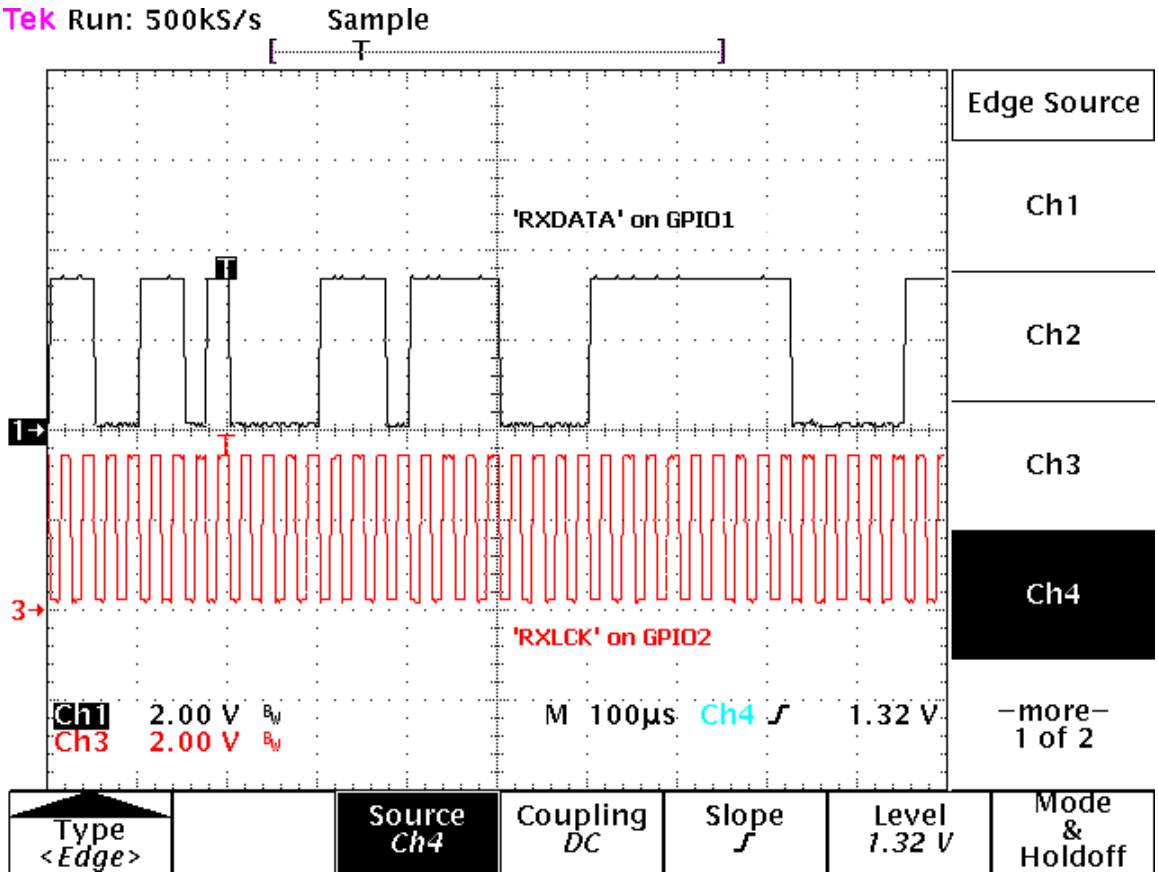

Figure 20. RXDATA / RXCLK Signals for PN9 Sequence

The user may note that AFC has been disabled in the above example script, with the following command:

```
# Disable AFC
S2 9D00
```

This is because the short number of “preamble” bits that may be found within a PN sequence is not sufficient in length to settle AFC.

4.1.3. RX CW Direct Mode BER with PN11 Sequence Modulation

The exact same test setup and test concept may be used to measure RX BER sensitivity with a continuously-looped PN11 sequence (instead of a PN9 sequence). The example script remains the same, except for the commands which define the Preamble Detection Threshold and the value of the Sync word. The pertinent excerpt of the modified script is shown below.

Script Name: RX_913.0MHz_BER_PN11_GFSK_40kbps_20kDev.txt

```
…
# Set Preamble Length = 4 nibbles (16 bits)
S2 B404
# Set Preamble Detection Threshold = 3 nibbles (12 bits)
S2 B518
# Set SYNC word length = 3 Bytes:  Sync Word 3 & 2 & 1
S2 B304
# The SYNC word values below may need to be modified for each PN sequence.
S2 B600
S2 B740
S2 B828
…
```

4.1.4. RX CW Direct Mode BER with PN15 Sequence Modulation

The exact same test setup and test concept may be used to measure RX BER sensitivity with a continuously-looped PN15 sequence (instead of a PN9 or PN11 sequence). The example script remains the same, except for the commands which define the Preamble Detection Threshold, Sync word length, and the value of the Sync word. The pertinent excerpt of the modified script is shown below.

Script Name: RX_913.0MHz_BER_PN15_GFSK_40kbps_20kDev.txt

```
…
# Set Preamble Length = 5 nibbles (20 bits)
S2 B405
# Set Preamble Detection Threshold = 4 nibbles (16 bits)
S2 B520
# Set SYNC word length = 4 Bytes:  Sync Word 3 & 2 & 1 & 0
S2 B306
# The SYNC word values below may need to be modified for each PN sequence.
S2 B600
S2 B703
S2 B8FF
S2 B9F7
…
```

4.2. RX Packet Direct Mode

In RX Packet Direct Mode, the RFIC is configured to receive a packet-like data structure (as opposed to a PN sequence), and to provide the RXDATA directly (i.e., in real-time) on a specified output pin. The data structure normally consists of a Preamble, a Sync word, and a data Payload. RX Packet Direct Mode differs from RX Packet FIFO Mode (discussed in the next section) in that the data bytes are not stored internally in FIFO register memory. Furthermore, automatic packet handling is disabled in Direct Mode; all bytes following the Sync word are considered to be data bytes and are simply output directly without further packet handling (such as checking for Header bytes, packet length count, CRC checksum calculations, etc.). Finally, in Packet FIFO Mode the RFIC turns off automatically after reception of the packet; in Packet Direct Mode the microcontroller must recognize when the packet has been received and must turn off the RFIC by SPI command. In this sense, there is no pre-defined “end” to the packet reception in this mode; the receiver turns off only when the microcontroller tells it to turn off.

The example script shown below will place the RF Test Card into Packet Direct Receive Mode at a center channel frequency of 913.0 MHz. The RX modem parameters are optimized to receive a GFSK-modulated signal at a data rate of 40 kbps with 20 kHz peak deviation. The RX packet parameters are configured to expect a packet structure

with a 64-bit Preamble, a 2-byte Sync word with a value of 2DD4h, and subsequent data bytes. The RX modem parameters used in this script are obtained directly from the EZRadioPRO Register Calculator worksheet. This script is appropriate for use with only DKDB1 TX/RX Split RF Test Cards; all other EZRadioPRO RF Test Cards are not recommended for use with this script.

Script Name: RX_913.0MHz_PacketDirect_GFSK_40kbps_20kDev.txt

```
#BATCHNAME RX 913.0 MHz Direct Mode 40kbps
# This script is appropriate for use with EZRadioPRO:
#  DKDB1 Split TX/RX RF Test Card
#  Do NOT use with DKDB0 AntDiv or DKDB2 Single Antenna RF Test Card(s)
#    (GPIO pins are not available for control of AntDiv or RF Switch)
# Set SDN Pin 20 = LOW
L6
# Set VDD:  1.8V=V54, 2.4V=V7E, 3.0V=V98, 3.3V=VA1, 3.6V=VA9
VA1
# Apply Software Reset
S2 8780
# Adjust Crystal for zero freq error (User may need to modify this value)
S2 8971
### Set Desired Receive Frequency = 913.0 MHz ###
S2 F575
S2 F6A2
S2 F780
# Select TR Data Clock out via GPIO, Direct Mode, GFSK.
S2 F143
### Set RX Packet Parameters ###
# Disable automatic RX Packet Handling, Disable CRC
S2 B000
# Set Preamble Length = 16 nibbles (64 bits)
S2 B410
# Set Preamble Detection Threshold = 5 nibbles (20 bits)
S2 B528
# Set Fixed Packet Length, No Header, SYNC word = 2 Bytes:  Sync Word 3 & 2
S2 B30A
# Set SYNC Word = 2DD4h (as an example)
S2 B62D
S2 B7D4
### Set RX Modem Parameters (from Excel Register Calculator) ###
# 40KBPS data rate, 20kHz dev, Mod Index=1, BW=83.2kHz, AFC Enabled
S2 9C02
S2 A064
S2 A101
S2 A247
S2 A3AE
S2 A402
S2 A591
S2 9D40
S2 9E0A
# 'S2 F218' for Si4432_V2, 'S2 AA1E' for Si4432_Bx
S2 F218
### Configure GPIOs ###
# Set GPIO0 = PREAMBLE_VALID output
#   (GPIO0 is not available as an output on Load Board SMA connector.  This
#   signal must be measured at the GPIO0 test point on the RF Test Card.)
```

```
S2 8B19
# Set GPIO1 = RXDATA output (with increased GPIO drive)
#   (GPIO1 is conveniently hard-wired to SMA connector on Load Board)
S2 8C94
# Configure GPIO2 = SYNC_OK output
#   (GPIO2 is conveniently hard-wired to SMA connector on Load Board)
S2 8D1B
### Turn Receiver ON ###
S2 8704
# Note that in Direct Mode, the RFIC does not automatically turn
#   off after reception of the packet, but instead remains in RX mode.
```

While many BER analyzers or arbitrary waveform generators may be capable of generating a PN sequence with the push of a single button, generation of the Preamble+Sync+Payload packet structure discussed above may require more effort. Possible methods of generation include the following:

- Manually programming an ArbGen or BER Analyzer with this required packet structure
- Using another Si443x RF Test Card in TX Packet FIFO Mode as the test signal source

It is a relatively complex task to measure BER when receiving packets (measurement of packet error rate or PER is much simpler). This is because the user generally does not wish to calculate BER over the Preamble, Sync, or Header fields, but instead over just the data Payload field. This requires the BER Analyzer or MCU to be able to distinguish between fields in the RXDATA stream, and thus requires advance knowledge of the exact packet structure.

As a result, the above example script is not intended for BER measurement and does not output the RXCLK signal; instead, this script configures the GPIOs to output other interesting signals related to the reception of the packet, such as the PREAMBLE_VALID signal and the SYNC_OK signal. As the signal names imply, these internal digital signals go active when the RX modem detects the successful reception of these respective fields in the packet structure. These signals are shown in Figure 21. Thus the primary purpose of this test example is to illustrate how the RX modem searches for, detects, and processes various fields within the data packet.

As mentioned previously, GPIO0 as an *output* signal is not available at an SMA connector on the Load Board; it will be necessary for the user to observe GPIO0 by connecting a scope probe directly to the GPIO0 test point on the RF Test Card.

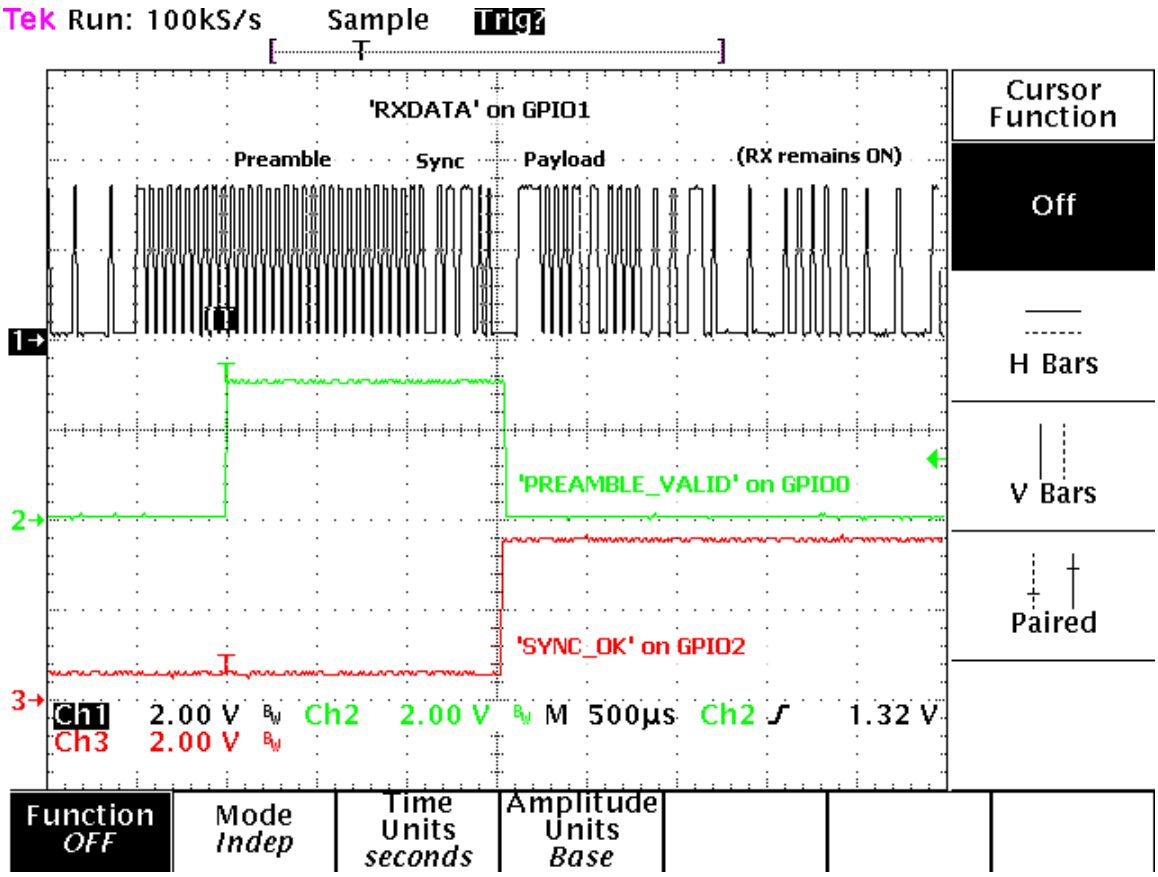

Figure 21. RXDATA and Packet Timing Signals in Direct Mode

4.3. RX Packet FIFO Mode

In RX Packet FIFO Mode, the RFIC is again configured to receive a packet-like data structure, but now stores the data bytes in internal FIFO register memory for later retrieval via the SPI interface. (Direct real-time output of the RXDATA on a physical output pin may *additionally* be obtained by configuring a GPIO pin appropriately, but it is not required.) The packet structure normally consists of a Preamble, a Sync word, optional Header bytes, a data Payload, and an optional CRC Checksum.

In FIFO Mode, an additional feature known as automatic packet handling may be either enabled or disabled. When disabled, all bytes following the Sync word are assumed to be data bytes and are stored in FIFO register memory. When enabled, the automatic packet handler processes the bytes following the Sync word to “strip off” such fields as the Header and CRC Checksum; only the data bytes of the Payload are stored in FIFO register memory. In both cases, neither the bytes of the Preamble or Sync fields are ever stored in FIFO memory; only the bytes considered to be “data” are stored to FIFO memory. (The RXDATA bits comprising the *entire* packet, including the Preamble and Sync fields, may continue to be observed as a direct real-time output on a GPIO pin, if desired.)

One advantage of Packet FIFO Mode is that there is a much more defined structure to the packet. This allows for automatic determination of the end-of-packet, as well as the quality of the packet (e.g., PACKET_VALID indication as determined by CRC checksum). In turn, these signals may be use to generate interrupts to the microcontroller, reducing processing requirements on the MCU. In Packet FIFO Mode the RFIC turns off automatically after reception of a valid packet; in Packet Direct Mode the microcontroller must recognize when the packet has been received and must turn off the RFIC by SPI command.

The example script shown below will place the RF Test Card into Packet FIFO Receive Mode at a center channel frequency of 913.0 MHz. The RX modem parameters are optimized to receive a GFSK-modulated signal at a data rate of 40 kbps with 20 kHz peak deviation. The RX Packet parameters are configured to expect a packet structure with a 64-bit Preamble, a 2-byte Sync word with a value of 2DD4h, no Header, a 4-byte Payload, and no CRC

Checksum. The RX modem parameters used in this script are obtained directly from the EZRadioPRO Register Calculator worksheet. This script is appropriate for use with *only* DKDB1 TX/RX Split RF Test Cards; all other EZRadioPRO RF Test Cards are not recommended for use with this script.

Script Name: RX_913.0MHz_PacketFIFO_GFSK_40kbps_20kDev.txt

```
#BATCHNAME RX 913.0 MHz Packet FIFO 40kbps
# This script is appropriate for use with EZRadioPRO:
#  DKDB1 Split TX/RX RF Test Card
#  Do NOT use with DKDB0 AntDiv or DKDB2 Single Antenna RF Test Card(s)
#    (GPIO pins are not available for control of AntDiv or RF Switch)
# Set SDN Pin 20 = LOW
L6
# Set VDD:  1.8V=V54, 2.4V=V7E, 3.0V=V98, 3.3V=VA1, 3.6V=VA9
VA1
# Apply Software Reset
S2 8780
# Adjust Crystal for zero freq error (User may need to modify this value)
S2 8971
### Set Desired Receive Frequency = 913.0 MHz ###
S2 F575
S2 F6A2
S2 F780
# Select TR Data Clock out via GPIO, FIFO Mode, GFSK.
S2 F163
### Set RX Packet Parameters ###
# Enable automatic RX Packet Handling, Disable CRC
S2 B080
# Set Preamble Length = 16 nibbles (64 bits)
S2 B410
# Set Preamble Detection Threshold = 5 nibbles (20 bits)
S2 B528
# Set Fixed Packet Length, No Header, SYNC word = 2 Bytes:  Sync Word 3 & 2
S2 B30A
# Set SYNC Word = 2DD4h (as an example)
S2 B62D
S2 B7D4
# Set Packet Length = 4 bytes (as an example)
S2 BE04
### Set RX Modem Parameters (from Excel Register Calculator) ###
# 40KBPS data rate, 20kHz dev, Mod Index=1, BW=83.2kHz, AFC Enabled
S2 9C02
S2 A064
S2 A101
S2 A247
S2 A3AE
S2 A402
S2 A591
S2 9D40
S2 9E0A
# 'S2 F218' for Si4432_V2, 'S2 AA1E' for Si4432_Bx
S2 F218
### Configure GPIOs ###
# Set GPIO0 = PREAMBLE_VALID output
#   (GPIO0 is not available as an output on Load Board SMA connector.  This
```

```
#   signal must be measured at the GPIO0 test point on the RF Test Card.)
S2 8B19
# Set GPIO1 = RXDATA output (with increased GPIO drive)
#   (GPIO1 is conveniently hard-wired to SMA connector on Load Board)
S2 8C94
# Configure GPIO2 = SYNC_OK output
#   (GPIO2 is conveniently hard-wired to SMA connector on Load Board)
S2 8D1B
### Turn Receiver ON ###
S2 8704
# Note that running this script will result in receiving only ONE RX packet,
#   after which the RFIC automatically turns off.  Subsequent RX packets
#   may be received by sending only a 'S2 8704' command; there is no
#   need to re-send this entire script.
```

By comparison of this script with that shown for RX Packet Direct Mode, the user will note that the primary difference(s) occur in the following commands:

```
# Select TR Data Clock out via GPIO, FIFO Mode, GFSK.
S2 F163
# Enable automatic RX Packet Handling, Disable CRC
S2 B080
```

Again, the user will need to create an RF test signal with the appropriate packet data structure. Possible methods of generation again include the following:

- Manually programming an ArbGen or BER Analyzer with this required packet structure
- Using another Si443x RF Test Card in TX Packet FIFO Mode as the test signal source

Once again, the above example script is not intended for purposes of BER measurement; instead, this script configures the GPIOs to output other interesting signals related to the reception of the packet, such as the PREAMBLE_VALID signal and the SYNC_OK signal. As the signal names imply, these internal digital signals go active when the RX modem detects the successful reception of these respective fields in the packet structure. These signals are shown in Figure 22. This plot also shows how the RFIC automatically turns off upon reception of a valid packet.

In the event that the user desires to receive an additional packet, it is not necessary to re-send the entire script shown above. This example script must be run *once* (to configure the RFIC for RX Packet FIFO Mode); subsequent packets may be received by simply turning the receiver on with the following one-line script:

```
### Turn Receiver ON ###
S2 8704
```

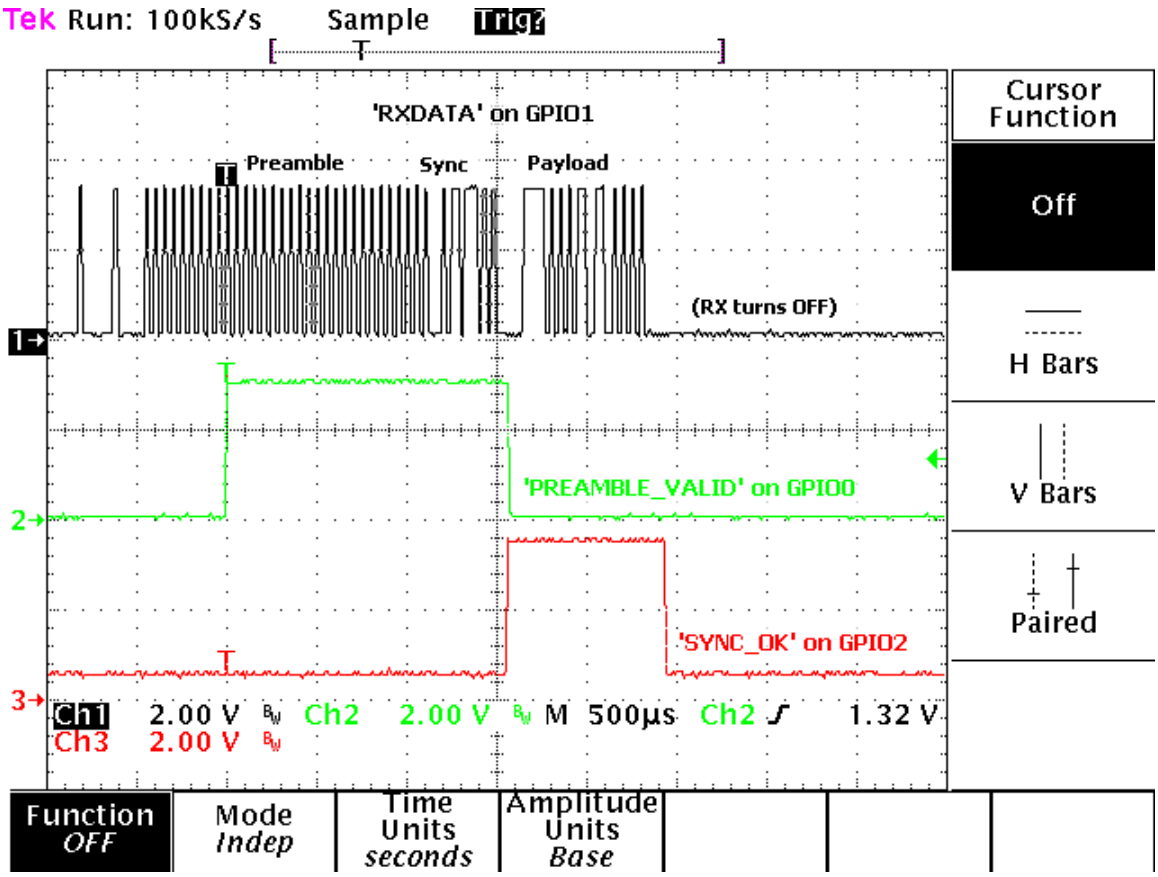

Figure 22. RXDATA and Packet Timing Signals in FIFO Mode

Additional internal digital signals may also be of interest, such as “pk_rx” (Packet RX, indicating the packet is being received) and “pk_valid” (Packet Valid, indicating a valid packet has been received). These alternative internal digital signals may be accessed via the digital test bus, as shown in the example script excerpt below. The resulting signals are shown in Figure 23.

```
# Set GPIO0 = Digital Test Bus
S2 8B0B
# Set GPIO1 = RXDATA output (with increased GPIO drive)
S2 8C94
# Set GPIO2 = Digital Test Bus
S2 8D0B
# Configure Digital Test Bus = 33h = 51d = ‘pk_rx’, ‘sync_ok’, ‘pk_valid’
S2 D133
```

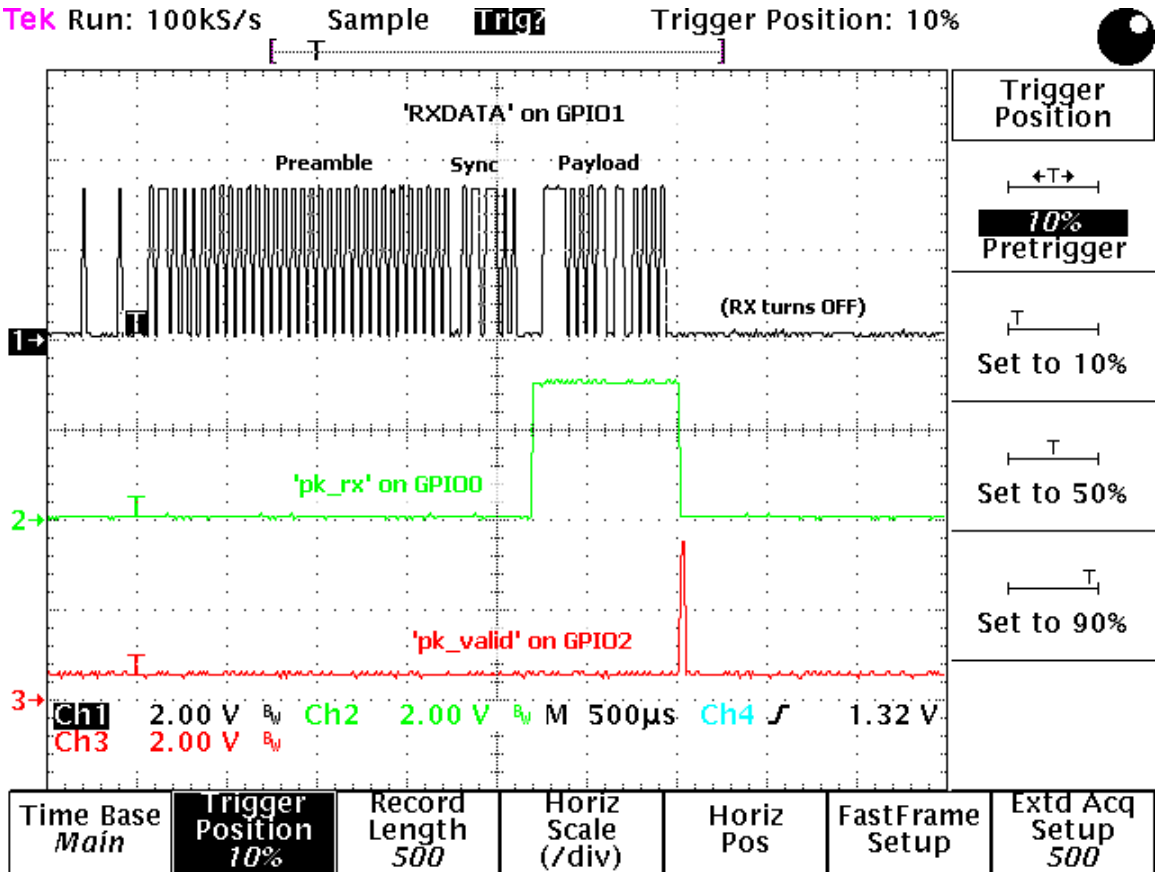

Figure 23. RXDATA and More Packet Timing Signals in FIFO Mode

In the example shown in Figure 23, the 4 bytes of Payload data that were used in the test packet structure were: 0Fh, AAh, CCh, AAh. It is apparent from simple visual inspection of the received Payload data bits that the test packet was received correctly. However, as the purpose of this section is to demonstrate Packet FIFO Mode, we also desire to read these data bytes from FIFO memory over the SPI interface. This may be done by creating the following separate script to read 4 bytes from FIFO memory:

```
# Read FIFO memory
S2 7F00
S2 7F00
S2 7F00
S2 7F00
```

If the user has enabled the data-logging window within WDS, the bytes read back over the SPI interface will be displayed within the log window, and may be verified against the transmitted Payload data bytes in the test packet structure.

Alternatively, the user could read the bytes in FIFO memory through use of the FIFO R/W tab in WDS; the received bytes may be displayed in the FIFO READ window.

5. Link Functional Tests

5.1. TX-RX Packet FIFO Mode (Link Test)

As a final functional verification test, it is possible to create a simple one-way link between two RF Test Cards, using only scripts executed from within WDS. Specifically, we will use one RF Test Card to transmit in TX Packet FIFO Mode, while the other RF Test Card will be configured to receive in RX Packet FIFO Mode. Two Load Boards with RF Test Cards may simultaneously be controlled from one computer and WDS window; the user may wish to review the WDS Chip Configurator user's guide for details on how to control multiple Load Boards and RF Test Cards.

The test setup is shown in Figure 24. If the two Load Boards and RF Test Cards are physically located only a short distance apart, the TX and RX antennas shown in the test setup below may be replaced by 50 Ω terminations; the residual radiation from the TX board will easily be sufficient to establish a link over a distance of a few feet.

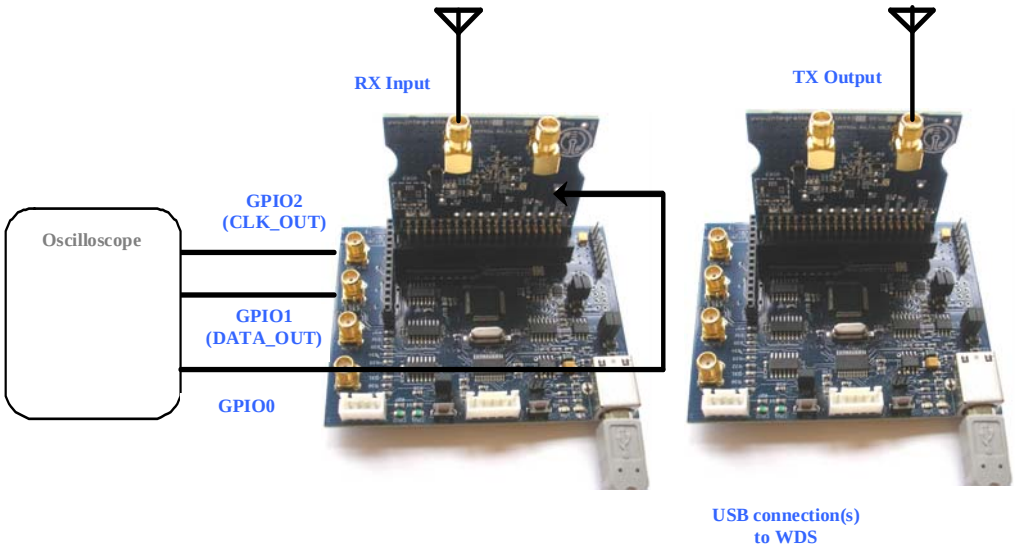

Figure 24. Test Setup for TX-RX Link Test

The required TX and RX scripts have been discussed in detail previously, and will not be completely re-shown here. For the RX board, the user should execute the following script:

Script Name: RX_913.0MHz_PacketFIFO_GFSK_40kbps_20kDev.txt

For the TX board, the user should execute the following script:

Script Name: TX_913.0MHz_PacketFIFO_GFSK_40kbps_20kDev.txt

The order of execution of the scripts is important. The RX script should be run first, so that the RX board is "waiting" for the arrival of the incoming TX packet. The TX script should then be run after the RX script.

If the user has set the oscilloscope to trigger on the GPIO0 signal (PREAMBLE_VALID in the example RX script), then the resulting display should match that shown previously in Figure 22. This verifies the functionality of a simple one-way link.

If the user wishes to transmit and receive further packets, it is not necessary to re-send the above two example scripts in their entirety (although there is nothing wrong with doing so). It is sufficient to simply send the following one-line RX script:

```
### Turn Receiver ON ###
S2 8704
```

and then the following one-line TX script:

```
### Turn Transmitter ON ###
S2 8708
```

This sequence should successfully transmit and receive one additional packet.

Simplicity Studio

One-click access to MCU tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

www.silabs.com/simplicity

MCU Portfolio
www.silabs.com/mcu

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com