Application Report
SWRA041–September 2004

Implementing a Bidirectional Frequency Hopping Application With TRF6903 and MSP430

Shreharsha Rao
Mixed Signal RF

ABSTRACT

The US ISM band (902-928 MHz) is a shared spectrum with many unlicensed radio devices interfering with each other. Spread spectrum systems are known for their interference rejection and anti-jamming techniques in multiple access systems. Federal Communications Commission (FCC) regulates the operation of unlicensed devices in the US ISM band. Under Part 15 of FCC regulations, frequency hopping systems are allowed to transmit at powers of up to +30 dBm EIRP. This higher power operation coupled with the benefits of spread spectrum systems makes frequency hopping an attractive option for the unlicensed radio devices in the US ISM band.

This application note discusses the implementation details to establish a bidirectional point-to-point and point-to-multipoint frequency hopping wireless link using Texas Instruments' TRF6903 single-chip multiband RF transceiver (SWRS022) and MSP430F449 (SLAS344), a 16-bit ultra-low power microcontroller from the Texas Instruments MSP430 family. This application is an extension of the single frequency wireless UART application (see SWRA039 for complete implementation details of the single frequency wireless link) to include frequency hopping.

For schematics, see SWRR001 . The complete firmware can be downloaded from the Texas Instruments ISMRF website at www.ti.com/ismrf.

Contents

1 Introduction 2
2 Frequency Hopping Systems 2
3 FCC Regulations for Frequency Hopping in the US ISM Band 6
4 Implementation of the Frequency Hopping System in Firmware 9
5 References 15
Appendix A Frequency Hopping Firmware - Code Snippet 16

List of Figures

1 **An Example of a Frequency Hopping Pattern** 3
2 **Block Diagram of a Frequency Hopping System** 4
3 **Architecture of a Point-Multipoint Frequency Hopping System** 5
4 **Frequency Hopping Transmit-side Block Diagram** 10
5 **GUI Used to Transmit and Receive Data** 11
6 **Data Packet Fragmented Into Frames For Frequency Hopping** 12
7 **Frequency Hopping Receive-side Block Diagram** 14

List of Tables

1 **FCC Restricted Frequencies from 30 MHz to 1GHz** 6
2 **US General Spurious Limitations** 7
3 **Operation Within 902-928 MHz (15.249) - Single Channel Devices** 7

4 **Operation Within 902-928MHz (15.247)- Frequency Hopping Devices** 8
5 **Hop Rate for Various Transceiver Bit Rates** 15

1 Introduction

The TRF6903 lends itself well for frequency hopping applications. The TRF6903 single-chip solution is an integrated circuit intended for use as a low-cost multiband FSK transceiver to establish a frequency-programmable, half-duplex, bidirectional RF link. The multichannel transceiver is intended for digital (FSK, OOK) modulated applications in the North American and European 315-MHz, 433-MHz, 868-MHz, and 915-MHz ISM bands.

The synthesizer has a typical channel spacing of better than 200 kHz and features a fully-integrated VCO. The TRF6903 features a PLL with a lock time of around 200 µs for a bit rate of 38.4-kbps NRZ. (See the *TRF6903 Design Guide* (SWRU009) for exact lock time values for various loop filter bandwidths). This ultra-fast lock time coupled with the feature of having no calibration procedure when switching frequencies makes the TRF6903 one of the fastest frequency hopping solutions available.

The frequency hopping application using the TRF6903 and MSP430F449 has been implemented as an extension of the single-frequency wireless UART application discussed in SWRA039 . The system and protocol definition, wireless transmission, reception and acknowledgement implementation are **identical** to the single-frequency wireless UART application discussed in SWRA039 . Thus, these details are not discussed in this application note. This application note discusses the specifics of developing a frequency hopping application in the US ISM band (902-928 MHz).

2 Frequency Hopping Systems

2.1 Introduction to Spread Spectrum Systems

Spread spectrum signals used for the transmission of digital information are distinguished by the characteristic that their bandwidth W is much greater than the information rate R in bits/s. The bandwidth expansion factor B_e = W/R for a spread spectrum signal is much greater than unity. The large redundancy inherent in spread spectrum signals is required to overcome the severe levels of interference that are encountered in the transmission of digital information over radio channels. The radios operating in the 900-MHz ISM band are subjected to lot of interference from other unlicensed devices in the same band and could significantly degrade the sensitivity of the operating radio.

Another important element employed in the design of spread spectrum systems is pseudo-randomness, which makes the signals appear similar to random noise and difficult to demodulate by the receivers other than the intended ones. To summarize the benefits of spread spectrum systems are:

- Combating or suppressing the detrimental effects of interference due to jamming, interference arising from other users of the channel, and self interference due to multipath propagation.
- Achieving message privacy in the presence of other listeners
- Hiding a signal by transmitting it at low power and thus making it difficult for an unintended listener to detect in the presence of background noise (low probability of intercept (LPI) signal)

Two types of spread spectrum systems exist. The direct sequence spread spectrum (DSSS) and frequency hopping spread spectrum (FHSS). In DSSS, a PN generator generates PN codes at a rate termed the chip rate which is much faster than the data rate. The data output at the data rate and the PN generator output at the chip rate are Modulo 2 added and fed to the PSK modulator. At the receiver, complex correlation properties of the PN codes are used to decode the message sequence. DSSS is much more complex to implement due to the stringent synchronization requirements and needs a coherent modulation technique like binary phase shift keying and hence is not suitable to be used with TRF6903 RF transceiver.

2.2 Frequency Hopping Systems

In a frequency-hopped spread spectrum system, the available channel bandwidth is subdivided into a large number of contiguous frequency slots. The US ISM band from 902-928 MHz lends itself well for low data rate frequency hopping systems. In any signaling interval, the transmitted signal occupies one of the available frequency slots. The selection of the frequency slots during each signaling interval is made pseudo-randomly. User-defined protocols can be used to determine the hopping sequence. Figure 1 shows an example of a random frequency hopping pattern.

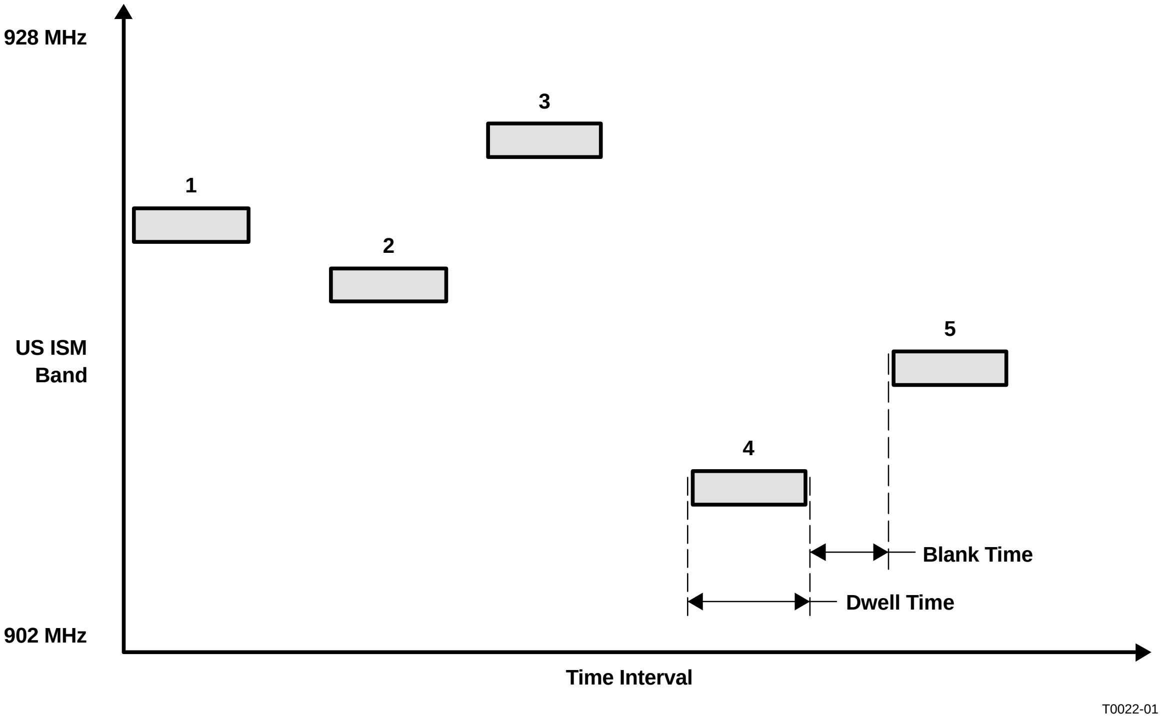


Figure 1. An Example of a Frequency Hopping Pattern

The small rectangles represent random carrier frequencies generated at successive time intervals. The time the transceiver is active (transmit or receive) is termed the dwell time. The time when the transceiver is configuring its registers to transmit or receive at another frequency is termed the Blank Time. The following events happen during each of these time periods.

Dwell Time

- Transmit (or receive) preamble, start bit, data sequence at a particular frequency in the hopping sequence.

Blank Time

- Pseudo random frequency generation.
- Configuring the TRF6903 Registers to operate at the randomly generated frequency.
- Waiting for PLL to Lock.

To conserve battery power, the Blank Time should be as low as possible.

A block diagram of the frequency-hopped spread spectrum system is shown in Figure 2.

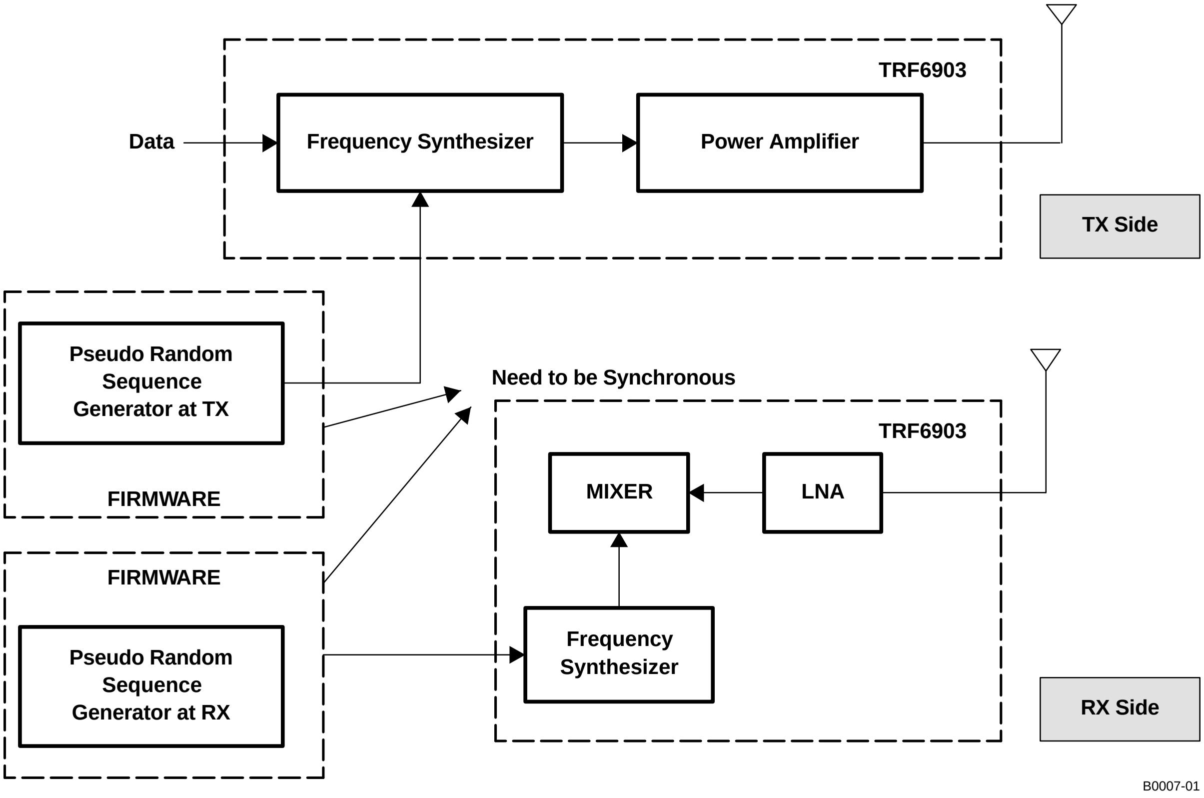


Figure 2. Block Diagram of a Frequency Hopping System

The modulation used with frequency hopping systems is usually binary FSK. As shown in Figure 2 the pseudo random sequence generator is used to control the output of the frequency synthesizer. The transmit carrier frequency hops pseudo randomly as determined by the transmit PN generator. This way random frequencies are generated as illustrated in Figure 1. The receiver needs to generate frequencies in the same pseudo random order to ensure proper demodulation of the signal.

At the receiver side there is an identical PN generator synchronized with the receive signal, which is used to control the output of the frequency synthesizer. This synthesized frequency is mixed with the incoming RF frequency to generate a fixed IF signal. The receive side pseudo random generator needs to be synchronized to the transmit side pseudo random generator to generate the proper IF signal for the demodulator.

The pseudo random generation of frequencies is usually implemented in firmware. The implementation details of the firmware and the means to achieve synchronization between transmit side and receive side are illustrated in Chapter 4. Frequency hopping systems are classified into two types depending on the rate at which hopping takes place. In slow frequency hopping, multiple data symbols are transmitted during one hop, whereas in fast frequency hopping there are multiple hops per data symbol. The slow frequency hopping system is used in this application.

2.3 Acquisition and Synchronization in Frequency Hopping systems

As explained in Section 2.2, since the transmit carrier frequency hops pseudo randomly, the receiver needs to generate frequencies in the same pseudo random order to ensure proper demodulation of the signal. Thus, there has to be time synchronization between the transmitter and receiver. This synchronization happens in following two phases:

- Acquisition – This is the initial phase where the receiver recognizes the transmitter.
- Tracking – This phase happens upon successful acquisition phase. In the tracking phase, the transmitter and the receiver need to be in continuous synchronization until data transmission is complete.

2.3.1 Acquisition

The complexity of the acquisition phase depends on the architecture of the frequency hopping system. If a point-to-point architecture is used, the acquisition is simple, whereas if point-to-multipoint architecture is used the acquisition can be complex. The acquisition protocol has to be designed keeping the system architecture in mind while making sure that protocol is secure and easy to implement in firmware.

For a point-to-point system, the communicating transceivers can be assigned unique transceiver ID's which are hard-coded in firmware. The transmitter sends a packet with the receiver transceiver ID as part of the header. The packet header transmission can take place using a pre-decided *acquisition channel*. The receiver constantly looks for packets in the same *acquisition channel*. Once it receives the header, the transceiver ID can be verified to complete the acquisition phase.

For a point-to-multipoint system, hierarchical definitions are needed. A group of transceivers are defined as a system and a unique system ID is assigned for all transceivers operating in that system. Communication can only happen between transceivers with the same system ID. Transceivers with the same transceiver ID but different system ID's will not communicate.

This is better illustrated in Figure 3.

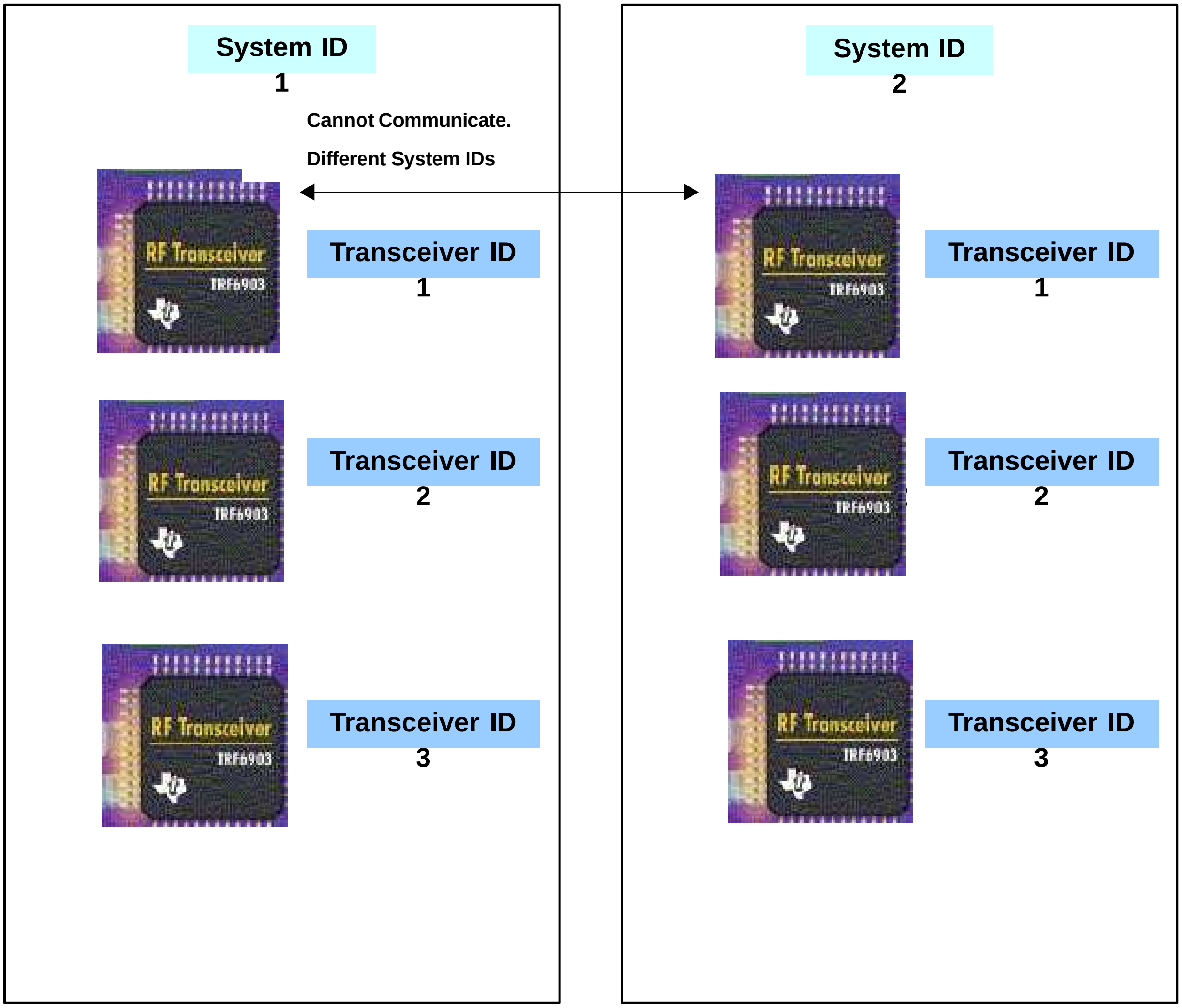


Figure 3. Architecture of a Point-Multipoint Frequency Hopping System

One commonly used acquisition method for point-to-multipoint systems is that the transmitter sends data on a randomly generated channel from a defined hop-set for a time duration which is equal to the hop dwell time + blank time of the of the receiver times the number of active channels in the hop-set. For example, if the receive hop dwell time (protocol defined) is 1.6 ms and blanking time is 0.4 ms, the total time the receiver stays on one frequency is 2 ms. Assuming 50 channels in the hop-set, the transmitter sends a 100 ms (50 x 2) training sequence (alternate 1's and 0's) at a randomly generated frequency. The receiver scans through all the 50 frequencies in the hop-set (for 100 ms) looking for activity. Valid data is generated when the receiver finds the transmitted channel. This way acquisition is completed.

This method is easy to implement and is recommended for systems which wake up periodically (for example sensor systems) and does not have a lot of activity. Since the acquisition period is large, this algorithm is not recommended for systems with lot of burst type data transmission.

2.3.2 Tracking

Once acquisition phase is complete the transmitter and receiver recognize each other and can communicate. In a frequency hopping system, the receiver should change frequencies in the same order as the transmitter. This needs both time and frequency synchronization and is implemented in the tracking phase. This is probably the most complex part of implementing a frequency hopping system. There are several ways of implementing frequency synchronization. The mostly widely used method is to store transmit/receive frequency pairs for each hop-set as a constant table in the microcontroller's memory. Both transmit and receive frequencies must use an identical hop-set for this method to work. Although this method is easy to implement, it has several drawbacks. The memory requirements for the microcontroller increase linearly with the number of hop-sets. The active hop-set need to be defined in advance etc. A simpler way of achieving frequency synchronization is explained in the Section 4.2.2 Random Frequency Generation. The firmware is written with this simple frequency synchronization method.

Time synchronization is usually achieved by using a fixed hop-rate at the transmitter and receiver. Since the hop-rate is constant, the number of bits (or packets) transmitted using a single frequency is constant. The receiver can the count the bits (or packets) to decide when to hop to the next channel.

NOTE

It has to be stressed that acquisition and tracking protocol has to be customized for the end application to meet the desired specifications. The protocols mentioned in these sections have to be used as a guideline only.

3 FCC Regulations for Frequency Hopping in the US ISM Band

In addition to all the advantages of using a frequency hopping system as explained in Section 2.1, perhaps the most important advantage of frequency hopping systems is the option to transmit at higher power levels (up to +30 dBm as specified by the FCC Part 15 Sec 15.247). FCC regulations for US ISM band are summarized in the following sections.

In the US, low date rate transceivers are governed by Federal Communications Commission (FCC). FCC regulations focus exclusively on the emissions of devices. Telecommunications rules in the US are codified in Title 47 of the Code of Federal Regulations (CFR). The FCC rules on low date rate devices are covered in Part 15 of Title 47.

The first part of the regulations on unlicensed transceivers is a list of restricted bands in which only low-level spurious emissions are allowed (§15.205). These restricted bands apply to the harmonics and sub-harmonics of transceiver outputs as well as the fundamental. The restricted frequencies from 30 MHz to 10 GHz are listed in Table 1.

Table 1. FCC Restricted Frequencies from 30 MHz to 1GHz

MHz	MHz	MHz	GHz
37.5 - 38.25	167.72 - 173.2	1660 - 1710	3.6 - 4.4
73 - 74.6	240 - 285	1718.8 - 1722.2	4.5 - 5.15

Table 1. FCC Restricted Frequencies from 30 MHz to 1GHz (continued)

MHz	MHz	MHz	GHz
74.8 - 75.2	322 - 335.4	2200 - 2300	5.35 - 5.46
108 - 121.94	399.9 - 410	2310 - 2390	7.25 - 7.75
123 - 138	608 - 614	2483.5 - 2500	8.025 - 8.5
149.9 - 150.05	960 - 1240	2655 - 2900	9.0 - 9.2
156.52475 - 156.52525	1300 - 1427	3260 - 3267	9.3 - 9.5
156.7 - 156.9	1435 - 1626.5	3332 - 3339	
162.0125 - 167.17	1645.5 - 1646.5	3345.8 - 3358	

There is also a general limitation on the level of spurious emissions (§15.209). These levels depend on frequency and are summarized in Table 2. Usually, the requirements on harmonics of the transmit frequency are relaxed, but if the harmonic falls in one of the restricted bands, it must meet the spurious limits in Table 2.

Table 2. US General Spurious Limitations

FREQUENCY RANGE (MHz)	MAXIMUM SPURIOUS EMISSION (dBm)
30-88	-55.25
88-216	-51.75
216-960	-49.25
Above 960	-41.25

If the harmonics of the transmitted frequency does not fall in one of the restricted bands outlined in Table 1, the harmonic suppression limit is only 20 dBc.

Operation within the 902-928 MHz band is covered in FCC Part 15.249 (Subpart C – Intentional radiators). This is summarized in Table 3.

Table 3. Operation Within 902-928 MHz (15.249) - Single Channel Devices

FUNDAMENTAL FREQUENCY (MHz)	FIELD STRENGTH OF FUNDAMENTAL (mVm)	FIELD STRENGTH OF HARMONICS (µV/m)	FIELD STRENGTH OF FUNDAMENTALS (dBm)	FIELD STRENGTH OF HARMONICS (dBm)
902-928	**50**	**500**	**-1.2**	**-41.2**
2400-2483.5	50	500	-1.2	-41.2
5725-5850	50	500	-1.2	-41.2
24000-24250	250	2500	12.7	-27.3

Field strength limits are specified at three meters. Emissions outside of specified bands (except harmonics) is the lesser attenuation of -50 dBc or 15.209.

A transmitter of constant level of power can produce electric fields of different strengths depending upon the transmission line and antenna connected to it. The electric field is what causes the interference, hence most of the Part 15 emission limits are specified in field strength.

FCC Part 15.247 specifies operation in the frequency hopping mode. General purpose data devices with frequency hopping do not have any restriction on the type of data or the duty cycle of the transmission. However, a spreading of the transmission spectrum through frequency hopping is required. The allowed UHF band for hoppers is 902 – 928 MHz.

The 15.247 rules are summarized below. (See Table 4)

- Channel spacing (-20 dB) minimum of 25 kHz and maximum of 500 kHz.

- Frequency hopping systems (15.247a1) must use at least 50 hopping frequencies if the 20-dB bandwidth is <250 kHz, having a duration of <0.4 s of a 20 s period, and maximum 1-W output power. (30-dBm EIRP).
- For bandwidths >250 kHz, at least 25 channels must be used, having a duration of <0.4 s of a 10 s period, and maximum 0.25-W output power. (24 dBm EIRP).
- Spurious emissions in a hopper need only be 20 dB below the fundamental, with the exception, of course, of the restricted bands. It is important to note that the third, fourth, and fifth harmonics of the 902 – 928 MHz band do fall within restricted bands, so their emission is limited to approximately –41.25 dBm.
- Single-channel general purpose data devices share the same 902 – 928 MHz frequency band as hoppers, just at a reduced power level. There is no hopping requirement. The maximum output power is approximately –1.25 dBm (see Table 3), and the maximum harmonic power is approximately –41.25 dBm. Spurious output other than harmonics must be either 50 dB below the fundamental, or meet the requirements in Table 4, whichever is the least restrictive.

Table 4. Operation Within 902-928MHz (15.247)- Frequency Hopping Devices

20-dB BANDWIDTH	NUMBER OF CHANNELS	MAXIMUM FUNDAMENTAL OUTPUT POWER (dBm EIRP)
<250 kHz	At Least 50	30
> 250 kHz	At Least 25	24

NOTE

The designer assumes all responsibility for compliance with applicable European, North American, or other governmental regulations in the use of the TRF6903 and other products made by Texas Instruments. Texas Instruments assumes no responsibility or liability for summarizing or interpreting regulations governing the use of wireless transmitters or receivers.

4 Implementation of the Frequency Hopping System in Firmware

This chapter discusses the implementation of point-to-point frequency hopping systems using the TRF6903 and MSP430F449. For the demo board schematic, see SWRR001 . The frequency hopping algorithm is implemented in firmware and is flashed on to the MSP430F449 microcontroller using the JTAG interface.

4.1 Downloading the Code

The workspace file for compiling the demo source code is `6903_FREQHOP_DEMO.eww`, developed with IAR compiler version 3.20.

The following steps show how to open the project file and program the MSP430.

1. Insert the two AAA batteries and slide the on/off switch to ON.
2. Start the Workbench (START->PROGRAMS->IAR SYSTEMS->IAR EMBEDDED WORKBENCH
3. Use FILE->OPEN WORKSPACE to open the workspace file at: <Installation root>\...\ `6903_FREQHOP_DEMO.eww`
4. Use PROJECT->REBUILD ALL to compile and link the source code. You can view the source code by double-clicking Common Sources and then double-clicking on the source files in the workspace window.
5. Use PROJECT->DEBUG to start C-SPY. C-SPY erases the device flash and downloads the application object file.
6. In C-SPY, use EXECUTE->GO to start the application.
7. In C-SPY, use FILE->EXIT to exit C-SPY.
8. In Workbench, use FILE->EXIT to exit Workbench.

The MSP430 is now programmed with the frequency hopping code and is ready to operate.

4.2 Protocol Definition

The baseband system and protocol definition, wireless transmission, reception, and acknowledgement implementation are **identical** to the single-frequency wireless UART application discussed in SWRA039 . The firmware discussed in this application note is capable of frequency hopping, while the firmware discussed in the SWRA039 application note is developed for a single-channel wireless UART application.

4.3 Frequency Hopping Algorithm

The flowchart of the frequency hopping algorithm at the transmit and receive side is shown is detail in this section.

4.3.1 Transmit Side Protocol

The transmit side flowchart is shown in Figure 4. The implementation details are explained in the following sections.

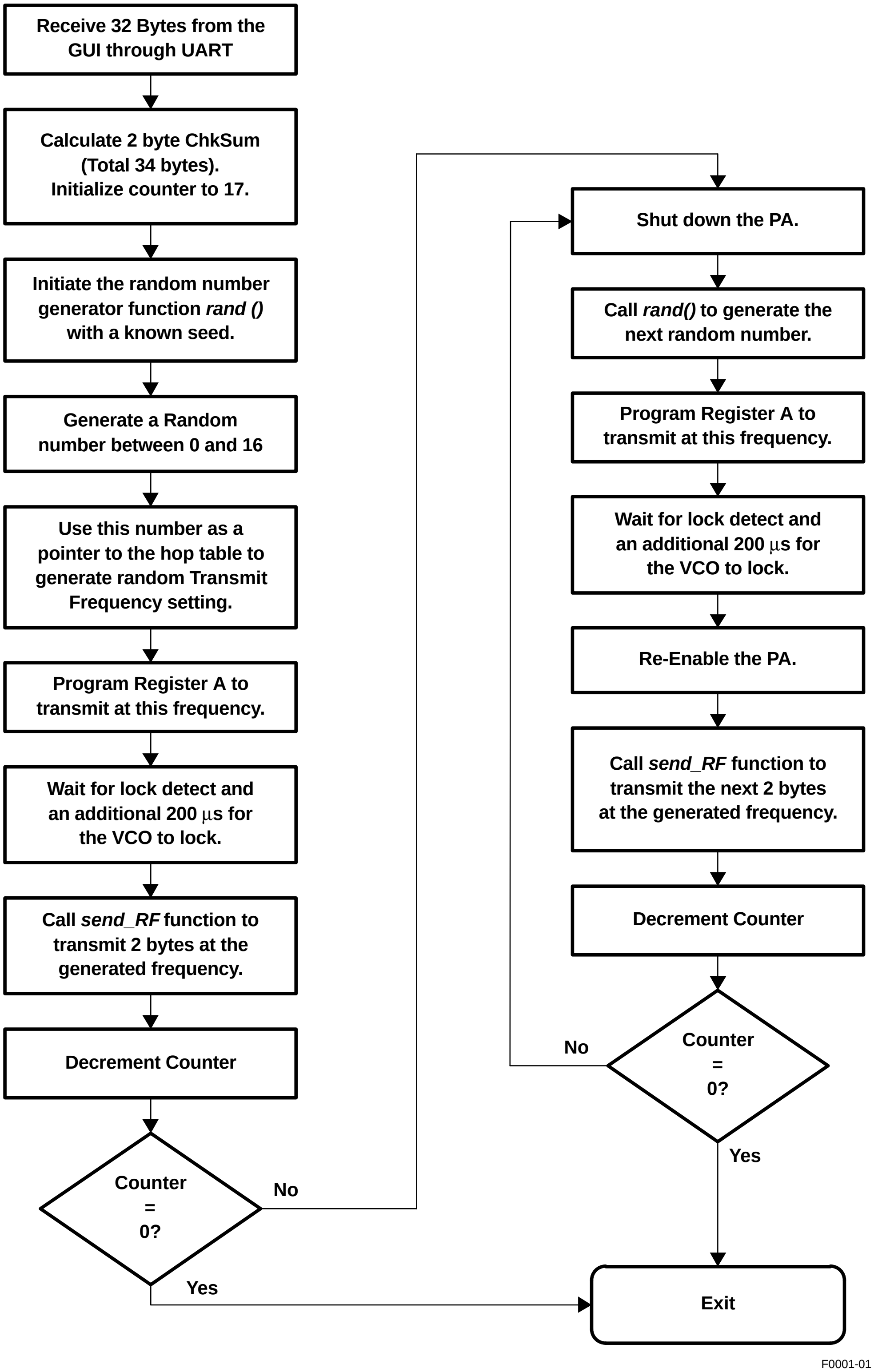
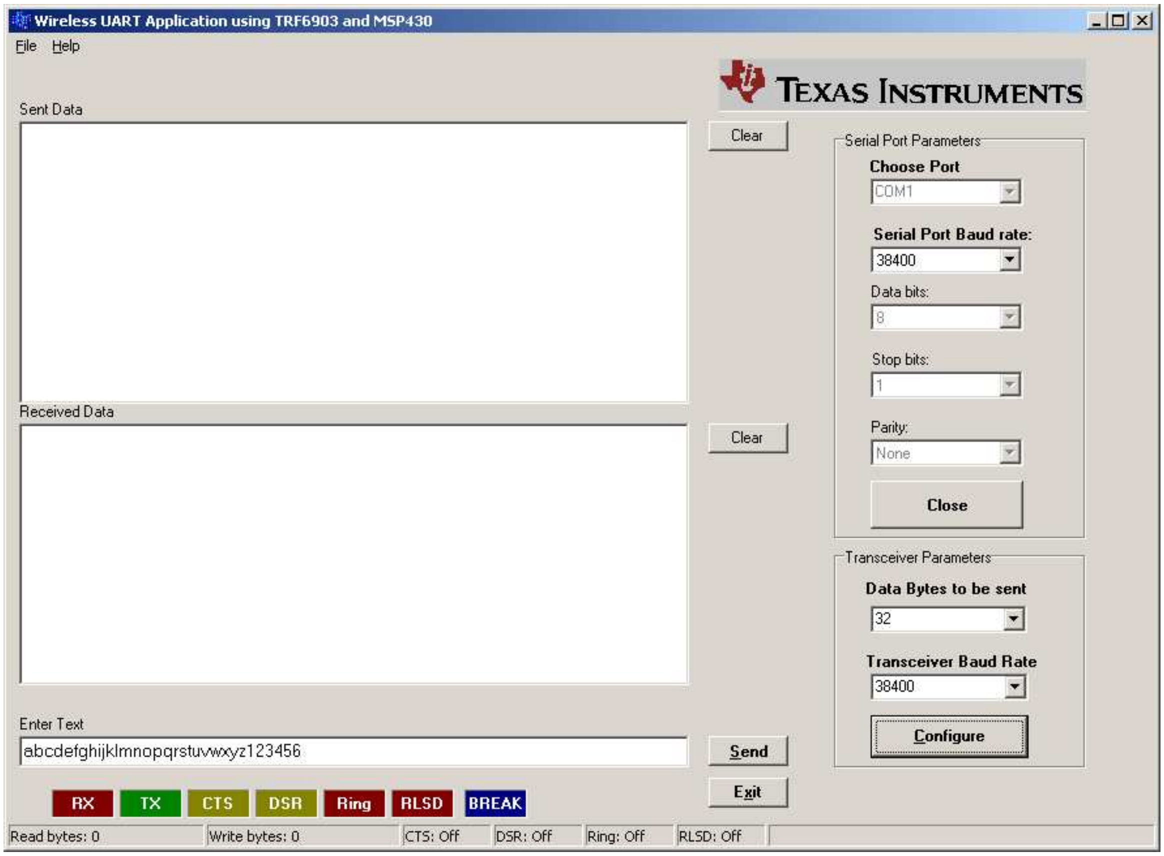


Figure 4. Frequency Hopping Transmit-side Block Diagram

A GUI shown in Figure 5 is used to transfer 32 bytes of data to the MSP430 UART using RS232.



Wireless UART Application using TRF6903 and MSP430

Choose Port: COM1

Serial Port Baud rate: 38400

Data bits: 8

Stop bits: 1

Parity: None

Data Bytes to be sent: 32

Transceiver Baud Rate: 38400

Enter Text: abcdefghijklmnopqrstuvwxyz123456

Read bytes: 0 | Write bytes: 0 | CTS: Off | DSR: Off | Ring: Off | RLSD: Off

Figure 5. GUI Used to Transmit and Receive Data

The MSP430 microcontroller stores the received data from UART in the transmit buffer (RAM). The 2-byte checksum is calculated and stored as part of the data packet. The data packet thus contains 34 bytes of data.

These 34 bytes of data are split into frames of 2 bytes each and each frame is transmitted using a different frequency. Thus, 17 random frequencies need to be generated to transmit these 34 bytes of data. This is illustrated in Figure 6. Each frame transmission is preceded by a training sequence (approximately 3 ms in length) and a start bit.

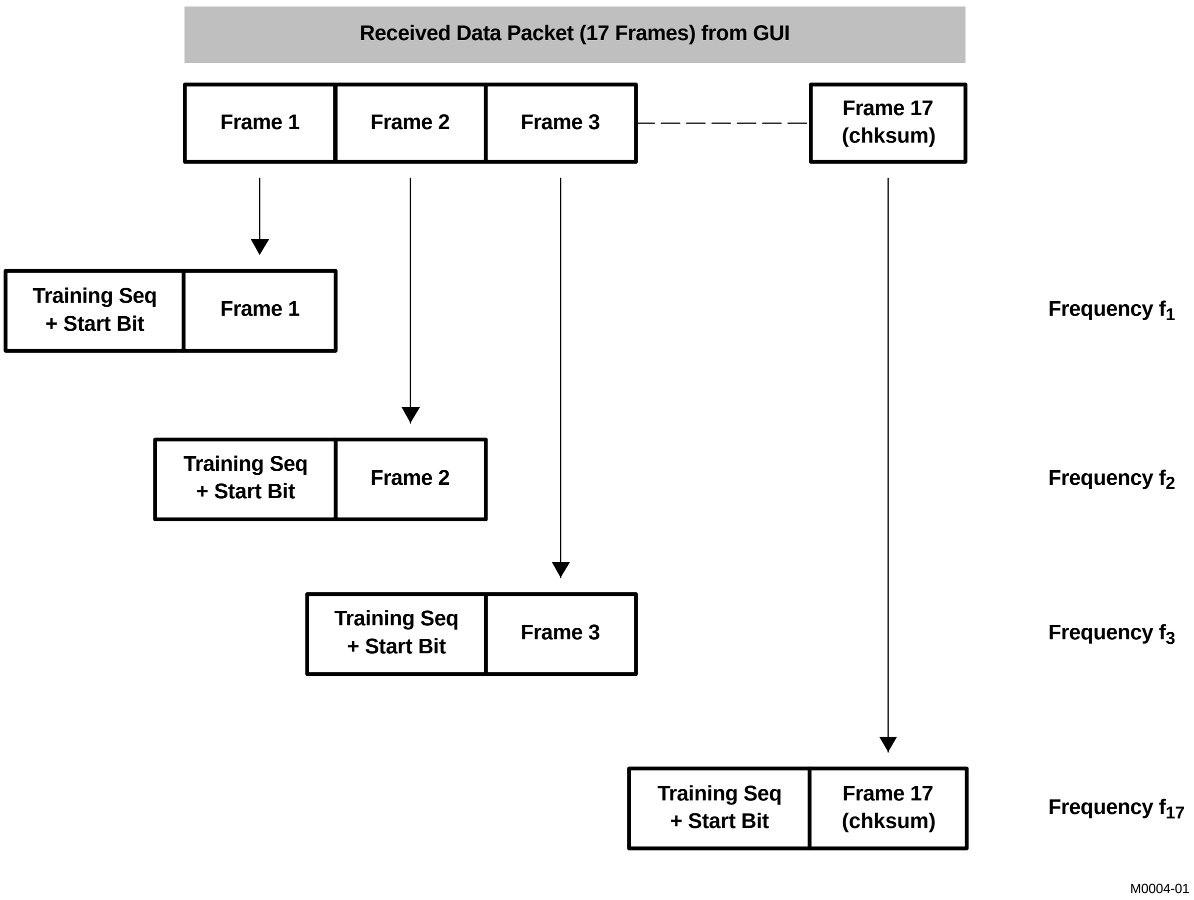


Figure 6. Data Packet Fragmented Into Frames For Frequency Hopping

4.3.2 Random Frequency Generation

Random frequency generation is accomplished by generating a uniformly distributed random number between 0 and 16 and using this as a pointer to the hop table. The hop table consists of 17 TX/RX frequency pairs uniformly distributed across the ISM band and are spaced approximately 1 MHz apart. The random number generation is accomplished using the function *rand ()* available in the IAR embedded workbench C library. The designer can write their own random number generation routines. The *rand ()* function generates a random number between 0 and 32768; this is divided by 2048 to generate a random number between 0 and 16.

The receive-side frequency generation has to be synchronous with the transmit-side frequency generation. This is accomplished by generating random numbers *with the same seed* in the transmit and receive side firmware routines. The hop table is stored in the header file `f_6903.h` in the firmware.

For low-cost applications where a low-end microcontroller (limited memory) is used, frequency generation and synchronization can be achieved without using a hop table which consumes memory. Instead of storing a hop table, the lowest frequency information is stored (902 MHz for US ISM band) and random offsets are used to generate the frequencies.

4.3.3 Steps to be Taken When Frequency Hopping at the Transmit Side

In the initialization stage the TRF6903 register A is programmed with the random frequency generated using the hop-table. The firmware routine then waits until LOCK_DETECT (pin 22) goes high. LOCK_DETECT goes high, when the VCO is locked to the programmed frequency to within a few kHz. It is recommended to wait for an additional 200 µs before send_RF can be called to transmit data. If LOCK_DETECT is not used (to free up a pin on the microcontroller), a 1-ms wait time after programming the registers is sufficient to initiate data transmission. The `send_RF` routine is then called to transmit the first frame along with the training sequence and start bit. See SWRA039 for implementation details of `send_RF` routine.

Once the first frame is sent, it is necessary to frequency hop and some procedures need to be followed to successfully implement frequency hopping. To reduce splatter when frequency hopping, it is essential to shut down the PA when programming the new frequency. The PA is re-enabled after the VCO is locked.

These steps are given below.

- Shut down the PA. (Using Register B bit 3).
- Call *rand ()* again to generate the next random transmit frequency.
- Program register A to transmit at this frequency.
- Wait for *LOCK_DETECT* or wait for 1 ms.
- Re-enable the PA.
- Call the `send_RF` function to transmit the next 2 bytes along with the training sequence and the start bit.
- Repeat these steps until all 17 frames are transmitted.

4.3.4 Receive Side Protocol

The receive side logic diagram shown in Figure 7. The implementation details are explained in the following sections.

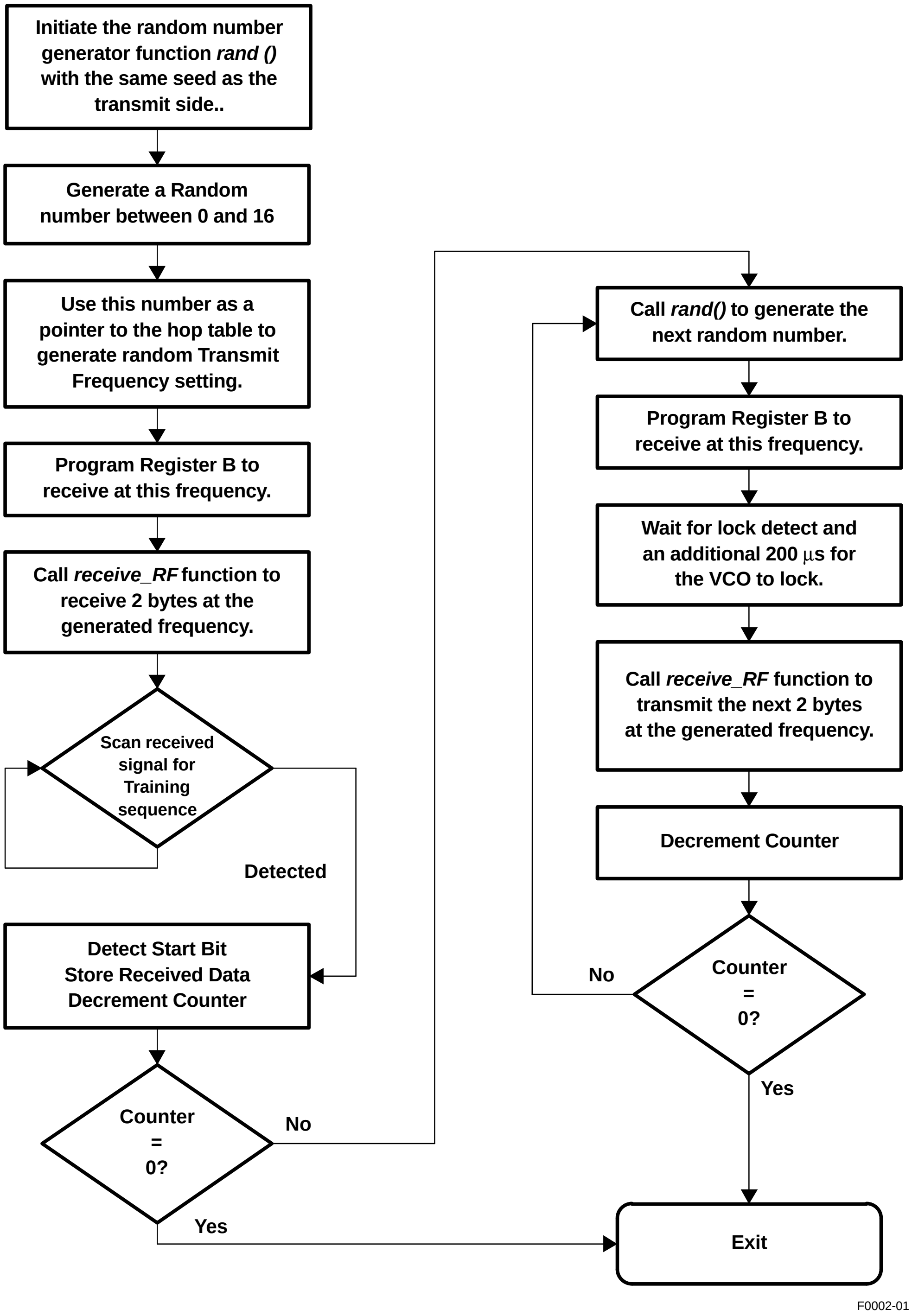


Figure 7. Frequency Hopping Receive-side Block Diagram

On the receive-side the main challenge is to establish time and frequency synchronization with the transmitter. To establish frequency synchronization, the receive frequency synthesizer needs to generate frequencies in the same random order as the transmitter. This is accomplished by calling the *rand ()* function with the same seed in both transmit and receive routines. For point-to-multipoint systems more complicated frequency synchronization techniques discussed in earlier chapters need to be used.

Time synchronization is considerably eased at the receiver if fixed hop-rate is used at the transmitter. The receiver changes frequency once a frame is received (training sequence, start bit, and 2 bytes data). To considerably ease time synchronization requirements at the receiver a fixed delay before each hop may be provided at the transmitter.

The `receive_RF` routine is then called to receive the first frame along with the training sequence and start bit. See SWRA039 for implementation details of `receive_RF` routine.

The hop rates for various transceiver bit rates implemented in firmware are summarized in Table 5.

Table 5. Hop Rate for Various Transceiver Bit Rates

RF BIT RATE (kbps)	TRAINING SEQUENCE (ms)	START BIT (µs)	HOP RATE (Hops/sec)
19.2	3	156.24	1200
38.4	6	78.12	2400
51.2	2.5	58.59	3200

5 References

1. Implementing a Bidirectional Wireless UART Application With TRF6903 and MSP430 (SWRA039)
2. TRF6903 Single-Chip Multiband RF Transceiver data sheet (SWRS022)
3. TRF6903 with MSP430 Demonstration and Development Evaluation Kit user’s guide (SWRU008)
4. TRF6903 Design Guide (SWRU009)
5. MSP430x44x data sheet (SLAS344)
6. MSP430x4xx User’s Guide (SLAU056)
7. Digital Communications, John.G.Proakis, 3rd edition

Appendix A Frequency Hopping Firmware - Code Snippet

A.1 Transmit Side Routine (Note: Complete Code Not Shown)

```
srand(1);  // Generate a random number between 0 and 32768
            f_sel=rand()/2048;   // Generate a random number between 0 and 16
            configure_trf6903();
            send_RF(buf.packetsize,buf.xmit[16-cnt2],48);


for (cnt2=1;cnt2<MAXWORD;cnt2++)
             {
               f_sel=rand()/2048;
                /* Shut Down the PA  when new frequency is programmed to reduce splatter*/
               trf6903.b.bit.PARXED = 0;
                configure_trf6903();


 /* Re-enable the PA  */
               trf6903.b.bit.PARXED = 1;
               configure_trf6903();



//write a wait routine for 200 usec atleast to allow for the VCO to lock
             gen_time_delay(200);  // Total delay 200*4 = 800 clock cycles
             send_RF(buf.packetsize,buf.xmit[16-cnt2],48);
            }
```

A.2 Receive Side Routine (Note: Complete Code Not Shown)

```
void freqhop_rcv(void)
{
 for (cnt2=1;cnt2<MAXWORD;cnt2++)
           {
            f_sel=rand()/2048;


/* Shut Down the PA  when new frequency is programmed to reduce splatter*/
            trf6903.b.bit.PARXED = 0;
            configure_trf6903();


/* Re-enable the PA/RX Chain  */
             trf6903.b.bit.PARXED = 1;
             configure_trf6903();


receive_RF(buf.packetsize,(buf.rcv+(cnt2)),16);


}
opstate|=rF_REC_FULL; //RF data received, has to be send to desktop via RS232
opstate|=rF_ACK_SEND; //initialize the acknowledge state
opstate|=RCVD;


}
```

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

Copyright © 2004, Texas Instruments Incorporated