

Frequency Hopping Systems

By K. H. Torvmark

Keywords

- *Spread Spectrum*
- *Frequency Hopping*
- *FHSS (Frequency Hopping Spread Spectrum)*
- *Anti-jam Systems*
- *CDMA*
- *Synchronisation*
- *Acquisition*
- *High-Reliability Communications*
- *Multi-path reflections*

Introduction

Spread spectrum systems are becoming more and more prevalent, also for ISM-band systems. As the ISM bands become more and more crowded, the anti-jamming properties of spread spectrum techniques are useful for ensuring reliable communications. All of Chipcon's RF chips are very well suited to the frequency hopping form of spread spectrum, as they

are equipped with fully-programmable, fast-settling frequency synthesisers.

This document outlines the basic principles of frequency hopping, as well as important implementation issues.

Spread spectrum techniques

Spread spectrum differs from a classical narrow-band or broadband system in that the signal energy is spread over a much wider frequency range, reducing the power spectral density of the signal and providing several advantages:

- Low Probability of Intercept, meaning that it is harder to detect the RF signal
- Higher tolerance to narrow-band noise sources, the anti-jamming property
- Reduction of sensitivity to interference from multi-path reflections
- Possibility of CDMA (Code-division multiple access) operation, where several co-operating transmitters using different frequency hopping patterns can transmit in the same frequency range without disturbing each other

Under FCC regulations, frequency hopping systems fulfilling certain requirements in the 902-928 MHz band are allowed to transmit using up to 1W output power. Using an external power amplifier, systems using Chipcon transceivers can achieve a range of up to several kilometres. See [2] for more information.

Direct-sequence spread spectrum

There are two forms of spread-spectrum techniques, direct-sequence (DSSS) and frequency hopping (FHSS). Direct sequence spread spectrum involves transmitting at a much higher rate than the data-rate, and convoluting the data with a spreading code. At the receiver, the signal can be decoded by using correlation techniques, comparing the incoming signal with the spreading codes. DSSS is more complex to implement than frequency hopping, and is not suitable to low-power systems due to the high data rates involved.

Frequency hopping

In this application note, we will only consider the frequency hopping form of spread spectrum, as this technique is more suited to relatively low-data rate, low-power systems. Frequency hopping, as implied by the name, is performed by changing carrier frequencies while communicating. In a typical system, the frequency hopping will be of the so-called slow variety, which means that several data symbols (bits) are transmitted during each hop. A rate between 50 and several hundred hops per second is practical. The lock time of the PLL when changing frequencies is 100us-200us (depending on the loop filter), while the time required to reprogram the needed registers using a 1 MHz clock is on the order of 50-60us. The time during a hop when data cannot be received or transmitted is termed the blanking interval. The dwell time is the time spent in each channel.

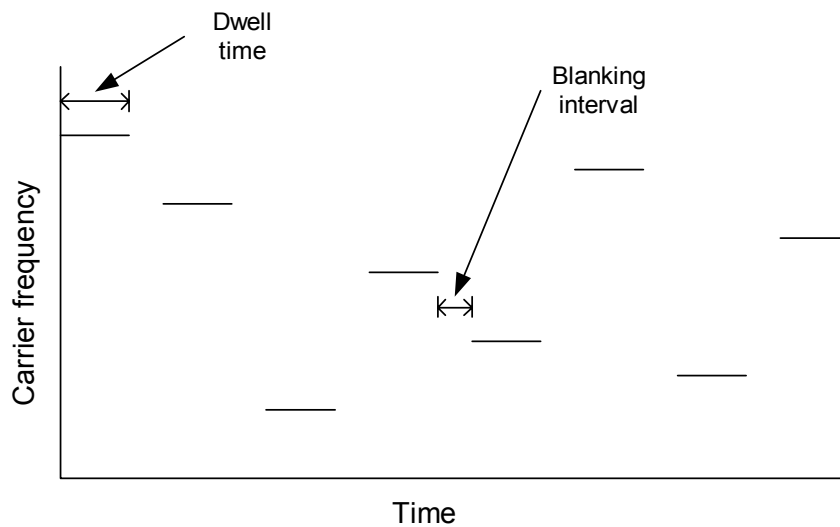


Figure 1 Frequency hopping terminology

Since spread-spectrum technology has its roots in military applications, much of the terminology refers to enemy “jammers” of varying complexity. In commercial systems, intelligent jamming is not a primary threat. Most of the time, the “jamming” signal will merely be another device trying to utilise the same frequency band for communicating. These devices will typically not be as devious as intelligent enemy jammers might be, so the security requirements can be eased a bit compared to military applications. The so-called “narrow-band jammer” is probably the most representative threat seen in civilian applications. Interference from multi-path reflections is also a serious threat. These reflections can cause large frequency- and location-dependent drops in signal strength. Frequency hopping combats multi-path reflections by ensuring frequency diversity.

Time and frequency synchronisation

Perhaps the single most challenging design issue when designing a frequency-hopping system is the question of how to synchronise the transmitter and receiver. This can be done in many ways, literature often mentions sophisticated correlation techniques, impractical in small systems using microcontrollers.

The receiver and transmitter should be provided with a table of channels that represent the allowable frequencies for frequency hopping. The number of channels will depend on the application. For 1W operation in the 902-928 MHz band the minimum number of channels required by the FCC is 50. The designer should use SmartRF Studio or the frequency calculation spreadsheet to calculate register settings for all the channels to be used, and then store these register settings in a look-up table in the microcontroller. For CDMA operation, the different co-operating systems will have tables each made up of unique channels, so that collisions will occur very rarely.

The next step is to decide how the transmitter and receiver will hop from channel to channel. To maximise the benefits of frequency hopping, the hopping should be performed pseudo-randomly. The transmitter and receiver might use pseudo-random number generators with the same seed number, or the transmitter might transmit the next channels as part of the data packet. Note that in this case, the transmitter should transmit information for several hops ahead, otherwise the receiver will lose lock completely if it misses a single data packet.

Once all of this is worked out, the synchronisation problem must be considered. The problem may be split into two parts. First the receiver must acquire the transmitter, and then it must be able to track it.

In the acquisition phase, the receiver must find the transmitter. Several approaches can be used. One is to use only a few channels as acquisition channels. This can be problematic if these channels are jammed. Another is that the receiver hops at a much slower rate than the transmitter. The receiver hop dwell times will typically be the transmitter dwell time multiplied by the number of channels. The receiver listens to each channel, trying to find data transmitted by the transmitter. Discovering if valid data is received is performed via standard software squelch methods, i.e. either using Manchester coding, the RSSI function of the CC1000 or via coding in the protocol. When valid data is received, the receiver and transmitter are synchronised in time, and the receiver can start hopping in synchronisation with the transmitter. A common synchronised clock is another method, this requires a very stable crystal oscillator.

In the tracking phase, the receiver must stay in synchronisation with the transmitter. To keep the complexity down, some constraints can be added. Having a fixed time interval between hops simplifies timing considerably. Most frequency hopping systems use a fixed hop rate. Both the receiver and transmitter can then count bits to know when to jump. Bit-level fine synchronisation is then either handled by hardware in the case of CC1000, or by using an oversampling algorithm, in the case of CC400 and CC900 (see [1] for more information).

Protocol

Protocol design is the most important part of the development for frequency hopping systems. A frequency hopping protocol will typically be more complicated than a conventional protocol due to synchronisation requirements. Often, the motivation to use frequency hopping will be a need for reliable data communication, so data error detection and correction will be implemented in the protocol as well.

A sophisticated protocol can perform many “tricks”, such as avoiding particularly noisy channels. This is termed adaptive frequency hopping. Even better, coding may be performed as part of the low-level transport protocol, so that the data is spread out on several hops. This means that if one channel is jammed, the data can still be reconstructed based on the data received on the other channels. Forward error correction (FEC) algorithms like Reed-Solomon coding are particularly interesting in this regard.

The system designer must balance the complexity of the algorithm versus the reduced error rate. Lots of issues will influence this decision; choice of micro-controller, complexity of software, power consumption, and parameters of expected interference sources are just some issues. It is difficult to provide specific recommendations for protocols, due to the variety of applications involved. Some general points may be made, however.

In an intermittently transmitting system, especially a battery-powered one, acquisition must occur for each transmission, and the acquisition time should be as short as possible. There are several ways to do this; reduce the number of channels, increase the hop rate, or implement a scheme to speed up acquisition.

In a system where very reliable data communication is needed, some form of data spreading and correction code should be used. Reed-Solomon coding, as mentioned earlier, is one such coding format. Including adaptive frequency hopping in the protocol to avoid jammed channels may be worth the extra protocol complexity.

In some applications, the low-probability of intercept characteristics of frequency hopping may be important. The protocol can then be written so that the different communicating units in the system exchange information about bit-error rates. This information can then be used to

lower the transmitted power to the lowest level that satisfies data error rate requirements, and thus adapting transmitted power to the conditions in which the system is used, lowering radiated emissions.

The data rate of the system must be seen in connection with the hop rate in a frequency hopping system, as it is the ratio between them that determines how many bits of data can be transmitted between hops. If maximum range is a requirement, the data rate should be set to 2.4 kbaud, as this gives the best sensitivity. Increasing the hop rate means that more bits are lost during the blanking interval, as the blanking interval is set by the PLL lock time, which is relatively independent on the data rate. On the other hand, increasing the hop rate lessens the demands on the data spreading, and makes it possible for the system to have a lower response time.

Implementation issues using Chipcon transceivers

The CC400, the CC900 and the CC1000 all include a flexible frequency synthesiser capable of setting frequency in 250 Hz steps.

Selection of which software squelch type to be used during acquisition will depend on the transceiver used, the CC400 and CC900 must use Manchester coding, so a squelch based on detecting Manchester code violations is a natural choice. The CC1000 provides RSSI, and the nature of the protocol will determine whether RSSI, Manchester code violations or protocol-based squelch is the best choice.

Configuration

The configuration of the radio chip should be as fast as possible. If the microcontroller has a synchronous serial interface available (an SPI interface, for example), this should be used. Ideally, the configuration time should be a fraction of a bit period. All of Chipcon's radio chips require only a few registers to be updated in order to change frequencies.

CC400/CC900

Once the CC400/CC900 receiver has locked onto the data stream, frequency hopping can be performed without sending a preamble after each hop. If synchronisation is lost, acquisition must be performed anew. The code for doing the frequency hopping may be integrated into the oversampling software.

CC1000/CC1050

In order to use sensitivity-optimised frequencies in the CC1000, the optimised frequency checkbox in SmartRF Studio should be selected when calculating the frequency table. A frequency table for the US 902-928MHz band is given in [3]. Also, the CC1000/CC1050 must be calibrated for each frequency that is to be used. The MCU should make a calibration table on startup, running calibrations for all frequencies and storing them in a table in memory. If the environmental conditions change appreciably, a recalibration should be performed, see [4] for more details on the conditions.

Later, when hopping to a new frequency, the calibration values stored in the table are written to the CC1000/CC1050 after setting the frequency. The routines needed for doing this is contained in [4].

When doing frequency hopping, a preamble in the start of each packet should be used in the acquisition phase. Once the receiver has acquired the transmitter, it should lock the averaging filter on a valid preamble. After this, the receiver does not need to re-lock the filter

during hopping. The transmitter can merely send a number of “dummy” bits corresponding to the blanking time, and the receiver can disregard an equal number of bits. Other than this blanking time, the system will not notice that it is frequency hopping at all! Note that if there are too many bits during the blanking interval, the bit synchroniser will be confused. For the higher data rates, a short preamble should be used to allow the bit synchroniser time to synchronise again. Typically, 6 bits will be enough.

The CC1000/CC1050 has two frequency registers, so the microcontroller can reprogram the inactive one while waiting to switch frequencies. Switching between the two frequency registers only entails updating the MAIN register.

Frequency table

A good idea is to implement a frequency table containing the settings needed for each separate frequency (frequency data for the CC400/CC900, frequency and calibration data for the CC1000). A pseudo-random bit-sequence can then be used as an index into this table to perform the hopping.

References

Cited references

- [1] Chipcon Application Note AN008 Oversampling and data decision for the CC400 / CC900. www.chipcon.com
- [2] Chipcon Application Note AN001 SRD Regulations. www.chipcon.com
- [3] Chipcon Application Note AN011 Programming the CC1000 for Best Sensitivity. www.chipcon.com
- [4] Chipcon Application Note AN009 CC1000 MCU Interfacing. www.chipcon.com

General references

- [5] M. K. Simon, J. K. Omura, R. A. Scholtz, B. K. Levitt : Spread Spectrum Communications Handbook, McGraw-Hill 1994.
- [6] O. Berg, T. Berg, S. Haavik, J. Hjelmstad, R. Skaug : Spread Spectrum in Mobile Communication, The Institution of Electrical Engineers, 1998
- [7] T. Hui : Bluetooth Frequency Hopping Scheme, National University of Singapore. Available from <http://opensource.nus.edu.sg/projects/bluetooth/bluetooth.html>.

This application note is written by the staff of Chipcon to the courtesy of our customers. Chipcon is a world-wide supplier of RFICs. For further information on the products from Chipcon please contact us or visit our web site. An updated list of distributors is also available at our web site.

Contact Information

Address:

Chipcon AS
Gaustadalléen 21
N-0349 Oslo,
NORWAY

Telephone	:	(+47) 22 95 85 44
Fax	:	(+47) 22 95 85 46
E-mail	:	wireless@chipcon.com
Web site	:	http://www.chipcon.com

Disclaimer

Chipcon AS believes the furnished information is correct and accurate at the time of this printing. However, Chipcon AS reserves the right to make changes to this application note without notice. Chipcon AS does not assume any responsibility for the use of the described information. Please refer to Chipcon's web site for the latest update.