A 2.4-GHz WirelessUSB™ Radio-on-a-chip For Human Interface Devices

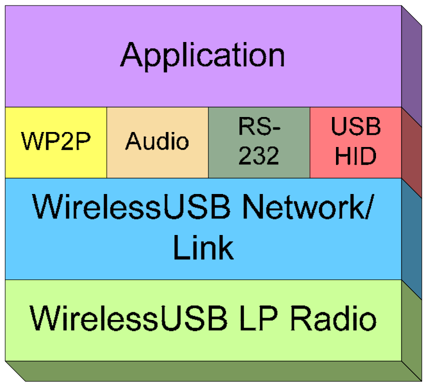
By (Ryan Woodings, Software Engineer, Cypress Semiconductor Corp.)

Executive Summary

Cycle times for designing wireless HID (Human Interface Devices) products have gone down dramatically in the past few years. Time to market has become key and taking a wireless keyboard and mouse into production from initial design phase now stands between 2-4 months. System engineers need their vendors to provide a chip solution that will help them meet the short design cycle. They also expect the chip solution to be noise tolerant and support long battery life.

Recognizing the needs of users today, Cypress recently introduced the WirelessUSB™ LP radio-on-a-chip to meet the demands of HID systems designers. WirelessUSB LP combines low power and cost, intelligent features, and an interference immune radio with a lightweight power-sensitive protocol to produce the industry's best-in-class interference immune 2.4-GHz RF solution. From the physical layer to the transport layer, WirelessUSB LP has been optimized for computer peripherals, sensors, remote controls, and other low-power, low data-rate devices. This document provides an overview of the WirelessUSB LP Radio and the WirelessUSB Protocol.

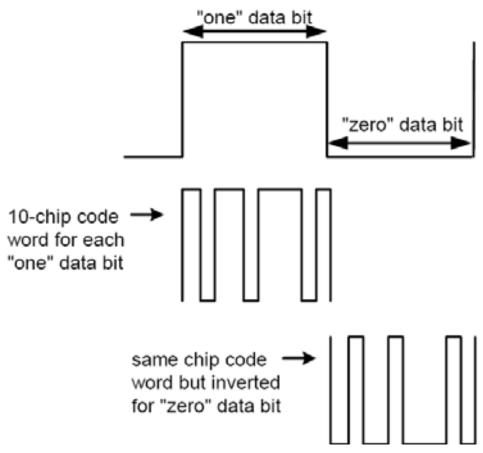
Figure 1. WirelessUSB Protocol Stack



Direct Sequence Spread Spectrum

WirelessUSB LP utilizes a 2.4-GHz direct sequence spread spectrum (DSSS) radio interface. DSSS uses a pattern for each data bit to be transmitted called a pseudo noise (PN) code. Each bit in the PN code is called a chip, and each instance of the PN code is called a symbol. Data is typically transmitted by transmitting the PN code as a 1, and transmitting the inverse of the PN code as a 0.

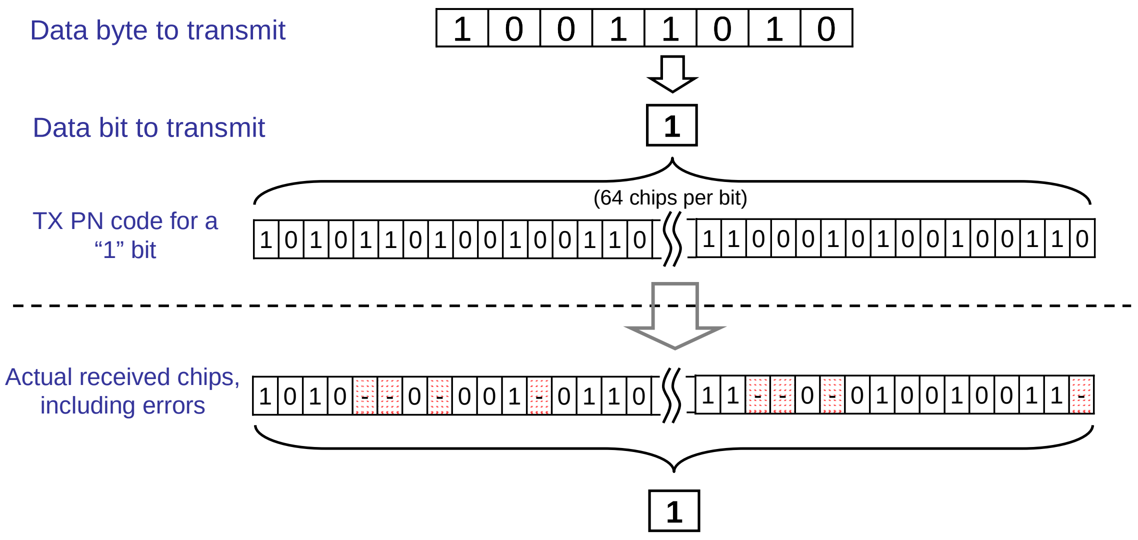
Figure 2. Data Encoded With Pseudo-Noise (PN) Code


"one" data bit
"zero" data bit
10-chip code word for each "one" data bit
same chip code word but inverted for "zero" data bit

Notice in **Figure 2** that the PN code is a binary signal that is produced at a much higher frequency than the data that is to be transmitted. The nature of this signal makes it appear that it is random noise. This allows DSSS signals to operate in noisy environments and reduces the interference caused by traditional narrowband signals. The longer the symbol is, the greater the probability that the original data can be recovered, and, of course, the more bandwidth required. WirelessUSB LP uses 64-chip and 32-chip PN codes, allowing data rates of 15-250 Kbps. WirelessUSB LP can also transmit data that is not encoded with a PN code at a data rate of 1 Mbps.

The receiver uses a correlator to separate only the desired coded information from all possible signals. A correlator is a special type of matched filter: it responds only to signals that are encoded with a specific PN code. Thus a correlator can be "tuned" to different codes simply by loading a different PN code. If one or more chips in the symbol are damaged during transmission, statistical techniques embedded in the radio can recover the original data without the need for retransmission.

Figure 3. Chip-Level Error Correction



Data Modes

The WirelessUSB LP radio receiver contains a 128-chip correlator, which is either split into two 64-chip correlators or into four 32-chip correlators. This means that the radio can listen for two separate 64-chip PN codes or four separate 32-chip PN codes. Instead of encoding a single data bit in each PN code symbol WirelessUSB LP can encode multiple data bits in one PN code symbol, allowing it to transmit at faster data rates while still using robust PN codes.

Table 1. Data Rate Table

Mode	Data Rate
GFSK	1000 Kbps
32-chip 8DR	250 Kbps
64-chip 8DR	125 Kbps
32-chip DDR	62.5 Kbps
64-chip DDR	31.25 Kbps
64-chip SDR	15.6 Kbps

Dynamic Data Rates and AutoRate™ Receiver

Although the primary design goal of wireless computer peripherals and other low-power wireless devices is data integrity, a very important secondary goal is that of conserving battery life. Dynamic data rates allow devices to adjust to other factors, such as interference, range, and data throughput by using the fastest data rate that still achieves the primary goal of data integrity.

In WirelessUSB LP three different PN code lengths are possible (None, 32, and 64); therefore in order to dynamically change data rates without changing the PN codes that are loaded into the radio, special PN codes called Multiplicative PN codes, have been created that can be used as two 32-chip PN codes or combined to create a 64-chip PN code. Using Multiplicative PN Codes allows WirelessUSB LP to transmit data at any supported data rate by simply setting the data rate register before transmitting; there is no need to load a different PN code.

The full power of Multiplicative PN Codes is realized when packet headers identify the data rate the packet was transmitted at. This allows the AutoRate™ Receiver in the WirelessUSB LP radio to interpret the incoming data at the specified data rate. This makes it possible for different devices to transmit data at different data rates and for devices to even change data rates on a packet-by-packet basis in order to adjust for interference and range. For example, a mouse and keyboard connected to a PC could transmit data at 250 Kbps in order to conserve battery life, while a presenter connected to the same PC could transmit at 125 Kbps in order to achieve greater range, and wireless thermostats and other sensors connected to the same PC could transmit at 15.6 Kbps to achieve maximum range.

Packet Buffers

The WirelessUSB LP radio contains two 16-byte buffers: one for transmission and one for reception. The transmit buffer allows a complete packet of up to 16 bytes to be loaded, and then transmitted with no further microcontroller intervention. Similarly, the receive buffer allows an entire packet of up to 16 bytes to be received with no microcontroller intervention until the packet is completely received.

Auto-Transaction SequencerTM

One of the basic methods to conserve power in RF devices is to reduce the amount of time actively transmitting or receiving data. Along with dynamic data rates WirelessUSB LP has added a transaction mode called the Auto-Transaction Sequencer™, which further reduces the amount of time each radio must be actively transmitting and receiving in order to send a packet and receive an acknowledgment. When transmitting a data packet, the radio automatically starts the crystal and synthesizer, enters transmit mode, and transmits the packet in the transmit buffer. After transmitting the packet the radio automatically switches to receive mode, waits for an acknowledgment, and then automatically reverts to sleep or idle mode when either an ACK packet is received, or a timeout period expires.

Similarly, when receiving in Transaction Mode, the device waits in receive mode for a valid packet to be received, and then automatically transitions to transmit mode, transmits an ACK packet, and the switches back to receive mode to await the next packet.

In each case, the entire packet transaction takes place without any need for microcontroller action; to transmit data the microcontroller simply needs to load the data packet to be transmitted and start transaction mode. Similarly, when receiving packets in transaction mode, the microcontroller simply needs to retrieve the fully received packet in response to an interrupt request indicating reception of a valid packet—invalid or error packets can be simply ignored.

Transmit Power Level

WirelessUSB LP can also conserve power by reducing the power of the transmitter as shown in the following table. Observe that the power can be reduced by more than 50% by transmitting at a lower transmit power level as opposed to transmitting at the maximum power level. Devices that are not at maximum range may reduce the transmit power level to conserve power.

CRC Seed

The WirelessUSB LP radio appends a 16-bit CRC to every transmitted packet to guarantee data integrity. In Transaction Mode only packets with a correct CRC (transmitted CRC matches calculated CRC for the packet) are acknowledged. The CRC is seeded with a predetermined CRC seed known by both the transmitter and receiver, to prevent the possibility of a packet being acknowledged by neighboring WirelessUSB devices using the same channel and PN code.

WirelessUSB Protocol

WirelessUSB is targeted at cost-sensitive, power-conscious computer peripherals and sensors. These devices typically connect to a powered bridge or hub that is not as cost and power-sensitive. Therefore, one of the main goals of WirelessUSB design has been to optimize the power and cost of the peripherals. In order to achieve these goals WirelessUSB uses a master/slave architecture wherein bridges and hubs are masters and peripherals and sensors are slaves. The master is responsible for monitoring interference, and selecting a quiet channel. Masters are typically in receive-mode and only send data to slaves when polled or in response to data from slaves.

Each WirelessUSB master can handle up to 263 slaves, with most computer peripheral devices containing 1-4 slaves and sensor networks containing at least a dozen slaves. WirelessUSB also allows for slaves with multiple masters and devices that can act as master and slave. These features allow the protocol to be expanded to multi-hop and mesh networks in the future. Slave devices typically are fixed-function (e.g. mouse, keyboard, temperature sensor) containing a single transport layer. Masters may be more complex to handle multiple transports.

Co-Location

The WirelessUSB protocol has been optimized for maximum co-location, ensuring that each desk in an office can use WirelessUSB-enabled peripherals without interfering with neighboring systems. This is accomplished by diversifying each device’s use of the frequency, time, and space domains.

The interference immunity algorithms in WirelessUSB have been optimized to avoid Wi-Fi devices without constantly changing channels due to interference from frequency hopping spread spectrum (FHSS) devices, such as Bluetooth. Interference from Bluetooth may cause WirelessUSB packets to be retransmitted, but due to the hopping nature of Bluetooth, WirelessUSB retransmissions will not collide with the next Bluetooth transmission because the Bluetooth device will have moved on to a different channel. Due to the frequency hopping nature of Bluetooth it typically will not cause enough consecutive high noise readings for the WirelessUSB master to change channels.

Conclusion

WirelessUSB LP's agile DSSS modulation along with smart features like AutoRate Receiver, Auto-Transaction Sequencer and on-chip packet buffers will help simplify product design cycle. Other features of WirelessUSB LP such as the integrated boost converter will help lower the overall BOM cost of the end product by at least $0.10.

Additionally, Cypress's excellent after sales support and complete range of kits will make WirelessUSB LP a dream solution for system engineers looking to design their next-generation HID applications.

References

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
http://www.cypress.com