

# Zeroing in on ZigBee (Part 1)

## Introduction to the Standard

*Will the ZigBee standard successfully emerge from the world of technical committees and marketing hype? If so, will it be accessible enough for you to use in simple low-cost projects? In this two-part series of articles, Pete investigates these questions and more.*

In order to be useful, a radio communication standard must be obtainable, affordable, and understandable. It must be able to be interfaced with no more than a few microcontroller I/O lines and be supported by inexpensive development tools. 802.11x is too expensive and complicated. UWB isn't there yet. Single-frequency solutions are affordable, but they're hardly reliable in the face of rapid changes in attenuation, multipath fading, and in-channel interference. Proprietary frequency-agile solutions are available, but this "valued added" approach has been developed by companies hoping to recoup their investments by making you pay for far more than the hardware. Because no one vendor offers a device in every application domain, proprietary solutions have incompatibility built right in.

What you need is an open standard in which the chip makers generate revenue by selling a sophisticated yet easy-to-use chip. The companies will charge for the foundry costs and not the IP in some clever but closed protocol whose details remain hidden and inflexible. Furthermore, the companies won't charge for the IP because the smarts are already open in the form of the IEEE 802.15.4 standard (the communication layers operating in the lower layers of ZigBee). The IP is equally available to any IC developer wishing to make a competing product. This leaves the chip makers' to strive to deliver the best standard implementation in silicon at the best price.

### ZigBee BASICS

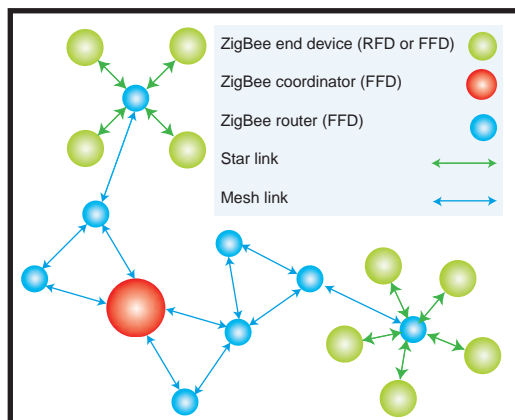
Zowie! What's ZigBee? A hot new personal area network radio communications standard with a catchy name, lots of hype, and no real product associated with it until years after the marketing push hits the mainstream? Not likely. I'm talking about ZigBee here, not Bluetooth. Actually, that's pretty unfair. People are always slighting Bluetooth, but the complaints are often specifically related to an application Bluetooth wasn't designed for. Remember, Bluetooth was originally designed to replace cables between cell phones, laptops, and other devices within a range of 10 m.

Some people say ZigBee got its name from the way bees zig and zag while tracking between flowers and relaying information to other bees about where

to find resources (router bees!). It is designed for mesh networking (see Figure 1). The applications are targeted toward groups of unattended wireless systems in homes, offices, and factories. ZigBee is optimized for low-cost, low-power systems. The compromise is fairly modest bit rates—a maximum of 250 kbps versus the 1 Mbps of Bluetooth version 1.2.

Mesh networking makes up for the limited power of each individual node by leveraging the ability to relay data through nearby cooperating nodes. This happens transparently and provides redundancy and reliability, assuming the density of nodes is high enough. It's a case of the value of the network growing at a greater rate than the rate at which you add nodes to it. The overhead of occasional network reconfiguration takes only a few tens of milliseconds.

Nodes can be full-function devices (FFDs), which embody all the 802.15.4 functionality and features. This allows them to act as a network coordinator or router. An FFD used as a coordinator needs sufficient memory to hold the network configuration, data, and processing power to self-configure the network in addition to its application task. At least one coordinator is required for a network to form. A router stores and forwards messages to and from devices that can't directly swap messages. A coordinator or router would use a lot more power than a simple node at the edge of the network and may



**Figure 1**—Although ZigBee eschews battery-wasting activity by limiting power output, it more than makes up for this by being clever at how data is routed. The full-function devices (FFDs) use the resources of reduced-function devices (RFDs) to self-organize into mesh, star, or tree network topologies. One caveat: this benefit relies on there being enough other nodes nearby.

require line power or be powered from a device with a substantial power supply. For example, a cell phone would be a good choice for a coordinator for a network carried entirely by a person.

Reduced-function devices (RFDs) are limited to a star topology and can only talk to a full-function device. They have a low level of complexity and are found at the edge of the network.

ZigBee uses direct sequence spread spectrum (DSSS) modulation in mixed-mesh, star, and peer-to-peer topologies (including cluster-free) to deliver a reliable data service with optional acknowledgments. The range per node is a nominal 10 m, but popular implementations have a single-hop range of up to 100 m per node line of sight (and farther if relaying through other nodes). ZigBee employs 64-bit IEEE addresses and shorter 16-bit ones for local addressing, which allows thousands of nodes per network.

Association, disassociation, and CSMA-CA channel access with an optional guaranteed time slot for high-priority, low-latency transmissions are transparently handled from the application's point of view, as is AES 128-bit security. Association is the process used to establish a device's membership in the network. With 16 channels at 2.4 GHz offering 250 kbps, 10 channels at 915 MHz offering 40 kbps, or one channel at 868 MHz offering 20 kbps, ZigBee provides modest bandwidth that enables multi-year battery life from a coin cell in designs with a low duty-cycle (less than 0.1%).

## PROTOCOL STACK

Let's take a closer look at the layered structure of the ZigBee protocol stack shown in Figure 2. The lower layers are imported from the IEEE 802.15.4 wireless personal area network (WPAN) standard, which was approved in May 2003. 802.15.4 is a general-purpose WPAN standard that incorporates the lower levels of communication. You can download the standard for free from the IEEE web site listed in the Resources section of this article.

Although designated as a WPAN, ZigBee stretches farther than your desk, so long as the nodes are spread over a wide area with adequate density. 802.15.4 consists only of the physical

(PHY) and medium access control (MAC) layers. ZigBee-specific layers manage routing, discovery, security, and other network-level functions. The ZigBee sublayers also present a sophisticated number of profiles for your application.

## APPLICATION LAYER

Let's start from the top and drill our way down. The ZigBee application layer is comprised of your application-specific code incorporating hardware drivers and whatever else your project requires. You write this into your ZigBee device object (ZDO) according to the standard. In the ZDO you must specify the function of your device within the ZigBee framework and indicate how to initiate and respond to events.

For example, you must define whether you're implementing a reduced function device (RFD) or whether your device is capable of coordinating other devices. You then must decide which type of network layer security to use. Finally, you must hook in the functions that respond to messages from the system. Like Bluetooth, there are a number of profiles that assist with the standardization of top-level device behavior. Smoke alarm manufacturers can adhere to a profile to make an alarm compatible with other devices associated with smoke alarm events.

## APPLICATION SUBLAYER

The application support sublayer, which is at the lowest level of the application layer, handles binding and discovery. The former involves matching devices based on how they're supposed to interact. For example, a light switch doesn't need to control a TV, but an audio-visual system remote can talk to the TV, a light dimmer, or a smoke alarm. The APS also relays messages from devices that cannot talk directly to each other. This is part of the mechanism that enables mesh networking.

## NETWORK LAYER

The network layer is next. As you'd expect, this layer is all about managing

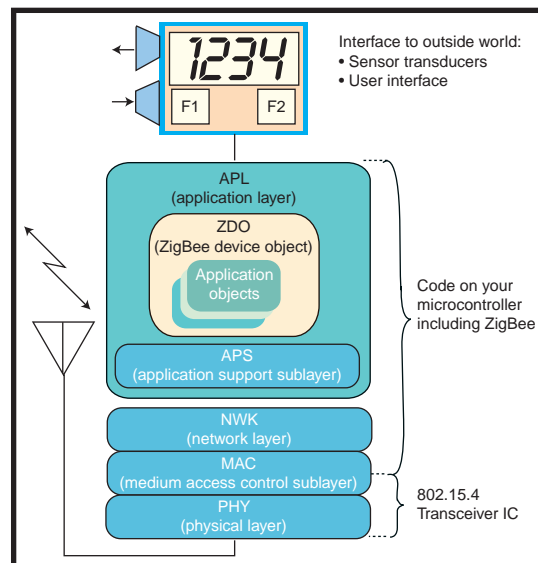


Figure 2—ZigBee provides network structure, routing, and security, while the more basic physical and MAC layers are provided by IEEE 802.15.4.

the network, which is a dynamically rearranging, reconfiguring, and self-healing beast. To maintain a self-managing network, the network layer must constantly keep track of nodes joining and leaving the network. If the node is a coordinator, it assigns an address to a joining node. If the coordinator leaves, another full-function device assumes its role.

Routing and security functions for frames are also implemented at this level. The protocol options may differ between nodes, so the network layer also must configure the protocol stack appropriately. The application layer sets the stack configuration for the network layer below (e.g., security settings).

## MAC LAYER

Discovery is the dynamic process that involves keeping track of other devices in range. ZigBee does this quickly. When you plug a USB device into your computer, or when you approach it with a Bluetooth device, it can take up to 10 s before you can start using it. Joining a network takes as little as 30 ms with ZigBee.

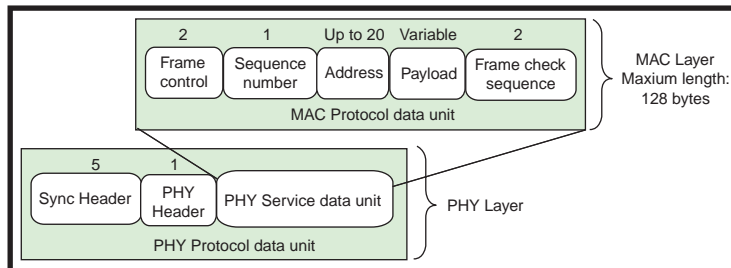
The network, application support, and higher functions of the MAC layer are provided as part of the software development tool chain. C code API functions from lower levels provide functions like setEncryption, sendPacket, Scan, and so on. packetReceived is a typical callback function that your code implements.

The MAC coordinates transceiver

access to the shared radio link. It also schedules and routes data frames. This provides address generation and address recognition, and it verifies frame check sequences. A data frame is shown in Figure 3.

To schedule the frame transmissions in Beaconless mode, 802.15.4 uses

a form of carrier sense multiple access collision avoidance (CSMA-CA). Each device listens before transmitting to decrease the likelihood of two transmissions occurring at once. If a transmitting node suspects that it has clashed with another transmission, it will roll back and reschedule itself to transmit later. Of course, the previously transmitted node will do the same, but the likelihood of the nodes rescheduling the same time slot is small because they base their retransmission delay on the output of a random number generator. If the network is busy, it could mean that the nodes waste a lot of time backing



**Figure 3**—The packet types defined in 802.15.4 are data (shown here) beacon, acknowledge, and MAC command packets. In most implementations, frame processing automatically happens in hardware on the same chip that incorporates the modem and RF transceiver functions.

off and continually trying again.<sup>[1]</sup> For this reason, Battery Life Extension (BLE) mode can limit the back-off exponent to a maximum of two.

With the CSMA-CA scheme in place, nodes only have to expend power transmitting when they have something to say. This enables huge savings in power compared to time-synchronized-only systems such as Bluetooth. In Bluetooth, devices have to keep transmitting periodically to remain synchronized with the network, even if they aren't sending application data. There are various modes to conserve power, but these parking, sniffing, and sleeping modes add a

great deal of complexity.

Of course, some devices periodically require guaranteed access at a high rate, and they will be designed with the increased energy capacity to do so. They may be part of mains powered equipment or simply have bigger batteries. To cater for this, there's an optional

mixed frame format called a superframe, which consists of 16 time slots of equal width chaperoned by a beacon (see Figure 4 on page 20). Any node can grab each of the first nine slots. This gives rise to the possibility of contention. The last seven slots can be reserved for individual nodes, which are then known as a guaranteed time slot (GTS). The coordinator may allow a single node access to more than one GTS in a frame if it has to send a lot of data per frame.

Low-powered devices can still use the beacon frame to gain guaranteed first-time access. The beacon superframes can occur with a period ranging from 15 ms

to 252 s. Peel-and-stick infrared alarm monitors might use this type of long period beacon frame to provide a presence-check heartbeat.

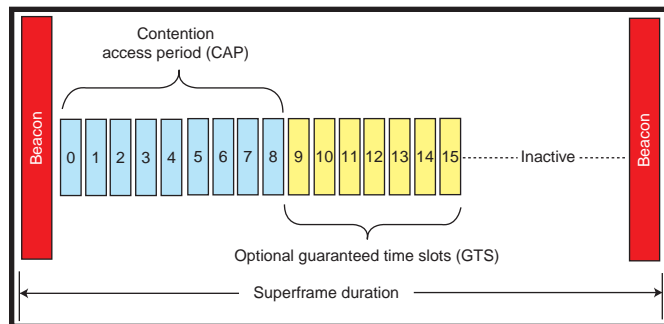
All of the higher-level stuff has been worked out at this point (locations, interconnections, action implementation, and responses). You have the PHY protocol data unit (bits to send), but how do you put them in the air? All prior layers are embodied by the code inside the same microcontroller used for your application.

## PHYSICAL LAYER

Now let's look at what happens in the physical layer inside the modem and transceiver IC. The physical layer takes care of encoding bits to send and decoding received bits with a base-band modem and radio transceiver. In fact, this isn't the entire truth. Things tend to get messy when you go to implement a conceptual model in an efficient manner. There are some raw resources included on the transceiver chip that are actually part of the MAC layer. These hardware resources off-load some of the work that otherwise must be performed on the microcontroller in software.

Other facilities on the transceiver IC have to do with information obtained at the physical layer but used at the MAC layer. For example, received signal strength indication (RSSI) is used for link quality indication (LQI) to control power settings. The clear channel assessment signal is used to implement CSMA-CA functionality.

Now let's set aside these extraneous features and get on with the core job of the transceiver. A DSSS modulator, in which groups of bits are represented by a symbol, generates the modulation of the raw data bits. The symbols are translated into a higher number of bits by mapping them through a look-up table of larger bit-sequences chosen for their mathematical properties. The desired properties include short-run DC bal-



**Figure 4**—If superframes are used, the coordinator will transmit a beacon frame to synchronize the attached devices, identify this particular network, and tell the other nodes how the frames are structured. Any devices wishing to reply in an ad hoc fashion using the CSMA-CA approach can reply in the contention access period (CAP).

ance, autocorrelation, cross-correlation properties, and enough apparent randomness to make the waveform appear as flat noise to a receiver that isn't supposed to be listening.

The reason for discreteness is that a nearby network needs to ignore the signal to concentrate on the transmissions from its own network. In systems where the chipping table constantly changes on a pseudo-random basis, on-air security is also a prime motivator.

In the 802.15.4 standard, the raw data bits are grouped by nibbles to represent symbols. Because 4 bits are represented at a time, there are 16 different symbols in the look-up table numbered from zero to 15. Each symbol corresponds to a 32-bit sequence called a chipping code. Figure 5 illustrates this process using the chipping code for the zero symbol.

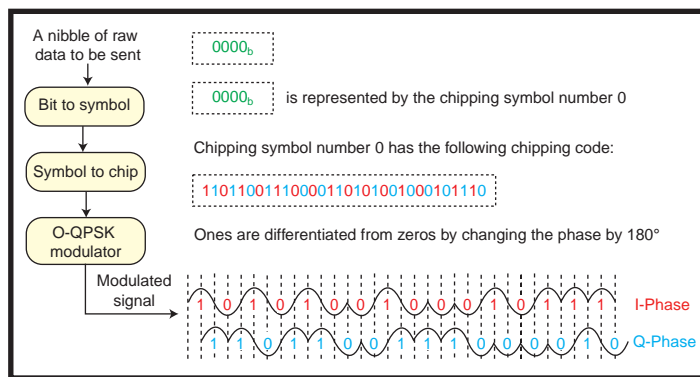
Each symbol now consists of a chipping code of 32 bits called chips, and the rate at which the signal changes has increased greatly, which spreads the signal over a wider bandwidth. After some filtering to reduce the bandwidth, the

chipping codes are presented to the modulator, which carries out half-sine pulse construction.

Offset quadrature phase-shift keying (OQPSK) is used for the 2.4-GHz physical layer. There are two sine-based carriers used in OQPSK. One is in-phase (I) and the other is in-quadrature (Q), which means it's offset by 90°. So, there are sine-based and cosine-based components with which to repre-

sent a symbol. This is advantageous because the chipping code can be split and the two halves can be sent simultaneously. The even chips are represented by the I component and the odd chips by the Q component. The I and Q waveforms are added together and amplified before they're sent through the transmit/receive switch to the antenna.

Data represented by multiple bytes is presented least significant byte first, except for fields associated with security, in which case it's the other way around. The entire process is reversed at the receiver, which is chip-synchronized with the transmitter and attempts to match one of the 16 possible codes to values in the datastream. The closest fitting chip sequence is selected using a statistics-based maximum likelihood technique. This results in the dispersing of the correlated signal in the frequency domain, and the dispersing of any single narrow band interference. This processing gain represents a mathematically powered improvement in the signal to noise ratio.



**Figure 5**—The "O" in OQPSK means the I and Q channels are offset by half a chip period. This limits the possible instantaneous phase-shifting to 90° (as opposed to 180° with straight QPSK).<sup>[2]</sup> This provides a more constant RF envelope and eases implementation of the power amplifier.

Figure 6 (see page 22) shows how the spreading and despreading rendered by modulation/ demodulation minimizes the undesirable effects. The horizontal axis is the frequency.

The chipping rate for the 2.4-GHz PHY is 2 million chips per second. Because 32 chips are sent for every 4 bits of real data, the effective data rate is as follows:

$$2 \times 10^6 \left( \frac{4}{32} \right) = 250 \text{ kbps}$$

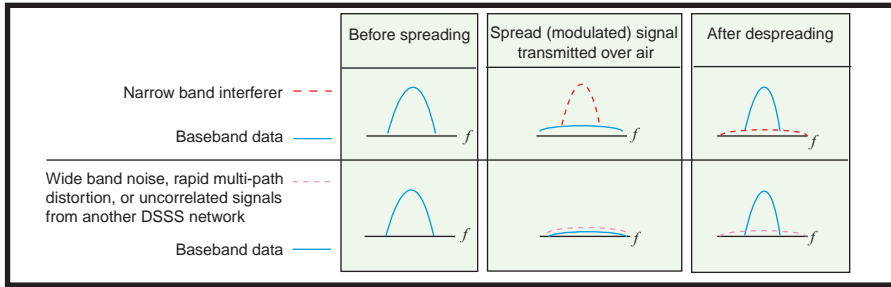


Figure 6—DSSS demodulation despreading is the true geek's version of an amplifier because the signal has been effectively amplified above the noise by brains instead of brawn.

For the 868- and 915-MHz physical layers, the modulation is a binary phase-shift keying (BPSK) and the chipping rate is 0.3 million chips per second. BPSK is simpler because the raw data bits simply alter the instantaneous phase of the carrier. However, the suffix-b proposal may introduce O-QPSK modulation to the lower bands. Figure 7 is a block diagram of a typical application.

## ZigBee ADVANTAGES

ZigBee is so low powered that a typical battery-powered node can wake up, check in, send data, and shut down in less than 30 ms. This leads to an extremely long battery life. For devices with a 30-s check-in period or more, the battery's shelf life will expire before the battery capacity runs out.

If a node is configured for use with a beacon frame and a guaranteed time slot, then on-air time is reduced to 3 ms. This can all be achieved with only one transceiver IC incorporating the PHY and some MAC layer functions and a light-weight task running on the same medium-powered 8-bit microcontroller used for the application. The flash memory requirement for a ZigBee device ranges from 16 to 60 KB depending on the device's complexity, the required stack features, and whether or not it's an RFD or FFD. This is about a quarter of Bluetooth's requirements.

AES 128-bit security and a sophisticated MAC layer supporting CSMA-CA, clear channel assess-

ment, link quality indication, optional acknowledgement, and packet freshness are built in. An addressing scheme can support more than 64,000 nodes per coordinator. Multiple network coordinators can be linked, which means extremely large networks are possible.

## RADIO STANDARDS

Thanks to Tom Cantrell, I don't have to cover the topic of emerging radio standards in too much detail. In "Radio Riot," Tom briefly described ZigBee, Bluetooth, UWB, 802.xx, and Z-Wave (*Circuit Cellar*, 167, June 2004). I'm interested in where ZigBee fits in.

First of all, let me be clear about the comparisons I'll make. I'll make some assumptions about what's important for your projects and then describe the best solution possible.

Sophisticated MAC and a spread spectrum modulation scheme should be built-in. This rules out Z-Wave and any garage door opener type of devices

that rely on OOK/ASK/FSK or require you to cook your own protocol for low-level access to the air interface. I want sophistication, but I don't want to deal directly with the complexity or have my application processor bogged down by low-level transceiver control. Secondly, I assume you want to integrate wireless capability as a small subsystem of a project that uses an 8-bit microcontroller. If you already have a laptop in your design, stop reading right here and plug in an 802.11 PCMCIA card.

Size and power consumption are also important. You should be able to place the transceiver IC and discretes directly into your design without having all the extra garbage associated with evaluation boards getting in the way.

I don't need 20-Gbps data rate or 20-mile range, but I want a bits in/bits out solution, which means that channel encoding, CRC checking, and link quality indication are handed to me on a plate. I also don't want too much of that tricky RF layout stuff. In previous attempts at using RF ICs, my PCBs went straight from my desk to the dumpster. Did I mention I want all this to cost less than \$15 per unit? I'm a demanding kind of guy.

Well, some of these criteria have been met, but not all at the same time. Frequency-hopping radio modems with an RS-232 interface are available, but they're still too expensive.

Other solutions place too much emphasis on the process of trying to transfer as much information as possible in the most sophisticated way. What about the less-sophisticated nodes such as light switches and thermometers that don't need to play video streams or transfer 3-MB MP3 files? What's more, these types of nodes potentially outnumber cell phone-like devices by an order of magnitude. I have three phone-type devices in my house, but 22 light switches.

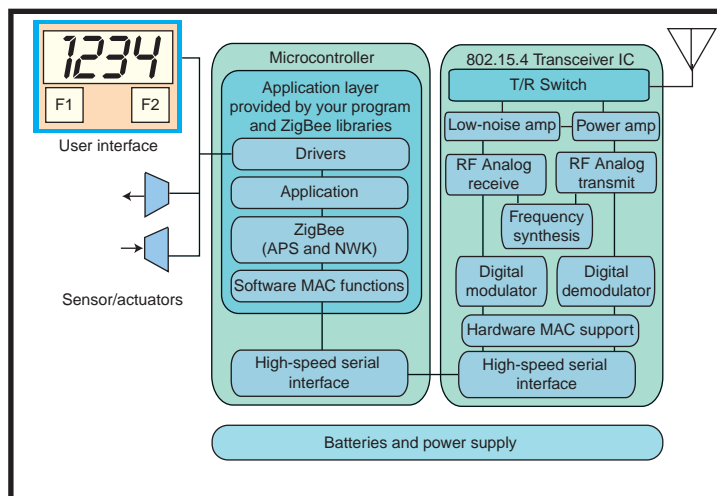


Figure 7—This block diagram of a complete ZigBee node closely follows the hardware implementation of your device in terms of the physical components. In the near future, manufacturers are expected to release a chip that combines the microcontroller and 802.15.4 blocks on a single die. Then an entire ZigBee node truly will be single-chip.

## WHY CHOOSE ZigBee?

Figure 8 shows where ZigBee sits in the pool of current and emerging wireless standards. The “hobby ability” color scheme indicates the likelihood of your being able to use the technology in projects for less than \$15 while being able to wire and code it into a project. The \$15 figure is a one-off cost, assuming you make your own two-layer PCB. Green on the hobby ability scale means the technology is in a format you can use without having to be one of the select few beta-testing partners of a silicon-forging research company. Beige means it’s emerging or isn’t in a format in which you can directly integrate the chipset to a low-end microcontroller for less than \$15.

The newer the technology, the higher the initial cost, and the higher the risk that it won’t become popular. For example, 802.11 used to be technically superior but a little too expensive and complicated to hook up to my PIC-style microcontroller. Today, things are different, but the price remains higher than \$20 and it won’t fit key fob-sized applications. The cost of a naked ZigBee IC plus the necessary discrettes beats those options on physical size and cost if you want to hook it up to a simple 8-bit MCU. However, ZigBee doesn’t compare when it comes to data rate.

At the other end of the spectrum are simple transceivers that use rudimentary modulation techniques such as

OOK, ASK, and FSK. These are easy to use for simple point-to-point links, but anything more complicated requires you to write a complicated protocol to be run in parallel with your application. Some of these low-cost transceivers are frequency-agile enough to implement simple FHSS. But again, implementing this yourself isn’t trivial and almost certainly demands a dedicated base band microcontroller. Even an expert could spend several months setting up the transceiver and coming up with a workable solution for the first few layers of the protocol stack.

For ready-to-go proprietary spread-spectrum solutions, MaxStream leads in the field with the most affordable spread-spectrum radio modems. The 9Xcite 900-MHz wireless OEM module is \$48 in single quantities (\$34 in volume). They’re easily configurable in point-to-point or point-to-multipoint modes, and they operate transparently with an RS-232, RS-485, or USB interface. This is the option to choose if your focus is to get something working within 30 min., and you want an FCC pre-approved solution. The price of proprietary spread-spectrum solutions has dropped over the last two years. If you need to increase the range to several miles in one hop, there are simple upgrade options for that too.

ZigBee’s range is restricted to your house or office. It isn’t designed for a high data duty cycle from each node. ZigBee is much less mature than pro-

proprietary spread-spectrum solutions. However, ZigBee might be a better option if you require the following: small size, cost sensitivity, low latency, low power, and interoperability. But the biggest reason to choose ZigBee is by far the wow factor of implementing cutting-edge technology that is the next big thing.

## PROJECT PREP

Leaving you with this mouth-watering summary of ZigBee’s capabilities isn’t fair, I know. Next month I’ll survey some available development resources for creating your ZigBee projects. I’ll compare affordable chipsets and accessible source code so you can start a practical project of your own. ☒

*Pete Cross lives in Hamilton, New Zealand, where he helps design online sensors that use a range of optical, electronic, and assay-based techniques for measuring biological components in fluids. You can reach him at [pete.cross@clear.net.nz](mailto:pete.cross@clear.net.nz).*

## REFERENCES

- [1] G. Lu, et al., “Performance Evaluation of the IEEE 802.15.4 MAC for Low-Rate Low-Power Wireless Networks,” Workshop on Energy-Efficient Wireless Communications and Networks, April 2004.
- [2] C. Langton, “All About Modulation Part I,” [www.complextoreal.com](http://www.complextoreal.com), 2002.

## RESOURCES

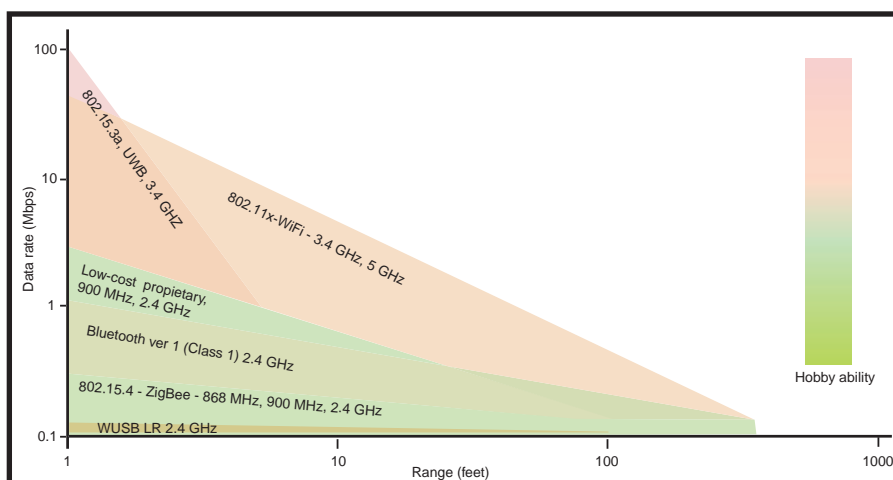
J. Adams, “Meet the ZigBee Standard,” Sensors, June 2003.

D. Benson and M. Gold, “Survey of Selected 802.xx Wireless Standards,” SRI Consulting Business Intelligence, August 2003.

IEEE 802.15 WPAN Task Group 4 website, [www.ieee802.org/15/pub/TG4.html](http://www.ieee802.org/15/pub/TG4.html).

IEEE Standards Association, IEEE 802.15.4-2003, <http://standards.ieee.org/getieee802/802.15.html>.

ZigBee overview, The ZigBee Alliance, [www.zigbee.org](http://www.zigbee.org).



**Figure 8**—802.15.4 zigzags its way around the other wireless options. Although it zips below almost all of the others in data rate, it zings rings around them in terms of the probability that you’ll be able to use a sophisticated radio modem in a project of your own at the chip level.