



## Column: Practical, Secure Embedded Wireless Communications Control Design/Industrial Networking

**Date: 1-11-2009**

**Author: Ned Lecky**

**Lecky Integration**

**ned@lecky.com**

**Work/Cell: 518-258-5874**

I'm working on an extended project for a wind power startup in which we are trying to tie redundant control and monitoring stations into a redundant central controller. We have several weather monitoring stations, for example, as well as redundant motion control systems used to position and monitor the turbine complex. The whole system is a critical controller since properly orienting the turbine blades in high winds is a safety-critical function.

Interconnect is a challenge- most of these stations are separated by 100 feet or more and lightning strikes are not only possible but inevitable and frequent. I decided to consider a wireless interconnection strategy since it would offer complete electrical isolation of the stations and even more: wireless interconnect would completely sidestep the leastglamorous, but most-common failure mode in digital systems, cable and connector failure. Especially in our wide range of environmental conditions, I was more worriedabout isolation, connectors and cables than pretty much anything else. So how about a wireless solution?

Not wait a minute. I'm a pretty good electronics designer and architect. It's not that I'm a genius- it's just that I've been doing this for a long time. Three engineering degrees, fifteen years of embedded software, another ten doing hardware designs, digital microcontrollers, FPGAs, analog circuitry, PCB layout. And for the last seven or so of those I've even added power electronics into the mix, working on controlling converters, inverters, and dealing with mobile and micropower applications.

But wireless? That's, like, with a radio, right?

My grandfather designed radio circuitry at Bell Labs for thirty years, but he can't help anymore. I don't really think a séance is the right format for a technical discussion, especially since psychics rarely have whiteboards in their offices. So I was going to have to figure this one out on my own.

Two issues dominated the choice: how to go wireless, and how to do it securely?

### **How To Go Wireless?**

WiFi, or IEEE802.11 networking is an option. The WiFi standard, however, is really designed to provide interconnectivity to the global TCP/IP network. It is also power hungry and PC-centric, and while it offers excellent security, the security is based on Internet security standards that are power, hardware, and/or CPU-intensive- not wellsuited to embedded system design.

IEEE802.15.4, a set of communication standards and applications commonly called ZigBee, is a relative newcomer to the wireless networking scene. It uses low-power spread-spectrum radios, typically in the 2.45 GHz frequency range, to interconnect devices in ad hoc networks. It is different from WiFi in that:

- 1) It is specifically designed to require much less power than WiFi
- 2) It includes it's own ad hoc networking hierarchy in which nodes may be a primary, an endpoint, or an intermediate router/endpoint
- 3) It continuously reevaluates signal strengths and reassigns routes and traffic to avoid failed (or powered-down) nodes
- 4) Nodes are commonly addressed by unique MAC addresses assigned during manufacture- every node is unique and cannot be mistaken for another
- 5) The communications protocol stack is much simpler than TCP/IP and is therefore much easier to implement on small microcontrollers

Many companies, like Digi, Microstrain and National Instruments, make complete Zigbee-based wireless systems for remote I/O. For the more hands-on integrators, Atmel, Digi, Freescale, Jennic, MeshNetics, NEC, Panasonic, Rabbit, and TI are just some of the major players making modules in the \$20-\$30 range that often include not just the radio but a C-programmable microcontroller: yes, a \$20 postage-stamp-sized board can be added to your design and you have a Zigbee-based embedded system.

### **How To Do It Securely?**

I'm using the Jennic JN5139 module product with an integrated SMA connector for antenna connection. The JN5139 is a low power, low cost wireless microcontroller integrating a 32-bit RISC processor, a fully compliant 2.4GHz IEEE802.15.4 transceiver, 192kB of ROM, RAM sizes from 8kB to 96kB, and extensive analog and digital peripherals. The device also integrates hardware MAC and AES encryption accelerators and mechanisms for security key and program code encryption.

The Jennic encryption coprocessor implements the NIST-approved Advanced Encryption Standard (AES) using a 128-bit nonce (Number Used Once) and a 128-bit key for encryption. The purpose of the nonce is to allow implementation of a counter, date, or otherwise disposable portion of the key that ensures that if an intruder simply repeats an earlier message, the decryption will fail since an old nonce would be embedded in the packet.

By using the AES coprocessor and a standard security model for the nonce/key pair, completely secure communications may be realized between the stations without dramatically increasing CPU load.

And what about jamming? 802.15.4 is based on a spread spectrum technique in which radios are frequency-agile in the 2.4 to 2.5GHz band: they jump from one frequency to another to avoid interference with other radios operating in the same region of space and frequency. This reliability feature also dramatically improves their jam-tolerance. A jamming system would have to transmit on frequencies spanning 100MHz-- from 2.4 to 2.5GHz-- to jam operation, and this is a difficult and costly proposition. Further, the module units support antenna diversity; this is the use of dual antennas which are located in different locations or orientations. The antennas can be alternately selected to avoid destructive interference nodes or blocked pathways in the environment.

My redundant control system is inherently easier to design with the wireless communications, since "cutover" between one weather station and another, or one motion control system and another, or even one main controller and another, only involves changing the addressing field in a transmitted packet. There are no muxes, relays, routers, or switches to interpose and control between the terminal stations. This eliminates software headaches as well as eliminating more single-point-of-failure hardware devices that so often complicate our best-designed redundant systems.



How to get started? To experiment with ZigBee, I'd certainly recommend starting with the \$500 Jennic Home Monitoring evaluation kit that contains five AA-powered evaluation boards, seven Zigbee radio modules, and a fantastic set of C-based development tools with enough sample code and documentation to get you started on becoming a ZigBee expert. While the offerings from many other vendors are excellent, I'm always in the mode of delivering full-featured applications very quickly, and having great documentation, copious sample code, and a fully-working set of demo hardware that can form the core of my own design is about the only way I can feel comfortable about trying something radically new. Give it a whirl!