numbers simultaneously. Speedups of between 10 and 500 are possible over the PC solution of the system.

CJ: FPGA projects, programming, neuroscience, graphics, physics. Where do you find the time for all of this tech work? Do you have any free time?

BRUCE: I do some of the work, but a large number of very good students do a lot of it. I supervise about a dozen Master’s of Engineering projects and undergraduate projects each year, outside of the two classes I teach. In my free time I do some forestry and gardening and try to live sustainably. We have 5.6 kW of photovoltaic solar and 3 kW of solar hot water collectors. We have chickens and honey bees and grow most of our own food.

CJ: Planning any other projects for the near future?

BRUCE: A collaboration with the Cornell Lab of Ornithology may result in some interesting bird-based ad hoc networks. I am working on the simulation of toadfish swim bladder sound production to try to understand some, apparently, chaotic sounds. I am designing a video analysis system for fish activity.

CJ: You interact with up-and-coming electrical engineers, embedded designers, computer scientists, and programmers on a daily basis. What would you say are the “hot” topics exciting this new generation of engineers?

BRUCE: Energy (control, production, and storage); biomedical instrumentation, game design; robotics; parallel or multicore, multithread computing; photonics and optical techniques in computing, and human-computer interaction.

CJ: What is the biggest “growth area” for the embedded industry?

BRUCE: Instrumentation: energy control, biomedical measurements. Geriatric applications for us baby boomers: smart walkers, web-attached pill dispensers, etc. And, of course, mobile devices. A

---

**Need-to-Know Info**

**Knowledge is power.** In the computer applications industry, informed engineers and programmers don’t just survive, they *thrive* and *excel*.

For more information about Bruce Land’s work, the Circuit Cellar editorial staff recommends the following content:

**Floating Point for DSP**
by Bruce Land
*Circuit Cellar 235, 2010*

For DSP and other fine-grained parallel operations, you need to pick a floating-point representation and implement five basic operations. The 18-bit floating point described here allows up to 70 floating-point multipliers and around 150 floating-point adders to be placed on an FPGA. Topics: DSP, Floating-Point Math, FPGA, Conversion, Matlab

Go to: www.circuitcellar.com/magazine/235.html

---

**Hybrid Computing on an FPGA**
by Bruce Land
*Circuit Cellar 208, 2007*

Bruce explains how to simulate the parallel functions of an analog computer on an FPGA. Now you can harness the advantages of parallel execution and a general-purpose CPU on the same chip. Topics: FPGA, CPU, Parallel Execution, Analog Computer, VHDL

Go to: www.circuitcellar.com/magazine/208.html

---

**First low-cost mixed signal oscilloscope!**

Oscilloscope, Spectrum Analyzer, Recorder • Logic Analyzer, Pattern Generator • 2:5 Analog Channels (12-bit, 1 MS/s) • 16 Digital Channels, Square and PWM Generator • Up to 128 Simultaneous DAQ Devices • Free Software Upgrades

---

**NEW! Lowest price USB isolator**

www.poscope.com
Is the Door Closed?
Why Every Safety-Critical Decision Matters

Part of developing a “robust” system involves ensuring your design is safe and secure. Whether you’re building a small embedded app or a complicated door-opening control system for a plane’s landing gear bay, you must protect both your design and the end users by choosing the proper parts and design techniques.

It may be very important to know if the door is closed. The answer is obtained by acquiring data reflecting the door’s status. Every control system needs data to act upon. In embedded controllers, most of the time, the data is provided by various detectors and sensors. Whether the controller itself is analog or digital is a secondary matter. The primary consideration is the type of sensor and its characteristics in terms of the operating environment. Let’s consider an example.

There are numerous mechanical devices on an aircraft whose configuration you must know at any given time. Doors are such devices: landing gear bay doors, passenger doors, and so on. You must also know the configuration of the up and down locks for the landing gear. To obtain their status, you must use some kind of a sensor—a limit switch, for example—to detect the position of the device you’re monitoring. Then, the resulting signal is acquired by the controller, digitized [if necessary], and processed. The trick is to deliver that important signal to the controller consistently and reliably.

Mechanical limit switches—such as micro switches—are rarely seen on aircraft. They are not very reliable in the harsh aerospace environment, and their maintenance by replacement is costly. Instead, proximity detectors are commonly used to detect the positions of moving mechanical parts. This column is not a proximity detection tutorial, but you need to understand the technical basics of proximity detection to follow my example. So, I’ll only briefly describe the principle of inductive proximity detectors as found in many real-world applications.

Figure 1 is a block diagram of a typical proximity detector. A proximity detector consists of an oscillator, a part of which is a sensing coil. As the metallic target attached to a mechanical part being monitored moves toward the coil, the coil impedance changes, and this results in a voltage or current change. That voltage or current change is decoded and the output driver is energized accordingly. The device is usually powered by two wires, and the current driving the device is monitored by the controller. This current is divided into five distinctive bands. Target far is usually a low-current region. Target near is usually a higher-current region. Both appear in Figure 1. Currents less than the minimum target far, higher than target near, or between the targets [noted in Figure 1] mean that the detector is not operating correctly and its data should be rejected. This way a proximity detector not only detects the position of the monitored component, but also indicates whether the device is functioning correctly—a crucial characteristic in safety-critical systems.

SWITCH OR SENSOR?
The proximity detectors come in two basic flavors. A “proximity switch” contains all the components in Figure 1 in one tiny package, interfacing with the controller with a twisted
wire pair. The controller needs only to monitor the current to derive the status. A “proximity sensor,” on the other hand, is merely the sensing coil, while the associated interface electronics are an integral part of the controller.

So which one of the two functionally identical devices should we use? The switch or the sensor? It depends. Make a wrong choice and you may regret it.

Proximity switches are simple to use and to design around. Plus, the total cost of implementation is low. Their interface bandwidth and impedance to the controller are low, so it’s easy to protect their signal from transients, electromagnetic interference (EMI), and all the nasty disturbances bouncing around the aircraft. Properly constructed, a proximity switch generates no measurable interference of its own.

The proximity sensor interface, on the other hand, is complicated. The sensing coil, far removed from the interface circuitry, must be connected by a shielded cable. It can generate unwanted emissions, the small reactance changes of the coil due to the target movement may be obliterated by noise and external interference. The system integration is time consuming and requires experience.

So, why would you even bother with the sensor? The proximity sensor has some important redeeming qualities. Only the sensing coil is exposed to the elements and not much bad can happen to a rugged coil in the tough environment of landing gear, inside a jet engine or a space shuttle. Heat it up to nearly a melting point, freeze it to cosmic cold, immerse it in water, shake it, hit it with a hammer—the coil will continue to work reliably. Try doing the same to the proximity switch! If there is a chance that the operating environment could unexpectedly exceed the relatively mild operating conditions of the proximity switch, you may be in for a costly controller redesign. A proximity switch on a passenger door in an aircraft’s inhabited environment will work just fine, but on control surfaces I’d want to see proximity sensors. I want to know that the thrust reverser doors, for example, are really retracted. Lives may depend on knowing this.

System design and its reliable data acquisition from its peripheral components require not just a thorough knowledge of the system and its operating environment, but also an understanding of the internal characteristics of the devices used to acquire the data. I used a simple, 1-data-bit example to illustrate the point, but the same consideration must be given to the selection of all systems’ peripheral devices.

**FUNCTIONALITY MATTERS**

It is easy to get carried away by the capabilities and sophistication of new technology, such as smart sensors, smart actuators, high-level languages, or distributed processing. One can declare a holy war on anything other than the most advanced methods, insisting that those and only those find their way into his design. Experienced engineers, however, know that the essential part of engineering is a trade-off. Performance, reliability, operating environment, maintenance, cost, and time to market all play a role. Fad architectures and elegant “gee-whiz” designs mean little if they’re not 100% satisfactory. There is no universal best solution, not even for functionally identical tasks like monitoring a status of a door. A good engineer keeps his options open and is not afraid to use even old, time-tested design solutions if they best satisfy the issues at hand. Then and only then can he deliver optimal, robust designs.

George Novacek (gnovacek@nexicom.net) is a professional engineer with a degree in cybernetics and closed-loop control. Now retired, he was most recently president of a multinational manufacturer for embedded control systems for aerospace applications in Canada. George wrote 26 feature articles for Circuit Cellar between 1999 and 2004.

---

**NEED-TO-KNOW INFO**

**Knowledge is power.** In the computer applications industry, informed engineers and programmers don’t just survive, they thrive and excel.

To learn more about George Novacek’s design tips and projects, the Circuit Cellar editorial staff recommends the following:

**Fault-Tolerant Electronic Systems**  
*by George Novacek*  
*Circuit Cellar* 162, 2004

All electronic systems fail. So, to be a successful designer of embedded systems, you must prepare for system failures and glitches. Topics: Fault Tolerance, Failure, Built-in Test, Redundancy

Go to: [www.circuitcellar.com/magazine/162toc.htm](http://www.circuitcellar.com/magazine/162toc.htm)

**Time-Triggered Technology**  
*by George Novacek*  
*Circuit Cellar* 155, 2003

Clearly, the older communications protocols used by the aerospace industry are becoming increasingly expensive to implement. But, as George explains, TTP technology could change the game. Topics: Time-Triggered Protocol, Data Bus, Frame

Go to: [www.circuitcellar.com/magazine/155toc.htm](http://www.circuitcellar.com/magazine/155toc.htm)
Embedded Systems

High-End Performance with Embedded Ruggedness

Unbrickable design

3x faster
and backward compatible with TS-72xx

TS-7800
500 MHz ARM9
- Low power - 4W@5V
- 128MB DDR RAM
- 512MB high-speed (17MB/sec) onboard Flash
- 12K LUT customizable FPGA
- Internal PCI Bus, PC/104 connector
- 2 host USB 2.0 480 Mbps
- Gigabit ethernet
- 10 serial ports
- 5 ADC (10-bit)
- Sleep mode uses 200 microamps
- Boots Linux 2.6 in 0.7 seconds
- Linux 2.6 and Debian by default

$229 qty 100
$269 qty 1

TS-TPC-7390
7" Color Touch Panel Computer
- Low Power, Industrial Quality Design
- Mountable aluminum frame
- 200 MHz ARM9
- 64MB SDRAM (128MB opt)
- 512MB Flash w/ Debian Linux
- 800 x 480 video core
- Dedicated framebuffer- 8MB RAM
- Audio codec with speaker
- Boots Linux 2.6 in less than 2 seconds
- Unbrickable, boots from SD or NAND
- Runs X Windows GUI applications
- Runs Eclipse IDE out-of-the-box

$449 qty 1

More Touch Panel Computers on our website

- Over 25 years in business
- Open Source Vision
- Never discontinued a product
- Engineers on Tech Support
- Custom configurations and designs w/ excellent pricing and turn-around time
- Most products stocked and available for next day shipping

Design your solution with one of our engineers (480) 837-5200