

Buried treasure: Unlock the processing power of wireless modules

Jacques Suire, Sierra Wireless



When looking to add cellular connectivity to an embedded system, many designers choose a wireless module like the one shown in Figure 1, since the device contains all the analog and digital circuitry necessary for producing, transmitting, and decoding data over a cellular network.

The developer can interact with the module using familiar interfaces like UART and USB, and doesn't have to be concerned with the complex analog aspects of cellular communications. Compared to a custom RF solution, a wireless module can shorten time-to-market, be less expensive to debug and prototype, and be more cost-effective in mass production.

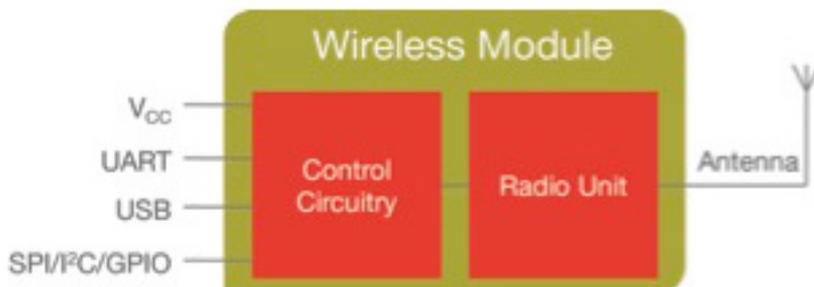


Figure 1. High-level block diagram of a typical wireless module

Oftentimes, the quickest way to add wireless connectivity is to use the module in combination with a standard microcontroller. This is especially true when starting with an existing design. The upgraded, cellular-equipped design can be ready to ship in very little time. However, since the wireless module and the microcontroller are usually the two highest-priced items in the bill of materials, using both in a system can be expensive.

Fortunately, designers have an alternative. Taking a closer look at the product offerings reveals that many wireless modules are actually capable of doing much more than just managing cellular communications. This is because wireless modules typically integrate a highly optimized chipset, originally designed for use in low-end

Buried treasure: Unlock the processing power of wireless modules

Published on Electronic Component News (<http://www.ecnmag.com>)

and mid-range cell phones, that includes a 32-bit ARM microcontroller.

Hidden talent

Figure 2 gives a more detailed look at the sample wireless module shown in Figure 1. The control circuitry is actually an ARM9 core, one of the most widely used control architectures in all of embedded.

The task of managing cellular communications typically uses less than 20% of the ARM9's total processing capacity, so that leaves excess available for doing a lot of other things. By taking advantage of this often overlooked processing power, designers can use the module as both the central processor and the modem, and eliminate the standalone microcontroller.

The fact that wireless modules usually include an ARM9 core isn't always obvious from the marketing literature. Wireless performance is often emphasized over processing power, with product flyers highlighting specs for cellular performance. The module's sub-components aren't always mentioned, and it can be difficult to find a block diagram. This is understandable, since wireless modules are, after all, designed to operate as drop-in solutions for cellular connectivity.

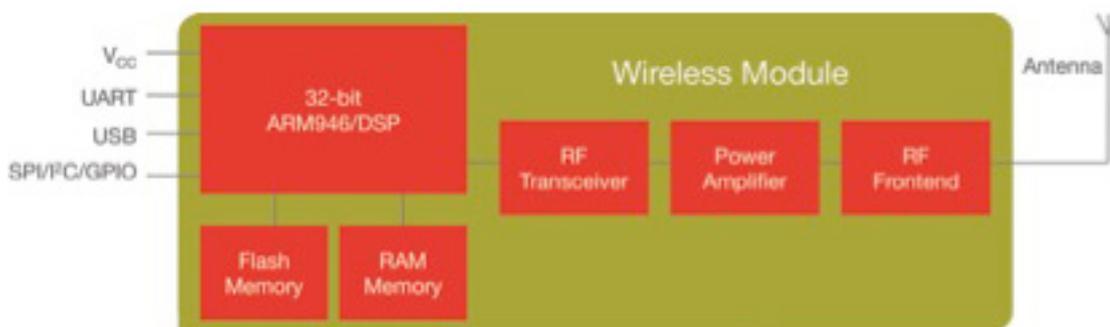


Figure 2. More detailed block diagram of wireless module, showing ARM9 core

On the other hand, you can't always assume that a given wireless module can take on a whole application, since not all modules provide access to the excess capacity of the ARM9. Some modules are configured to operate more purely as modems, with only limited support for customization. These commodity-like products use the same basic chipset as fully programmable modules, but are shipped with only a subset of the ARM9's pins connected to the external package. The full potential of the ARM9 core remains inaccessible.

Hardware performance

The three modules listed in Table 1 are all currently available and have already been configured as microcontroller/modem combinations in large-scale deployments. Specific hardware requirements will, of course, vary by application, but the specs given in Table 1 can act as a guide.

Buried treasure: Unlock the processing power of wireless modules

Published on Electronic Component News (<http://www.ecnmag.com>)

Table 1. Sample integrated features and performance characteristics

	Module A	Module B	Module C
Air interface	GSM/GPRS	GSM/GPRS/EDGE	HSDPA
Processor	ARM946/DSP	ARM946/DSP	ARM926EJS/QDSP4
Core frequency	26 MHz	26 or 104 MHz	184 or 230 MHz
I/O voltage	1.9 to 2.8 V	1.8 to 2.8 V	1.8 V
Standby power consumption	2.5 mA	2.4 mA	< 2.0 mA
UART	2	2	1
USB 2.0	1 (full-speed)	1 (full-speed)	1 (high-speed)
SPI	1	2	1
I ² C	--	1	--
GPIO	44	44	7
Temperature range	-30 to 70 °C	-40 to 85 °C	-40 to 85 °C
Footprint	25 x 25 x 3 mm	30 x 40 x 4 mm	25 x 30 x 2.35 mm

Software programmability

The wireless module should give maximum flexibility for configuring the ARM9 core. Most embedded systems benefit from a real-time operating system (RTOS) to ease the application design. The best option is a pre-emptive RTOS that supports a familiar programming language, such as ANSI C or C++. Multi-tasking functions can help simplify the software design and maintenance. The RTOS should also be available with a rich, PC-based Integrated Development Environment (IDE) that can create a project, perform downloads, and debug the design. All the software should stay royalty-free to keep the total cost of ownership low.

Look for other time-saving tools, too, such as additional libraries running on the RTOS, since these can make wireless development much easier. For example, libraries might include Internet protocols to exchange data with a server, security protocols and services to protect your sensitive data locally and during a server connection, or location services to calculate system positioning.

Management Services

Some wireless modules are also available with a cloud-based management service that lets you monitor and upgrade deployed devices remotely, using a web portal. Management services can be useful with any system that has a wireless module, but they're particularly helpful when the entire application is on the module, because over-the-air monitoring and updates can be used to evaluate and modify the application itself, not just the telecom-related functions.

Conclusion

Using the excess processing capacity of a wireless module to replace the system microcontroller can yield a cellular-equipped design that is smaller, more efficient, and much less expensive to produce. Not all wireless modules can be configured as application microcontrollers, so choose carefully. It's important to consider the hardware specs and options for software programming, and, ideally, to find a module supported by management services that lower the cost of deployment and maintenance.

Buried treasure: Unlock the processing power of wireless modules

Published on Electronic Component News (<http://www.ecnmag.com>)

Source URL (retrieved on 07/23/2014 - 6:49am):

http://www.ecnmag.com/articles/2012/07/buried-treasure-unlock-processing-power-wireless-modules?qt-video_of_the_day=0