Embedded modems: The choice is hard

Is a hardware or a software modem the best for your application?

By Paul Brown, Silicon Laboratories

Choosing between hardware and software modems means managing cost and risk. Technology options impact international compliance, size, features, speed, and cost. They also impact the level of necessary customer support. An outline of the major factors in play can help you determine the best modem for your application.

The first step in choosing the best modem technology for an embedded application is understanding clearly the available system resources and any limitations that will impact your choice (see sidebar “Modem Basics”). A full-featured hardware modem consumes virtually no system resources. It contains a controller, a DSP, data, and program memory. The embedded-system host processor communicates with the modem through a serial or parallel interface and must pass to the modem only the initial setup strings, which are commands that control the modem and the data. The modem processor handles all modem functions, including equalization, echo cancellation, modulation, demodulation, error correction, and compression, without involving the host processor or the embedded operating system. The modem simply sends and receives data. The choice of a serial or parallel interface for the hardware modem depends strongly on other system-interface requirements, the processor, available interfaces, and any difference in system software complexity, overhead associated with the interface selection, or both.

You implement software modems, on the other hand, on the host processor. They must run concurrently with any other software, such as the user interface, error correction, and compression that are active during data communications. System memory stores the soft modem code; it must be large enough to hold the modem and other application code and fast enough to provide proper execution of all software. Additionally, you must fully understand the capabilities and limitations of the operating system. Many embedded operating systems are simple, adequate only for running the existing application software, and many have significant difficulty performing the multitasking necessary to run time-critical modem code concurrently with other required software.

Modem code requires guaranteed interrupt timing and adequate interrupt duration to guarantee a maximum processing latency. You must take care to develop a detailed understanding of the effective processor MIPS the modem will require. Modem code executes many multiply-accumulate operations. DSPs are designed to perform this type of operation and can complete a multiply-accumulate operation on each clock cycle. A general-purpose processor may require three to five clock cycles to perform a multiply-accumulate operation. A “30-MIPS processor” may really be capable of only 6 to 10 MIPS with respect to the modem code.

Other important considerations are the AT command set and response codes. Windows applications require the ITU (International Telecommunications Union) standard V.250, but few embedded modems strictly follow that standard. A hardware modem comes with a predefined AT command set that includes provisions for all features that the modem supports. Embedded soft modems tend to have fewer features and therefore require a less detailed AT command set. You must carefully consider the AT command set, response codes, and timing if you intend to have an alternative modem solution, such as a higher speed option. In such cases, provisions will likely be necessary in system code to adapt to the peculiarities of different modems’ commands, response codes, response times, register definitions, and value ranges.
Most modems will identify themselves with an ATI command that allows the system software to choose the correct instruction set.

The system clock becomes critical in soft-modem applications. To meet ITU interoperability requirements, the modem clock must be stable to at least 100 ppm. Therefore, the system crystal or clock source must be accurate to within 100 ppm, including initial accuracy, temperature drift, aging, and loading capacitor tolerances.

**Table 1** gives approximations for the MIPS, program memory, and data-memory requirements for typical modem functions on a DSP processor. MIPS requirements for features such as V.42, V.42b and V.32b, for example, are additive, because they are active simultaneously. MIPS requirements for different modulations, such as V.32b, V.22b, and Bell 212, are not additive, because multiple modulations are not active simultaneously. The program-memory requirements are additive for all functions the modem code will perform. Every modem requires an AT command parser and modem operating system. The MIPS listed in the table can vary significantly depending upon the architecture, instruction set, and precision of the target host processor.

**DAA technologies**

Like the choice between hardware and software modems, the choice of DAA (direct-access-arrangement technology) is a key system decision. The DAA impacts modem performance, regulatory compliance, feature set, pc-board area, and cost. The primary function of the DAA is isolating the SELV (safety extra-low-voltage)-system circuitry from the high-voltage TNV (telephone-network voltage). The relatively simple “tried-and-true” transformer DAA is the oldest method of providing this isolation (Figure 1). Transformers have typically been used for very basic, no frills, low-speed modem applications without enhanced protection, international compliance requirements, or space limitations. The problems with transformer DAAs include the difficulty of adding features, such as parallel phone-off-hook detection or Caller ID. Typically, international compliance requires different component-population options for many countries, which increase costs.

Hardware modems typically come with an integrated DAA solution. Selecting a hardware modem, therefore, also means selecting the DAA technology. Software modems, on the other hand, offer more DAA-choice flexibility, because there is no predefined DAA solution. Although a transformer-based DAA can work with a soft modem, some features, such as the parallel phone-off-hook detection that solid-state DAAs include, add cost to a transformer DAA and complexity to soft-modem code. Future feature upgrades are difficult or impossible with a transformer DAA without a complete hardware redesign. Therefore, it is not the best choice to limit the inherent flexibility of a software modem with an old and inflexible DAA technology.

On the other hand, capacitively coupled DAAs offer significant advantages in terms of modem performance, software-programmable country support, enhanced safety/surge performance, and a rich feature set. They are small, cost-effective, and suitable for all modem speeds. Capacitively coupled DAAs offer a variety of features, including Caller ID, on- and off-hook intrusion detection, valid-line detection, and overcurrent detection. Another versatile feature of capacitively coupled DAAs is their ability to work interchangeably with hardware and software modems with the simple substitution of a system-side interface device. The line-side (TNV) circuitry is unaffected. This news is good, because most of the compliance-related circuitry is on the line side. Figure 2 illustrates the relatively minor differences between the circuitry for a hardware modem and the circuitry for a software modem. What’s interesting is that you can make the hardware/software modem selection as a board-stuffing option.

Other DAA technologies fall between the transformer DAA and the completely capacitively coupled DAA. Some recent “solid-state” DAA offerings use both capacitive and transformer coupling. This approach offers some enhancements over the traditional transformer approach
but still requires a transformer and significantly more external components than a pure capacitive-isolation technique. Additionally, this technology is tied to a particular hardware modem and is not readily adaptable to a software-modem solution. If you are considering doing your initial design with a hardware modem and transitioning to a soft-modem solution to reduce costs, choosing the best DAA technology is crucial to make the transition smooth.

**Regulatory compliance: Check the credentials**

All modems, whether hardware or software, are connected to the PSTN (Public Switched Telephone Network) and are subject to government regulation in every country in which they are used. New products require initial certification. Regular (usually annual) certification checks are performed for the life of the product to ensure production units remain compliant. Failure to comply with the regulatory requirements can result in significant economic penalties. Although efforts such as the CTR21/TBR21 standard have made significant progress over the last decade to reduce the differences in requirements between countries, particularly in Europe, dissimilarities remain. The bottom line is that any modem solution must comply with the regulatory requirements of the country in which it will be used in and, at least, have provisions for compliance in all countries where use is anticipated.

You can divide regulatory compliance into three categories. The first encompasses the technical requirements for equipment to properly interoperate with the PSTN in a given country. The many differences and exceptions make having a global transformer DAA nearly impossible. The only universal DAA solution must be software-programmable to adapt to the special country requirements without hardware modifications. Programmability strongly favors capacitively isolated DAAs.

The next group of requirements is EMI/EMC (electromagnetic-interference/electromagnetic-compatibility) compliance. Fortunately, this group is largely standardized. The modem can contribute to radiated emissions itself or, with an improper system design, by radiating nonmodem interference. EMI/EMC compliance is a strong function of the DAA technology, pc-board layout, and system design.

The last group of regulatory requirements is safety compliance. It focuses on preventing dangerous voltages from coupling to the low-voltage side of a system, where an operator may come in contact with them. DAA isolation technology, discrete component selection, pc-board-trace widths, trace spacing, pc-board-to-chassis spacing, and good system design are required to pass safety requirements and to provide users with proper protection.

Finally, changes to the modem code or DAA circuitry affect certification. Here, the programmability of a capacitively isolated DAA is a significant advantage. Manufacturers have pretested the programmable features to comply with the country requirements, and no hardware design change is necessary.

**Upgradeability**

There are many reasons for upgrading an embedded modem, including bug fixes, new product-feature additions, speed enhancements, and hardware- to software-modem conversion. Careful planning can make these changes relatively straightforward.

If you plan to upgrade to a higher speed modem, whether it’s a hardware or a software modem, check to see whether the AT commands and register settings are compatible for the different speeds. If the AT command set is the same except for commands relating to the higher speed, the impact on user interface/application software will be minimal. Also verify that the S-Register and other register definitions and value ranges are compatible. Upgrading an existing embedded system to a higher speed software modem could be very expensive or impossible depending on system resources and the DAA implementation. And the option does not even exist if there are too few MIPS or too little program or data memory on the host.
processor to run the higher speed code. Using an unnecessarily powerful processor and providing extra memory in the case of a speed upgrade might be necessary in the future. However, it could also be expensive enough to make soft modems far less attractive. If you need a new modem chip for higher speed operation in a hardware modem, be sure the higher speed chip is pin- and register-compatible. This feature will allow speed upgrades as a stuffing option and minimize regulatory problems.

Hardware-modem solutions, such as the one shown in Figure 2, are extremely versatile. Not only can you upgrade the modem speed by simply choosing the appropriate modem chip, you can change this modem into a software-modem DAA solution by substituting a DAA interface chip for the modem chip. The line-side circuitry surrounding remains completely unchanged.

Costs and risks

The need to quickly develop and introduce new products places increasing demands on embedded-systems developers to get it right the first time. This need favors a tried and true hardware-modem solution. On the other hand, cost is driving dial-up-capable embedded systems toward adopting soft modems, following the path that PCs blazed. The idea of piggybacking a software modem on the host processor for free is attractive. Reality, however, can be quite different.

Although they initially seem simple and inexpensive, embedded soft modems, more so than their PC counterparts, carry hidden costs and risks. Considerations including the operating system, processor loading, additional memory requirements, development and support costs, and time to market make the decision difficult. Hardware-modem chip sets are proven building blocks and can accelerate product introduction by eliminating the need to design, test, and run the gauntlet of regulatory approvals on a new, untried soft-modem implementation. Hardware modems can even offer a soft-modem migration path as a stuffing option. Hardware modems, however, may carry a slightly higher unit cost. When choosing an embedded modem solution, the following considerations can help you manage the risk/reward ratio.

Hardware modems have a leg up when it comes to risk. They are a complete solution to the embedded-communication problem and are proven with reference designs and recommended pc-board layouts that you can cut and paste into any design. Typically, manufacturers have extensively tested these products in laboratories, and consumers have tested millions of units in the field. Manufacturers, therefore, have likely identified and corrected operational and performance problems. Hardware modems also have a proven track record in international-compliance certifications. The fact that the hardware-modem solution has passed certification testing in the past makes it much more likely that your modem implementation will pass on the first attempt.

One of the most significant advantages of a hardware modem is comprehensive one-stop application-engineering support. The modem chip, modem code, and DAA solution come from a single source. Additionally, the complete solution is well-documented with data sheets and application notes. Hardware modems also offer the simplest solution for future speed upgrades. Even if the system host processor has sufficient MIPS and memory to allow a software speed upgrade, and the design includes an appropriate DAA technology, a software-modem upgrade will require rigorous testing. Hardware modems have minimal impact on system software, are feature-rich, and can be very flexible with global compliance.

The cost of this reduced risk is the slightly higher price tag for the modem hardware. The best way to manage hardware-modem cost is to use the lowest speed modem adequate for the application at hand with pin-compatible chip upgrades for future speed enhancements. This strategy will also help to minimize future development costs.
A software modem, on the other hand, may appear less expensive, but be certain you evaluate all costs before going soft. A software modem that you port to your host processor and proprietary operating system is a new and largely untested product. Even if others have successfully used versions of the soft-modem code in the past, the integration onto your processor with your operating system, application software, system clock, and DAA is new and unproven. Ask for a comparison of performance data for the soft modem with data for commercially available stand-alone modems. Require that the modem-code vendor performs a full suite of CCR (call-connect reliability) and throughput (or bit/message-error rate) tests on your finished system and compare the results with previous implementations of this modem code and commercially available modems. The data will give you a good benchmark of where your modem implementation ranks with others.

The cost advantage of a software modem is the trade-off between the cost of the hardware-modem chip and the software-modem costs, including license fees, software development and integration, extra host-processor MIPS and memory, additional application software and operating-system complexity, and additional support costs. Certification costs can also be higher, because the new soft-modem software has not previously been certified, at least in its present form.

The more experience a modem solution has in the real world, the lower the risk of difficult support problems. It can be difficult to replicate a customer’s problem, find a solution, and translate that solution into a code update.

The choice between a hardware and a software modem for an embedded system is difficult and requires a full understanding of the risks and a well-thought-out plan to manage them. In most cases, a hardware modem offers the most compelling solution with the fewest risks. Table 2 summarizes the major risks and suggested strategies you can use to manage that risk. If a software modem truly offers a significant cost advantage and is the best alternative, choose the best DAA technology for the long term so you can reuse as much of your engineering investment as possible on future products and minimize redesign effort. Finally, be absolutely certain you have comprehensive customer support no matter which modem implementation you choose.

Author’s biography
Paul Brown is a modem-applications-engineering manager at Silicon Laboratories, where his responsibilities include modem-product development and customer support. He has a BSEE from San Jose State University and has completed graduate work at the University of Colorado.
Figure 1—The relatively simple, “tried-and-true” transformer DAA is the oldest method of isolating the SELV (safety extra-low-voltage)-system circuitry from the high-voltage TNV (telephone-network voltage).
Figure 2—The differences between the circuitry for a hardware modem and the circuitry for a software modem are relatively minor.

Table 1—Approximate system requirements for modem functions

<table>
<thead>
<tr>
<th>Function</th>
<th>MIPS</th>
<th>Program memory (k words)</th>
<th>Data memory (k words)</th>
</tr>
</thead>
<tbody>
<tr>
<td>V.92*</td>
<td>50</td>
<td>67</td>
<td>12.5</td>
</tr>
<tr>
<td>V.90**</td>
<td>38</td>
<td>55</td>
<td>12.0</td>
</tr>
<tr>
<td>V.34</td>
<td>38</td>
<td>40</td>
<td>12.0</td>
</tr>
<tr>
<td>V.32b</td>
<td>18</td>
<td>14.5</td>
<td>5.2</td>
</tr>
<tr>
<td>V.22b</td>
<td>5.0</td>
<td>6.0</td>
<td>0.5</td>
</tr>
<tr>
<td>V.21</td>
<td>4.0</td>
<td>2.0</td>
<td>0.5</td>
</tr>
<tr>
<td>V.42 and V.42b</td>
<td>8.0</td>
<td>8.5</td>
<td>8.5</td>
</tr>
<tr>
<td>V.44</td>
<td>8.0</td>
<td>3.0</td>
<td>5.0</td>
</tr>
<tr>
<td>Basic AT commands and parser</td>
<td>&lt;1</td>
<td>5.5</td>
<td>4.0</td>
</tr>
<tr>
<td>OS and miscellaneous</td>
<td>3.0</td>
<td>1.0</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Note: *V.92 includes V.90
** V.90 includes V.34

Table 2—Risk management
<table>
<thead>
<tr>
<th>Risk</th>
<th>Hardware modem</th>
<th>Software modem</th>
<th>Risk management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development cost/time</td>
<td>Low</td>
<td>High</td>
<td>Know your system-resource limitations and modem requirements. Plan future products to reuse engineering effort.</td>
</tr>
<tr>
<td>Certification cost/time</td>
<td>Low</td>
<td>Moderate</td>
<td>Get copies of test reports and review for potential problems in advance of actual certification testing.</td>
</tr>
<tr>
<td>Support cost</td>
<td>Low</td>
<td>Moderate to high</td>
<td>Have a clear customer-support plan in vendor contracts or agreements.</td>
</tr>
<tr>
<td>IP royalty cost</td>
<td>Low</td>
<td>Low to high</td>
<td>Be sure code cost includes IP cost and negotiate a meaningful indemnification agreement.</td>
</tr>
<tr>
<td>Customer-support effort</td>
<td>Low</td>
<td>Moderate</td>
<td>Check modem production history and formal test results compared with commercially available stand-alone modems.</td>
</tr>
<tr>
<td>Bug Fix</td>
<td>Low</td>
<td>Moderate</td>
<td>Have a clear and well-thought-out field code update capability and method.</td>
</tr>
</tbody>
</table>

This is a sidebar to the article that can be included if there is room:

Modem basics

Modems have three basic building blocks: a controller, a DSP (data pump) and a DAA (direct access arrangement). Figure A illustrates a typical modem block diagram.

The controller provides several vital functions, including AT command parsing; DAA control; connect sequence control; DCE (data-communication-equipment)-protocol control; the DTE (data-terminal-equipment) protocol, such as speed, data length, start and stop bits, and parity; ring detection; DTMF (dual-tone multifrequency) control; call-progress monitoring; error correction; and data compression. The controller and DAA
handle special features, and the controller handles all interaction between the host processor and the modem.

The DSP (data pump) is primarily responsible for modulation, equalization, and echo cancellation. In fact, the DSPs in most of today’s hardware modems are powerful enough to perform both the data-pump and the controller functions. The DAA is necessary to connect the DSP to the telephone network, provide protection to the user, and implement special features.

Figure A depicts the three basic modem configurations. In the controller-based (hardware-modem) configuration, the modem hardware implements all modem functions. In a controllerless modem, the host processor handles the controller functions, and a separate DSP implements the data-pump functions. Finally, in a software (soft) modem, the host processor implements both the controller and the data pump. The host processor communicates with the hardware-modem controller through a serial or parallel interface with a standard AT command set.

Hardware modems are self-sufficient and require no processing power from the host aside from issuing commands to cause the modem to dial, hang-up, or read its status. Controllerless modems have dedicated hardware for the data pump and DAA but implement the controller function, including the AT command parser and UART, error correction, and compression, as software on the host processor. Software or “soft” modems place a significant resource load on the host processor and system memory and have operating-system implications due to the time-critical processing required to perform the modulation, demodulation, and echo-cancellation functions.