

# Embedded.com

## Five things to keep in mind when selecting an embedded mobile/consumer SoC

Sorting through system-on-chip design options such as power management, performance, peripherals, memory, and bus structure

By Ron Stieger, BSQUARE Corporation

If you were to take the case off a newer smartphone or PDA you would undoubtedly find a SoC processor inside. That's because SoC processors often provide the fastest route from design to the marketplace, offer tremendous flexibility in adding features in future designs, and can offer cost, form factor, and other advantages that would be impractical or impossible to realize with a system involving many discrete components.

But the advantages of SoC processors are not limited to mainstream consumer applications. These processors can pay sizable dividends in embedded applications of all sorts. The key is understanding which SoC processor is appropriate for your embedded device and giving thought early on to the best way to integrate it.

SoC applications processors are designed to integrate a large number of peripherals with the processor core. These processors, which are typically built around a RISC core " commonly ARM, though MIPS and other cores are also used " provide memory management units and other features required by Windows CE, Linux, and other embedded operating systems.

Supported peripherals typically include a display controller, static and dynamic memory controllers, various serial interfaces (e.g., UART, SPI, USB), and add-on slot controllers (e.g. PCMCIA, CompactFlash, SDIO). Some SoC processors may also include keypad controllers, multimedia accelerators, and other features.

With all these peripherals, there is a great deal of complexity in building a platform around a SoC processor. There are usually multiple ways to map the functional blocks to the I/O pins. The operating system needs to understand what is being used and how, which means drivers need to be customized to match the schematic design. There are also low-power modes that you will want to take advantage of.

Here are five things to consider when selecting SoC processors for your embedded systems:

### 1. Choose Wisely

Choosing a SoC processor is the most important decision you will make and requires careful consideration. There are many options available, but there are ways to quickly focus on what is

the best for your application.

- The first question you should ask is whether a particular SoC processor is even available for your device. Some SoC manufacturers sell only in tremendous volume to a handful of mass-market device makers. Their chip may be functionally perfect for your device, but it may not be available to you. Depending on your volume and procurement strategy, you may require a processor that is available through distributors. Start your decision-making process by making a list of SoC processors that you can buy.

Second, look at what markets each SoC processor is targeted toward. For example, a particular processor may be loaded with a variety of high-end multi-media features. That's great if you're building a device that will make use of those features. If not, those capabilities may simply drive up your cost and cut into your margin without providing any benefit.

You may even want to create a spreadsheet where you list the features you need in one column and mark which SoC processors meet those needs. If you aren't able to find a part that has all the features you want " and only those features " you may need to make tradeoffs between the cost of the processor and the cost of external peripherals to provide the missing features.

Finally, look at the level of support available. Using a SoC processor is not easy. There is a lot to learn, and learning it on your own will take time " time that you may not be able to afford in your rush to market. You will want some level of support and a reference board to help you speed up the design process, test custom circuitry, develop software, and finalize your design. Whether they come from the processor manufacturer itself or a third party, make sure a quality reference board and other forms of support are available for any SoC processor you consider.

### 2. Start with a Reference

The availability of an appropriate reference design or development platform (Figure 1, below) should be one of the key factors you use to decide on a SoC processor. A reference design gives you a key head start for developing application software to run on your embedded device. And it gives you a very good start in developing your own hardware design.

You can see a working design in action before you build any-

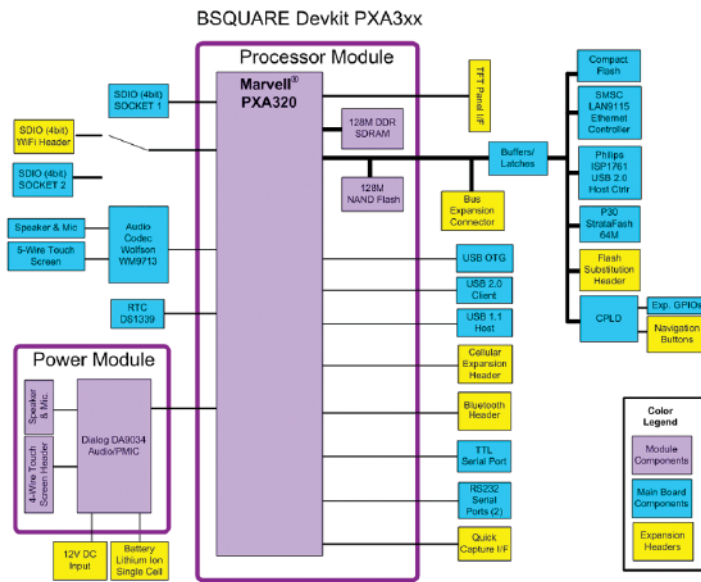


Figure 1

thing. You can see how someone successfully incorporated various components, and use their framework as a starting point for your own design. With a reference design, someone else has already navigated the quirks of the hardware to build a functioning device. All you need to do is figure out how to customize it to meet your specific needs.

Just as the SoC processor decision needs to be made based on the needs of your particular device and the functions the processor offers, you also need to consider your needs in choosing a reference design. Reference designs vary quite dramatically, ranging from small form factor packages containing little more than the processor to more robust tools with large displays and circuitry that's easier to access.

What type of reference makes sense for you depends on where you are in your development process. Do you want to put the reference design on a bench and test different electrical approaches? If so, you should go with a larger development platform. If you have a base board design complete, then a simple processor module may be sufficient. If you require a handheld form factor, other references may be available.

You should also evaluate how easy it is to test custom hardware with any given reference design. Does it offer the expansion interfaces you're going to need to test your custom parts? You may not need these expansion interfaces if your embedded device is fairly standard. If your device is highly customized, however, these interfaces will prove critical.

### 3. Don't Neglect the BSP

A good board support package (BSP) is also vital for rapid development. Because today's processors have such tremendous potential for customization, there is a tremendous amount of work to be done to get an operating system running on them. Choosing a SoC processor that has a production-quality board support package will give you a good foundation to begin the customization work.

The benefits of the board support package are the same as with the reference design. In all but the most exceptional of cases, your drivers will require some degree of customization. But with the BSP you can customize drivers that already work, rather than trying to figure out how to build your own drivers in the first place. It frees you from making arbitrary decisions about how your drivers should be set up and from the trial-and-error debugging process needed to make everything work.

Using a tool chain that allows easy modification of the board support package makes the task even easier. You will likely have to remap some of the pins if you are using the functional blocks of the processor differently, and you may want to eliminate drivers for peripherals you are not using. Rather than manually editing the driver source files to make any modifications, you can edit a single configuration file that will select the appropriate drivers, modify them as necessary, and compile a BSP for you.

Finally, make sure you consider the quality of the board support package. Sample or demo-quality BSPs may be fine for showing off hardware, but you will be stuck with a lot of optimization work and long test and debug cycles to make sure you have a stable platform. A production-quality BSP that has already gone through a rigorous quality assurance process can greatly accelerate your project.

### 4. Consider Power Management

If you are designing a battery-powered device, power management is at the top of your list of concerns. Even if your device is line-powered, you may want to reduce thermal dissipation or minimize power use for environmental reasons. While you can work to maximize power efficiency yourself, it's far more efficient to choose a SoC processor that is supported by an integrated power management solution. An appropriate power management integrated circuit (PMIC) may be provided by the same manufacturer as the SoC processor or by a manufacturer that specializes in power and analog components.

Modern application processors feature dramatically improved power management capabilities, with many power domains giving you the ability to match consumption with the task at hand. Integrated power solutions are better able to take advantage of those additional power domains than solutions involving discrete components. This is due to their economies of scale, including the sharply reduced need for silicon packaging and the ability to share functionality that would otherwise be duplicated between several different voltage regulators.

Some of these solutions even go beyond power management and incorporate audio codecs, LED drivers, and other analog functions. This provides an integration bonus "integrated digital in your SoC processor and integrated analog in your PMIC!

As with the reference design, it's far easier to build a highly efficient device when somebody has already done the most complicated work for you. That reduces the debugging challenge and gives you more time to focus on the features that make your device unique.

## 5. Consider the Future

Just as you don't want a SoC processor with frills that you would never use in a million years, you also don't want a SoC processor that you're going to outgrow next week. The hardest part of any embedded project is coming up with the basic design, and ideally you should be able to use that design for a few years, even as you add features.

If you use a SoC processor that has only the functionality you need today, you will need to select a different processor if you want to add any capabilities, such as additional peripherals. That requires developing a new embedded design from scratch. Adding any features "no matter how minor" becomes a costly endeavor.

As you begin your design work, you should think about what you may realistically want to add to the device in two to three years, and you should use a SoC processor that includes those features. Don't go overboard. For example, if you're designing a data entry device, you probably won't need multimedia accelerators, so why pay for them. But you may want to add Bluetooth capabilities.

Choosing a SoC processor that has those capabilities now even

if you don't plan to use them right away makes the upgrading process less painful. It can mean the difference between an application software change to use a supported feature (or, at worst, a minor board spin to add an additional connector) and a complete platform redesign.

## Conclusion

A SoC processor gives you unprecedented functionality and power-savings capabilities in a compact design. But there are many choices. The key to getting the most out of the processor is giving serious thought about what you need today and in the future, and carefully selecting the processor and support options that will meet those needs. The result can be a highly efficient embedded device with rapid time to market.

*Ron Stieger works at BSQUARE Corporation as a Hardware Engineering Manager on embedded reference designs. Ron has over ten years of experience in embedded system design and management at BSQUARE, Fluke Networks, Terabeam and AstroTerra.*



BSQUARE is a solution provider to the global embedded device community. Since 1994, BSQUARE has completed hundreds of successful projects and is a trusted partner to smart device makers worldwide. Our teams collaborate at any stage in device development with a portfolio including software and hardware development, systems integration services, reference designs, board support packages, middleware, and applications. As a full service provider, BSQUARE also offers Windows® Embedded licensing expertise.

BSQUARE Corporation 110 110th Ave. NE, Suite 200 Bellevue, WA 98004  
888-820-4500 sales@bsquare.com www.bsquare.com

BSQUARE is a registered trademark of BSQUARE Corporation.