Introduction

A wide range of semiconductor products based on ARM® Ltd. cores have become technology and market leaders over the past two decades. Cell phones, feature phones and smart phones are perhaps the most widely known ARM success stories. But specialized computers, industrial equipment, home entertainment equipment and other embedded applications also run on ARM processor cores, sometimes in tandem with other processors such as digital signal processor (DSP) cores from Texas Instruments (TI).

TI became an important adopter of ARM intellectual property when it integrated ARM cores and TI DSP cores into dual-core devices for cell phones. Later, when ARM broadened its strategy by licensing its cores to major semiconductors vendors for integration into standard microcontrollers (MCUs) and microprocessors (MPUs) TI was in the vanguard. The move to include standard architectures has significantly increased ARM’s presence in the embedded systems market.

In 1993, TI began licensing ARM cores and has shipped over five billion ARM-based products. The first multicore applications processor, the ARM9™-based OMAP1510, was launched by TI in 2002; and the first Cortex™-A8 multicore products, TI’s OMAP3 processors, were introduced in 2006.

Similarly, TI has integrated generations of ARM cores into its DaVinci™ video processors and its MCUs that target automotive applications. As a result of this relationship, TI ARM products offer robust peripherals and an excellent system performance.

In its most recent extension of the ARM relationship, TI’s latest Sitara family of MPUs offer the first Cortex-A8 processor-based solutions optimized for industrial applications. One the first series of products, the Sitara AM35x MPUs, offer multiple device packages, industrial temperature options, peripheral integration, graphics capabilities and high computational performance at sub-one watt power levels.

Performance, price and value

ARM’s diverse and growing processor family provides a wide range of price and performance options with energy consumption and footprint specifications that are often leaders in their respective applications.

The architecture’s popularity is due in no small part to the fact that ARM architecture provides embedded systems designers with an exceptional choice of development tools, operating systems (OSs) and application options. Although ARM offers its own support for board development, in the standard MPU and MCU market, TI offers even more.

Over the long relationship with ARM and knowledge on different applications, TI has invested heavily in system integration, software, tools and evaluation kits to optimize performance of its ARM-based products. As a result of this relationship, TI ARM products offer robust peripherals and excellent system performance.

The flexibility and freedom that designers receive from ARM and its semiconductor partners is accompanied by a formidable requirement; the system-level design team must select an OS that best suits its hardware and target application. With over 50 choices of OSs that are supported on ARM, this can be a daunting task.
There are, however, some top-level decisions that at least provide a framework for choosing an OS. These include the answers to simple questions such as whether the application requires a memory management unit (MMU).

Often a particular application interface is mandated, and that can also help thin the field. Other aspects of choosing an OS are more nuanced and involve issues such as cost, time to market and special features required by the application. This white paper will provide a guide for making the right OS decision.

**OS continuity**

The familiarity that the system-level design team has with a particular OS is one of the most important criteria. A high comfort level is an important consideration regardless whether the team has previously designed on ARM®-based hardware or not.

By working with a familiar OS, design teams will shorten design cycle time and are highly likely to produce a higher-performing system. If the design team has been developing a key application on a particular OS for a long time, the issues of cost, real-time operating system (RTOS) versus high-level OS, and others have already been decided.

Under these conditions, the primary question is whether there is a version of the OS that runs on ARM platforms.

If the application is new or if a decision has been made to move to a new OS, there are many factors to be considered. A few significant factors are listed below with brief descriptions.

- **Overall cost.** Includes direct costs such as licensing, training, technical support and royalties as well as hardware-related factors such as memory requirements.
- **Royalties** has been included in this list in addition to overall cost because in high-volume applications, in particular, margins can be quite thin even without paying royalties. In addition, the thought of sharing revenue long after the product has shipped rubs engineering managers the wrong way.
- **Real-time capability.** This is a tricky consideration because “real time” seems to be in the eye of the beholder. For example, an [Embedded.com](https://www.embedded.com) survey found that 90 percent of developers interviewed considered their current project had real-time requirements but in quantifying their real-time deadlines, the results covered a few orders of magnitude. The mean time was about 400μs with about 75 percent choosing deadlines of under 1ms.
- **Software tools availability and performance.** Ease of use and the ability to deliver a high-performing system are obviously critical criteria for evaluating an OS. The decision is complicated, however, by the fact that high-performance and user-friendly tools do not necessarily go together, so a tradeoff has to be made. In addition, commercial OSs tend to have better tool support than open-source OSs such as Linux®, which brings OS acquisition cost into the decision.
- **Hardware/CPU compatibility.** While still an important consideration, the compatibility issue is greatly reduced when an ARM processor is used because ARM’s market leadership is an incentive for other hardware vendors to engineer compatibility into their products.
• **Code size/memory usage.** Memory dominates the bill of materials in virtually all embedded designs so efficient memory usage for both code and data is essential to reduce costs. ARM cores have extensive support for efficient memory utilization which makes this less of an OS issue with ARM. Its Thumb®-2 instructions, for example, include a mix of the original 16-bit Thumb instructions and new 32-bit instructions that provide equivalent functionality to the 32-bit ARM instruction set.

• **Networking capability.** Networking support is important for developers who are reluctant to write protocol stack code themselves. This is actually a majority because network protocol stacks are complicated and regulated. They also tend to be pass/fail coding projects. Either they meet specification or they don’t. Quality is seldom an issue, so there’s little room for improvement or creativity.

• **Documentation and technical support.** These two criteria have been combined because they both argue for choosing a commercial OS rather than an open-source OS.

• **Supplier’s reputation.** OSs, particularly RTOSs, have become something of a cottage industry because it does not take much of an investment to get into the business. The 2010 EETimes Embedded market study named 25 commercial OS vendors and acknowledged it was only a partial list. Of those 25, no product garnered more than 19 percent of the “market share” as measured by respondents saying they used the OS. At the time, the top seven included VxWorks®, Windows® CE, Red Hat Linux, Embedded NT, DSP/BIOS™ Kernel, QNX, and MontaVista Linux.

**Reducing design cost**

In engineering, cost is always a major consideration and in most embedded designs it is the overriding consideration (assuming, of course, that performance goals are met). Cost can be segmented into three categories:

• Non-recurring engineering design cost, which includes software licenses, technical support and salaries paid to the design engineers and programmers

• Recurring costs such as bill of materials (BOM) cost, which includes hardware components and packaging

• Royalties, which although certainly a recurring cost, are associated with software licensing

The major impact on BOM cost that can be associated with an OS decision is typically the Flash and RAM memory requirements. The general guidelines are straightforward for these requirements including full featured OSs, such as WinCE and Linux while some commercial OSs require more Flash than OSs built around a real-time kernel with the goal of minimizing external resources, such as Flash for program memory and RAM.

The Embedded Linux OS can be adapted to run with as little as 256 KB of ROM and 512 KB of RAM. The total memory requirement is always, of course, dictated by the application. In many instances, RTOSs from niche vendors can do even better.

A related BOM cost consideration for teams designing with ARM processors is support for hardware features that are unique to ARM because peripheral devices may be needed to implement the features.
Often, support for a specific OS feature or application component is an important consideration in selecting an OS. Real-time capability has already been discussed in some detail. Three other common requirements related to the application that can dictate the choice of an OS are:

- **Safety critical applications.** When a system failure can lead to potentially catastrophic events, it must typically meet official certification requirements. From a processor perspective this usually means a high level of predictability, which can be enabled by certain OS features.
- **Multitasking.** Multitasking can be either preemptive, in which the OS manages time slots for applications; or, cooperative where the OS waits for the program to give back control.
- **Support for a specific device class.** USB is a common example.

Most embedded system designers tend to think in terms of having limited options when it comes to OSs. They can pay a royalty on every device shipped by using a pre-integrated OS/Framework/Application combination such as WindRiver’s VxWorks or Microsoft’s WindowsCE. Alternatively, they start with a royalty-free Linux OS kernel and add available framework/applications to create the complete system solution.

The build-your-own option requires more software development time but is attractive when the system-level product has thin profit margins. For design teams that prefer Linux but want more support, a number of companies including WindRiver, MontaVista and Timesys, offer Linux plus frameworks and applications as commercial distribution.

A relatively new option is Google’s Android™ OS because it promises a royalty-free OS/Framework/Application combination. Although originally intended for smartphones, Android is proving to be much more versatile and is gaining ground as an OS for eReader tablets, digital media adapters and other consumer devices.

**Embedded Linux OS**

Linux for embedded systems (Embedded Linux OS) differs from the proprietary model followed by Microsoft and other software OS vendors. Linux is not developed, maintained and owned by a single company — and it does not require royalty payments. It is community supported and although the kernel and basic tools are held in common, additional tool chains, applications, configurations and kernel patches are available.

Embedded Linux OS incorporates many improvements over the Standard Linux developed for PCs, including better real-time performance, a simpler and smaller kernel, and support for a range of CPU structures such as ARM.

Design teams that opt for a do-it-yourself (DIY) version of Embedded Linux OS can start at the kernel level by downloading code at [www.kernel.org](http://www.kernel.org). Alternatively, they can find the desired port version for their chosen ARM CPU at [www.arm.linux.org.uk](http://www.arm.linux.org.uk). This process is very time-consuming and difficult. The procedures for configuring the OS and then adding applications and drivers to the DIY kernel is complicated, particularly for embedded systems.
The linux.onarm.com website provides open-source components, middleware and utilities used to build a Linux mobile software stack on an ARM processor. Libraries, utilities, multimedia components and Mozilla and Webkit web browsers are included.

ARM and some of its partners (including TI) has created a non-profit entity named Linaro to improve Linux offering on ARM platforms. Linaro will create a common software foundation for software stacks and distributions to land on and provide the best open-source tools for developers to develop on. The focus is on low-level software around the Linux kernel that touches the silicon, key pieces of middleware that enable new markets and tools that help the developer write and debug code. Linaro aims to maximize the potential of the latest features of ARM-based processors, helping provide optimized performance in a lower-power envelope. Linaro is both a community and a supported project with engineering teams spread globally. More information is available at: www.linaro.org/

TI provides a big head start for design teams (compared to the DIY approach) by providing a full set of tools and development board packages, which will be described later.

Commercial Linux distributions
Other notable Linux distributions that support ARM processors include:

- Android™
- Meego
- Ubuntu
- WindRiver Linux
- Timesys Linux
- MontaVista Linux

Android is worthy of special mention here if for no other reason than that it was developed by Google specifically for the ARM architecture. Android offers designers two major benefits including:

- The architecture and versatility of the Android software stack
- The size of the IP and services ecosystem that has rapidly materialized around Android.

Android’s modular architecture enables the rapid inclusion, replacement or addition of open-source or proprietary content to deliver the required functionality. The modularity has the additional advantage of facilitating the rapid integration of on-chip differentiators such as hardware video codecs and integrated radios.

The relationship continued with Android 2.0 which runs on Cortex™-A8 class devices and takes advantage of the latest features such as ARM’s NEON™ general-purpose SIMD engine for processing multimedia data types. ARM has created a Solution Center for Android that has more than 40 partners offering a wide range of solutions and services.
OS support for TI's Sitara™ MPUs

TI offers a broad range of software development support for its Sitara family of Cortex™-A8-based microprocessors, including both community and commercial versions of Linux, Android™, WinCE and leading RTOSs.

Support is either presently available or will be available in the near term for the Sitara AM35x, AM1x and AM37x MPUs. In addition, support for OMAP35x applications processors, which are also based on the Cortex-A8 MPU, are continued.

Community supported and commercial versions of Linux are already available on all Sitara MPUs.

There is no shortage of tools. Companies offering software applications development environments for Linux and a list of vendors is available on TI’s website. CodeSourcery’s Sourcery G++ software development environment is of particular interest to TI customers. Sourcery G++ Lite Edition is sponsored by ARM, Ltd. and used internally by TI and is available free with TI evaluation boards and from the CodeSourcery web site. Sourcery G++ commercial editions provide a completely compatible upgrade path for users who require an IDE, additional tools and support from CodeSourcery. TI provides an evaluation version of CodeSourcery commercial tools with evaluation boards.

Android support is available for Sitara MPUs. In January 2010, Google released the Éclair software development kit (SDK) for version 2.1 of the Android OS. TI currently supports this version of Android on all Cortex-A8 based Sitara products.

Android is particularly valuable for applications that depend on display and touch support. V 2.1 offered several enhancements of interest to embedded designers including support for more screen sizes and resolutions, Bluetooth® 2.1, a new browser UI, and Microsoft Exchange support. As new SDK versions are released, TI Android support will evolve accordingly.

Android on OMAP35x, Sirtara AM35x and AM37x MPUs – codenamed rowboat – is a free, open-source project for all TI customers as well as developers and other parties. Development on this platform will extend to other platforms.

TI has traditionally supported WinCE and TI-sponsored support for the Sitara family is readily available for optimized versions of WinCE 6.0 R3 and Windows Embedded Compact 7.0. Developers can download a board support package (BSP) from TI’s product website including drivers, an application framework and a graphics development package.

Several system integrators and consultants that specialize in WinCE development also have extensive experience with TI processors and can be helpful to developers with limited time. They include Bsquare and Logic PD in the Americas, Adeno and MPC Data in Europe and the Americas, and Mistral, which operates worldwide. Contact information is available from TI.

Once an embedded system is ready to go market the developer’s company must acquire a runtime license and certificate of authenticity from a Microsoft Authorized Embedded Distributor.

Developers of Sitara-based applications requiring a RTOS have several commercial versions to choose from including Green Hills® Software’s Integrity®, VelOSity™ OSs, WindRiver’s VxWorks®, QNX’s Neutrino™,
Mentor Graphics’ Nucleus®, and Express Logic’s ThreadX® OS. All of the companies support the OMAP35x platform and most have either announced plans for supporting selected members of the Sitara family or already support them.

**Conclusion**

Selecting an OS for a specific application that will run on an ARM processor entails the consideration of many desired features, technical requirements and cost parameters. While many OSs support the ARM architecture, design teams can quickly narrow the list by applying the guidelines discussed in this article.

The good news is that when a TI Sitara MPU, or any other ARM-based MPU, is selected to power your embedded system, a wealth of software and board development support is available from TI regardless of the OS you choose.
# IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal and regulatory requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or “enhanced plastic.” Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer’s risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use. TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

<table>
<thead>
<tr>
<th>Products</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amplifiers</td>
<td>amplifier.ti.com</td>
</tr>
<tr>
<td>Data Converters</td>
<td>dataconverter.ti.com</td>
</tr>
<tr>
<td>DLP® Products</td>
<td><a href="http://www.dlp.com">www.dlp.com</a></td>
</tr>
<tr>
<td>DSP</td>
<td>dsp.ti.com</td>
</tr>
<tr>
<td>Clocks and Timers</td>
<td><a href="http://www.ti.com/clocks">www.ti.com/clocks</a></td>
</tr>
<tr>
<td>Interface</td>
<td>interface.ti.com</td>
</tr>
<tr>
<td>Logic</td>
<td>logic.ti.com</td>
</tr>
<tr>
<td>Power Mgmt</td>
<td>power.ti.com</td>
</tr>
<tr>
<td>Microcontrollers</td>
<td>microcontroller.ti.com</td>
</tr>
<tr>
<td>RFID</td>
<td><a href="http://www.ti-rfid.com">www.ti-rfid.com</a></td>
</tr>
<tr>
<td>RF/IF and ZigBee® Solutions</td>
<td><a href="http://www.ti.com/lprf">www.ti.com/lprf</a></td>
</tr>
<tr>
<td>Audio</td>
<td><a href="http://www.ti.com/audio">www.ti.com/audio</a></td>
</tr>
<tr>
<td>Automotive</td>
<td><a href="http://www.ti.com/automotive">www.ti.com/automotive</a></td>
</tr>
<tr>
<td>Communications and Telecom</td>
<td><a href="http://www.ti.com/communications">www.ti.com/communications</a></td>
</tr>
<tr>
<td>Computers and Peripherals</td>
<td><a href="http://www.ti.com/computers">www.ti.com/computers</a></td>
</tr>
<tr>
<td>Consumer Electronics</td>
<td><a href="http://www.ti.com/consumer-apps">www.ti.com/consumer-apps</a></td>
</tr>
<tr>
<td>Energy</td>
<td><a href="http://www.ti.com/energy">www.ti.com/energy</a></td>
</tr>
<tr>
<td>Industrial</td>
<td><a href="http://www.ti.com/industrial">www.ti.com/industrial</a></td>
</tr>
<tr>
<td>Medical</td>
<td><a href="http://www.ti.com/medical">www.ti.com/medical</a></td>
</tr>
<tr>
<td>Security</td>
<td><a href="http://www.ti.com/security">www.ti.com/security</a></td>
</tr>
<tr>
<td>Space, Avionics &amp; Defense</td>
<td><a href="http://www.ti.com/space-avionics-defense">www.ti.com/space-avionics-defense</a></td>
</tr>
<tr>
<td>Wireless</td>
<td><a href="http://www.ti.com/wireless-apps">www.ti.com/wireless-apps</a></td>
</tr>
</tbody>
</table>

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2010, Texas Instruments Incorporated