Introduction

This whitepaper is for developers new to TI ARM®-based processors, as well as for experienced developers wanting to better understand the various ARM architectures. Beginning with an overview of ARM technology and available processor platforms, this paper will then explore the fundamentals of embedded design that influence a system’s architecture and, consequently, impact processor selection. However, selecting the right processor involves more than just estimating performance requirements. Software often represents a majority of the development investment, both in terms of cost and time, and so the optimal processor for an application also has the right development resources supporting it. Finally, an application-specific perspective is required to determine how the needs of particular markets also affect processor and software requirements. After reading this whitepaper, developers will be able to identify an appropriate ARM processor for their current application, as well as which development resources they need to immediately begin processor evaluation and design.

Why TI ARM processors?

ARM is a supplier of core processor technology and offers several licensable processor cores to meet a wide range of application requirements. By basing designs on ARM technology, OEMs are able to leverage the widest diversity of any processor technology available today: more than 30 billion ARM-based processors have been sold, with more than 16 million sold each day (Source: ABI Research).

Before getting into too much detail, it is important to note that the ARM architecture, by itself, is primarily a computing core that silicon manufacturers license to augment their own technology and build processor platforms. The most successful manufacturers bring value by obtaining an architectural license to modify the core, as well as design a complete processor or system-on-chip (SoC) around the ARM core, not just by integrating peripherals and memory but adding application-specific features that enhance performance and power efficiency. Thus, while many vendors design their own processor offering an ARM core, the resulting processors are not equal. In fact, the differentiating technology around the ARM cores is what has made TI the world leader embedded processors with ARM technology.

TI has designed devices with ARM cores since 1993 and has developed an unparalleled expertise and IP with its customers to maximize performance and reliability while minimizing cost and power consumption. Integrated high-performance TI analog functionality, coupled with deep system expertise and a robust software ecosystem, is what makes TI ARM processors excel. TI also has a large IP profile of ‘application specific’ accelerators which it puts into these systems, such as a digital signal processor.

Scalability is key to TI's ARM portfolio – both hardware (pin compatibility) and software. Given that software can drive more than 50 percent of a product’s total development costs, OEMs need to be able to develop multiple products on a single development effort. Processor platforms like TI’s that span several generations of code-compatible ARM architectures make highly leveraged development possible. To minimize development costs, TI and its partners continue to make substantial investments beyond just the ARM core to create the industry’s leading suite of development tools and resources. This comprehensive offering of the hardware and development software TI provides with each ARM core – peripherals, memory, advanced processing engines, software, tools – is part of what sets TI’s ARM technology above others:
Six ARM-based processor platforms contain devices as inexpensive as $1 with performance as high as 5 GHz, OEMs can power a design across a wide-ranging product line, from low- to high-end, while leveraging common processor architecture, application code and base hardware design.

With TI’s broad ARM portfolio, there are also system solutions for applications, ranging from smart grid and industrial systems, to advanced infotainment devices, to advanced, high-performance multicore wireless infrastructure and supercomputing applications.

Integrated industry-leading peripheral technology on each of TI’s ARM-based processors, gives OEMs access to high-performance analog solutions, advanced power management capabilities, specialized memory technologies and efficient connectivity peripherals.

Specialized hardware: from digital signal processing engines, analytics and advanced video/voice accelerators to safety features critical for standards, such as IEC 61508, are used to create processors optimized that accelerate design, increase performance, and boost power efficiency all at highest overall value.

Extensive hardware and software development tools from TI and its partners accelerate product development, enable state-of-the-art development to extract more performance (e.g., algorithm optimizations, bandwidth analysis) and minimize time-to-market by simplifying the code development process and enabling advanced troubleshooting so that OEMs can focus their design efforts on their core competencies.

All levels of application complexity are supported with a variety of real-time and high-level operating systems to simplify managing system tasks.

System-level tools for specific applications, including application code, optimized peripheral drivers, development kits, reference designs and functional demonstrations, jumpstart development with design elements that drive robust products to rapid completion.

With more than 500 feature-rich ARM devices to choose from based on a wide range of ARM cores, developers can be confident that they will find a processor with the right combination of performance, price, peripherals, power consumption and specialized hardware to meet the needs of a wide range of applications, including renewable energy, industrial automation and control, transportation, consumer electronics, medical and more.

The first step in selecting the right ARM processor for an application is to understand the three primary ARM core families and the basic differences between them:

**ARM Cortex™-A:** Applications processors based on the Cortex-A series core offer low power, along with exceptional 32-bit performance. This architecture supports multicore configurations, as well as optional advanced floating-point capabilities.

The Cortex-A offers up to 13-stage technology with 1.5-2.5 DMIPS/MHz per core with advanced branch prediction. Its Neon integer and floating-point SIMD engine enable advanced media performance for applications.
The Cortex-A series is tuned for memory streaming with 1-2 cycle cache access, pipe-lined loads and stores and coupled (or integrated) level-2 caches.

With its high-performance and advanced memory, Cortex-A applications are well-suited for any product that needs to incorporate and run an advanced operating system, such as Windows, Linux or Android. In addition, Cortex-A processors provide additional benefits through a few available technology extensions — TrustZone® security extensions for secure computing applications and Jazelle® technology for accelerating execution environments, such as Java, .Net, MSIL, Python and Perl.

Because they are designed to serve as an application processor, common applications for Cortex-A-based processors include digital TV, home gateways, auto infotainment systems, industrial automation, basestations and cloud computing.

- **ARM Cortex-R**: Developed for real-time applications where low power and responsive-interrupt handling are needed, processors based on the Cortex-R provide exceptional performance. High performance is enabled with features such as high clock frequency, micro-architecture, dual-core multi-processing (AMP/SMP) configurations and hardware SIMD instructions for very high-performance DSP and media functions.

  For real-time control, the ARM Cortex-R core uses tightly-coupled low-latency memory, which allows low-latency response to real-time events. These real-time response features provide high DMIPS/MHz benchmark and clock frequency.

  Unique to the ARM Cortex-R core is the memory protection unit (MPU), dual-core lockstep configuration and automated parity checking and error correction for hard and soft errors. These distinctive features are what set these cores apart for safety applications, such as automotive braking systems, renewable energy, safety, mass storage controllers, networks and printing. For example, Texas Instruments (TI) utilized the ARM Cortex-R technology, as a dual lockstep core, on its Hercules™ micro controllers (MCUs) targeted at functional safety applications.

- **ARM Cortex-M**: This architecture, perfect for 16- to 32-bit processors, was designed for applications requiring an MCU providing fast and highly deterministic interrupt management with the smallest footprint and low power consumption. If the end product needs to enable features, ARM Cortex-M cores are perfect. They run at a lower MHz with shorter activity periods and its special sleep-mode architecture.

  The Cortex-M offers a high-density instruction set and has smaller RAM, ROM and Flash requirements, which means smaller amounts of software code to lower silicon costs. If your product needs robust features while offering low-power modes, this is the processor.

  Cortex-M-based processors are ideal for most MCU-based applications, including mixed-signal devices, smart sensors portable health and fitness devices, metering and automotive electronics.
As stated before, the overall performance, power and price of an ARM-based processor are determined by the silicon vendor. As the industry leader in ARM technology, TI offers six platforms of ARM-based processors, each with a range wide of cores to address varying performance, power, and cost requirements across nearly all application markets (see Figure 1). The breadth of TI’s portfolio provides comprehensive application coverage so OEMs know they can find a device that provides an optimal balance of performance and peripherals. Providing a thermal advantage, TI’s ARM processors consume low amounts of power, which precludes the designer from having to implement heat sinks or fans, which means bill of materials and cost reduction.

Keystone multicore processors: Based on the ARM Cortex-A15 core, Keystone processors have multiple cores on a single chip for high-performance applications. Processing efficiency is complemented by a best-in-class interconnect and 2x-DDR3/3L memory interface to eliminate internal throughput bottlenecks. Further enhanced with integrated packet processing, these are TI’s highest performing ARM processors, running at up to 1.4 GHz on each of the four cores (cumulative total of 5.6 GHz), making them ideal for applications such as wireless infrastructure, base stations, cloud computing and supercomputing.

OMAP™ processors: These processors are known for their low power yields in tandem with best-in-class performance, supporting multiple high-level operating systems, such as Linux, Android and Windows Embedded. With designs based on the Cortex-A8, -A9 and –A15 cores, the OMAP family includes graphics and display accelerators for multimedia images in several mediums and supports a diversity of application requirements, including industrial, automotive infotainment, touchscreens and video applications. OMAP 4 smart multicore processors, for example, employ two ARM Cortex-M3 cores to offload real-time control processing from the processor’s two main ARM Cortex-A9 cores. The architecture also integrates unique programmable accelerators – an embedded DSP, imaging and vision IMX accelerator and display controller –

Figure 1 – TI’s ARM Portfolio
that provide greater flexibility and future-proofing compared to hardwired accelerators. Dual-channel memory provides unparalleled bandwidth support for high-definition multimedia and displays and support for LPDDR2. For even higher performance, OMAP 5 smart multicore processors have the same attributes as OMAP4; however, they are based on the ARM Cortex-A15 and ARM Cortex-M4 and include a two-stage memory management unit (MMU), support for LPDDR2 and DDR3, as well as virtual interrupts to enable virtualization of multiple, guest operating systems and applications utilizing a hypervisor architecture. OMAP5 runs at 1.7 GHz for each ARM Cortex-A15 processor.

**Sitara™ ARM processors:** The Sitara family, based on the ARM Cortex-A8 core, runs at up to 1.0 GHz, offering the best performance, peripherals and memory for the price. Sitara processors have an integrated touchscreen controller and graphics engine to support the advanced user interfaces today’s consumers have come to expect from electronic products, especially power-sensitive portable and handheld devices. Ideal for industrial, smart grid, portable navigation, and educational consoles, Sitara processors are backed by extensive support for both the Android and Linux, as well as Windows Embedded operating systems and a variety of real-time operating system (RTOS) platforms through TI design network partners.

**Hercules™ ARM MCUs:** Primarily based on the ARM Cortex-R4, Hercules MCU offer advanced hardware safety features such as dual lock-step cores, as well as a variety of software-based safety capabilities so OEMs can implement reliable functionality in applications that require a high degree of safety, including industrial, medical, automotive, transportation and renewable energy systems. Starting at $4, Hercules MCUs have up to 3 MB Flash and offer several hardware-, software- and tools-based features to help designers simplify compliance with safety standards such as ISO 26262 and IEC 61508 in their functional safety applications.

**Tiva™ C Series MCUs:** Specifically designed to meet the needs of home, building and industrial automation designs, Tiva C Series MCUs are focused on control, communications and connectivity and built around an ARM Cortex-M4F and integrated with a wide selection of peripherals and connectivity options (i.e., CAN, USB) appropriate for computing applications. With up to 80 MHz and 256 KB Flash plus powerful floating-point capabilities, the Tiva C Series architecture provides tremendous value with pricing as low as $1.

**C2000™ DSP + ARM Cortex-M3 MCUs:** C28x + ARM® Cortex™-M3 brings together connectivity and control by combining a ARM Cortex-M3™ core with C2000’s C28x core on one device. With C28x + ARM Cortex-M3, applications such as solar inverters and industrial control can keep the benefits of separating the communication and control portions while maintaining a single-chip solution. In addition, The C28x + ARM Cortex-M3 enables safety certifications in your system via enhanced hardware and safety features.

Important to note, TI offers an integrated 2D graphics engine on its Keystone, Sitara and OMAP processors that can handle this task. For applications requiring an elevated level of graphics capabilities, 3D engines, along with appropriate graphics middleware frameworks, are also available on some of these processors. 2D and 3D graphics engines have the dual advantage of offloading graphics rendering from the processor to support additional tasks and improve the responsiveness of the UI. The cost of implementing graphics — including increased processor and display cost — must be considered too.
Choosing the right processor is a critical decision as it defines a product’s capabilities. For example, when a processor is correctly sized, it supports the full suite of a product’s functions and leaves headroom for growth. Over-specifying the processor, on the other hand, will drive system cost up unnecessarily. A mis-chosen processor may run out of performance during the design cycle, leading to key features being eliminated or downsized. In a worst-case scenario, an improperly sized device in the field may not have the capacity for a critical update.

Once the correct processor is identified, attention turns to the development tools. A strong tool suite can accelerate design, test and debug, but a processor that lacks a strong development environment will result in developers fighting their tools. This can delay a product’s launch or require key differentiating features to be dropped in late design cycles. A final consideration is the ecosystem of production-ready or near-ready software, the experienced design houses and third-party manufacturers who can actually accelerate design by providing professional products and services.

Advanced ARM-based devices and systems-on-chip (SOCs) often incorporate application-specific accelerators or targeted input/output (I/O) interfaces. These can facilitate or hinder development, depending on how robust the special features are. Robustness incorporates everything from testing to documentation of the APIs or configuration options. Important software considerations include:

- The maturity and completeness of the tool chain and the development environment. A mature tool chain can substantially speed design by helping developers identify potential issues early in the design cycle and resolve late-cycle hardware issues through software workarounds. In contrast, a bare-bones development system will hinder development as developers struggle to make their tools work for them.
- Availability of a range of development boards which provide a means for jumpstarting application development while hardware prototypes are designed. Such boards can reduce development time by weeks or months.
- Use of an operating system (OS) to manage the multitude of tasks a system must perform. Feature-rich software development kits supporting each OS should include a wide range of example applications and middleware including communication stacks, graphics libraries and security framework.
- Efficient multiprocessing tools manage multi-processor, multicore and multi-threaded programming in a manner that makes much of the complexity transparent to developers.
- An organized development community that is embraced by the silicon provider can provide support among its members through shared code and expertise.
- System profiling and performance analysis tools provide extended lifetime of the device by getting more out of the part without having to buy more expensive or newer ones.
Not all development tools are created equal. This is one area where it is important to evaluate the options available for a processor. Several general factors to consider during an evaluation include:

**Integrated Development Environment (IDE):** An IDE brings many of the core development tools into a single environment and transparently handles sharing data between tools. This simplifies the learning curve compared to having to learn the interface for multiple tools. For TI’s ARM-based processors, developers can use TI’s powerful Code Composer Studio™ IDE or one of several other IDEs available from TI partners. In addition, Code Composer Studio and Google’s Android development tools, among others, are based on Eclipse, an open source environment. This means developers can leverage continuous innovation and support from the open source community.

**Tight integration:** All IDEs are not equal. Some IDEs are really nothing more than a shell that launches several disintegrated tools. Without tight integration, transitioning from one tool to another can be time consuming, adversely impacting the efficiency of the entire software development process. Many of TI’s partners design their tools to integrate with Code Composer Studio to provide seamless functionality with the entire suite of capabilities.

**Operational performance:** With C being used in more and more systems, today’s compilers are expected to produce code that meets or exceeds the capabilities of a developer writing in assembly. How well code is compiled, however, varies greatly from compiler to compiler. Industry benchmarks are available that show how efficient a particular compiler is in terms of execution and code size. These benchmarks can also show how much more efficient one ARM processor is compared to another.

**Processor-specific optimizations:** General-purpose tools produce general-purpose code. To get the highest performance possible, tools need to create code that recognizes and exploits all of a processor’s integrated capabilities. TI works diligently on its own and with its partners to create tools that are optimized for each of TI’s ARM-based processors. This gives developers the confidence that their tools are producing code that maximizes the efficiency of the processor with minimal, if any, hand-optimizations of the system at the lowest hardware levels.

**Multicore support:** TI offers several multicore ARM-based architectures backed by tools that support design of multicore systems in an efficient and intuitive manner. Rather than treating each core independently, developers are able to debug multiple cores as a single system. For TI’s DSP + ARM devices, both the DSP and ARM applications can be developed and debugged in a single IDE with the ability to correlate debug information to facilitate identifying issues that involve both cores.

**Visibility:** In general, the greater the visibility into a system, the faster and more efficient the code that can be developed. In addition, visibility is essential for validating system reliability.

**Technical support:** Processors and tools supported by a single vendor may be limited in their value, depth and availability. With its large portfolio of ARM-based devices and extensive ecosystem relationships, TI is able to provide unparalleled support, ranging from community to professional, for ARM-based design at all levels required by OEMs, including professional design services.
EVMs are an important element of TI’s ARM processor portfolio. Most EVMs comprise a printed circuit board (PCB), incorporating the processor surrounded by relevant support circuitry and interfaces. Generally available at low cost, EVMs enable developers to quickly assess and verify the capabilities of a particular processor. In addition, EVMs provide an initial test platform upon which developers can begin product design before their own hardware is available. EVMs also provide a convenient way to reproduce design issues that are discovered and can help validate the integrity of the OEMs design or identify a systematic issue that needs attention. As a result, the availability of an EVM can drastically reduce development time.

TI works with its partners to create a variety of EVMs, many designed for specific applications. Each provides a base hardware platform from which developers can jumpstart their own designs. Comprehensive software support provides an out-of-box experience, enabling developers to immediately evaluate a processor’s capabilities with little to no coding. Application-specific EVMs may supply in-depth application code, as well as a complete software development kit (SDK), full tool chain, and support for a high-level operating system. This enables OEMs to focus on differentiating their products with value added features rather than having to expend design resources and time developing systems from the ground up.

As the number of threads and real-time interrupts in an application increases, so does the value of having an operating system (OS). Applications that begin to exceed 128 KBytes in code size will likely benefit from the multitasking capabilities of an OS.

When system determinism is required, a real-time operating system (RTOS) can provide a reliable framework to accomplish this. In an extreme example, an embedded braking system that misses a critical real-time deadline could result in an automobile crash. An RTOS focuses on responsiveness to ensure that time-sensitive tasks are performed reliably. TI developers can choose from RTOS providers in the industry, including VXWorks (WindRiver), Nucleus (Mentor Graphics), QNX/Rim, and Green Hills.

For applications with less restrictive responsiveness requirements — sometimes referred to as “soft real-time” — a high-level OS (HLOS) will most likely provide more value because of the wide range of general-purpose application-level tasks it supports. In general, an HLOS will abstract more functionality and automatically perform more tasks for developers than an RTOS. However, it will do so at the expense of a larger footprint and greater processing overhead. Examples of HLOSes that have been optimized for TI ARM-based processors include Linux and Android.

Simple embedded applications with limited functionality and memory are not likely to require an operating system. Applications with a code size less than 32 KBytes typically gain little value from an OS compared to the percentage of available resources an OS requires. For systems like these, TI offers TI Ware for many of its ARM-based processors including StarterWare™, StellarisWare® and TivaWare™ software. TI Wares help simplify development by providing device abstraction layer libraries, peripheral programming examples and board-level application code.

Applications with a high-speed communication channel like Ethernet or USB or a sophisticated user interface that must operate in parallel to the main application are also likely candidates for an OS. For example,
when interoperability with networks and other devices is required, it makes sense to use an OS that provides a mature, proven implementation. This not only simplifies development of basic system functions, it can provide a comprehensive implementation with protocol stacks that support the many protocol and class drivers that the system may need to communicate reliably with other devices.

Another aspect of cost to consider is the overhead (i.e., CPU utilization) and footprint of the OS (i.e., memory) as these will impact hardware cost. If the design team is not familiar with the OS, this will increase design time and cost, although these can be reduced through training and the use of consultants.

Today, developers using TI-based ARM processors have a rich range of options, from commercially available operating systems like Windows CE, open source alternatives like Linux and royalty-free options like Android. The choice of OS will influence the general direction that software development will take. Commercial OSes commonly offer ease-of-use and are supported with modules, ready-to-integrate subsystems and technical support. Typically, these advanced features are fee based.

With an open source OS, there are no royalties or fees and support comes from the community. The development team takes responsibility for integrating third-party software into their system, a cost that should be comprehended up front, along with the experience of the integration team. A balancing advantage of an open source OS like Linux is that the fast-moving Linux community usually ensures that Linux drivers are the first to take full advantage of new hardware features.

OEMs can use a commercial version of an open source OS if they decide they want a different level of support or want to have confidence that all of the tools have already been integrated and tested with each other.

When choosing a commercial OS or one supported by a vendor, OEMs will often have a choice of licensing methods including a one-time per-project fee, subscription based, a per-unit royalty or a combination of these. Which one to choose depends upon application volume and the company’s business model.

**Software Development Kits (SDK)**

An SDK is a software development package providing the tools and software developers need to begin coding. SDKs are often offered free of charge and enable developers to quickly get a feel for what it would be like to develop code for a specific processor.

Many of TI’s SDKs offer an extensive software reference design for a particular application with most of the essential components of the system’s software are already in place. Examples include a multimedia framework with codecs or I/O stacks for a communication interface using an integrated USB or Ethernet port. SDKs abstract many of the underlying hardware implementation details, thus enabling developers to immediately focus on designing top-level application software.

TI and its partners know how important SDKs can be to developers and have invested in assuring that the SDKs include structured and well-documented APIs at each level of the system software. This enables developers to use the SDK in a way that facilitates programming and re-configurability of the supporting software architecture. This consistency across code from TI and its partners reduces development time significantly.
Code reuse and debugging

Code reuse has become an essential factor in software development, both in terms of product longevity and product line extension. Many OEMs employ a product line strategy where the same base design is leveraged across a wide range of price points with varying degrees of functionality, processing power and features. To minimize development cost and time-to-market software, code should be designed so that it can be reused across the line.

As each step up in functionality likely requires a step up in processing power, a product line will utilize multiple processors. To enable reuse, software compatibility between processors is needed. Note that this applies not just to application code but to firmware design and other assets such as security features and UI graphics.

In addition, while most of an application will likely be written in C, depending on the segment. Certain low-level optimizations (e.g., boot, driver/kernel) may have been made to target a product for a specific market segment and are in C. High-level applications are many times written in Java or C++. For example, Qt used in GUI application development is C++ while Android is primarily Java and C++. Well-architected and fully documented APIs are part of ensuring the scalability of software over multiple platforms and processors.

When reuse is important, OEMs should consider the breadth of a processor family. If the high-end of the processor family can only serve as a mid-end product, development will have to migrate to a different processor, in the worst case from a different supplier for the advanced products in the OEMs portfolio. Depending upon the processor architecture and maturity of the development environment, this increases the complexity of design and can severely limit code reuse.

TI recognizes the need to help OEMs maximize their software development investment. Each family of TI ARM processors offers a wide range of capabilities to make it possible for OEMs to support an entire product line with a single architecture. However, silicon-level compatibility, while necessary, is not the only requirement for extending software life over time. Software reuse does not come without planning for reuse, including designing code in a modular fashion, creating an API that is carefully documented, verifying the quality of the code, minimizing overhead and managing a software repository that enables developers to easily discover reusable components. Compatibility and reuse must thus be extended through the entire development chain.

To this end, TI designs its tools and provides software components, wrappers, frameworks and other capabilities that enable developers to carry reuse through the many levels of software – function, module, subsystem and application – to maximize the reuse of firmware and code between applications and even alternate TI processor architectures.

Debugging is as much a part of the development process as is writing code. TI works with its partners to design code that facilitates the debugging process by architecting the overall system and composition of individual processing nodes to provide as much visibility into code as possible. In addition, data from both software-based and hardware-based debugging tools is correlated so that developers can isolate faults.
more quickly and precisely. For many of TI’s ARM processors, TI offers emulators with advanced on-chip debugging, like processor trace (PTM) and system trace (STM) along with an array of statistic collectors to emit information that emulator receivers can catch and analyze. TI offers a full line of emulators to provide real-time hardware and software visibility into a system. These emulators are integrated with TI’s Code Composer Studio IDE so they can work in concert with the entire development tool chain.

**Middleware frameworks**

Middleware frameworks can accelerate design by providing a base software architecture from which to build an application. They provide a set of predefined and fully developed software functions in a format that is straightforward to integrate into designs with minimal time and effort.

Each framework specializes in a certain type of functionality, enabling developers to treat complex technologies like video or graphics more like components that are added to a system rather than as complex features that must be developed from the ground up. GStreamer, for example, enables developers to create a multimedia processing stack based on different modules or plug-ins, such as video decoding or audio processing. It is quite common for a single design to utilize several frameworks.

Frameworks also abstract hardware-specific implementation details so that developers don’t have to create base firmware. Such abstraction also prevents code from being locked to a specific processor architecture, thus enabling developers to more easily migrate code to another processor if system requirements change.

A wealth of software resources is available to developers using TI’s ARM-based processors. When evaluating middleware options, developers need to keep in mind that the choice of OS will have a significant bearing on which resources will be readily available. For example, when using Windows CE, resources will be limited to those provided by Microsoft and its partners. In contrast, a Linux-based system has access to a variety of open source resources, although some of these resources may need to be ported. Android is Linux-based, so most of these resources can be applied to Android-based designs.

TI understands that OEMs have different business models and design philosophies and that many companies are actively participating in open source software development. In general, using open source code requires software teams to port tools and frameworks to their processor of choice. To minimize the investment required to use open source code, TI has worked with its partners to ensure that a wide range of open source middleware frameworks and tools have been ported to their ARM-based devices, including OpenMax, GStreamer, OpenGL, and OpenCV. TI has also integrated frameworks into many of its software development kits to accelerate application development.

When consumers compare similar products, they often base a large part of their buying decision on how easy the device is to use. From this perspective, how sophisticated the graphics are behind a UI is crucial to differentiating a product as from its competition.

How a UI is developed depends upon the capabilities of the processor. Processing requirements for graphics may be high so some form of hardware acceleration that offloads the main ARM core(s) will be needed. For example, when graphics and video appear at the same time, they must be blended.
Developers have a wide range of tools available to match their graphical requirements, including both commercial and open source options. OpenGL (khronos.org), for example, is a cross-platform framework independent of any particular OS that supports both 2D and 3D graphics. A variety of frameworks is available specifically for Linux, including Qt, X-11, GTK, and DirectFB, each offering a different level of hardware abstraction. Numerous commercial OSes, including Windows CE, VxWorks and QNX, include an optional graphical framework. For applications with only limited graphical requirements, TI's software libraries offer support for applications with less stringent graphical requirements, providing graphic primitives such as lines, rectangle, and circles, and features such as checkboxes and sliders.

TI has worked with its partners to ensure that their code has been optimized to utilize each processor's specialized graphics engines for higher processing efficiency and simpler system integration.

**Security**

Security can take on several dimensions in an ARM-processor-based embedded system. Four common areas of a system that may require secure capabilities are:

- **Communications:** Devices that connect to a network are likely to be vulnerable to hacking, snooping, viruses, malware and a slew of other types of attacks. Security is used both to authenticate transactions and to protect data while it travels over public networks to prevent identity theft and unlawful intrusions on privacy.

- **User data:** For many electronic devices, such as point-of-sale (POS) devices, the security of user data is important. In some cases, such as certain medical devices, security is mandated. It is important to note that for some applications, especially medical and financial applications, user data needs to be protected from tampering by users themselves. Consider a smart meter where user data determines how much energy has been used and, consequently, how large the bill will be.

- **Protected content:** Devices that support multimedia playback will likely need to support various forms of content protection and digital rights management (DRM) mechanisms.

- **Field upgradeability:** Devices that can be reprogrammed and/or updated in the field may be targets for reverse engineering. For example, consider an industrial system where a software update costs thousands of dollars. Without security, unscrupulous users could update systems without paying the appropriate fees, resulting in substantial loss of revenue for manufacturers. Security is also essential for devices that must provide reliable operation, such as wireless infrastructure devices, which might be the target of hackers wanting to shut down critical public or enterprise operations.

Security technology can be extremely compute-intensive, depending upon the application. Thus, it is imperative to understand the vulnerabilities of a system and plan for them from the very beginning of the design cycle so that a processor with the proper resources can be selected. Key to implementing adequate security without unnecessarily raising system cost is understanding 1) what needs to be protected, 2) from whom and 3) the operational cost of a compromised system.
Developers also need to consider how to secure the system from physical attacks. For example, some systems can be compromised using standard development tools such as a debugger connected to a processor’s debug channels unless these channels are disabled for field operation. Applications requiring more security may need to implement secure boot capabilities to prevent firmware from being overwritten to hijack a system. The most secure systems may require additional measures, such as tamper detection of voltages and clocks.

Many of these security features need to be implemented in hardware given that they are either extremely compute-intensive or are designed to prevent a software attack from hijacking the system. Important features to consider include:

**Cryptographic acceleration:** With a dedicated cryptographic processor, the majority of encryption, hashing, authentication and true random number generation functions can be offloaded from the main programmable cores.

**Secure Boot:** By creating a “trusted root” in software, OEMs can prevent a system’s code from being overwritten, cloned or reverse engineered. Secure boot capabilities, such as those on the Sitara ARM Processors, include protecting code residing in external Flash through encryption, ensuring that the processor only runs authorized software, and preventing third-party manufacturers from exposing a system’s sensitive data.

**Secure run-time:** Secure run-time capabilities enable a system to prevent secure code and data from being exposed to non-secure partitions of a system. For example, a secure watchdog timer such as that on the Sitara AM37x Processors, is only accessible within the secure partition, thus preventing application code from corrupting the secure watchdog. Similar capabilities are available to protect DMA transactions and storage of cryptographic keys, among other data.

### Multiprocessing

Multiprocessing refers to a system that is running multi-threaded software and/or using multiple cores. Many applications can benefit from multiprocessing, and not just in terms of more processing resources. For example, a C2000™ real-time control processor with a DSP with an ARM Cortex-M3 core enables precision algorithm and control with the DSP and communications with the ARM Cortex-M3. Battery-operated devices using the combination of a high-performance processor core in conjunction with a more simple processor core can yield improved power consumption compared to a system designed around a monolithic core.

Multiprocessing applications require significant coordination between threads and cores, including arbitration of shared resources. The system will also need to support communication between threads and cores without allowing a corrupted thread to corrupt the memory of another thread. A hardware-based interconnect, featuring either message passing or shared memory communications, is used to manage data synchronization, coherency and consistency.

OpenMP, supported by TI and ported to many of its processors, is the de facto standard tools environment for shared memory parallel programming. Its high-level constructs substantially simplify and even automate many multi-threaded tasks. It achieves this by allowing developers to specify a parallelization strategy for software detailing to the compiler how particular regions of code are executed by different threads.
As processors increase in performance and capabilities, applications become increasingly complex as well. Today’s smartphones, for example, are more complex than desktop PCs in many ways. Even when the base functionality of a device is simple, the requirements of the rest of the system must still be taken into account when evaluating a processor’s architecture.

**Industrial Applications**

Industrial applications can span the range of requirements. A low-level automation node might control a limited number of programmable logic controllers (PLCs) and have only limited graphical needs that can be met by a graphics primitives library (e.g., lines, circles). Typically, these applications require less than 150 MIPS and are ideally addressed by the Tiva C Series MCU family based on the ARM Cortex-M4. This Tiva C Series architecture also offers a Programmable Real-Time Unit (PRU), a 32-bit processing engine with GPIO that can implement custom logic and often serves as an extended I/O block.

More advanced industrial applications will require more advanced functionality to be able to perform tasks such as switching views of inspection cameras and issuing commands to alter processes on an assembly line. A control console could easily receive data from hundreds of devices among the underlying architecture layers. An efficient processor, then, will need to support graphical/video capabilities as well as have a rich variety of I/O and connectivity options. For these applications, the Sitara Processor platform offers ARM9-based processors for applications requiring from 150 to 500 MIPS and Cortex-A8-based processors for equipment requiring even more processor power. Many industrial applications require the use of an ASIC or FPGA to support industrial protocols, such as EtherCAT. TI eliminates that needs with the integrated programmable real-time unit – industrial control subsystem (PRU-ICSS) found on Sitara Processors. This programmable subsystem gives customers the flexibility to support various industrial protocols while not increasing their design costs.

**Motor Control**

Traditionally, motor control applications have required two processors: a DSP to control the motor and an MCU to manage systems communications. The C2000 DSP + ARM Cortex-M3 MCU offers a high level of determinism and peripheral integration, enabling it to serve as a single-chip solution for motor control applications. In fact, a 50 MHz Tiva C Series MCU can perform more work than many other MCUs running at 100 MHz. In addition, Tiva C Series MCUs maximize Flash utilization by storing the RTOS, peripheral and configuration code, Advanced Encryption Standard (AES) tables, and Cyclic Redundancy Check (CRC) functionality in ROM.

Both Tiva C Series MCU and Sitara processors offer functionality important for motor control, including very fast GPIO, pulse width modulation (PWM), quadrature encoder inputs and multiple high-performance ADC channels. TI offers comprehensive support for motor control applications through its MotorWare™ and C2000 controlSUITE™ libraries.

**System Management Controller**

Many systems, including industrial PCs and communications equipment, require a system management controller to monitor the health of the system and take proactive action, such as turning on a fan when the operating temperature is too high. With the rising complexity of today’s systems, 8-bit MCUs have neither the...
peripherals nor sufficient processing capacity to serve in this role without an expensive external FPGA or ASIC.

Tiva C Series MCUs based on the Cortex-M4 core provide a cost-effective, low power 32-bit platform capable of operating across industrial temperatures (see Figure 2). Built on a 65nm process, Tiva C Series MCUs achieve standby current as low as 1.6 uA and has a wakeup time of 500 us or less. By integrating the analog, digital and industrial interfaces required – including a platform environment control interface (PECI), low pin count (LPC) bus, multiple PWM outputs, high-performance ADCs, and CAN, USB, multiple UART, I2C and SSI/SPI interfaces – a single Tiva C Series MCU can provide a low-cost, single-chip solution.

In addition, by implementing configuration details in software, including customizable interface placement and routing, the resulting system design can accommodate many design variations with simple firmware changes that would otherwise require hardware-based modifications in FPGA- or ASIC-based designs. TI further simplifies design by providing a comprehensive SMC library to complement the TivaWare library.

![Figure 2 – Tiva C Series MCUs](image)

**Safety-Critical Applications**

Systems whose reliable operation is critical to the preservation of human life, including many medical, industrial and transportation applications, must meet stringent functional safety standards such as IEC 61508 and ISO 26262. In addition to following best practices, developers need a processor with safety features that can simplify safety-critical system design, reduce software overhead and speed certification. The result is a more reliable system at a lower cost and faster time-to-market.

Safety takes a two-prong approach. First, the system verifies that it is operating without fault. When a fault does occur, the system detects the fault and takes appropriate recovery measures.

TI’s Hercules MCUs use a “safe island” approach to balance cost without compromising safety (see Figure 3). Rather than unnecessarily verifying the operation of every aspect of the device, Hercules MCUs focus on the critical set of elements required to guarantee functionality while correct execution of software is continuously verified by hardware diagnostics. Specifically, multiple layers of diagnostics are in place to detect failures in power, clocking and reset circuitry. The result is a best practices balance between safe operation and safety overhead.

To verify core operation, Hercules MCUs operate dual ARM Cortex-R4 cores in lockstep. Effectively, both ARM cores receive the same input and must produce the same output. Any variance is detected near instantly.
with no impact on core performance and little penalty in terms of power consumption. To address common mode failure concerns (i.e., both cores experience the same fault), TI has taken measures including temporal diversity (i.e., operating the cores 1.5 or 2 cycles out of phase) and utilizing a voltage guard ring.

Hardware-based logic built-in self-test (LBIST) provides very high diagnostic coverage of the cores on the transistor level with the result of reducing safety software overhead up to 30 percent. Similarly, TI’s programmable memory BIST (PBIST) provides transistor coverage on all SRAM, including memory in peripherals. Memory is also protected using error correcting code (ECC) technology augmented by address and control parity capabilities.

The Human-Machine Interface (HMI)

One of the key differentiating features in electronic devices and systems today is their user interface. For any level of sophistication in the HMI, a graphical user interface (GUI) is required. TI provides a plethora of tools to facilitate HMI creation, ranging from frameworks with simple line and rectangle primitives to full-blown frameworks supporting 3D graphics, animation and textures. These tools greatly simplify interface design, enabling developers to drag-and-drop elements onto a virtual display and interact with the interface as a user might.

Graphic HMIs require significant memory and processing resources. GUI frameworks intended for desktop or smartphone applications, for example, require a minimum of a 300 MHz processor several MBytes of RAM. Thus, it is unwise to consider processor and GUI framework selection independently of each other.

There are many approaches to graphics development. Specialist GUI frameworks, like Prism from Bluewater Embedded, provide many capabilities while minimizing memory requirements (i.e., only 60 to 200 KBytes). However, these require developers to write programs to implement the GUI. Compare this to

![Figure 3 – Hercules Safety MCUs](image)
an environment like Adobe Flash which requires MBytes of RAM but eliminates the need to write any code. Android offers a well-integrated GUI framework with good touchscreen support. Linux has numerous options, including Qt, X-11, GTK, and DirectFB. These Linux frameworks are not part of the Linux kernel, and they can be time-consuming to learn to use for inexperienced developers. Many commercial OSes also have integrated GUI support, including Windows CE, VxWorks and QNX.

The default for a GUI framework is to utilize the main system’s programmable ARM cores. However, even simple 2D graphics can consume the majority of available cycles. A processor with an integrated graphics accelerator like the Sitara and OMAP processor platforms can substantially improve performance by shifting this load to a dedicated, optimized graphics core.

**Multimedia**

A system that requires multimedia functionality – video, audio, imaging or graphics – will have several characteristics that impact processor choice:

- **Supported codecs and decoders:** Each codec has different performance and memory requirements
- **I/O:** A system may need multiple digital and analog interfaces to support the different ways users will view or interact with content, such as S-Video, HDMI and DVI
- **Real-time constraints:** Teleconferencing and streaming video impose different latency and jitter requirements compared to simple video playback
- **Concurrency:** A system may have real-time functions that it needs to support in parallel to multimedia functions such as a high-data-rate communications channel for streaming applications
- **Quality:** Includes bit rate, frame rate, and display resolution. As each of these increases, so do processing and memory requirements
- **Number of channels:** Processing several channels of multimedia data can strain internal processor resources, including system buses
- **Latency:** Each processing stage adds latency that can disrupt sensitive applications like streaming audio or videoconferencing. Care must be taken to optimize the processing pipeline to ensure total processing time meets real-time requirements

Video playback and capture place tremendous processing and memory bandwidth requirements on a processor. MegaBytes of data need to be manipulated for every video frame or still image. TI processors offer a range of integrated functionality to accelerate design of multimedia applications, including audio and video processing capabilities, security and encryption processing, and integrated interfaces.

OMAP processors, for example, leverage an optimized camera software architecture to bring new use cases to Android-based applications while avoiding interconnect bottlenecks. The architecture of the OMAP 4 requires 47 percent less memory bandwidth for 1080p video playback compared to the GPU-based composition approach used by other leading ARM-based processors.

All new camera APIs in Android 4.x (e.g. 4.0 – Ice Cream Sandwich, 4.1-Gingerbread and 4.2 Jellybean) are fully supported by the OMAP platform and boosted by real-time enhancement algorithms for unprecedented image quality from a raw CMOS sensor. Therefore, OEMs can be assured that Android products built on OMAP will support the broadest set of features in the standard Android Camera Application, including advanced autofocus, low light support, zero shutter lag, face detection and tracking and video editing.
The OMAP architecture features an innovative Tiler memory management unit (MMU) to seamlessly blend buffer management across an OMAP device’s multiple cores. It achieves this by providing contiguous virtual memory to non-contiguous physical memory to reduce memory fragmentation of encoded and decoded video buffers. In addition, buffers can be addressed in 0, 90, 180 and 270 degree orientations, thus eliminating compute-intensive frame rotation and mirroring.

On OMAP processors, digital rights management content is protected end-to-end by firewalls, decryption, decoding, and rendering from the rest of the system. Specifically, the DRM agent is executed in a Trusted Execution Environment (TEE) that enforces strict boundaries within the processor, isolating it from external and even internal snooping.

**Gesture and Face Recognition**

3D vision capabilities are the foundation of several emerging technologies. Hand tracking, for example, is increasingly becoming a part of many HMIs, giving users more freedom and flexibility in how they interact with systems. Face recognition, used for personal and environmental security, is also finding traction in a range of markets. Augmented reality, where virtual objects interact with world objects, is changing the face of electronic gaming. TI’s OMAP and Sitara ARM-based processors are ideal for 3D vision applications, providing an optimal combination of hardware accelerated processing capabilities with a powerful development environment.

For hobbyists and serious engineers alike, one of the best ways to begin learning to develop with ARM is to take a hands-on approach using a development kit. TI designs its software and hardware development kits with ease-of-use in mind. Many of even the most advanced kits can be set up for evaluation within just minutes.

TI offers a wide range of evaluation boards for developers to choose from, complete with application-specific development tools to accelerate the design of a wide range of applications and functions. For entry-level development, TI recommends the following low-cost platforms:

**Tiva C Series LaunchPad:** This evaluation platform for the Tiva C Series ARM Cortex-M4 MCU is available for $12.99 and provides a low-cost way for developers to experience ARM development for themselves. With on-board emulation and a full development environment, developers have everything they need to program and debug a system. The LaunchPad platform also supports TI’s BoosterPacks, a series of modular plug-in boards that can stack on top of the LaunchPad baseboards and provide additional functionality, including wireless, capacitive touch, LED lighting and more. [www.ti.com/launchpad](http://www.ti.com/launchpad).

**BeagleBone:** Based on the Sitara AM335x ARM Cortex-A8 processor, the BeagleBone development board provides an excellent introduction to both the ARM architecture and open source development. It is supported by a complete Linux operating system with full-featured web servers, native compilers and scripting languages, video analytics libraries and much more. In addition to executing the Linux kernel,
many complete operating and development environments are supported by community and third-party developers, such as Android, OpenEmbedded, Windows Embedded, QNX, Ubuntu, Symbian, Debian, Fedora, Gentoo, FreeBSD and others. BeagleBone is well-suited for various mobile systems, robotics applications, web servers, Internet-enabled kiosks, media centers, home automation applications, thin clients, digital signage and many other types of low- and mid-range embedded applications. With its diverse universe of its open-source resources and own open source community at www.beaglebone.org, the BeagleBone is recognized as a stable, readily available development platform in its own right. www.beagleboard.org

Finally, developers need to consider the level of differentiation they expect to achieve through software. The combination of libraries, reference applications and frameworks that TI supplies can provide a large percentage of the required software for an application. This enables OEMs to enter a new market quickly.

To differentiate their products within a competitive industry, many OEMs will want to develop their own “special sauce,” as it is commonly called (see Sidebar, “Open Source: The Customization Challenge”). Given that a growing percentage of product functionality is implemented in software, this differentiation often takes the form of a new algorithm or process as developed by the OEM. For example, an OEM may have developed a more efficient algorithm to control a motor. To create a product that stands out because of superior battery operating life, an OEM may choose to optimize their software for power efficiency. To increase ease-of-use, the OEM may have developed an intuitive GUI.

The possibilities are endless, but only if the development tools provide enough visibility and the middleware framework in use is flexible enough. If not, developers may spend far too much time simply integrating their specialized code to work with the rest of the system.
By partnering with TI and ARM, developers can be confident that they will be able to design the most cost-effective systems in the least amount of time. The ability to leverage the industry’s broadest portfolio of ARM-based embedded processors means systems can be designed around a processor with the optimal combination of performance, peripherals and power efficiency. And, supported by the most extensive software and hardware development ecosystem available, developers can minimize development investment and time-to-market.

For more information on TI’s ARM portfolio, visit www.ti.com/arm. Support is available from the TI Engineer-to-Engineer community at www.ti.com/e2e. To order today, visit www.estore.ti.com. For a comprehensive list of whitepapers on using TI ARM-based processors in a wide range of applications, including industrial, safety, smart energy, communications, security, tablets, motor control, and many others, visit: www.ti.com/arm-whitepapers
IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment. TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

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

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

Reproduction of significant portions 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. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

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

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have not been so designated is solely at the Buyer’s risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products

<table>
<thead>
<tr>
<th>Audio</th>
<th><a href="http://www.ti.com/audio">www.ti.com/audio</a></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>OMAP Applications Processors</td>
<td><a href="http://www.ti.com/omap">www.ti.com/omap</a></td>
</tr>
<tr>
<td>Wireless Connectivity</td>
<td><a href="http://www.ti.com/wirelessconnectivity">www.ti.com/wirelessconnectivity</a></td>
</tr>
</tbody>
</table>

Applications

| Audio and Transportation | www.ti.com/automotive |
| Communications and Telecom | www.ti.com/communications |
| Computers and Peripherals | www.ti.com/computers |
| Consumer Electronics | www.ti.com/consumer-apps |
| Energy and Lighting | www.ti.com/energy |
| Industrial | www.ti.com/industrial |
| Medical | www.ti.com/medical |
| Security | www.ti.com/security |
| Space, Avionics and Defense | www.ti.com/space-avionics-defense |
| Video and Imaging | www.ti.com/video |
| TI E2E Community | e2e.ti.com |
| TI E2E Community | e2e.ti.com |

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2013, Texas Instruments Incorporated