WHITE PAPER

Atul Verma,
Business strategy and development manager,
Multicore processors
Texas Instruments

TEXAS INSTRUMENTS

Get Into the Zone: Building Secure Systems with ARM® TrustZone® Technology

Abstract

From consumer devices such as smartphones, tablets or set-top boxes to infrastructure elements such as base stations for the wireless cellular network or supervisory control and data acquisition (SCADA) systems, security is an important consideration in not only preventing theft but also, in some cases, catastrophic failures. In an increasingly connected world, where even cars will soon be connected to each other and to the communications infrastructure, new security concerns that were nonexistent just a few years ago are coming to the forefront. With the stakes as high as ever, hackers and rogue programmers have every incentive to go to painstaking depths to cause maximum impact and damage. Embedded systems will be the focus of next-generation attack vectors; as a result, a flexible, cost-effective and programmable security solution is needed to defend against these increasing attacks.

Introduction

Both the application and the environment where the system is executing will determine one or more of its critical security objectives. For instance, while authentication, encryption and repudiation measures of security are crucial for a system handling commercial transactions, physical protection is paramount for the security system guarding a nuclear power generation facility. Many tenets of security are extensively covered in the literature and beyond the scope of this paper. The main focus of this paper is to draw attention to aspects of security that are becoming increasingly important for those types of embedded systems which have been driven to a large extent by interconnectedness and openness. This paper also highlights ARM TrustZone technology, a system-wide approach to security on high-performance computing platforms, and describes how this technology can be used to build secure systems. TrustZone is supported by TI's KeyStone architecture. This paper explains various architectural features of KeyStone that are important in realizing the TrustZone framework. In addition to hardware, software plays an equally important role in supporting TruztZone's functional elements. The paper briefly touches upon those TrustZone features provided by TI and other features that can be implemented by developers.

Driving forces

Within the realm of embedded devices, several factors are elevating security concerns and making device trustworthiness a key design element:

- **Widespread use of open-source software.** Use of open-source software in embedded systems development is rapidly increasing as the Linux™ operating system (OS) spurs open-source adoption across a variety of industry segments. Apart from the OS, open-source software is available in almost every application area, such as databases, vision libraries and communication stacks, to name a few. Low cost, time-to-market advantages and rich feature sets are some of the leading factors behind the industry's adoption of open-source development. Despite all the advantages, open-source software is inherently more vulnerable than closed proprietary systems because open source software is widely available to anyone, including hackers, who are more than willing to figure out new ways to exploit the software's weaknesses.

- **Interconnectedness of devices.** Embedded devices are increasingly connected to the Internet and to each other, creating valuable information networks. However, this added value comes at a price. While an isolated, unconnected device can be quite secure, any device connected to most networks will require additional capabilities to protect it against external attacks.
- **Widespread use of embedded devices.** Embedded components such as microprocessors and microcontrollers are making their way into an ever increasing number of new applications, ushering in an era of the Internet of Things (IOT). Even appliances such as washing machines and refrigerators are being fitted with smart processors to provide advanced diagnostic capabilities and other convenient features. This trend is driving up security concerns, especially for devices employed in applications that deal with sensitive privacy issues and personal data.

Security scenarios for embedded systems

Security scenarios vary according to end application. For example, security considerations for a point-of-sale (POS) terminal may be quite different from those applicable to a traffic surveillance system. Embedded systems, because of their physical accessibility, also raise some unique security concerns. An Internet server adequately protected by firewalls has little to worry about from lab attacks, but this is a real threat for a small cellular base station deployed on user premises. Moreover, the value of the intellectual property (IP) in these devices makes them an enticing target for sophisticated attacks. The following section highlights some broad security concerns that can plague manufacturers of embedded products, although the list below should not be considered definitive:

1. **Running unauthorized software.** By running unauthorized software, competitors can bring to market knock-off products at lower prices, having spent little or nothing on hardware design. This is also of great concern to mobile phone manufacturers and carriers, who subsidize the purchase price of mobile devices through the consumer's commitment to a long-term service contract. By running unauthorized software, a mobile phone purchased at a subsidized cost can be used on another network or even in another country, thus depriving the original carrier of legitimate revenue. The implications of hackers running unauthorized software on embedded devices can be far more serious in other areas such as military and aerospace applications, where dangerous equipment and even state secrets can fall into unfriendly hands.
2. **Reverse engineering.** Several tools and techniques, many of them free and readily available on the Internet, can be used to reverse engineer embedded software to extract useful and valuable information. Disassemblers – tools for extracting file systems – are a prime example. These advanced utilities

can also be employed to further analyze how critical software elements are implemented. Not only does reverse engineering make the system vulnerable to attacks but it also exposes the vendor's IP to copycat implementations.

3. **Unintentionally running malicious code.** This is, of course, related to the interconnectedness of embedded devices, which, in turn, is important for the system's flexibility and upgradability. Malicious code or a trojan can be installed on an embedded device as the result of a compromised server, man-in-the middle attack (MITM) or an internal disgruntled worker attack. Depending upon the intent, this code can cause disruptive behavior, tamper with legitimate software or steal valuable information.
4. **Secure transactions.** Access codes, PINs, bank account numbers and other sensitive information are needed to conduct secure transactions over the Internet. If not adequately protected, malicious software code can steal this information and run up unauthorized charges.

What is needed for embedded devices is a comprehensive solution that addresses many of these security concerns at the system level. Device vendors need to provide a secure and scalable process by which OEMs are able to exchange security keys. Vendors also need to be able to program these keys into their embedded devices. In addition, elements inside the device and system software need to provide a framework that recognizes security attributes and provides controlled access. TrustZone from ARM Ltd. is one such framework. It spans silicon and software to provide a strong foundation for building secure solutions.

Introduction to TrustZone

Traditional approaches to security in embedded environments require a separate security processor with its own carefully controlled access and execution environment. Although highly secure, this approach suffers from a number of disadvantages, most notably high system cost and a lack of programmability. TrustZone technology, which alleviates these disadvantages, refers to security extensions implemented by ARM in a number of its cores, including the Cortex™-A15 processor. TrustZone is a system-wide approach in which security begins in the execution environment and permeates throughout the system's buses and IP blocks. Because it is hardware based, TrustZone provides a robust foundation upon which the upper layers of secure software can be built. Trusted Execution Environment (TEE) is one such implementation by Global Platform. It is discussed in the next section.

TrustZone provides the following foundational elements that are essential for hardware-based security:

- **Separate secure environment.** TrustZone splits processor cores into two virtual cores, one operating in a normal world and the other working in a secure world (Figure 1). This mechanism essentially creates another level of execution privilege in addition to the traditional demarcation of user and kernel modes. Transitions between the two worlds are carefully controlled by monitor mode software. In addition, each virtual processor has access to its own virtual memory management unit (MMU) so that clear separation between normal and secure page table translations can be maintained. Cache memories also have additional tag bits to distinguish between content cached by either secure or normal world cores. By this means, access to secure cached content from normal world masters can be denied.

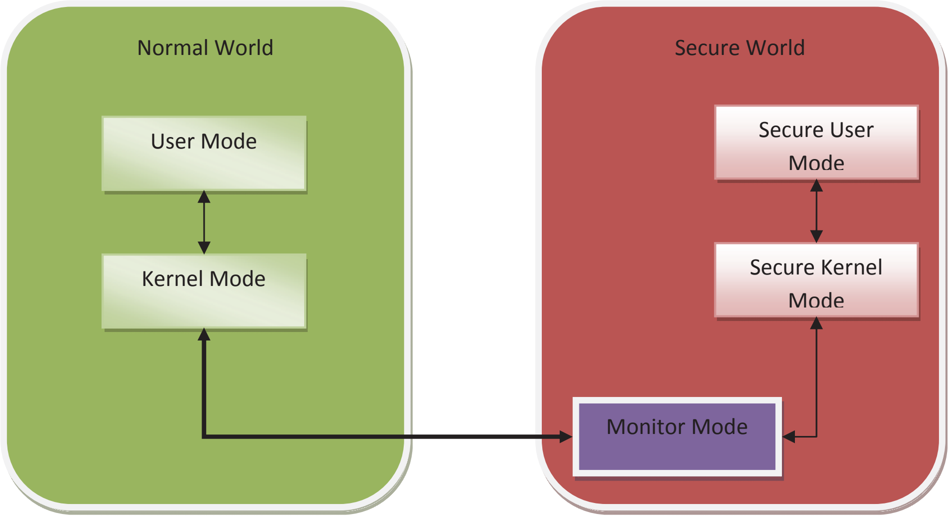


Figure 1: Normal and secure world environment[1]

- **Security aware bus and peripherals.** The internal bus has security awareness built in so that both secure and non-secure access can be easily distinguished. As a result, conditional access to different bus masters in the system can be authorized or not. Similarly, different peripherals can be made secured. For example, only secure masters might be allowed access to certain peripherals. With this combination of bus and peripheral security features, a keyboard, for example, might be made secure, allowing access to only legitimate secure software and preventing any malicious code from trapping keystrokes and stealing sensitive information.
- **Secure interrupts.** To aid in the design of sophisticated secure solutions, TrustZone provides secure interrupts for interfacing with secure peripherals. These interrupts can also be assigned higher priorities to guard secure world software against denial-of-service attacks.
- **Security aware debug.** While it is important to have the ability to debug secure software, it is equally important to make sure that not everyone who is able to access a device via the JTAG port will also have access to secure code. Therefore, the TrustZone architecture provides separate signals to control secure and normal world software debugging such that secure world debugging can be enabled when the device is in a physically trusted location – for example, where trusted software is being developed – and disabled in production devices.

While TrustZone provides the fundamental ingredients for hardware-based security, software layers are needed above it to fully utilize these features and build a robust secure solution.

Building secure solutions

Just as the hardware infrastructure is key to laying a robust foundation for security, so too is the software architecture in creating secure solutions. Although there are several possibilities for the software architecture, the details of the security design and its complexity is largely determined by the application's use cases and the value of the assets that are being protected. Regardless of the overall software architecture, three key pieces of software are needed to implement a secure solution: secure boot for starting the device, platform software to manage the secure and normal worlds, and lastly, secure applications to provide service to the user.

Secure boot

The starting point for any secure solution is when the device first boots up because a device that starts up in a compromised environment can potentially circumvent all other runtime security measures. Since security should start as early in the boot process as possible, on-chip ROM code plays a key role in the secure boot process. It brings up a trusted software image that is signed with the OEM's private key. The public counterpart of the OEM's private key is programmed into the device only once during manufacturing. A trusted OEM image of the software subsequently boots the secondary boot loader which in turn boots the high-level operating system. This scheme allows for setting up a complete chain of trust. That is, a first-level component can have another embedded public key to validate the next-level component that it is trying to load and so on. Such a secure chain of trust can be extended all the way to the loading of secure applications. Figure 2 depicts a typical secure boot flow.

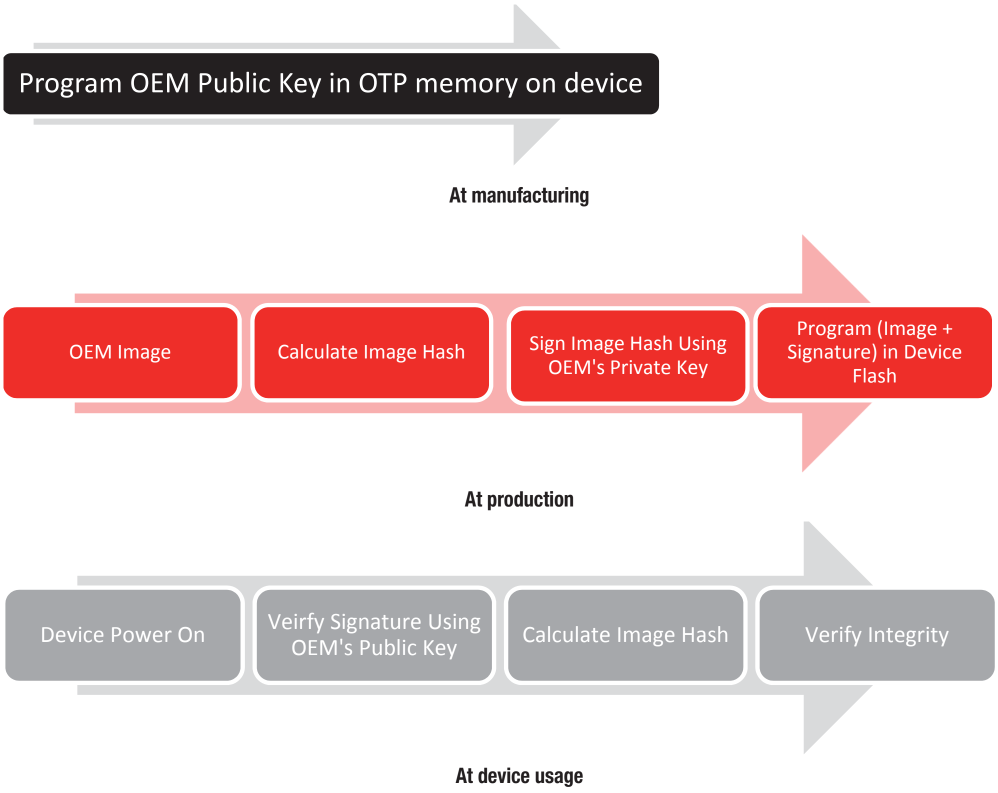

Program OEM Public Key in OTP memory on device
At manufacturing
OEM Image
Calculate Image Hash
Sign Image Hash Using OEM's Private Key
Program (Image + Signature) in Device Flash
At production
Device Power On
Veirfy Signature Using OEM's Public Key
Calculate Image Hash
Verify Integrity
At device usage

Figure 2: Sample secure boot flow

Platform software

Since TrustZone provides hardware capabilities to separate normal and secure worlds, it follows that any software architecture utilizing these features will implement similar partitioning. In this model, the secure world handles all security-related functions, including interfacing with secure peripherals. The normal world handles all other tasks. Secure world also hosts all secure applications and may provide services that are carefully brokered through monitor code to clients residing in the normal world. One key element here is the restriction that the secure world can only run code that has gone through extensive vetting. It is not possible to run arbitrary code downloaded as an application from the Internet, for instance, inside the secure world. With this scheme any damage from malware or trojans is confined to the normal world, thus limiting the scope of attacks against the device.

The secure world does not have to implement complex software to provide meaningful security. In some cases, excess software actually hinders security goals by hiding additional vulnerabilities. The choice of security applications software and their implementations will depend on the user application and usage scenarios. Implementations can range from a sophisticated fully preemptable OS to a set of passive libraries providing on-demand services to the normal world. The TrustZone architecture does provide a secure timer and security-aware interrupt controller for building a preemptable secure OS.

Applications

Recognizing the difficulty in porting secure applications, ARM has released a standardized application programming interface (API) called TrustZone API (TZAPI)[2]. This API makes higher layer applications transparent to the underlying TrustZone implementation. By building on this work, GlobalPlatform has defined a Trusted Execution Environment (TEE) API (Figure 3) that is split into two parts:

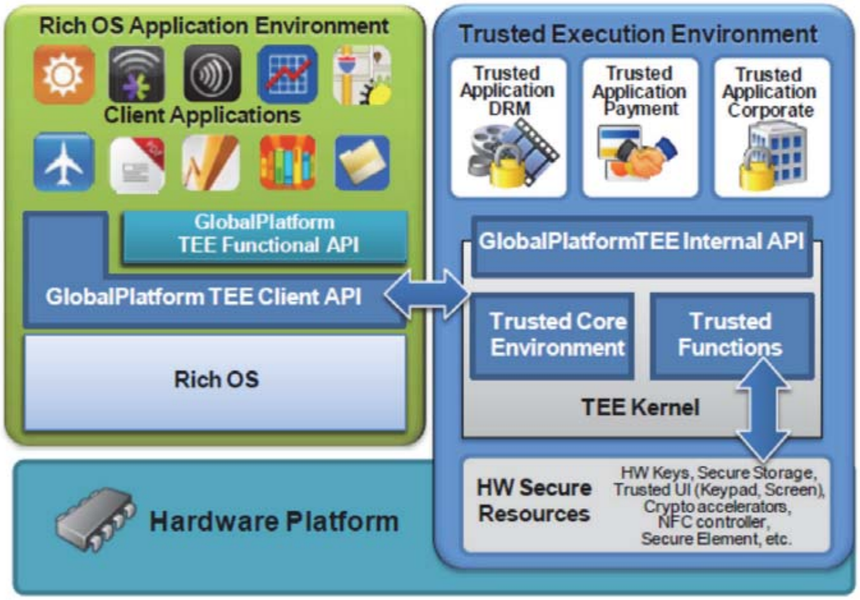

Rich OS Application Environment
Client Applications
GlobalPlatform TEE Functional API
GlobalPlatform TEE Client API
Rich OS
Trusted Execution Environment
Trusted Application DRM
Trusted Application Payment
Trusted Application Corporate
GlobalPlatformTEE Internal API
Trusted Core Environment
Trusted Functions
TEE Kernel
HW Secure Resources
HW Keys, Secure Storage, Trusted UI (Keypad, Screen), Crypto accelerators, NFC controller, Secure Element, etc.
Hardware Platform

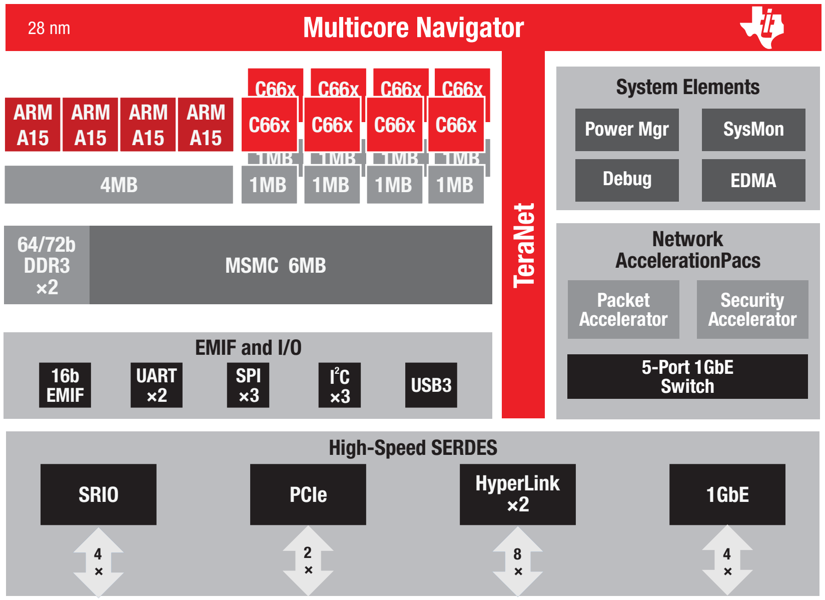
Figure 3: TEE Architecture as defined by GlobalPlatform[3]

- The TEE Client API runs in the normal world and provides a portable way for applications to communicate with and utilize services from the secure world
- The TEE Internal API resides in the secure world and provides a standardized way for secure tasklets and applications to access features provided by the secure OS

As shown in Figure 3, the secure environment stores all sensitive information (this means that the hardware will only allow access to masters which have a secure bit turned on in their bus access) and also provides an execution environment for trusted applications. Trusted applications gain access to security resources through the TEE Internal API.

KeyStone II architecture

KeyStone II refers to the System-on-Chip (SoC) architecture for high-performance devices from Texas Instruments featuring an integrated ARM Cortex-A15 cluster. One such device from the KeyStone II family is shown in Figure 4.



40mm × 40mm package

Figure 4: TI KeyStone II-based 66AK2H12 SoC

ARM licensees may choose not to implement all the security extensions that are part of TrustZone. TI's KeyStone II family of SoCs implements a large number of security features in addition to those specified by TrustZone. Some of these capabilities include:

- **CorePacs**. ARM Cortex-A15 cores inside the KeyStone II architecture implement security extensions specified by ARM as part of the TrustZone framework. In addition, TI's C66x DSPs, if present, also implement security extensions that enable different memory protection regions.

- **Secure boot.** ROM code inside KeyStone II devices supports a staged boot process that is capable of booting the device from encrypted and signed code images stored on external media.
- **Security aware interconnect.** A high-speed internal bus called the Teranet carries security attributes so it can appropriately distinguish between secure and non-secure masters. This, in combination with memory protection units, can be used to selectively allow only secure access to some peripherals.
- **Key storage.** KeyStone II devices provide multiple programmable EFUSE areas to safely store various customer keys and/or their hashes. One-time programmable (OTP) memory is also available for users to program other ancillary information. In addition, TI specifies an open manufacturing flow so that OEMs can program their own keys in a secure manner even when manufacturing is performed by third parties or contract manufacturers.
- **Debug.** KeyStone II devices can disable the JTAG interface completely or provide restricted access to enable or disable secure debugging.
- **Secure interrupts.** Interrupts can be classified as secure or normal in KeyStone II devices, empowering secure world software with advanced capabilities.
- **Scrambled external access.** Without incurring any performance penalty, all external memory interfaces support a scrambled data bus that is useful in preventing various kinds of lab attacks.
- **Security accelerators.** KeyStone II devices have on-chip security accelerators that support a variety of encryption schemes, including AES, DES, 3DES, SHA1, SHA2, MD5 and True Random Number Generation (RNG). In addition to aiding secure software, these accelerators can be used by boot code to support a secure boot process and efficiently decrypt a software image.

While the user’s needs for a secure solution vary by application, it is easy to see that KeyStone II devices provide the hardware elements necessary to build a robust security solution. Secure boot and an open manufacturing flow may be sufficient for a large number of users. For a fully secure kernel implementation, users can either leverage a free open-source implementation called Open Virtualization[4] or tap into a strong network of knowledgeable partners to build advanced secure kernel capabilities.

Conclusion

Just as the Internet and e-commerce brought into focus security concerns and various kinds of attack vectors, embedded systems will bring into focus a new set of security scenarios. The reason for this is the trend toward making everyday devices smart and interactive. Their increased processing power together with connectivity makes these embedded devices an enticing target for attacks. The TrustZone architecture from ARM is an elegant and cost-effective solution that addresses many of the security concerns for embedded systems in a holistic manner. The KeyStone II family of devices from TI not only offers high performance in a wide variety of applications, but it also provides many security extensions and secure capabilities to build and deploy solutions that are protected.

Learn more by visiting **www.ti.com/multicore**.

References

[1] http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf

[2] http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.prd29-genc-009492c/CACCICDE.html

[3] http://www.globalplatform.org/documents/GlobalPlatform_TEE_White_Paper_Feb2011.pdf

[4] http://www.openvirtualization.org/index.html

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		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265