An Introduction to Software Development for Functional Safety on TI Processors

A functionally safe electronic system is one that responds as expected for every set of inputs. Functionally safe systems are developed and validated against well-defined market-specific functional safety criteria. The umbrella standard for the majority of application-specific functional safety development in electronics is IEC 61508. It specifically addresses electrical, electronic and programmable electronic safety-related systems. Many market specific safety standards and guidelines have been derived from IEC61508:

- ISO 26262 for automotive passenger vehicles
- EN50128 for software development of railway applications
- IEC61513 for nuclear power plants
- IEC61511 for the process industry and associated instrumentation
- IEC62061 and ISO 13849 for machinery electrical control systems
- IEC62304 for medical systems

COTS (Commercial-Off-The-Shelf) software (such as a Real-Time Operating System (RTOS)) developed according to IEC61508 provides a valid baseline on which programmers can develop functional safety software for most market sectors. However, developers should seek to gain clarity from their suppliers on the additional safety requirements specific to the target market segment. For example, DO-178B for airborne systems uses many of the same principles of functional safety software development as IEC61508, but it pre-dates IEC61508 and takes a different approach to safety certification. So while companies supplying solutions certified to IEC61508 have a good start point for functional safety software development to DO-178B, you will generally need to seek partners with specific competence in executing this standard.

Independent assessment is an important part of functional safety system development. In some market segments, assessment and certification is undertaken directly by notified bodies (such as the US Food and Drug Administration), whereas in many others these functions are undertaken by parties independent of the developer. These are often private companies who have developed a reputation for technical competence, consistent quality, and rigorous execution of their assessments. Companies such as TÜV and exida have garnered strong global reputations, such that the use of their names in conjunction with a functional safety certification claim can be a major marketing benefit to the developer.

Safety Integrity Level (SIL)

In the early stages of functional safety development a hazard and risk assessment of the target application is performed. This process is used to determine the level of acceptable risk for the application. Similar levels of acceptable risk are quantified using Safety Integrity Levels, or SILs. IEC 61508 offers SILs from 1 to 4, with 4 being the highest safety integrity level. In general, the lower the acceptable risk, the higher the SIL target, and the more challenging the requirements for product development.

Hardware, software, and systems can be developed according to IEC 61508 targeting a specific SIL. All three development types must comply with a number of requirements to manage systematic failure. Hardware and systems are also evaluated via quantitative metrics to determine probability of failure per hour and the safe failure fraction in the end application. Software has no inherent failure rate, and as such is only evaluated systematically.

A SIL rating cannot be achieved except by evaluating a completed and assessed system. For each component in the system, it can only be said that the component is “developed according to SIL x” or “suitable for use in SIL x systems.” Further, the SIL rating achieved by a system is only as strong as its weakest link – a single SIL 1 design element in a product could limit the achieved SIL of a system which otherwise achieves SIL 3. In the case of a COTS product, the end application is not known and assumptions must be made on the system level design and safety requirements. For this reason, it is advantageous for the COTS developer to target development to the requirements of the highest SIL anticipated to be implemented, and for the system integrator to carefully review component safety documents before selecting a COTS product.

So where does this leave software? Part 3 of IEC 61508 deals with software in the context of the system and is one of the mandatory aspects of the standard that needs to be addressed in order for a system to achieve a SIL rating. Unlike hardware, software has neither potential to wear out nor any random failure modes. In effect, one could argue that perfectly crafted software never fails! And therein lies the SIL objective for software – it’s all about the level of systematic test and development process wrapped around the software development process that determines the SIL to which software can be assessed, or to put it another way, how perfectly crafted it is. The more rigorous and systematic the development process, the higher the SIL rating that can be achieved. SIL3 tends to be the highest standard to which a broad range of systems is developed. SIL4 is an ultra-high safety objective much more rarely sought by system procurers. SIL4 is technically impossible to achieve without multiple channels (such as two or more instances of dissimilar software) – thus a single component like an RTOS capable of SIL3 performance cannot on its own provide the basis of a

solution to a SIL4 system objective. However, two or more RTOSs sourced from separate suppliers, each assessed capable of SIL3 performance, could be architected to run in a single system to achieve a SIL4 system rating.

Using COTS software assessed as being capable of SIL3 performance does not guarantee your system will achieve a SIL3 certification. However, COTS software developed for functional safety should come with all the documented development and test processes. When coupled with a functionally safe hardware design, you have a significantly increased chance of meeting the functional safety assessment criteria for your system. The supplied documented systematic software development practices provide an excellent proven guideline for the application software development process; the application developer can re-use these development guidelines if they do not have their own already in place.

A Systematic Approach to Risk Mitigation

The end objective of any system developed to meet safety standards is not to pass the assessment test, but to actually deliver a system that is functionally safe. This means that when (because it's not a matter of if, there is always a chance of failure) something fails in the system, it does so in a manner that does not put persons at unacceptable risk. However, we as humans, both as users of systems and as developers of them, are imperfect beings – the Darwin Awards testify to that! It is inconceivable that we can develop a perfectly safe system with zero risk. One of the ultimate objectives is to be able to stand up in a court of law and say "we did everything conceivably possible to ensure this system was safe." What is your defense? It can only be one thing – a systematic proof that the system was developed to take into account every known failure condition that could lead to an unsafe situation, to avoid those which can be avoided, and to mitigate those you cannot either avoid or detect. The reason it has to be systematic is because that's the only way to address the fact that even the developers of the system are prone to error and mistakes. Systematizing the process mitigates that reality.

The safety mechanisms selected to address an identified risk can be wide and varied. Just because the system you are developing is a complex piece of programmable electronics does not mean that every risk can be addressed by a bit of clever software and sharp engineering. If you are controlling a laser cutter, and there's a chance that it could be directed by the computer in a manner that may put life at risk, you don't necessarily need to develop clever software to identify and mitigate that risk. A superior solution may be to put a shield or physical movement inhibitor around the laser – even though this may not be an example of functional safety per IEC 61508. This is why there are market-specific functional safety standards with application-specific criteria. Each one brings to bear an understanding of the market-specific acceptable risks and challenges to the safety assessment process.

Let's take the simple example of selecting the programming language to use for your software development. Can you use C? There is a body of opinion that implies the answer is no and drives developers towards 'safer' languages like ADA – yet time has shown that this argument does not hold water. In fact, many SIL3-capable COTS software products, such as an RTOS, were developed in C. The key is to develop systematic processes for use of C that mitigate the risk factors associated with it. In fact, many of the riskier facets of coding in C can be directly addressed by using an RTOS and its underlying functionality. However the challenge for a COTS vendor of an RTOS is that the customer in a functionally safe system development may have to select a specific compiler other than the one chosen by the RTOS vendor in their assessment process, perhaps due to constraints from the assessor or due to the demands of other 3rd party software which must be integrated. Does using a different compiler invalidate the SIL3 capability of a COTS RTOS? Well the answer is – it depends – if the vendor develops their certification process with a tools-independent perspective, then it becomes possible to select any compiler. This step should only require a change impact analysis followed by re-execution of the existing SIL3-accepted test suites in order to demonstrate confidence in the compiler output to the safety assessor. It is the anticipation of this sort of issue, combined with accessibility to an in-depth knowledge and understanding of the challenges of software development in a functionally safe system development, that can add such huge value to working with both hardware and software suppliers that have developed their components according to the target functional safety standard.

The Hardware/Software Integration Challenge

Integrating software onto a specific processor and hardware environment is one of the more challenging areas of a system to develop and assess to demanding certification requirements. Just defining how and where software is placed in memory is one critical component of the process, but then defining how the I/O resources and processor time is allocated and being able to systematically define this to a safety assessor, becomes a huge challenge. It is one of the main reasons why most systems with a significant proportion of software in their designs decide to use a pre-validated RTOS, whose task is to manage these resources on behalf of the application in a manner that has been tested to meet the most rigorous safety assessment criteria.

Selecting an MCU, rather than selecting a processor and integrating your own peripherals, is also a key decision. The I/O and memory layout and management are fixed across multiple designs when you select an MCU. It can therefore be more readily assessed systematically and probabilistically for its ability to help meet hardware SIL

requirements in a system design as it integrates most of the challenging areas of hardware development.

TI has worked with SAFERTOS® vendor, WITTENSTEIN High Integrity Systems, to provide two separate approaches to pre-integration for functionally safe system development.

SAFERTOS® pre-integrated into ROM – TI LMS3S9B96

SAFERTOS® is pre-certified suitable for use in IEC 61508 SIL3 applications. Working with TI, it was the first SIL3 RTOS to be available in the market pre-integrated into ROM.

What this means is that the WITTENSTEIN's Design Assurance Pack (DAP) delivered with this integrated solution has already been tested in the context of memory utilization and I/O integration of this device, considerably reducing the end user's integration time and risk and generating significant savings in the functional safety certification process. This level of certified pre-integration removes one of the more challenging tasks from the long list of certification preparation work that developers would otherwise have to do. With the RTOS software in ROM the application developer cannot do anything but use the pre-tested code in exactly the way it was originally tested, thus removing that part of the code base from the re-test burden.

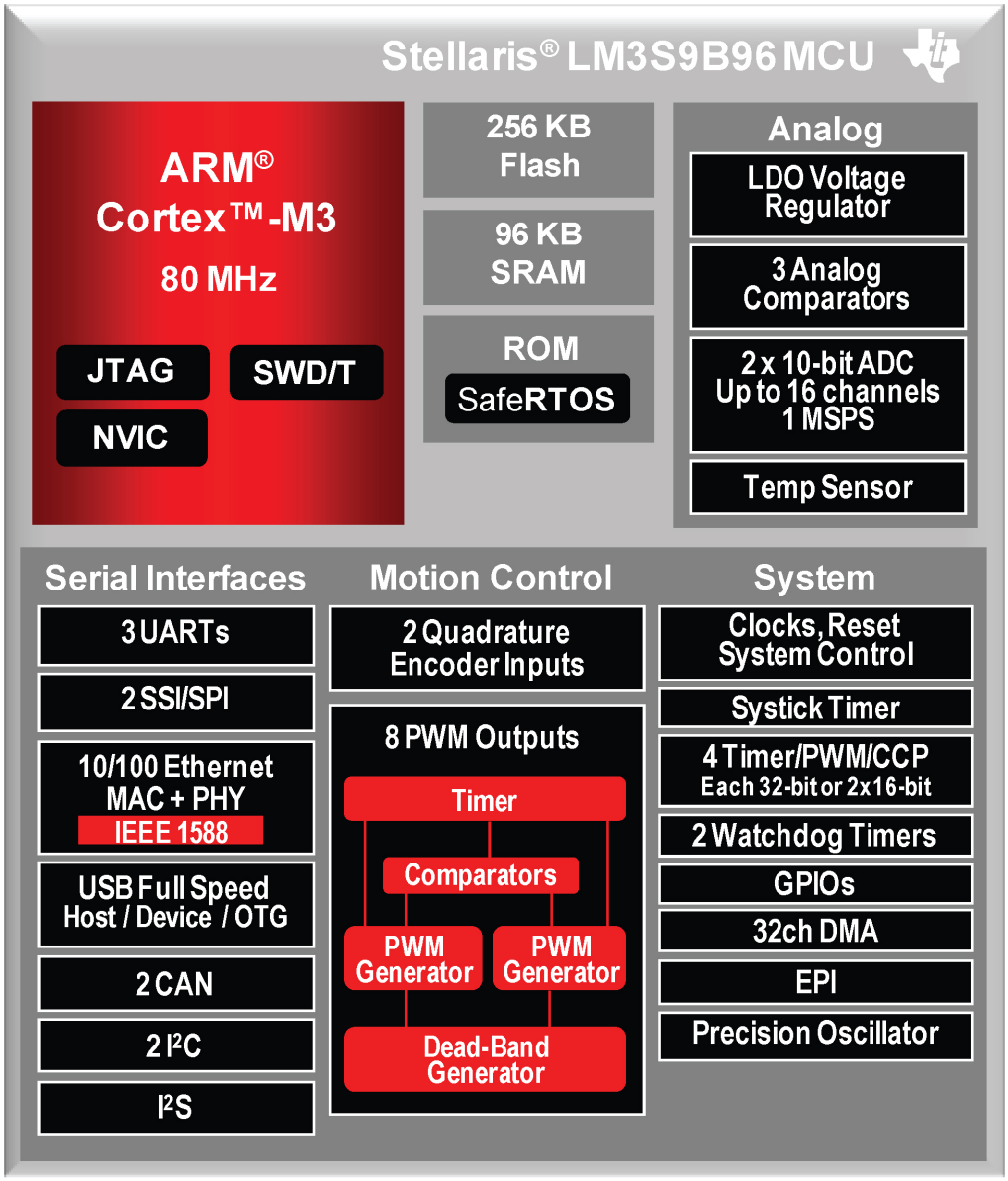

Fig.1 The Stellaris LM3S9B96 MCU with SAFERTOS® in ROM

SAFERTOS® for this integrated solution has also been pre-validated for compliance with FDA510(K) Class III device standards and EN62304 for medical devices, making it suitable for applications such as diabetes pumps, infusion systems and hypertension monitoring systems and other medical devices.

The LMS3S9B96 comes in both QFP and BGA packages and is suitable for a wide range of industrial and control applications that can benefit from SAFERTOS® pre-installed in ROM direct from the TI factory.

SAFERTOS® Ports to the Hercules™ family of MCUs

The TMS570LSxxx and RM48x are flagship safety processors for TI, select examples of which have been independently assessed as suitable for use in IEC 61508 SIL3 applications by exida. These processors were designed from the ground up to address the systematic risks associated with MCU development as well as manage random failures seen during operation. The product family features lock-step dual-cores, which means both CPUs process the same code-and-data stream and a fail-safe detection circuit compares the results. Both families of processor are fully supported by SAFERTOS®. WITTENSTEIN High Integrity Systems also plan a port (as of Q3 2011) to the lower cost TMS470M family of devices which feature a common set of safety certified memory and I/O but with a single core processor solution.

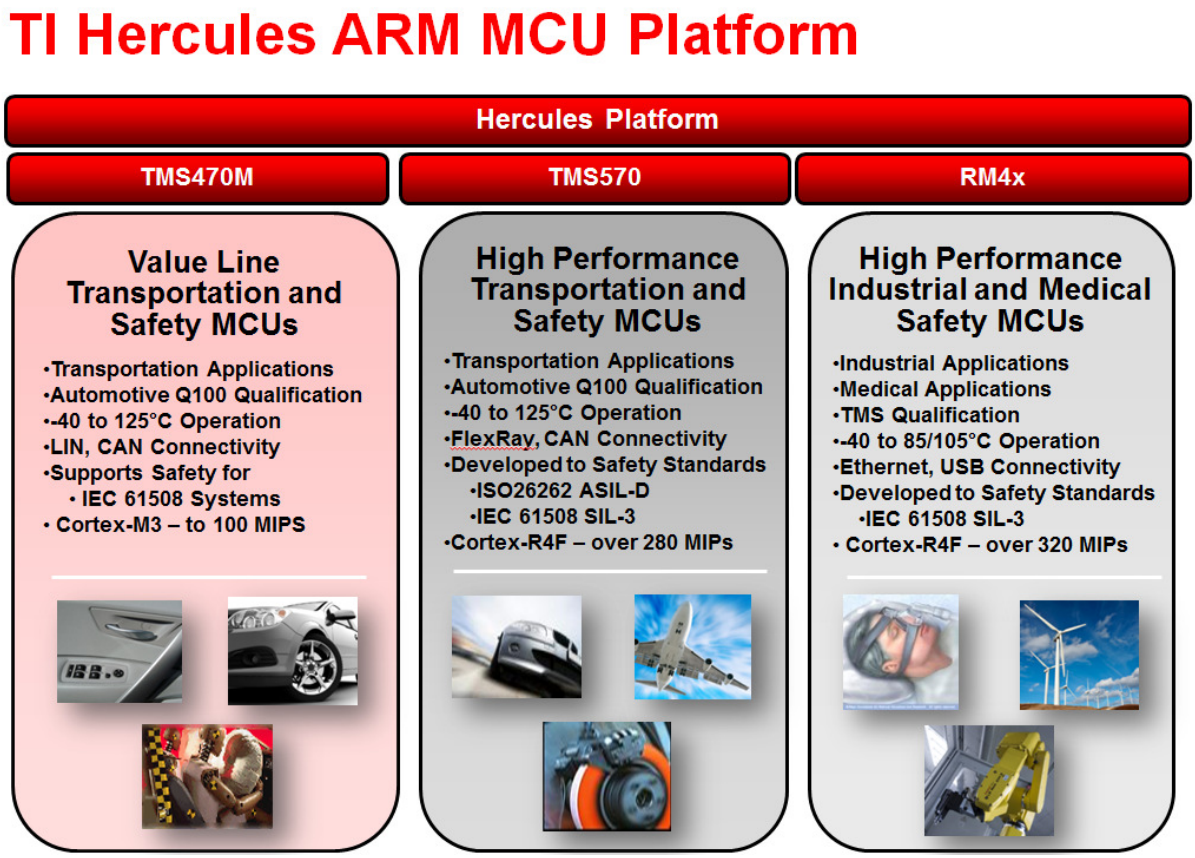

Fig.2 The Hercules ARM Family of safety MCUs

As functionally safe MCU designs, TI went much further with the memory and I/O integration and capabilities than other MCUs; they can be configured to use ECC to automatically correct 1-bit errors in memory on the fly and detect and flag 2-bit errors on SRAM, Flash, and memory interconnect. Even the peripheral queues and buffers have runtime parity checking. They also include a memory CRC checker in hardware to enable developers to validate the integrity of code in memory at any time. On top of all this, they include dedicated RAM and CPU checkers, thus removing the need for the development of software tests that then have to be integrated into the general memory of the existing system and validated for effectiveness. This sort of separation is important to developers of stringent safety critical systems as it avoids the problem of using RAM to test RAM – which came first the validated RAM or the RAM memory tester application?

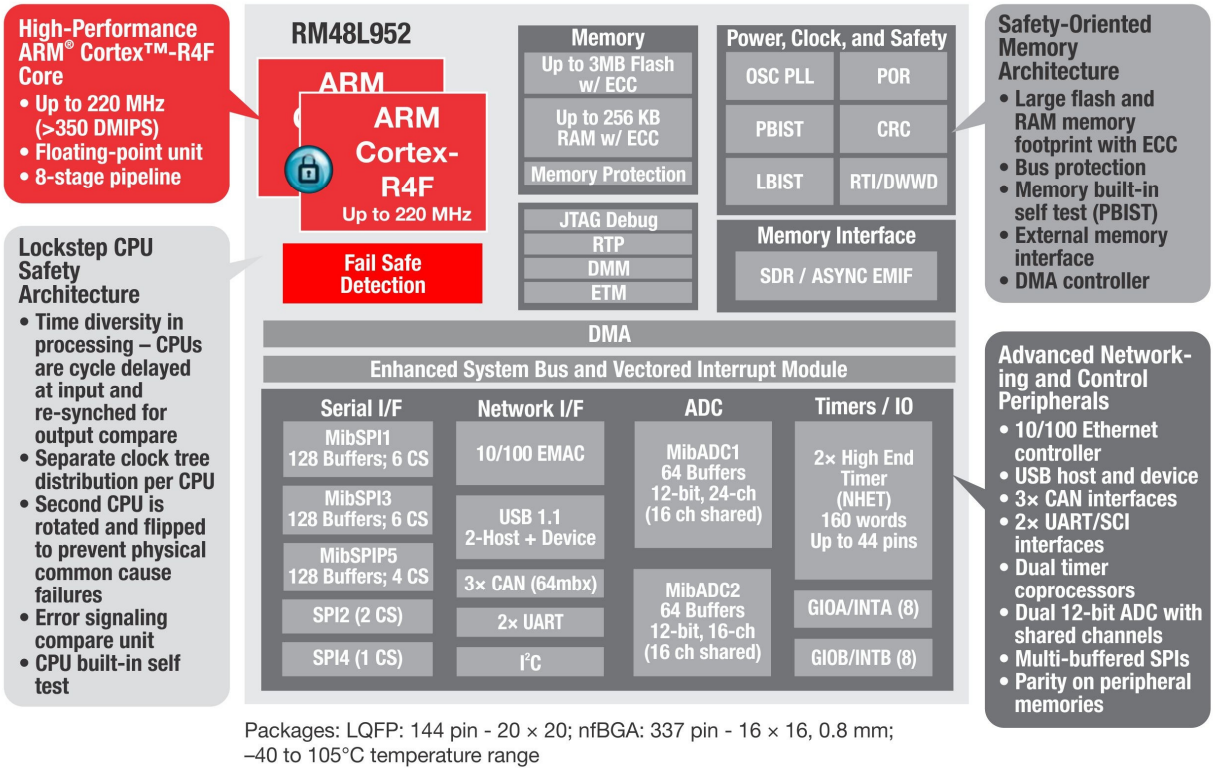

Fig. 3 – RM48L952 for safety critical industrial and medical applications

Combining a processor designed from the ground up for functional safety systems certification with an RTOS similarly designed and developed with the same objectives in mind is a huge de-risking proposition for safety critical systems developers as they prepare their case for the independent certifying body. SAFERTOS® has been ported to these comprehensive devices and together they provide an ideal platform for safe system development in transportation, rail and aerospace for applications such as braking, electronic motor control (power steering, electric powertrain), battery management, and driver assistance.

Summary

The partnership between TI and WITTENSTEIN High Integrity Systems delivers an accelerated development platform for functionally safe systems. The combined expertise in hardware systems and software development gives the best possible start to those initiating their first safe system development and a trustworthy base for those already experienced in complex functionally safe system.

About the Authors

Sue Cozart - Applications Engineer, Stellaris® Microcontrollers, Texas Instruments
Sue is an applications engineer for Texas Instruments' Stellaris® ARM® Cortex™-M microcontrollers. For over 15 years, Sue has worked in the semiconductor industry in a variety of engineering, marketing, and managerial roles. Sue was part of the team at Luminary Micro, the creators of the Stellaris MCU platform. Sue began her career at Motorola, working with 68K and ColdFire microprocessors. Sue holds a BS in Engineering Physics and an MBA from the University of Colorado. Sue can be reached at sue.cozart@ti.com.

Andrew Longhurst - Engineering Manager, WITTENSTEIN High Integrity Systems
Andrew has worked in the Medical, Aerospace and Automotive since 1993 and has worked for WITTENSTEIN since 2000 developing flight qualified systems, with a focus on Quality Assurance. Andrew became Engineering manager in 2006 for WITTENSTEIN aerospace & simulation Ltd. Andrew holds a BEng in Electrical & Electronic Engineering and an MSc in Robotics & Automation. Andrew can be reached at andrew.longhurst@wittenstein.co.uk.

Karl Greb – Functional Safety Technologist, Transportation and Safety Microcontrollers, Texas Instruments
Karl is a 15 year veteran of TI having worked in product test, applications engineering, architecture, and systems engineering roles for a variety of end equipment including high reliability mass storage and automotive. Karl is a member of the ISO 26262 standard international working group as well as the SAE Automotive Functional Safety Committee. Karl holds a B.S. in Computer Engineering from Texas A&M University. Karl can be reached at kgreb@ti.com.

IMPORTANT NOTICE

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

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

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

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

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

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

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

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

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

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

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