Technical White Paper

Leveraging USB Capabilities on the AM261x Microcontrollers

Shaunak Deshpande

ABSTRACT

The demand for efficient, flexible, and reliable fast communication interfaces in embedded systems is escalating, particularly in industrial and automotive applications where real-time data processing, remote management, and user interaction are needed. The Texas Instruments AM261x microcontroller stands out in this landscape by offering advanced USB capabilities that facilitate seamless connectivity and robust data handling. Utilizing the robust open-source TinyUSB stack and a hardware USB 3.0 controller IP, the AM261x enables the implementation of multiple USB classes, makes sure that developers can meet the diverse application needs.

This white paper explores the pressing market requirements for USB functionality in embedded systems and discusses the design challenges that engineers face in developing solutions that are both efficient, user-friendly, and at the same time easy to implement and maintain. By examining the hardware and software capabilities of the AM261x, alongside real-world use cases and applications, this document highlights how the microcontroller effectively addresses these challenges.

Table of Contents

List of Figures

Trademarks

1 Introduction

Universal Serial Bus (USB) has become a ubiquitous standard for connectivity due to its versatility, ease of use, and ability to support high-speed data transfer. The Texas Instruments AM261x microcontroller is engineered to leverage USB functionality effectively, integrating advanced features that allow developers to create systems capable of performing complex tasks while maintaining high levels of reliability and user accessibility.

The AM261x microcontroller is has a noteworthy integration of the USB hardware controller along with TinyUSB stack, which provides robust support for various USB classes. By offering features such as Device Firmware Upgrade (DFU) and Communications Device Class (CDC) along with other device classes that can be enabled quickly via the TinyUSB stack. The AM261x empowers developers to create systems that are not only efficient but also responsive to the evolving needs of users and industries. The USB PHY on AM261x supports USB 2.0, which means that the following features of USB 2.0 can be used:

- **Data Transfer Rate**: Supports a maximum data transfer rate of 480Mbps (High-Speed).
- **Backward Compatibility:** Fully compatible with USB 1.1 devices.
- **Improved Power Management:** Allows devices to draw up to 500mA from the bus.
- **Multiple Device Classes:** Supports various device classes such as CDC, DFU. Can support classes such as NCM, RNDIS, MSC, and so forth.
- **Hot Swappable (plug-n-play):** Enables devices to be connected and disconnected without powering down the system.
- **Cable Length:** Supports cable lengths of up to 5 meters for standard USB cables.
- **Point-to-Point Connection**: Utilizes a point-to-point architecture for direct communication between devices.
- **Increased Bandwidth:** Offers enhanced bandwidth to accommodate high-speed devices.

1.1 Real-World Applications for USB

The industrial landscape is rapidly evolving, driven by the need for more connected devices that can operate efficiently and autonomously. As industries integrate automation and Internet of Things (IoT) technologies, the following key market needs have emerged:

- **Enhanced Inter-connectivity**

  With the rise of Industry 4.0, the ability for devices to communicate with each other and with central control systems has never been more critical. USB serves as a reliable interface for establishing these connections, enabling devices to share data in real time and coordinate actions effectively. USB can be interfaced with various other technologies such as universal asynchronous receiver/transmitter (UART), CAN, Ethernet, and so forth, all of which are inherently supported on the AM261x devices. The USB 2.0 supports speeds up to 480Mbps
- **Remote Monitoring and Management, Firmware Upgrades**

  As organizations strive for operational efficiency, the need for remote management capabilities has grown. Embedded systems must support secure and straightforward methods for firmware updates, configuration changes, and diagnostics without requiring physical access. This is essential for minimizing downtime and reducing maintenance costs.
- **Real-Time Data Processing and Analytics**

  Data-driven decision-making is a cornerstone of modern industrial practices. Embedded systems need to log, process, and transmit data in real time to facilitate timely insights and operational adjustments. USB's high-speed data transfer capabilities enable efficient data handling, making it an ideal choice for applications that require quick access to critical information. With high speeds and support for multiple connections,
- **User-Friendly Interaction**

  For embedded systems to be effective, they must also be user-friendly. Designers need to create interfaces that allow users to interact with devices easily, whether for configuration, calibration, or data retrieval. USB provides a simple and standardized way to connect human-machine interfaces (HMIs) and other peripherals, streamlining the user experience.

- **Security and Compliance**

  With increased connectivity comes the necessity for enhanced security measures. As devices become more interconnected, the risk of unauthorized access and data breaches rises. Embedded systems must incorporate secure communication protocols, ensuring that firmware updates and data exchanges are protected. The USB capabilities of the AM261x enable secure authentication and access control measures, safeguarding sensitive operations in critical infrastructure applications.
- **Scalability and Future-Proofing**

  Finally, as technology continues to evolve, there is a pressing need for embedded solutions that are scalable and adaptable. The AM261x's support for multiple USB classes allows developers to future-proof their designs by easily integrating new functionalities and addressing emerging market demands without overhauling existing systems.

1.2 Acronyms Used in This Document

- CAN - Controller Area Network
- CDC - Communications Device Class
- DCD - Device Core Driver
- DFU - Device Firmware Upgrade
- DMA - Direct Memory Access
- HID - Human Interface Device
- HMI - Human-Machine Interface
- IoT - Internet of Things
- LwIP - Light Weight TCP/IP stack
- Mbps - Mega bits per second
- NCM - Network Control Model
- OTG - On The Go
- RNDIS - Remote Network Driver Interface Specification
- SoC - System-on-Chip
- TRBs - Transfer Request Buffers
- UART - Universal Asynchronous Receiver Transmitter
- USB - Universal Serial Bus

2 AM261x USB Offering

USB 2.0 supports a high speed interface supporting speeds of 1.5Mbps (Low speed), 12Mbps (Full speed) and 480Mbps (high speed). A USB descriptor can either be a device or Host (Interface). The AM261x supports both modes. The AM261x supports various USB classes for a variety of applications such as HID, CDC, DFU, Audio/Video device, Mass storage etc, using different types of endpoints. These endpoints include Control, Bulk, Interrupt and Isochronous endpoints. The AM261x can support all the four endpoints and various application classes. The below sections discuss the USB on AM261x briefly.

2.1 AM261x USB Hardware

The AM261x has a hardware USB controller that supports USB 3.0 with a software configurable architecture, has an internal DMA controller, descriptor caching, multiple transfer queueing support, Power saving features, Dynamic FIFO memory allocation for endpoints, low MIPS requirement and Interrupt moderation (in host mode) as well. This feature packed IP is integrated on the AM261x to work with the ARM R5F core. Figure 2-1 shows how the controller is interfaced with the R5F and the USB 2.0 PHY along with the memories it uses.

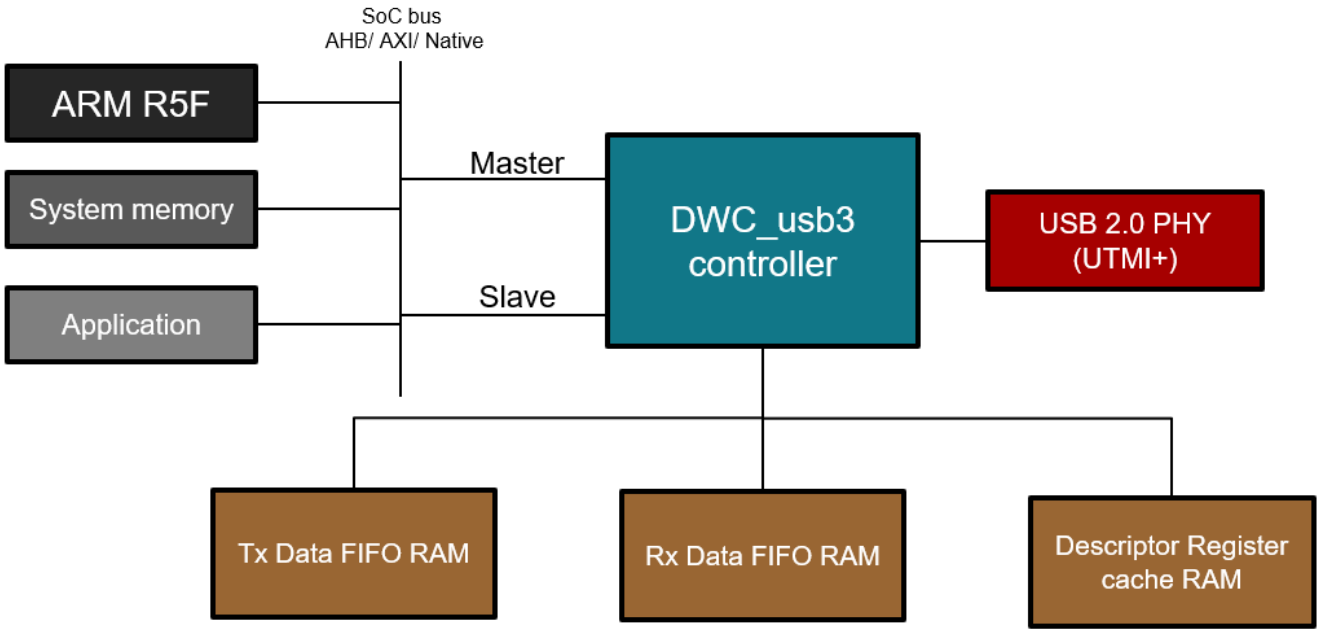

Figure 2-1. AM261x Hardware Block Diagram

The USB on AM261x supports a Dual-Role-Device which supports 15 IN, 15 OUT endpoints and a bi-directional EP0 endpoint. The USB Subsystem provides four programmable interrupts and a miscellaneous interrupt line for other events. The USB subsystem works very well with high-bandwidth applications and portable-electronic devices.

The USB low-level driver is only involved in setting up the transfer descriptors and data structures. The software creates the transfer request buffers (TRBs) and allocates event buffers and data buffers in system memory. The hardware handles the complete data packaging and routing to pipes. The internal DMA controller and BMU handle the core DMA operations and data buffering requirements. The TRBs only contain the buffer pointers to the actual data buffers, the length of the data and the TRB control block data. The below image shows how a software initiated and configured transfer occurs.

The USB on AM261x can be statically configured to be the Host or Device at power-on. The OTG switch cannot be made on the run-time and is not supported in the hardware. The embedded PHY does not support the OTG features of USB-OTG standard (ID pin detection and VBUS detection). The AM261x has a USB Micro-AB port for connection to the external world.

2.2 AM261x USB Software

The robust driver libraries provide seamless interaction between the microcontroller and USB peripherals, simplifying development. USB, sometimes perceived as a rather complex protocol, has been made easy to work with by TinyUSB. The TinyUSB stack is a very feature-rich open-source USB stack that is thread-safe, memory-safe with no dynamic run-time allocation of memory. With the TinyUSB stack, the developer can easily create applications using the USB classes provided in the TinyUSB stack. The TinyUSB stack already has some examples in-built inside the source code for both device and host modes. For example, the stack contains examples for HID host controller, which can be easily ported out of the stack and built in the MCU_PLUS_SDK. To support classes such as RNDIS with TCP/IP, TinyUSB stack can also be configured to work with LwIP stack.

The low-level USB driver can be divided in two parts, the system-on-chip (SoC) porting layer, which has information specific to the port on AM261x and the actual USB Device Core Driver which implements the complete USB functionality. The application developer can either directly use the driver without the involvement of the middle-ware stack (TinyUSB stack) and write applications over it, OR, the middle-ware stack can be used which eases the development process a lot.

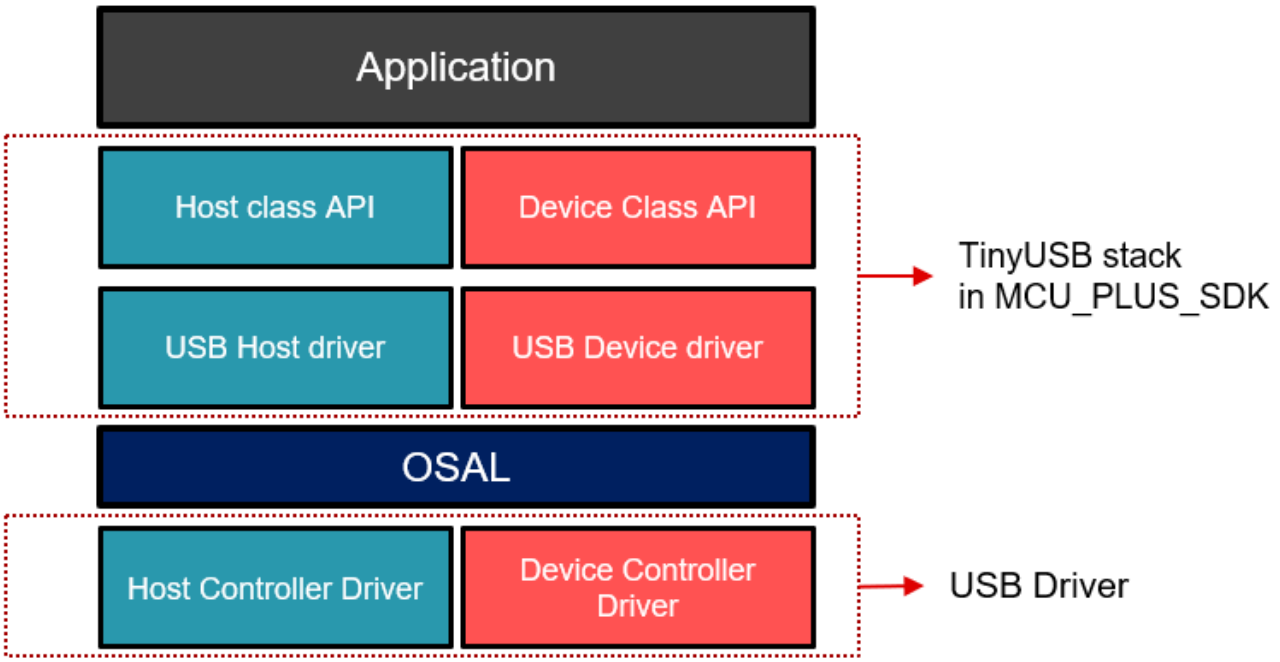

Figure 2-2. AM261x USB Software Block Diagram

The TinyUSB stack supports the following classes: Audio, Video, Bluetooth, CDC, DFU, HID, MIDI, MSC, Network, USBTMC, Vendor, and so forth, but not all of these are feasible to be run on AM261x (for example, *Bluetooth*® class cannot be enabled due to no Bluetooth hardware on the AM261x). To enable a new class that is not supported in the SDK out-of-box, the developer just needs to:

1. Create a tusb_config.h file, describing the size of endpoints, the class, size of Tx and Rx buffers, and so forth.
2. Have the Device driver support all the required endpoints and transfer callback functions.
3. Have an application built with the required USB class.
4. Rebuild the TinyUSB stack and link it to the application.
5. Rebuild the application.

With the steps above, the users can quickly enable different USB classes that are not a standard SDK offering.

2.2.1 Software Architecture

Figure 2-3 explains the architecture of the USB offering on AM261x devices. The MCU_PLUS_SDK v10.00 has the DFU and CDC applications available out-of-box that use the underlying TinyUSB stack classes. The USB applications and classes are supported over both FreeRTOS™ and NoRTOS and the TinyUSB porting layer and device driver layers are OS agnostic, meaning the user gets freedom to choose between NoRTOS and FreeRTOS, then choose the USB class from the stack and build a USB application seamlessly.

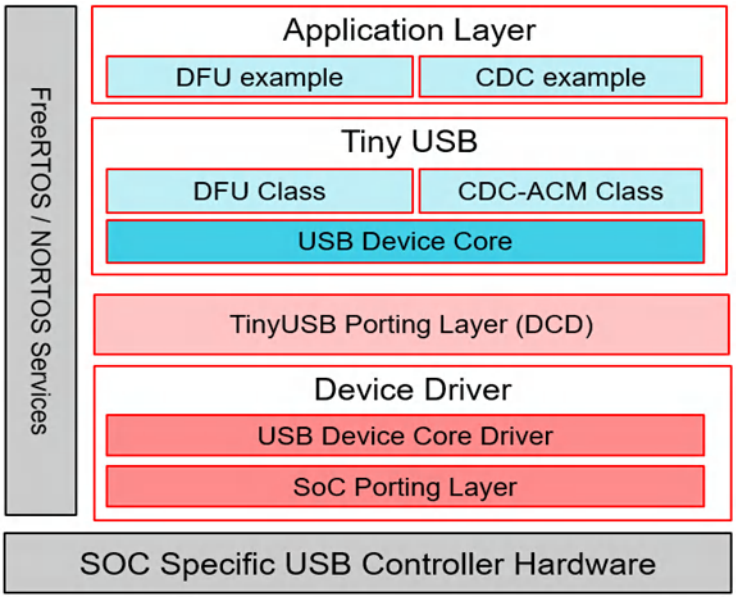

Figure 2-3. Software Architecture

The USB DCD layer has APIs defined to open, close, configure endpoints, initialize the device class driver and the USB driver, configure interrupts, enable or disable interrupts. The DCD layer also has input and output completion call back functions along with functions to submit a transfer request and general debug logging functionality.

2.2.2 AM261x USB Application Example

Figure 2-4 shows how the AM261x acts as a USB device running the DFU application. The DFU functionality can be used to update the firmware binary via USB in field. This means, there will be no down-time and enhanced firmware can be loaded to the MCU. The host machine having the USB host runs a dfu-utility program. This program is used to send files from the host PC to the AM261x device acting as USB device.

More details about the dfu-util commands can be found here: AM261x MCU_PLUS_SDK USB DFU example.

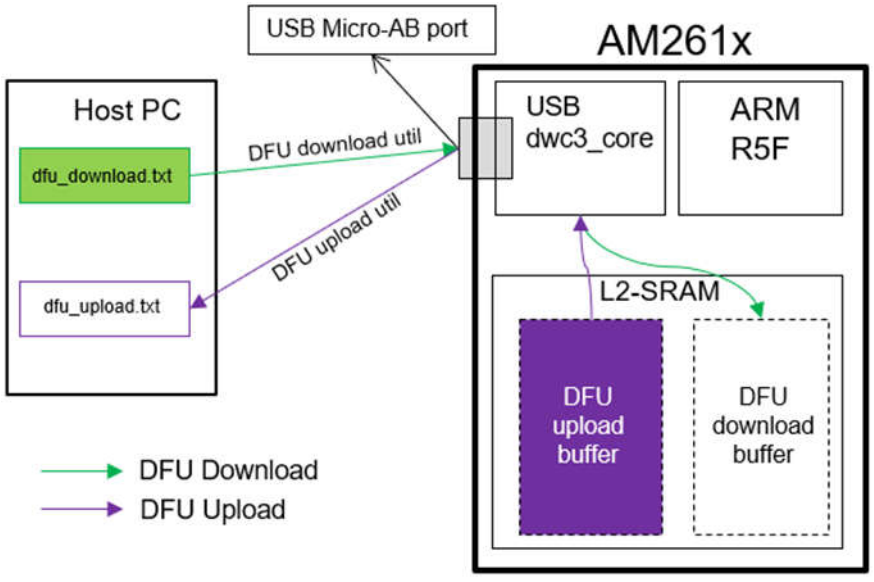

Figure 2-4. AM261x USB DFU Example

2.2.3 USB DFU Bootloader

The AM261x devices also support a DFU boot mode meaning the devices can receive applications (single and multi-core images) over USB DFU. The DFU class from the TinyUSB stack is used, which implements a dfu task. This task receives the application image and stores it into the device RAM. After this is completed, the application is then booted using standard bootloader APIs.

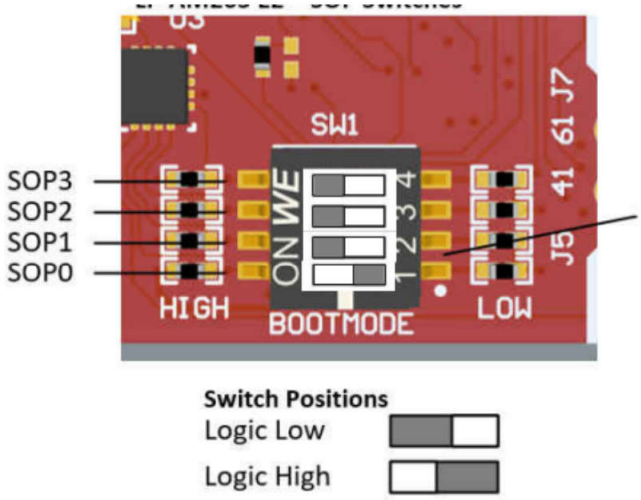

Figure 2-5. AM261x USB DFU Bootmode

USB bootloader is supported via USB SBL UniFlash application or the SBL DFU python scripts.

Read more about USB Bootloaders here:

- SBL DFU: SDK example link
- SBL DFU UniFlash: SDK example link

3 USB Use Cases and Applications

This section discusses some possible application areas where the AM261x USB functionality can be leveraged to build real-world applications.

- **Field Firmware Updates (DFU Mode)**

  The AM261x supports DFU mode, allowing for seamless firmware updates in the field. This capability is critical for minimizing downtime and ensuring devices remain secure and functional. Enhancements and bug fixes delivered directly to devices without the need for physical access.

  Applications: EV Charging Stations, Industrial controllers: Regular firmware updates to improve charging algorithms and security features.
- **Diagnostics and Communication (CDC Mode)**

  Utilizing the CDC, the AM261x can establish multiple virtual COM port connections, facilitating diagnostics and data communication.

  Applications: Motor Control Units, test equipment: real-time monitoring and diagnostic capabilities for motor performance and health, Simplified data logging and analysis processes through USB connectivity.

- **USB for Data Logging and Transfer**

  The AM261x enables efficient data logging and retrieval via USB, making it suitable for applications requiring local data storage.

  Applications: Portable data loggers and Smart Energy Meters: Easy data transfer for field analysis and reporting, Data collection for consumption analysis and performance optimization.
- **USB for Configuration and Calibration**

  USB functionality facilitates straightforward configuration and calibration of devices, making it accessible for field technicians.

  Applications: Medical Devices: Calibration of equipment to ensure compliance with health standards. Industrial Sensors: Configuration adjustments to optimize performance in varying operational conditions.
- **USB as a Human-Machine Interface (HMI)**

  The AM261x can connect to HMIs via USB, allowing for user interaction and control of embedded systems.

  Applications: Automated Machinery: Operator interfaces for controlling and monitoring system performance. Healthcare Equipment: User-friendly interfaces for patient data entry and system management.
- **USB for Peripheral Device Connection**

  The AM261x allows for the connection of various USB peripherals, enhancing the functionality of embedded systems.

  Applications: Factory Automation Systems (for example): Integration with barcode scanners and other peripherals for improved operational efficiency. Consumer Electronics: Connectivity with external storage and input devices.
- **Secure Device Authentication and Access**

  USB can be leveraged for secure authentication, ensuring that only authorized personnel can access sensitive system functions.

  Applications: Building Automation Systems: Controlling access to critical systems through secure USB tokens. Critical Infrastructure: Safeguarding essential operational systems against unauthorized access.
- **Portable Device Integration:**

  The AM261x enables integration with portable devices, facilitating data transfer and device management.

  Applications: Portable Medical Devices: Easy transfer of patient data for monitoring and analysis. Field Test Equipment: Rapid data offloading for immediate review and adjustments.
- **USB as a network adapter:**

  The AM261x USB can act as a virtual Ethernet adapter by enabling the RNDIS class of USB device. This protocol simulates Ethernet link over USB.

  Application: USB to Ethernet bridge for data transfer between different devices, remote data transfer in industrial setup.

The USB capabilities of the AM261x microcontroller empower embedded systems to meet the evolving demands of various industries. By addressing critical market needs such as real-time communication, firmware updates, and user accessibility, the AM261x stands out as a versatile solution for developers. The integration of hardware and software solutions, combined with its robust USB implementation, positions the AM261x as an essential component for modern embedded applications.

4 References

- AM261x product page: AM2612 product page
- AM261x USB MCU+ SDK documentation: AM261x USB SDK documentation
- AM261x USB DFU example: USB DFU example
- AM261x USB CDC example: USB CDC example
- USB types of classes:Defined class codes
- USB 2.0 Spec: USB 2.0 specification
- TinyUSB stack: TinyUSB Stack

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

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