WHITE PAPER

Bhargavi Nisarga
Systems engineer

Keith Quiring
Applications engineer

Mione Plant
Product marketing
Texas Instruments Incorporated

TEXAS INSTRUMENTS

Designing with ultra-low power MSP430™ USB microcontrollers

Bringing power efficiency and simplicity to your USB-enabled designs

Introduction

The ubiquity of USB makes it an extremely attractive interface for applications requiring connectivity to a PC or other host device for configuration, periodic downloading of data, or firmware updates. Often these devices are portable, such as medical or industrial tools which remotely collect data that needs to be uploaded at a later time. As these devices are portable, the final USB implementation must be both cost-effective and power efficient. TI's MSP430F55xx, MSP430F56xx, and MSP430F66xx microcontroller (MCU) families with embedded full-speed USB (12 Mbps) provides leading ultra-low power consumption combined with high-performance analog and other smart integrated peripherals. This paper provides an overview of some of the key USB challenges faced by a designer and how TI's MSP430 USB devices and supporting tools – including the new MSP-EXP430F5529LP LaunchPad Evaluation kit – help overcome these challenges to enable developers to focus on using USB in their application rather than learning USB as a technology.

The MSP430™ USB Microcontroller

To meet the needs of developers, TI provides full-speed USB to the F5xx and F6xx series of MSP430 microcontrollers (MCU). By integrating USB with a powerful performance MCU (up to 25 MHz), large Flash and RAM memory integrated intelligent peripherals (including an on-chip ADC, comparator, hardware multiplier, DMA controller, temperature sensor and other peripherals), and superior power management, the MSP430 is the ideal MCU for implementing USB in embedded applications (see Figure 1).
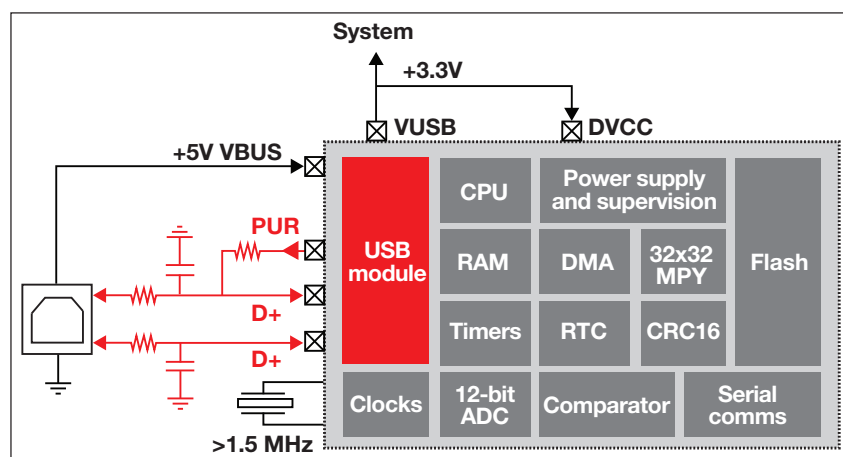

Figure 1. Integrating USB with the powerful performance and high integration of the MSP430 architecture creates the ideal microcontroller for implementing USB in embedded applications.

With the simple addition of a USB connector and a few discretes, MSP430 USB MCUs are a complete solution for applications that require USB connectivity and analog peripherals, while being ultra-low power. From a software standpoint, TI provides API stacks supporting the most common device classes.

The MSP430 USB MCUs are based on TI's proven MSP430 architecture, the F5xx and F6xx. Each supports 1.8 to 3.6 V operation, with clock speeds up to 25 MHz and integrated, programmable power supervision on the order of 200 nA. New clock sources further maximize tradeoffs between power, speed, precision and cost, and flash writes/erases can be performed across the full range of Vcc. In addition to offering a wide range of flash memory sizes (8 to 512 KB), and up to 66 KB of RAM, USB-enabled MSP430 devices also have an additional 2 KB RAM dedicated to USB that, when USB is disabled, is available for general purpose use.

Accelerated development and ease of use

TI recognizes that consumers – and engineers – have come to expect that USB "just works." In truth, while USB looks as simple as a UART or SPI port, the protocol is not trivial to implement. And unlike the UART or SPI interfaces, compliance is a primary USB design consideration, even in the simplest applications. For example, a host can suspend an attached device at any time. Devices must also be able to handle "surprise removals." For developers who have not been through this process before, such unexpected considerations can require extra development time and lead to unexpected delays.

Part of the value-add of the MSP430 MCU's USB support is that much of the underlying complexity of implementing USB is managed through an intuitive API stack. The API stacks are designed for fast absorption by the developer. Like USB itself, they contain much of the complexity "under the hood," masking the developer from unnecessary hassle to help speed development time. The stack source code is open for those developers wishing for complete control. For each stack, a complete Programmer's Guide is provided that serves as a reference for the API function calls and also describes the underlying concepts with clarity.

Further, TI provides the MSP430 USB Descriptor Tool. This tool serves as a "control panel" for the API stacks, allowing for quick configuration. It automatically creates the descriptors that every USB device must report to the host, based on user input. This saves the developer considerable time and provides peace of mind that the descriptors are done correctly.

Stacks are available for the most common device classes and are supplied without additional charge. While developers should not underestimate what is required to create a robust USB interface, TI has significantly shortened the USB learning curve. In this way, developers are freed of much of the burden of learning the intimate details of USB as a technology and instead are able to focus on using USB to increase the value and usability of their applications.

Multiple device classes

Part of the simplicity offered by the MSP430 USB API Stack is support for these device classes:

Communications Device Class (CDC): The CDC presents the USB port to the PC application as a standard COM port. COM ports are popular interfaces that are flexible, fast and simple to use. Because it uses bulk transfers, the CDC provides high bandwidth with a reasonable amount of simplicity. The chief disadvantage of using the CDC is that developers must distribute a simple file to end users that allows it to be associated with the CDC driver built into Windows. Fortunately, the "new device detected" installation in Windows is fairly straightforward and a minor step already well-accepted by end-users.

Human Interface Device (HID): While HID is commonly thought of as primarily for mice and keyboards, it is a flexible device class suitable for a wide variety of applications. TI's HID API efficiently generalizes HID capabilities, thereby eliminating the complexity associated with HID reports by allowing developers to access the interface in the exact same way they would a CDC device/COM port. While it has limited bandwidth (up to 64 KB/s), it requires no file to be distributed as CDC does, and it loads silently in Windows with no installation process required.

Mass Storage Class (MSC): MSC is the device class used to implement the highly successful USB "flash drives", as well as digital cameras and flash card readers. Since it is designed for moving large amounts of data, it provides higher bandwidth, similar to that of CDC. The tradeoff is more complexity – for example, developers will need to implement a file system – and the use of more code space. Like HID, MSC devices load silently in Windows without the need of an installation process. TI offers the MSC API layer at no cost.

Personal Health Device Class (PHDC): PHDC is designed for portable medical and wellness devices to be able to send measurements to USB hosts such as personal computers, cell phones, etc. The Continua Health Alliance has released guidelines for interoperability between various types of devices implementing the USB standard. Texas Instruments offers a USB platform for agent devices that has been certified by the Continua Health Alliance after having passed a rigorous testing procedure. Customers can use the software stacks of this platform to reduce development time for devices that will comply with the medical industry standards such as the Continua Health Alliance.

Given the wide variety of configuration possibilities for software handling file systems, the different media types, and Flash management (i.e., through wear-leveling and other techniques), developers have the flexibility of buying a commercial implementation or using one of the many open-source systems available for the MSP430 MCU.

Of the device classes, developers should begin by considering HID. If an application can work within the available 64 KB/s bandwidth, HID will often be the most cost-effective choice given that users can plug it in and use it without the need of an installation. Since the Windows installation process can sometimes lead to problems for the user, avoiding it can lead to a reduction of support calls and returns from customers.

The TI USB API stacks support three data transfer types: control for USB-level control/status data; interrupt for low bandwidth, fixed latency data; and bulk for high bandwidth, variable latency data. Usage of these data types is determined by the device class; so, developers need not concern themselves with most of the details associated with the different types.

With these data types, the MSP430 MCU's USB can support any application requiring control/configuration, firmware updating, or that needs to transfer bulk data (as opposed to streaming data). The MSP430 MCU does not support isochronous (high bandwidth, fixed latency) data, so it is not appropriate for streaming audio/video applications.

Just as the MSP430 microcontroller is flexible, so is its USB functionality. Any USB device contains a certain number of what are called endpoints. Support for multiple endpoints allows for composite USB devices which can have more flexible communication with hosts. For example, a device that uses the MSC for bulk data transfers and HID to manage control and status is comprised of 3 input and 3 output endpoints. The MSP430 architecture supports up to 8 input and 8 output endpoints, which provides sufficient capacity for most applications without overburdening cost.

Ultra-low power Efficiency

By its nature, the MSP430 architecture has been optimized for low power operation, both when USB is in use and when it is not. For example, MSP430 devices have five low power modes that enable designers to extend battery life in portable applications. MSP430 MCUs deliver high performance at the lowest power consumption with an active power consumption as low as 160 uA/MHz and 1.5 uA in standby mode coupled with fast wakeup from standby (less than 5 microseconds) and a supply voltage as low as 1.8 V operation. The onboard DMA controller can also save significant power when communicating with a battery-powered host.

One of the advantages of USB in embedded applications is the ability to power devices over the interface. Ideally, battery-powered devices can maximize operating life by powering the device over the bus when attached to a host. As USB hosts source 5 V over the bus, an LDO is required to drop the voltage to the 3.3 V typical of ICs. MSP430 devices simplify power design and conserve board space by integrating an efficient LDO as well as associated pull-up functionality. In addition to allowing MSP430 MCUs to operate directly from 5 V, integrating the LDO and pull-up resistors reduces component count and eliminates $.15 to $.20 relative to discrete implementations. By keeping the USB power and other MSP430 modules' power management separate, the USB module can always be powered as long as the USB device is connected to the host. This also enables the USB module to auto-powerup the MSP430 MCU via USB power even if the device battery is dead or non-existent.

MSP430 microcontrollers are designed to operate within the power limits of the LDO and can even source power off the USB bus to power the entire system. By driving the 3.3 V output externally (VUSB), the MSP430 MCU can supply the system with up to 12mA (see Figure 2), and eliminate the need for a system LDO as well. For high-current systems (requiring more than 12 mA of current) or for applications where it makes sense to power a device from battery even when connected via USB, the MSP430 MCU offers the flexibility of bypassing the integrated LDO and driving DVcc from an external supply or regulator (see Figure 3). TI offers a variety of external LDOs well-suited for low-cost (TLV70033), low-power (TPS78233), low-noise (LP5907), and low-noise, higher-current (TPS73433/735) applications. TI also offers its own dual push-button reset controller, the TPS3421. Although MSP430 MCUs come already equipped with a push-button interrupt, an external push-button reset controller may serve as an extra line of security. The TPS3421 remains isolated from the MSP430 MCU, ensuring the ability to reset so long as it is connected to a power supply. It also relieves the MSP430 MCU of the processing power required to monitor a push-button interrupt. Along with its low power properties and multiple timing configurations available, the TPS3421 is a great companion part for MSP430 MCUs.
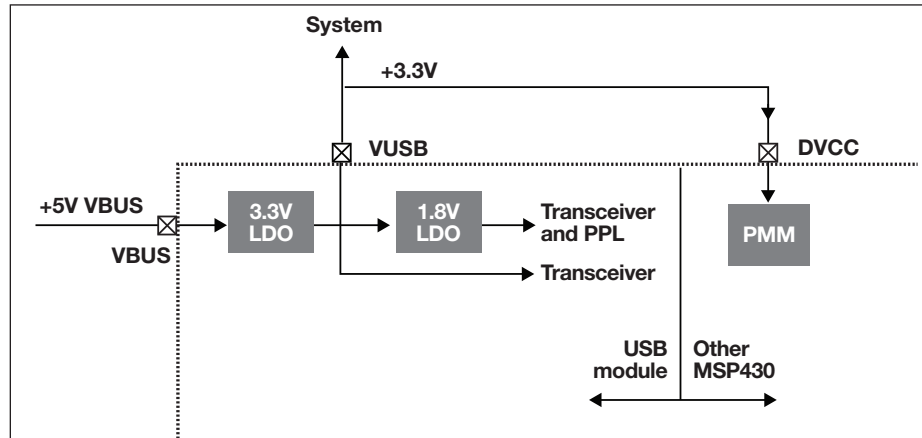
Figure 2. MSP430 USB microcontrollers can source power off the USB bus to power the entire system with 12 mA.
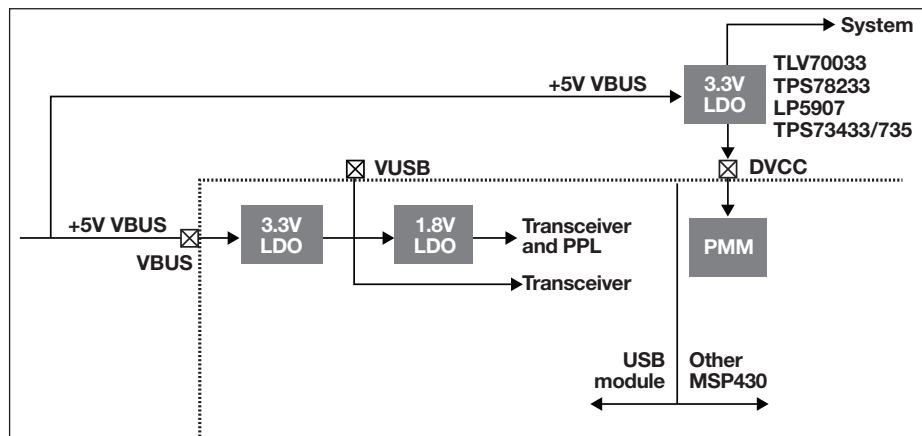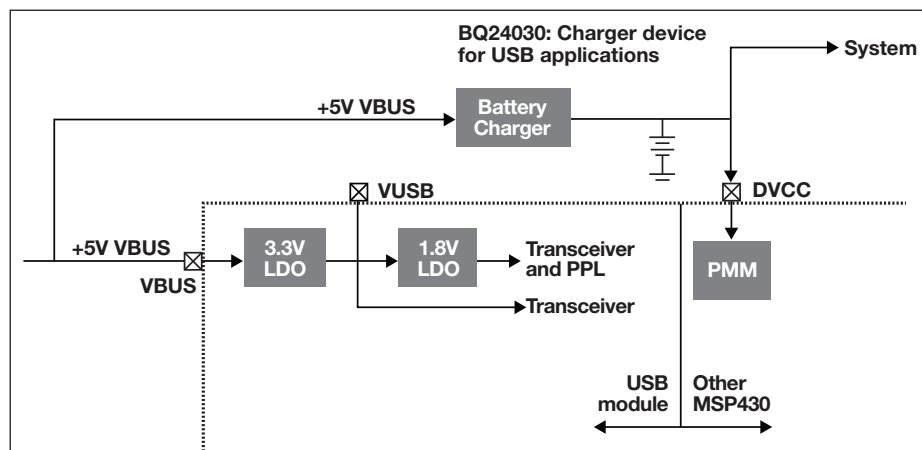

Figure 3. For high-current systems or for applications where it makes sense to power a device from battery even when connected via USB, the MSP430 MCU offers the flexibility of bypassing the integrated 3.3V LDO and driving DVcc from an external supply.

Figure 4. Power over USB can also be used as a primary source to charge batteries. TI's BQ2407x/3x family of chargers have been specifically designed for USB battery charging applications.

The 5V USB bus power can also be used as a primary source to charge batteries (see Figure 4). In this configuration, DVcc is always sourced from the battery whether the USB port is plugged into a host or not. When the port is plugged in, the charger receives power via USB to recharge the battery. TI's BQ2407x/3x family

of chargers have been specifically designed for USB battery charging applications. Power over USB can also be used as a primary source to charge batteries. TI's BQ2407x/3x family of chargers have been specifically designed for USB battery charging applications.

The low-speed USB used in mice and keyboards, for example, in general is not appropriate for any application processing enough data that it requires a modern, general-purpose MCU. Specifically, transmitting at a slower data rate wastes bus bandwidth and consumes more power by requiring the MCU to stay active for significantly longer transmission periods. Likewise, high-speed USB supplies too much bandwidth unless an application needs to support large audio or video transfers. Full-speed USB is a much better fit for most embedded applications.

While being an OTG appears to be an appealing option, it is not appropriate for many applications. Embedded hosts must be able to source 8 mA to attached devices, and this requirement has led many companies to reconsider their intention to support OTG host capability, especially in applications where long operation life is required from a single-cell battery. For applications that do require OTG support, TI offers solutions within its MCU portfolio.

In general, USB devices are less expensive, simpler, and faster to develop than embedded hosts. The MSP430 microcontroller family with USB was optimized to meet the needs of USB devices without burdening these applications with the added complexity, additional memory, integrated peripherals and larger power supplies required to implement USB hosts.

Taking care of details

In order to deliver on its promise to simplify USB, TI provides a wide range of tools and software to assist developers in quickly coming up to speed on using and implementing robust USB solutions. In addition to its API stack, TI also offers:

USB Descriptor Tool: The USB Descriptor Tool plays a key role in simplifying USB design and implementation. This GUI-based tool automatically configures the USB stack to reflect the specific requirements of a particular application, handling the management of descriptor fields including VID, PID, strings, how much power to draw from the host, and so on. Rather than require developers to delve into the details behind the various USB descriptor fields by breaking into the USB stack and writing code to support their application, the Descriptor Tool gathers the required information from developers and automatically generates the appropriate software modifications to the API stacks, requiring no further work from developers. In addition, the USB Descriptor Tool is intended to make the high-level stack configuration of composite devices much simpler.

Bootstrap Loader (BSL): TI's BSL is another important tool offered to developers who require field firmware update capability. One of the common uses of USB is to push updates to deployed devices. For example, a doctor could plug a medical instrument into a PC and have it quickly and automatically update itself with new features or bug fixes. The BSL Tool simplifies the updating process for developers by converting a firmware image file into a self-contained PC executable that can be delivered to end customers. All MSP430 devices are BSL-equipped and when implemented over USB, updates can be made securely even when the device is not powered (i.e., the microcontroller is powered via USB), enabling fast and efficient production-line programming without having to install batteries. For developers, the most complex part of the bootstrapping process will be to customize the BSL executable with their own logo.

Real Time Operating Systems (RTOS): The USB API may be used with real time operating systems such as SYSBIOS, ucos-II and III and Free RTOS. SYSBIOS is a real time operating system from Texas Instruments and supports a wide range of MSP430 devices. It provides a wide range of services such as:
• Pre-emptive multitasking
• Hardware abstraction
• Real-time analysis
• Memory management
• Configuration tools
SYSBIOS is very scalable and can be configured to fit in memory-constrained devices such as MSP430 MCU.

SYSBIOS is integrated into Code Composer Studio and one can very easily create SYSBIOS projects within CCS. All SYSBIOS tools are integrated into CCS as eclipse plugins.

VID Sharing Program: TI also offers developers to participate in its VID Sharing Program. Every USB device requires a Vendor_ID (VID) and Product_ID (PID). For companies making only a few devices, TI can provide a VID and a unique PID to bypass the investment of time and money required to register a VID with the USB consortium.

By taking care of the "odds and ends" of implementing USB through the USB Descriptor Tool, Bootstrap Loader, API stack, and other support software from its extensive third party network, TI has simplified the process of working with USB. By providing most of the underlying foundation required for USB, TI's MSP430 MCUs enables developers to focus on the value-added components of their application without having to concern themselves with the myriad implementation issues involved with a complex interface like USB.

High level of integraion

In addition to integrating USB, these new controllers also integrate a number of other features and peripherals beyond the standard MSP430 architecture that further simplify development, including:

Programmable PLL: This flexible, programmable PLL can adapt to a wide range of crystal frequencies, enabling developers to choose a frequency based on application-relevant criteria such as cost, other components within the system, or whether that frequency is needed elsewhere in the system for another purpose.

Very Low Power Oscillator (VLO): The VLO enables developers to keep the USB module running when the host is suspended. This is important because the USB module must be able to recognize when the host wants to wake it. Current drawn by the VLO in this manner is in the sub-uA range.

Comparator_B: MSP430 USB microcontrollers also offer a comparator for the generation of hysteresis without the need for external components. Many applications require the ability to monitor an input against two thresholds, such as for battery charging and capacitive touch interfaces. Typical comparators can only monitor one threshold and must be configured to watch for the rising or falling threshold and switch to the other threshold when it is crossed. Known as Comparator_B, the new comparator is a versatile reference generator using R-ladders capable of generating 32 different voltage reference levels. This approach avoids the need for external components as well as the constant power drain when external resistors are used. Comparator_B operates in three modes, ultra-low power (0.1 uA typical to 0.5 uA max), normal (10 uA typical to 30 uA max), and high speed (40 uA typical to 65 uA max) for optimizing power consumption to the application.

Port Mapping: A unique feature of USB-based MSP430 microcontrollers is the port mapping controller. With port mapping, developers can dynamically reconfigure digital outputs such as Timer PWMs or SPI/I2C interfaces across a specific range of pins. Such mapping enables flexibility of signal routing during board design, allowing designers to move signals to the other side of the IC as required. Each digital output can be mapped to multiple output pins, which can be useful in cases, for example, where the same Timer PWM is required on multiple pins. Port mapping also eases the challenges associated with migrating from one family of devices to another in terms of pin-to-pin compatibility.

Getting started MSP430 USB microcontrollers is very easy, especially considering the introduction of the low

Getting started with MSP430

cost MSP-EXP430F5529LP LaunchPad Evaluation kit. The MSP430F5529 LaunchPad is an easy-to-use flash programmer and debugging tool for the MSP430F552x microcontrollers, including on-board emulation, available today for $12.99. Also available are the MSP430F5529 USB Experimenter's Board (MSP-EXP430F5529) development platform, and MSP430F55xx USB 80-Pin Target board (MSP-TS430PN80USB). With the proven tool chain of MSP430 MCUs and the comprehensive MSP430 USB Developers Package (MSP430USBDEVPACK), developers can introduce USB to their applications with little effort. In addition, TI's many third parties supply a wide range of software and hardware to speed development and accelerate time to market.

Developers can choose from three families of USB-enabled MSP430 microcontrollers, each with a flexible roadmap and options that scale to meet a variety of embedded application requirements:

Mid-range applications: The F552x / F551x supplies 64 to 128 KB Flash and 4 to 8 KB (+2 KB) RAM, as well as Comparator_B functionality. F552x \ F551x devices are sampling now.

High-end applications: The F563x / F565x / F663x / F665x are perhaps the most feature-rich, integrated MSP430 devices ever produced. With 128 to 512 KB Flash and up to 66 KB (+2 KB) RAM, this device has six DMA channels, an RTC backup mode (enabling the RTC to operate at less than 1 uA even when Vcc is lost), and many other integrated peripherals.

Low-end applications: The F550x provides cost-effective USB with 16 to 32 KB Flash and 4 KB + 2KB RAM as well as a 10-bit ADC and Comparator_B functionality.

TI's new MSP430 USB microcontrollers reduce system cost, offer superior power efficiency to improve battery life, facilitate fast implementation, and are easy to use. In addition to allowing developers to focus on their application rather than USB as an enabling technology, these new controllers also lower system BOM by integrating several advanced peripherals and modules that increase performance and improve power consumption while reducing component count. With its comprehensive supporting software and hardware, TI reduces the USB learning curve from weeks to hours, making it cost-effective – and simple – to introduce USB into a wide range of embedded applications.

Get started today at www.ti.com/msp430usb

IMPORTANT NOTICE

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		