The ultra-low-power USB revolution:

Bringing power efficiency and simplicity to USB for portable embedded applications

Texas Instruments

Bhargavi Nisarga
Applications Engineer
Texas Instruments

Keith Quiring
Applications Engineer
Texas Instruments

Les Taylor
MSP430 Product Marketing Manager
Texas Instruments

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.

TI's MSP430F55xx microcontroller (MCU) family 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 challenges of developing with USB and highlights how the MSP430F55xx and supporting tools enable developers to focus on using USB in their application rather than learning USB as a technology.

Introducing the MSP430™ USB microcontroller

To meet the needs of developers, TI has introduced full-speed USB to the MSP430F5xx series of MSP430 microcontrollers (MCU). By integrating USB with powerful performance (up to 25 MHz), large memory (both Flash and RAM), 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 designed for MCU for implementing USB in embedded applications (see **Figure 1**).

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 three of the most common device classes.

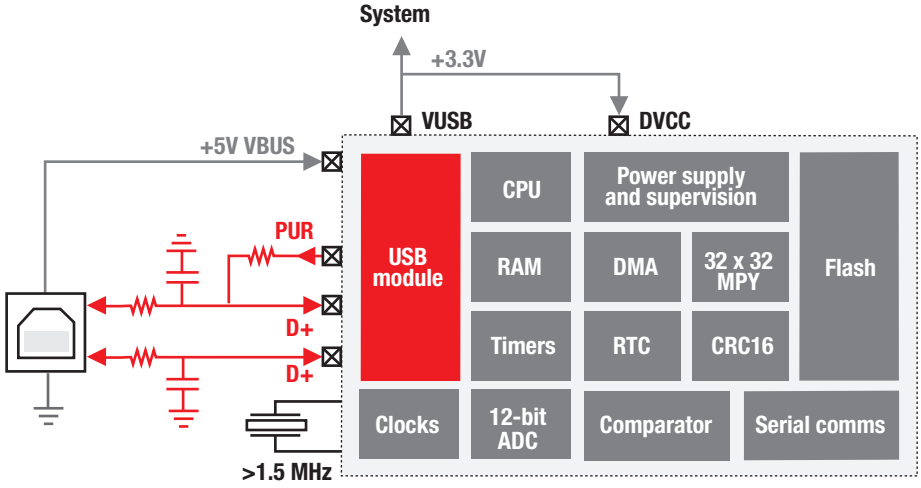

Figure 1. *Integrating USB with the powerful performance and high integration of the MSP430 architecture creates the ideal microcontroller for implementing USB in embedded applications.*

The new MSP430 USB MCUs are based on TI's MSP430 architecture, the MSP430F5xx. 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 (16 to 256 KB), USB-enabled MSP430 devices also have 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. Part of the simplicity offered by the MSP430 USB API stack is support for three device classes:

Multiple 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. 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 MCUs.

Of the three 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'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 does not support isochronous (high bandwidth, fixed latency) data, so it is not appropriate for streaming audio/video applications.

Just as the MSP430 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 µA/MHz and 1.5 µA 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 via USB power even if the device battery is dead or non-existent.

MSP430 MCUs 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 12,mA (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 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 (TLV733P), low-power (TPS7A05), low-noise (LP5907), and low-noise, higher-current (TPS7A90/1/2) applications.

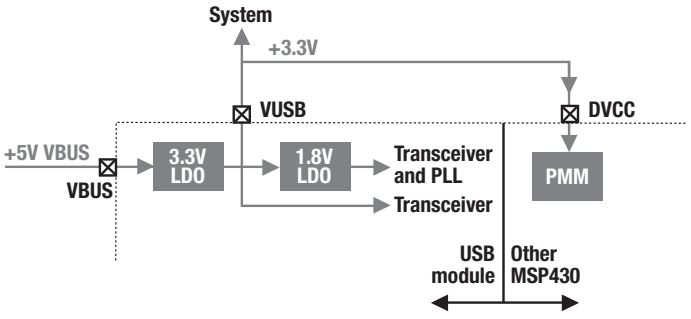

Figure 2. *MSP430 USB microcontrollers can source power off the USB bus to power the entire system with 12 mA.*

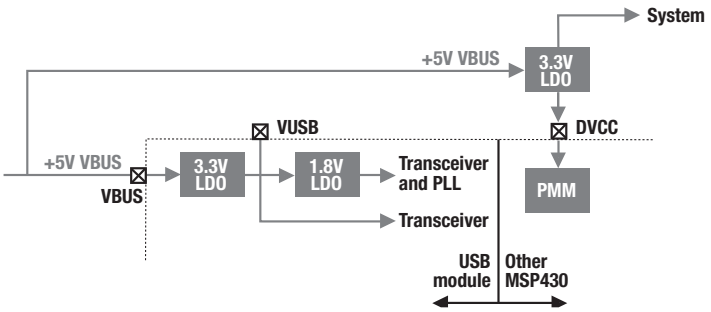

Figure 3. *For high-current systems or for applications where it makes sense to power a device from battery even when connected via USB, the MSP430 offers the flexibility of bypassing the integrated 3.3V LDO and driving DVcc from an external supply.*

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 BQ2560x and BQ2407x charger families have been specifically designed for USB battery charging applications. For more USB battery chargers, please visit ti.com/chargers

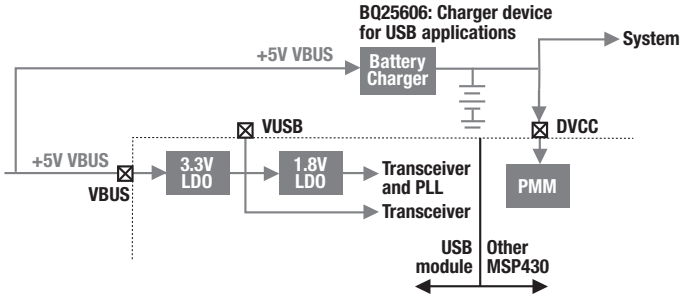

Figure 4. *Power over USB can also be used as a primary source to charge batteries. TI's BQ2560x and BQ2407x charger families have been specifically designed for USB battery charging applications.*

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 Stellaris MCU portfolio.

In general, USB devices are less expensive, simpler, and faster to develop than embedded hosts. The MSP430 MCU 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.

Bootloader (BSL):

TI's Bootloader - formerly Bootstrap Loader (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 Bootloader 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 MCU devices are BSL-equipped and when implemented over USB, updates can be made securely even when the device is not powered (i.e., the MSP430 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.

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, Bootloader, 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 enable 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 integration

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-µA range.

Comparator_B:

MSP430 USB microcontrollers also offer a new 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 µA typical to 0.5 µA max), normal (10 µA typical to 30 µA max), and high speed (40 µA typical to 65 µA 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 as sociated with migrating from one family of devices to another in terms of pin-to-pin compatibility.

Getting started with USB

Getting started with MSP430 USB microcontrollers is very easy. The MSP430F5529 LaunchPad™ development kit includes USB support. Samples of the MSP430F551x and MSP430F552x MCUs in an 80-pin LQFP package are available separately. With the proven tool chain of MSP430 devices and the comprehensive USB support package, developers already familiar with MSP430 MCUs 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 MSP430F552x/F551x MCUs supply 64 to 128 KB Flash and 4 to 8 KB (+2 KB) RAM, as well as Comparator_B functionality.

High-end applications:

The MSP430F563x/F663x MCUs are feature-rich. With 128 to 256 KB Flash and 16 KB (+2 KB) RAM, this device has six DMA channels, an RTC backup mode (enabling the RTC to operate at less than 1 µA 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. The F550x is expected to sample in early 1Q2010.

TI's 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.

Device class tradeoffs

	CDC	HID	MSC
Simplicity on host	Yes	Yes	Yes
Simplicity on MSP430	Yes	Yes	No
Avoids user install process	No	Yes	Yes
High bandwidth	Yes	No, 64 KB/s	Yes
Code size	4 KB	4 KB	10 to 20 KB

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), 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, 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 (www.ti.com/legal/termsofsale.html) 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.

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