



Integrated USB Connectivity with MSP430™ MCUs



Agenda

- **MSP430's USB Module**
 - USB features & architecture details
- **MSP430 Devices with USB**
 - How many MSP430 devices will have USB, and when?
- **MSP430's USB support**
 - How TI will make your MSP430/USB development easy & fast!



MSP430's USB Module: Features & Architecture



USB in the Market

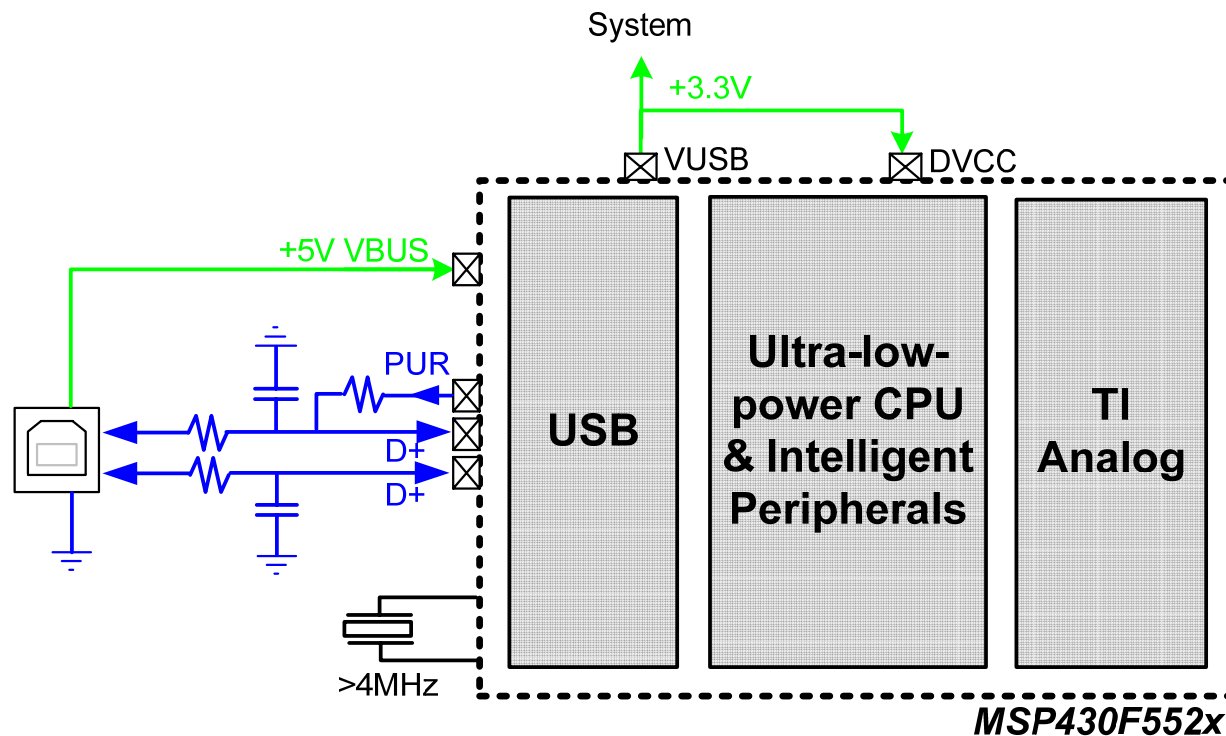
- USB has become prolific in the market
 - Everything from cell phones to devices that keep your coffee warm
- Standardized user experience has made it successful
 - OEMs see it as an easy interface for the end user
- USB is a mature technology
 - Allows us to take advantage of the best features in the market – and add our own advancements





MSP430 USB System

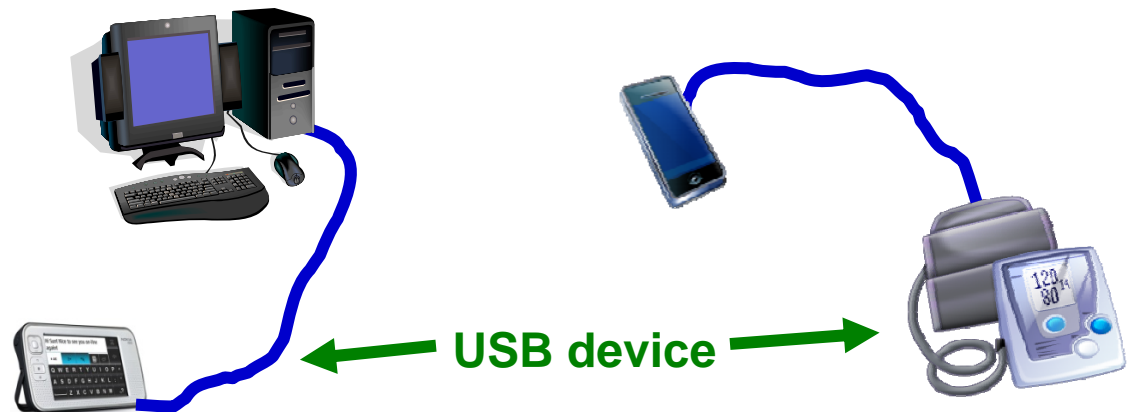
- Ultra-low-power, TI analog, & now USB – on one chip
 - TI is the industry leader in analog
 - MSP430 is the industry leader in ultra-low-power MCUs
 - Add USB & an LDO: a single-chip solution





Bus Speed/Role

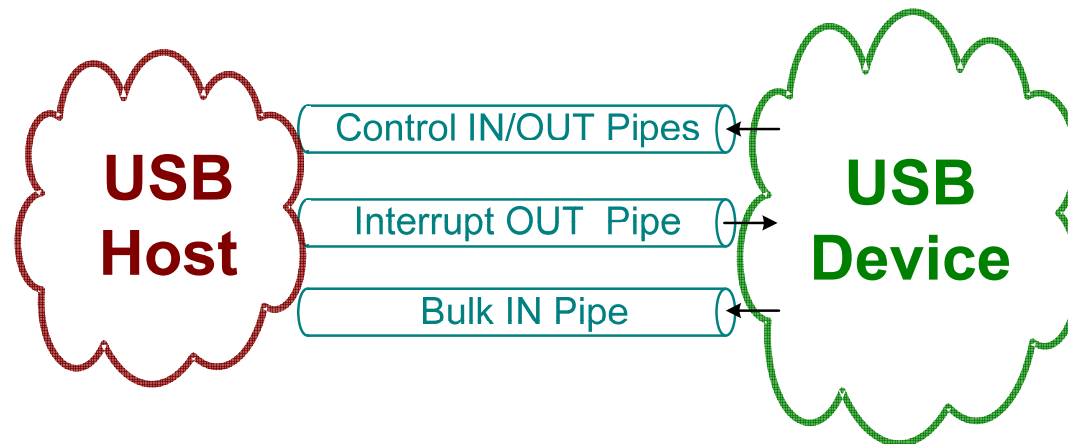
- MSP430's USB is a full-speed *device*, for applications such as:
 - Uploading data from a mobile device to a host (PC, cell phone, PDA, etc.)
 - Downloading a firmware update from a host
 - Controlling/configuring the device from a host
- Integrated transceiver





Transfer Types & Endpoints

- Supports control/interrupt/bulk transfers
- Eight input and eight output endpoints
 - Complete flexibility!



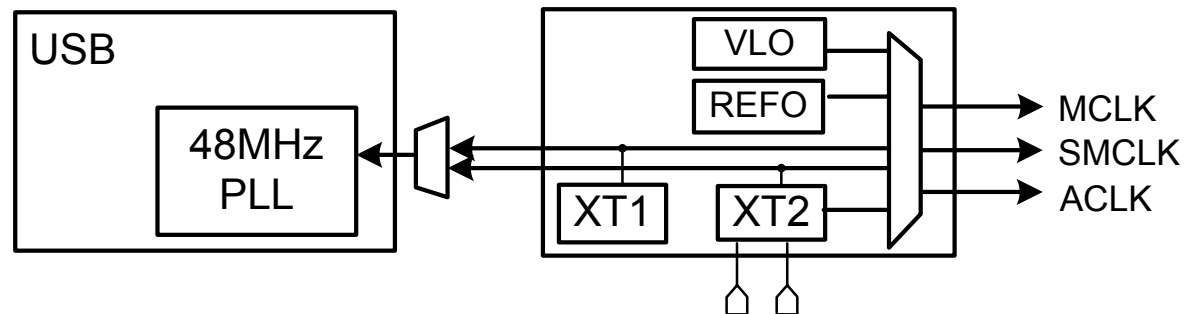


-
- The diagram illustrates the power supply and control logic for the USB module. A +5V VBUS input enters from the left, passing through a switch (represented by a box with an 'X') and a dashed vertical line. It then splits into two paths: one goes to a 3.3V LDO, and the other goes to a switch labeled VUSB. The output of the 3.3V LDO goes to a 1.8V LDO. The output of the 1.8V LDO is connected to the Transceiver & PLL and the Transceiver. The output of the VUSB switch is connected to the System and the Transceiver. A DVCC switch is connected to the System and the PMM. A dashed horizontal line separates the USB Module (containing the LDOs and Transceiver) from the Other MSP430 (containing the PMM). A dotted vertical line and a double-headed arrow at the bottom indicate the boundary between the USB Module and the Other MSP430.



USB Clocking

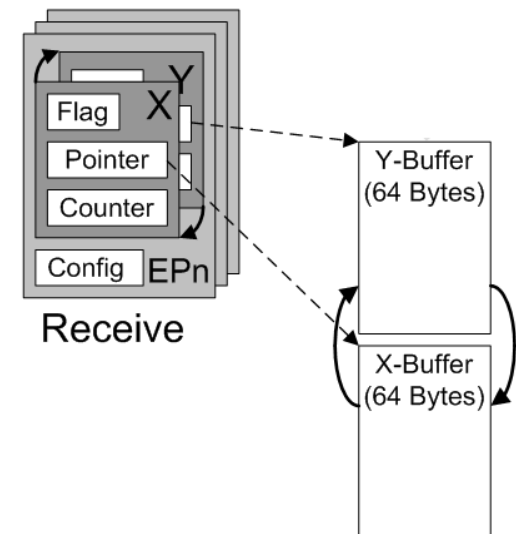
- USB clocked from HF crystal
 - Or, input a precise clock from another device
- Integrated PLL revs it to 48MHz
 - Usually powered down during USB suspend, & re-activated when resumed
- Can also use clock for other system functions
- PLL is very programmable – can use many reference freqs.
 - Choose one used elsewhere in the system
 - Or choose the cheapest crystal you can find
 - >4MHz (crystal) or >1.5MHz (bypass mode)





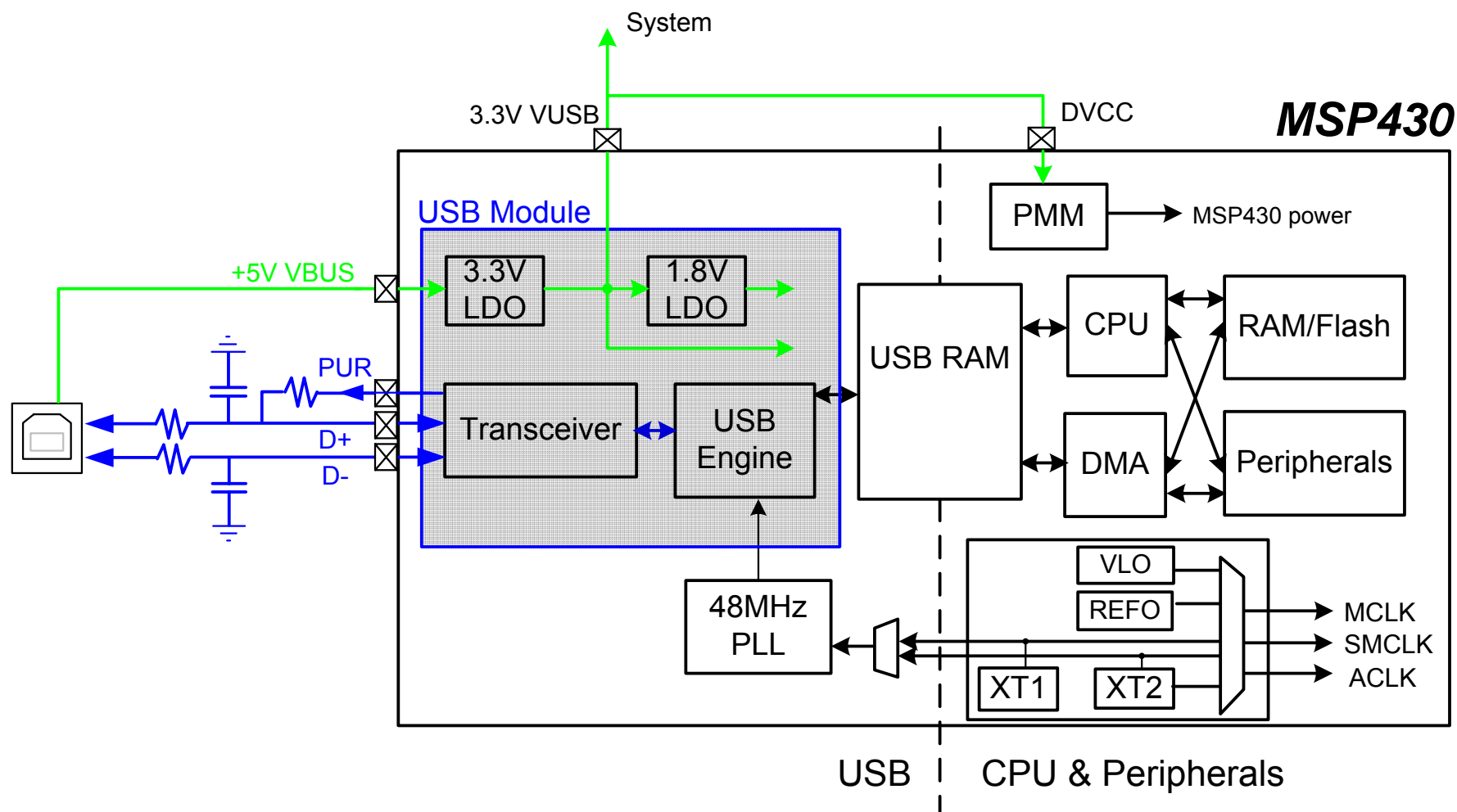
USB Memory & Data Transfer

- 1904 bytes of configurable buffer space for IN/OUT endpoints 1-7
- Each endpoint uses an automatic double-buffered scheme
 - Automatically toggles between X/Y buffers
 - Aids throughput
- Each endpoint described by six registers
 - Endpoint type
 - X/Y buffer addresses in USB RAM
 - Buffer size
 - Number of bytes in X/Y buffers
- CPU/DMA loads data in/out of buffers; automatically TX/RX over USB





Block Diagram & Data Flow





MSP430 Devices with USB

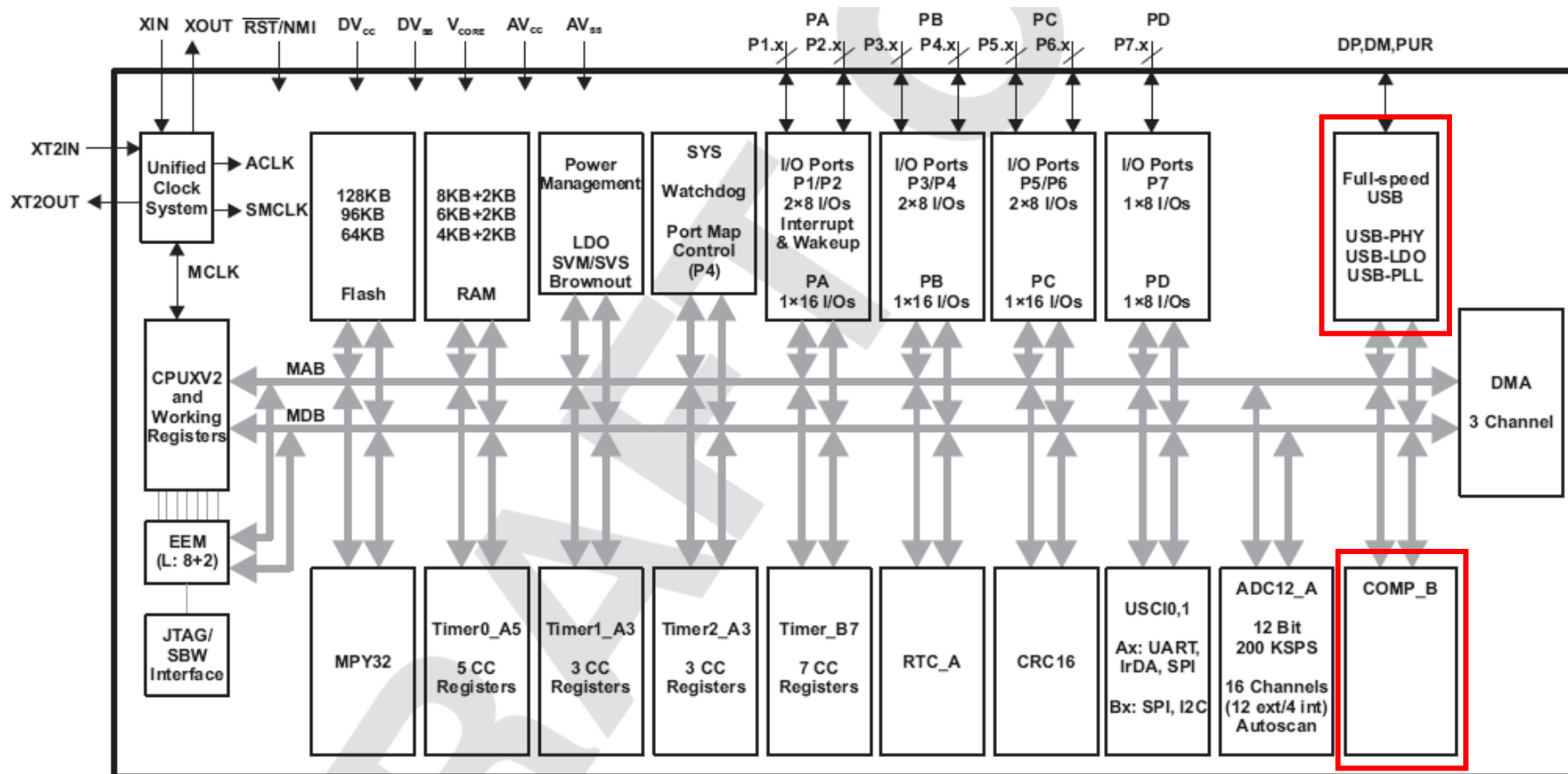


MSP430 Devices with USB

- MSP430s with USB will be in the F5xx/6xx family
- A sampling of features new to this family...
 - 1.8-3.6V operation, with speeds up to 25MHz
 - Wide range of flash memory sizes (8-256K)
 - Integrated, programmable power supervision (~200nA)
 - Flash write/erase across the full Vcc range
 - New clock sources, further maximizing tradeoffs between power, speed, precision, & cost
- Three new USB families in the next year...
 - Mid-range (F552x): RTM January 2010, [sampling NOW](#)
 - High-end (F563x/663x): RTM June 2010, sampling December 2009
 - Low-cost (F550x): RTM May 2010, sampling January 2010



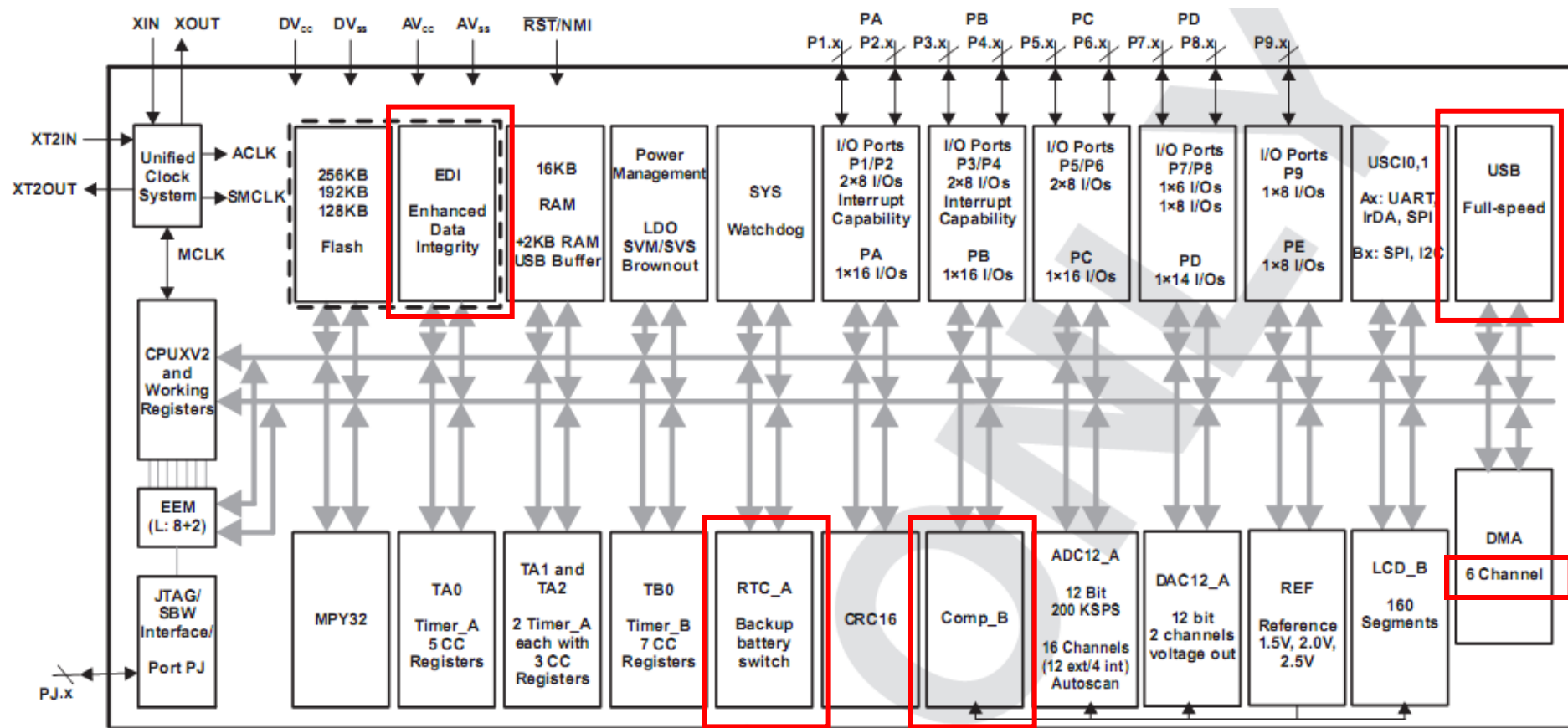
F552x Block Diagram



■ New



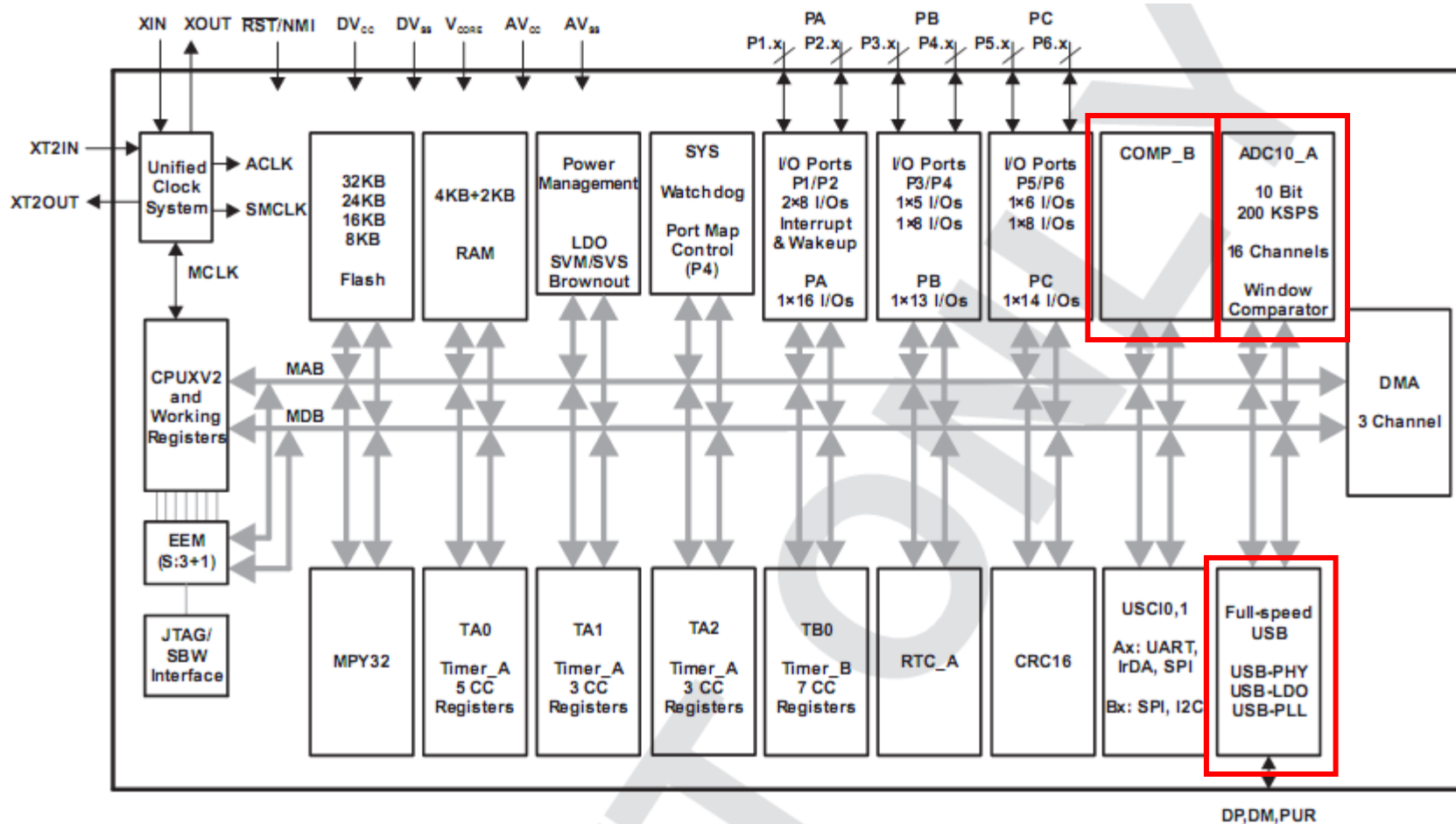
F563x/663x Block Diagram



■ New



F550x Block Diagram



■ New



Derivative Device Summary

Product	Program (KB)	SRAM (KB)	16-Bit Timers	Common Peripherals	ADC	Additional Features	Packages
MSP430F663x	128 to 256	8 to 16	4	WDT, RTC, DMA(3-6), MPY32, Comp_B, UART, SPI, I2C, PMM (BOR, SVS, SVM, LDO)	12-bit	USB, EDI, DAC12 , LCD , Backup battery switch	100QFP, 113BGA
MSP430F563x	128 to 256					USB, EDI, DAC12 , Backup battery switch	
MSP430F552x	32 to 128	6 to 8			-	USB, 25 MIPS	80 PN, 64 RGC, 80ZQE
MSP430F551x	32 to 128	4 to 8					
MSP430F550x	8 to 32	4			10-Bit		48 RGZ, 48 PT, 64 RGC



MSP430 USB: Support Collateral and Approach



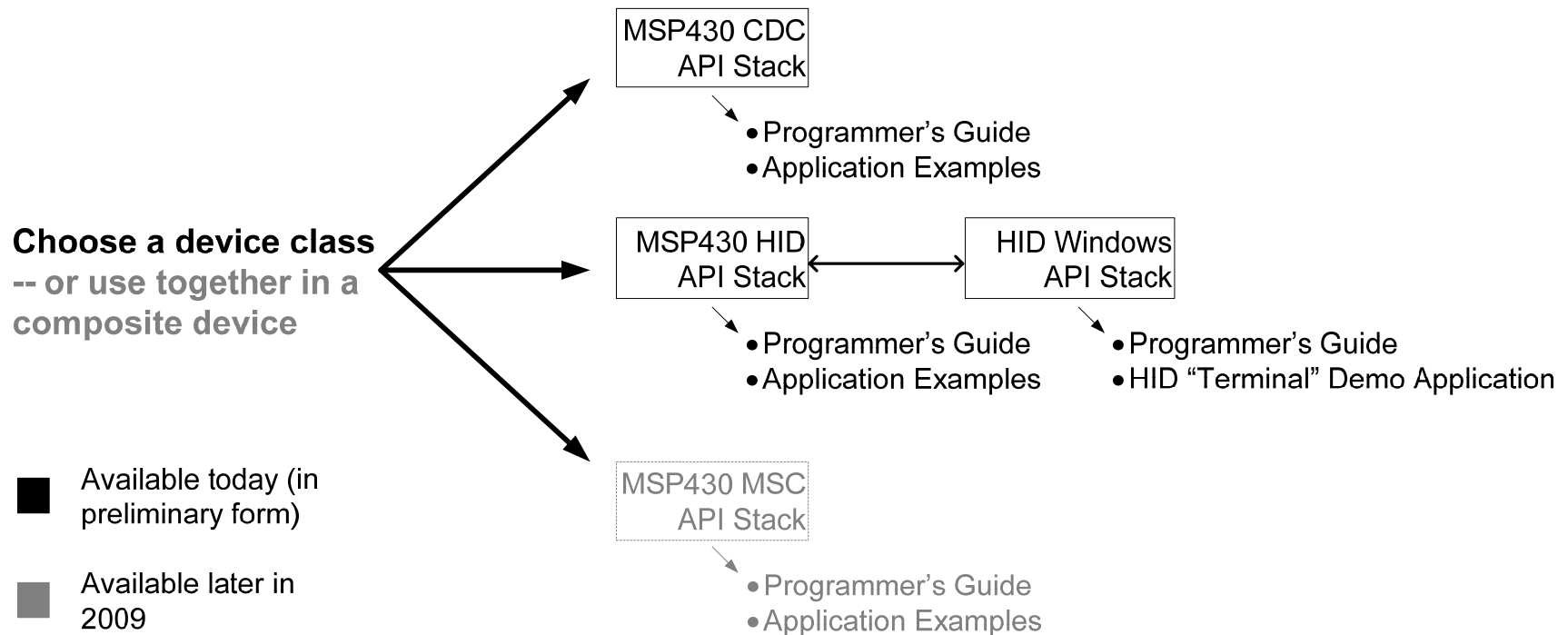
Support Approach & API Stacks

- USB presents challenges compared with other interfaces
 - A lot of protocol
 - Must think about compatibility with a wide range of USB hosts
 - Connection can be terminated (“surprise removal”) or suspended at any time
- But, the right silicon support CAN make it fast & straightforward
- The *MSP430 USB Tool Suite* is designed for ease-of-use
 - Let's you focus on *using* USB, rather than *learning* USB as a technology
 - Stretch goal: to make using USB similar to using a UART



MSP430 USB API Stacks

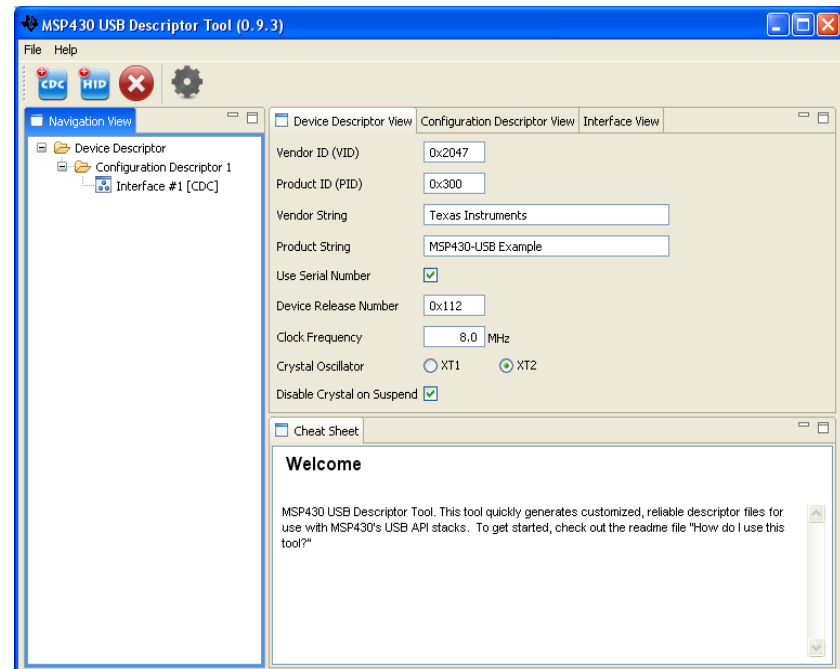
- Primary support tool: API stacks
 - Libraries based on standard USB device classes
 - Source made available; editing not necessary for most apps
 - Each stack supported by a complete Programmer's Guide
 - Stacks will exist for the most common device classes (CDC/HID/MSC)





MSP430 USB Descriptor Tool

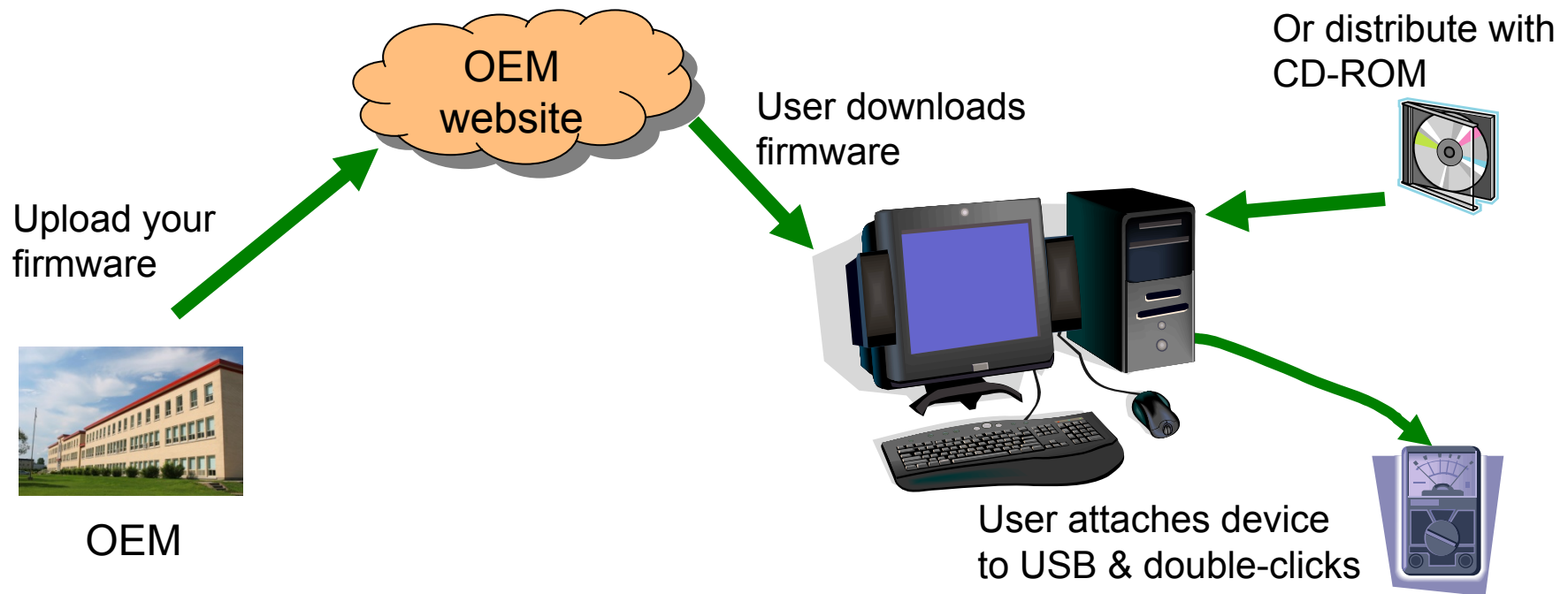
- Descriptor data structures are tedious & error-prone
 - One error & enumeration fails; or worse, appears to succeed but doesn't
 - Debugging them takes time
 - Which settings to use? Compatibility with hosts?
- TI provides a Windows utility to make descriptors easier
 - Automatically generates C code from GUI inputs
 - Descriptors & stack configuration
 - A help pane advises the impact of each decision
 - Composite support will be added 4Q09
 - Available for download today
 - **Speeds USB development**





USB Field Firmware Updates

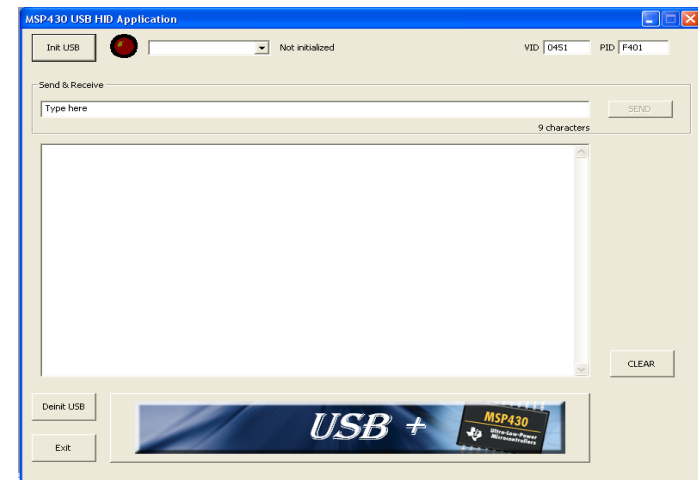
- USB is integrated into our on-chip bootstrap loader (BSL)
 - Enables end user to update firmware in the field
 - Customizable Windows project; use it to generate end user GUI
 - Just insert firmware image, customize the look, & build
- Also can be used in factory programming





MSP430 HID Windows API

- HID device class has many advantages for general use
 - Loads silently on host
 - No device install to go wrong, no need for admin rights, no need to distribute files to the end user
 - Small flash/RAM footprint
- Unfamiliar to many programmers; requires use of “reports”
- But MSP430’s HID *datapipe* system compensates for these
 - Special function calls in the MSP430 API stack
 - A Windows API is provided that makes HID easier to implement
 - HID Demo App demonstrates the API, can be used for eval





Collateral (1)

- Device documentation
 - Preliminary F552x datasheet
 - F5xx Family User's Guide
 - USB Module User's Guide
- MSP-TS430PN80USB FET Target Board w/ samples
- Special IDE versions for F552x
 - IAR Kickstart
 - CCE
 - Install, then apply F552x patch
- MSP430F552x Code Examples
 - Brief examples for non-USB peripheral functions





Collateral (2)

- F5529 USB Sample Kit – Getting Started & Support Files
 - *Starting a USB Design with MSP430* → 20-page overview of MSP430 USB
 - *Getting Started* – detailed walk-through for using FET target board demo
- MSP430 USB Descriptor Tool
 - Help pane is integrated; readme file in the zip
 - If you have trouble running it – install latest Java Virtual Machine
- MSP430 Field Firmware Update Demo
 - *.exe demo for firmware update
 - Visual Studio project coming soon
 - *USB Field Firmware Updates on MSP430 MCUs* → 20-page overview
- MSP430 USB CDC+HID API Stacks
 - API stack for CDC & HID
 - Programmer's Guide for each



Thank you!