MSP430 Design Workshop

STUDENT GUIDE

Texas Instruments

***MSP430 Design* Workshop**
Revision 4.01
February 2015

Important Notice

Revision History

October 2013 – Revision 3.0 (based on MSP-EXP430F5529 USB Launchpad)

November 2013 – Revision 3.01

January 2014 – Revision 3.02

February 2014 – Revision 3.10 (MSP-EXP430F5529 & MSP-EXP430FR5969 Launchpad's)

July 2014 – Revision 3.21

Jan 2015 – Revision 4.00 (based on F5529, FR5969 and FR4133 Launchpad's)

Feb 2015 - Revision 4.01

Mailing Address

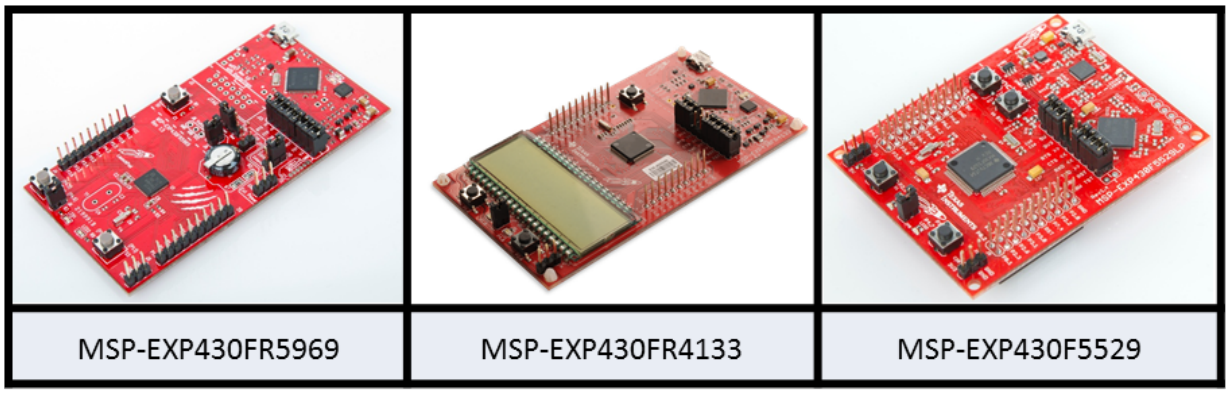
Texas Instruments
Training Technical Organization
6500 Chase Oaks Blvd – Bldg 2
M/S 8437
Plano, Texas 75023

Introduction to MSP430

Introduction

Welcome to the MSP430 Workshop. This workshop covers the fundamental skills needed when designing a system based on the Texas Instruments (TI) MSP430™ microcontroller (MCU). This workshop utilizes TI's integrated development environment (IDE) which is named Code Composer Studio™ (CCS). It will also introduce you to many of the libraries provided by TI for rapid development of microcontroller projects, such as MSP430ware™.

Whether you are a fan of the MSP430 for its low-power DNA, appreciate its simple RISC-like approach to processing, or are just trying to keep your system's cost to a minimum … we hope you'll enjoy working through this material as you learn how to use this nifty little MCU.



MSP-EXP430FR5969	MSP-EXP430FR4133	MSP-EXP430F5529

Chapter Topics

Administrative Topics

A few important details, if you're taking the class live. If not, we hope you already know where your own bathroom is located.

Administrative Topics

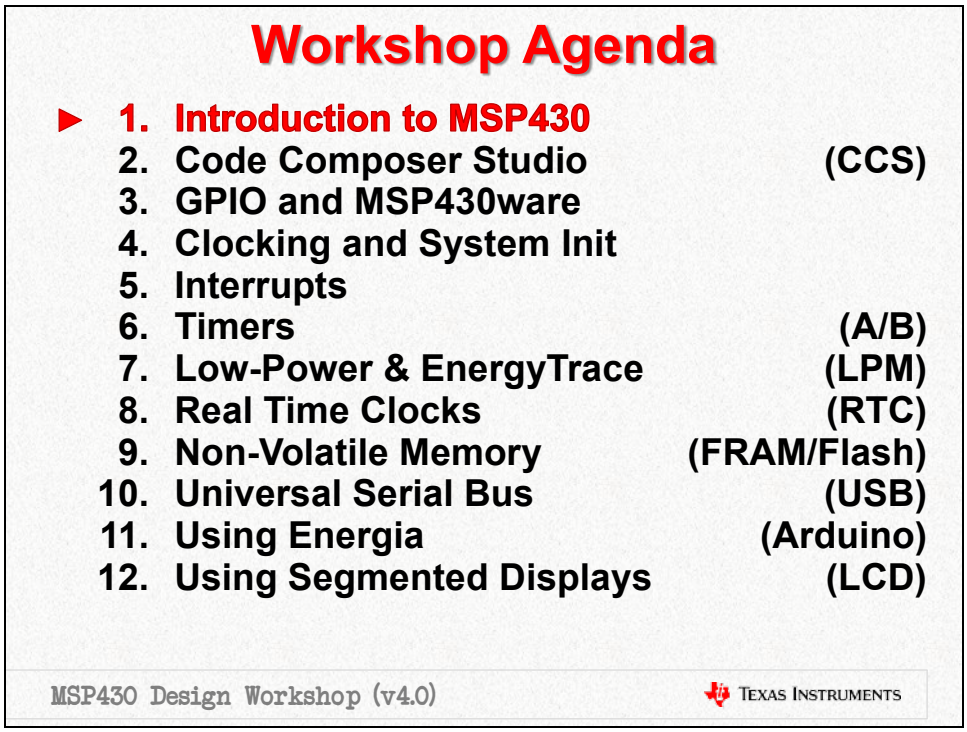
- Tools Install & Labs
- Start & End Times
- Lunch

- Course Materials
- Name Tags
- Restrooms

- Mobile Communications
- Questions & Dialogue (the key to learning)

Workshop Agenda

Here's the outline of chapters in this workshop.



Chapter 1: *"Intro"* Provides a quick introduction to TI, TI's Embedded Processors, as well as the MSP430 Family of devices.

Chapter 2: *"CCS"* introduces TI's development ecosystem. This includes:

- Code Composer Studio (CCSv5)
- Target software, such as MSP430ware and TI-RTOS
- TI's support infrastructure, including the embedded processors wiki and Engineer-to-Engineer (e2e) forums.

Chapter 3: *"GPIO"* This is our introduction to programming with MSP430ware; specifically, the DriverLib (i.e. driver library) part of MSP430ware. We start out by using it to program GPIO to blink an LED (often called the "embedded systems version of 'Hello World'"). The second part of the lab reads a Launchpad pushbutton.

Chapter 4: *"Clocks"* This chapter starts at reset – in fact, all three resets found on the MSP430. We then progress to examining the rich and robust clocking options provided in the MSP430. This is followed by the power management features found on many of the '430 devices. The chapter finishes up by reviewing the other required system initialization tasks … such as configuring (or turning off) the watchdog timer peripheral.

Chapter 5: *Interrupts* … do you use interrupts? Yep, they're one of the most fundamental parts of embedded system designs. This is especially true when your processor is known as

the king of low-power. We examine the sources, how to enable, and what to do in response to … interrupts.

Chapter 6: *Timers* are often thought of as the lifeblood of a microcontroller program. We use them to generate periodic events, as one-shot delays, or just to wake ourselves up every once in a while to read a sensor value. This chapter focuses on Timer_A – the primary timer module found in the MSP430.

Chapter 7: *Low Power Optimization* – shows the basic steps for lowering power usage. Following the ULP (ultra-low power) Advisor, we can find ways to minimize power in our code. Energy Trace is a new tool for measuring power and, on the 'FR58/59xx devices, examining the states of peripherals and clocks.

Chapter 8: *Real-Time Clocks* provides a very low-power timer to keep track of calendar, time and alarms.

Chapter 9: *Nov-Volatile Memory* – provides persistant storage, even when power is removed from the device. Most MSP430 devices contain either Flash or FRAM non-volatile memory.

Chapter 10: *USB* – Universal Serial Bus is an ideal way to communicate with host computers. This is especially true as most PC's have done away with dedicated serial and parallel ports. We attempt to explain how USB works as well as how to build an application around it. What you'll find is that the MSP430 team has done an excellent job of making USB simple.

Chapter 11: *Energia* is also known by the name "Arduino". *Energia* was the name given to Arduino as it was ported to the TI MCU's by the open-source community. Look up the definition of Energia – and let it 'propel' your application right off the Launchpad.

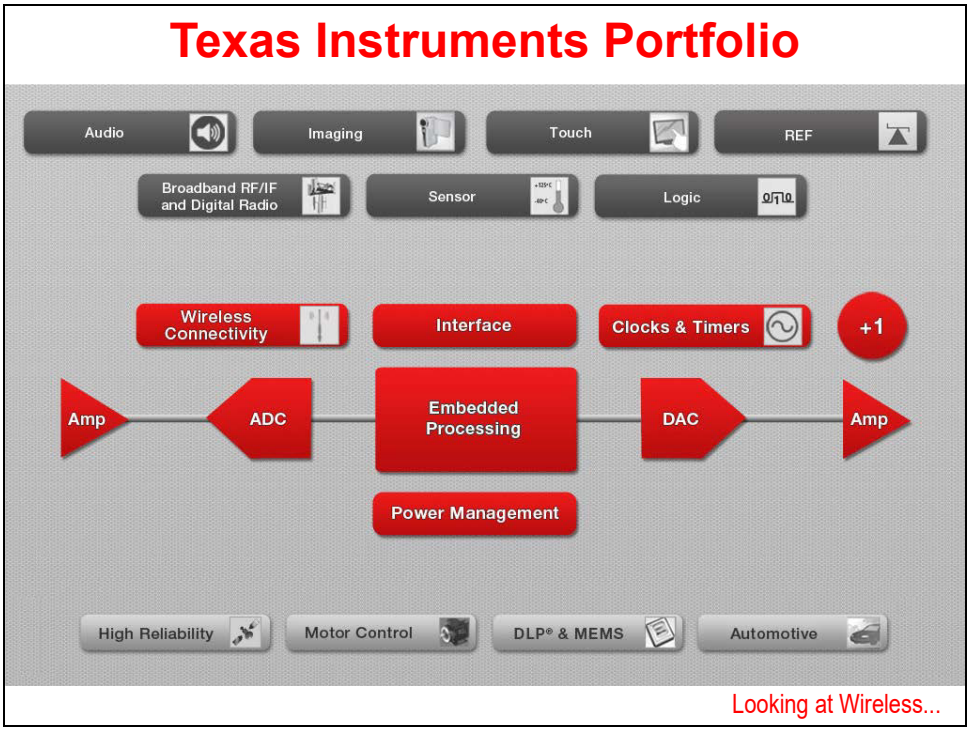
Chapter 12: Segmented LCD's (Liquid Crystal Displays) provide a convenient, low-power way of communicating with your system end-users. The 'FR4133 provides the lowest power LCD controller in the market. This chapter introduces you to LCD's in general, then to the specifics of using TI's LCD_E controller found on the 'FR4133 and its launchpad.

TI Products

TI's Entire Portfolio

It's very difficult to summarize the entire breadth of TI's semiconductor products – it's so far reaching. But, maybe that's not to be unexpected from the company who invented the integrated circuit.

Whether you are looking for embedded processors (the heart of following diagram) or all the components that sit alongside – such as power management, standard logic, op amps, data conversion, display drivers, or … so much more – you'll find them at TI.

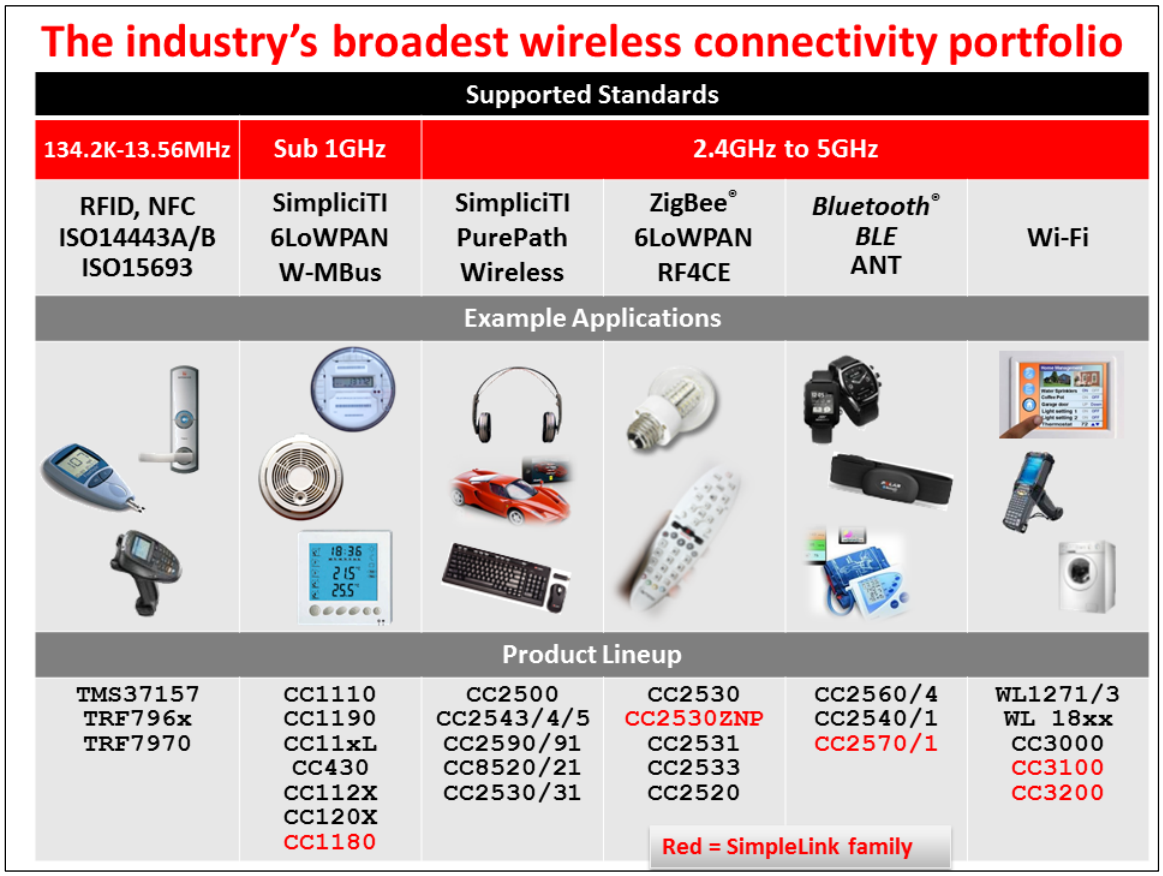


Before taking a closer look at embedded processors, we'll glance at one of the hottest growing product categories … TI's extensive portfolio of wireless connectivity.

Wireless Products

Wireless devices let us talk through the air. *Look ma, no wires.*

What protocol or frequency resonates with you and your end-customers? Whether it's: near-field communications (NFC); radio-frequency ID (RFID); the long range, low-power sub 1-GHz; ZigBee®; 6LoPan; Bluetooth® or Bluetooth Low Energy® (BLE); ANT®; or just good old Wi-Fi – TI's got you covered.



Many low-end, low-cost MCU designers have longed for a way to connect wirelessly to the rest of the world. TI's wireless devices and modules make this possible. No longer do you need a gigahertz processor to run the various networking stacks required to talk to the outside world – the TI SimpleLink line handles this for you … meaning that any processor that can communicate via a serial port can be networked. Drop a CC3000 module into your design and you've enabled it to join the *Internet of Things* revolution.

Check out TI's inexpensive, low-power and innovative wireless lineup!

TI's Embedded Processors

Whether you are looking for the MSP430, which is the lowest power microcontroller (MCU) in the world today … or the some of the highest performance single-chip microprocessors (MPU) ever designed (check out Multicore) … or something in between … TI has your needs covered.

Microcontrollers (MCU)				Application (MPU)		
MSP430	**C2000**	**Tiva C**	**Hercules**	**Sitara**	**DSP**	**Keystone**
16-bit **Ultra Low Power & Cost**	32-bit **Real-time**	32-bit **All-around MCU**	32-bit **Safety**	32-bit **Linux Android**	16/32-bit **All-around DSP**	32-bit **Massive Performance**
MSP430 ULP RISC MCU	• **Real-time C28x MCU** • **ARM M3+C28**	**ARM Cortex-M4F**	**ARM Cortex-M3 Cortex-R4**	**ARM Cortex-A8 Cortex-A9**	**DSP C5000 C6000**	• **C66 + C66** • **A15 + C66** • **A8 + C64** • **ARM9 + C674**
• Low Pwr Mode ▪ 250nA (RTC) ▪ 770nA (LCD) • Analog I/F • USB and RF	• Motor Control • Digital Power • Precision Timers/PWM	• 32-bit Float • Nested Vector IntCtrl (NVIC) • Ethernet (MAC+PHY)	• Lock step Dual-core R4 • ECC Memory • SIL3 Certified	• $5 Linux CPU • 3D Graphics • PRU-ICSS industrial subsys	• C5000 Low Power DSP • 32-bit fix/float C6000 DSP	• Fix or Float • Up to 12 cores 4 A15 + 8 C66x • DSP MMAC's: 352,000
TI-RTOS	**TI-RTOS (k)**	**TI-RTOS**	**3rd Party (only)**	**Linux, Android, TI-RTOS Kernel**	**C5x: DSP/BIOS C6x: TI-RTOS (k)**	**Linux TI-RTOS (k)**
Flash: 512K FRAM: 128K	512K Flash	1MB Flash	256K to 3M Flash	L1: 32K x 2 L2: 256K	L1: 32K x 2 L2: 256K	L1: 32K x 2 L2: 1M + 4M
25 MHz	300 MHz	120 MHz	220 MHz	1.35 GHz	800 MHz	1.4 GHz
$0.25 to $9.00	$1.85 to $20.00	$1.00 to $8.00	$5.00 to $30.00	$5.00 to $25.00	$2.00 to $25.00	$30.00 to $225.00

To start with, look at the **Blue**/**Red** row about ⅓ the way down the slide. The columns with **Red** signify devices utilizing ARM processor cores. If you didn't think TI embraces the ARM lineup of processors, think again. TI is one of the leaders in ARM development, manufacturing and sales.

Jumping to the 3rd column, the **Tiva C** (Tiva Connected) processors are probably the best all-around MCU's in use today. The 32-bit floating point ARM Cortex-M4F core can be connected to the real-world by a dizzying array of peripherals. They provide a near-perfect balance of performance, power, and connectivity.

On the other hand, if you're building safety critical applications, the **Hercules** family of processors is what you should key in on. Whether your customers appreciate the safety of dual-core, lockstep processing or the SIL3 certification, these processors are a unique mix of ARM Cortex-R4 performance combined with TI's vast SafeTI® knowledge.

Moving up to what ARM calls their 'Application' series of processors, TI set the processing world on fire (figuratively) when they introduced the **Sitara** AM335x. That you could get a $5 processor which runs Linux, Android or other high-level operating systems was jaw-dropping. We probably didn't make some PC manufactures happy – we've seen many of our customers replace bulky, power-hungry embedded PC's with small, low-power BeagleBoard-like replacements. This device was the inflection point – it's started a new direction for embedding high-level host systems.

And if you're looking for the high-end **ARM Cortex-A15**, we've got that too. Take your pick: do you want one … or up to 4 A15 cores on a single device? And these multi-core devices also pack the number crunching of TI's C66x line of DSP cores. When high-end performance processing is critical to your systems, look no further than TI Multicore.

But as one student asked, *"If ARM is so great, why do you make other types of processors?"*

While ARM is probably thought of today as the best all-around set of processor cores, there are areas where it can be improved upon.

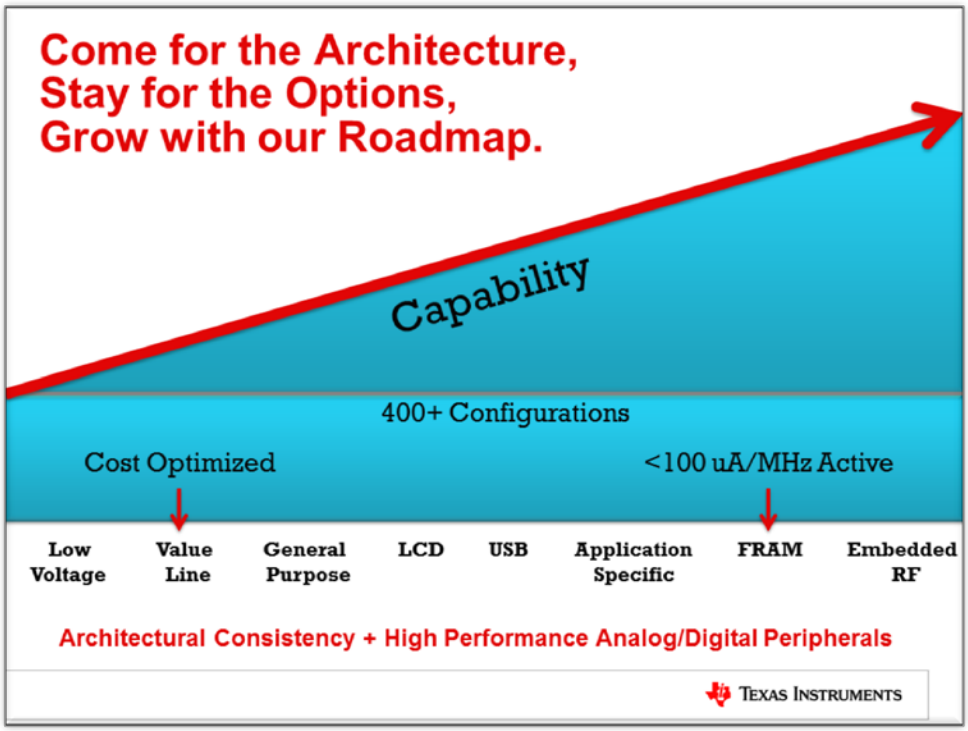
Driving to the *lowest-power dissipation* is one of those areas. In the end, the venerable **MSP430** is not to be outdone on the low end. As the MSP430 teams says, Ultra Low-Power (ULP) is "in our DNA". You know you're doing something right when the 10-year shelf-life of the battery ends up self-dissipating before you run it dry with your MSP430 design. It's just hard to beat an MCU designed from the ground up as a low-power CPU. That said, it's also hard to beat the MSP430's simple, inexpensive, high-performance RISC engine.

The **C2000** family has set the standard for control applications. Whether it's digital motor control, power control or one of the many other control-oriented MCU applications, this CPU really crunches the data. You might also see a little Red in this column. That's to indicate that even a good DSP-based microcontroller can use a little bit of ARM to get a leg-up in the industry. We've coupled an ARM Cortex-M3 along with the C28x core to make a stellar processing duo. Use the ARM to run your networking and USB stacks – all the while the C28x core is taking care of your system's real-time processing needs. Sure, you could buy two chips to implement your systems (we'll happily sell you a C28x along with Tiva C), but these devices integrate them both into a singular device.

Finally, TI is known by many as the center of **DSP** excellence. While these CPUs often get lost in all the hoopla surrounding ARM today, when it comes to real-time systems, a good DSP is hard to beat. Whether you're implementing a low-power system (look to **C5000** DSP's) or need the number crunching performance of the **C6000**, these devices still cannot be bested in the world of hard real-time, low-latency, highly deterministic applications. As mentioned earlier, the highest performing C6000 DSP cores have been combined into the awesome performance of Multicore. You can get up to 8 CPU's on a single device; make them all C66x DSPs – or match four C66x CPU's up with four of ARM's stunning Cortex-A15's for a performance knock-out punch.

MSP430 Family

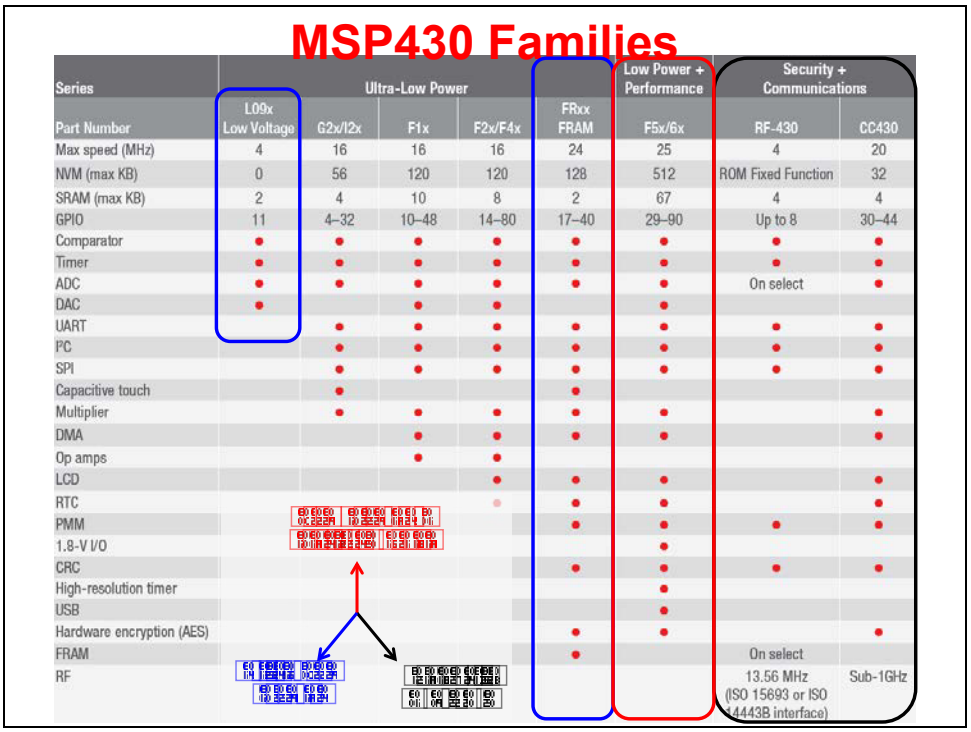
As stated, low-power is ‘in our DNA’. Though, it’s not all the MSP430 is known for.



One vector of new products has continued to integrate a wide range of low-power peripherals into the MSP430 platform. Look for the products in the MSP430 F5xx, F6xx and FR5xxx families. Also, the CC430 family adds the unique touch of on-chip integrated RF radios.

A second vector of development is driving the cost out of your designs. Look no further than the Gxxx Value Line series of devices. The goal is to provide highly integrated, low-power, 16-bit performance in an inexpensive device – giving you a new choice versus those old 8-bit micros.

And finally, the new MSP430 Wolverine series of devices is once again setting new standards for low-power processing. Sure, we're only topping our own products, but who else is better suited to enable your lowest power processing needs? Utilizing the FRAM memory technology, the FR5xxx Wolverine devices combine the lowest power dissipation with a rich integration of peripherals.



Series	Ultra-Low Power					Low Power + Performance	Security + Communications	
Part Number	L09x Low Voltage	G2x/I2x	F1x	F2x/F4x	FRxx FRAM	F5x/6x	RF-430	CC430
Max speed (MHz)	4	16	16	16	24	25	4	20
NVM (max KB)	0	56	120	120	128	512	ROM Fixed Function	32
SRAM (max KB)	2	4	10	8	2	67	4	4
GPIO	11	4–32	10–48	14–80	17–40	29–90	Up to 8	30–44
Comparator	•	•	•	•	•	•	•	•
Timer	•	•	•	•	•	•	•	•
ADC	•	•	•	•	•	•	On select	•
DAC	•		•	•		•		
UART		•	•	•	•	•	•	•
I²C		•	•	•	•	•	•	•
SPI		•	•	•	•	•	•	•
Capacitive touch		•			•			
Multiplier		•	•	•	•	•		•
DMA			•	•	•	•		•
Op amps			•	•				
LCD				•	•	•		•
RTC					•	•		•
PMM					•	•	•	•
1.8-V I/O						•		
CRC					•	•	•	•
High-resolution timer						•		
USB						•		
Hardware encryption (AES)					•	•		•
FRAM					•		On select	
RF							13.56 MHz (ISO 15693 or ISO 14443B interface)	Sub-1GHz

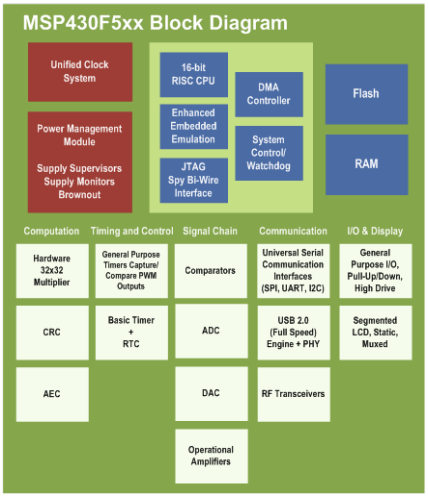
Here's a quick overview of the device we'll be using in this workshop. The MSP430F5529 is part of the F5xx series of devices and is found on the new 'F5529 USB Launchpad.

F5xx Key Features

Ultra-Low Power
- 160 µA/MIPS
- 2.5 µA standby mode
- Integrated LDO, BOR, WDT+, RTC
- 12 MHz @ 1.8V
- Wake up from standby in <5 µs

Increased Performance
- Up to 25 MHz
- 1.8V ISP Flash erase and write
- Fail-safe, flexible clocking system
- User-defined Bootstrap Loader
- Up to 1MB linear memory addressing

Innovative Features
- Multi-channel DMA supports data movement in standby mode
- Industry leading code density
- More design options including USB, RF, encryption, LCD interface



Looking at the 'FR59xx...

MSP430FR58xx/59xx

MSP430FR58/59xx
Ultra Low Power
16-bit MCU

16MHz

TEXAS INSTRUMENTS

Memory
- FRAM (32/48 / 64 KB)
- RAM (1 or 2 KB)
- MPU

Debug
- Real Time JTAG
- Embedded emulation
- Bootstrap Loader

Accelerators
- 32x32 Multiplier
- DMA (3 Ch)
- CRC16
- AES256 Encryption (FR59xx)

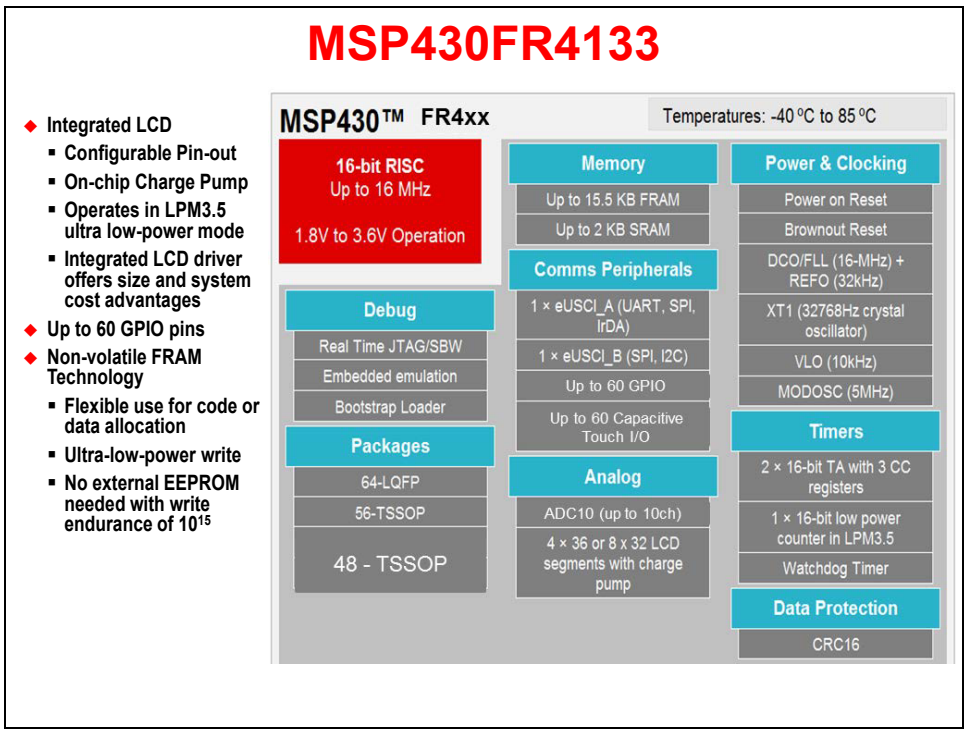
Serial Interfaces
- 3 Serial Interfaces (eUSCI)
- 2 UART + IrDA or SPI
- 2 I2C or SPI

Analog
- 12-bit SAR ADC (up to 16 ch)
  - Differential inputs
  - Window comparators
- Comparator (Comp_E)
- Vref (REF_A)

Timers
- Watch Dog Timer (WDT_A)
- Real Time Clock (RTC_B)
- Two 16-bit w/3 CCR (TA0, TA1)
- Two 16-bit w/2 CCR (TA2, TA3)
- One 16-bit w/7 CCR (TB0)

Connectivity
- Up to 40 GPIO (Interrupt/Wake)
- Cap touch IO

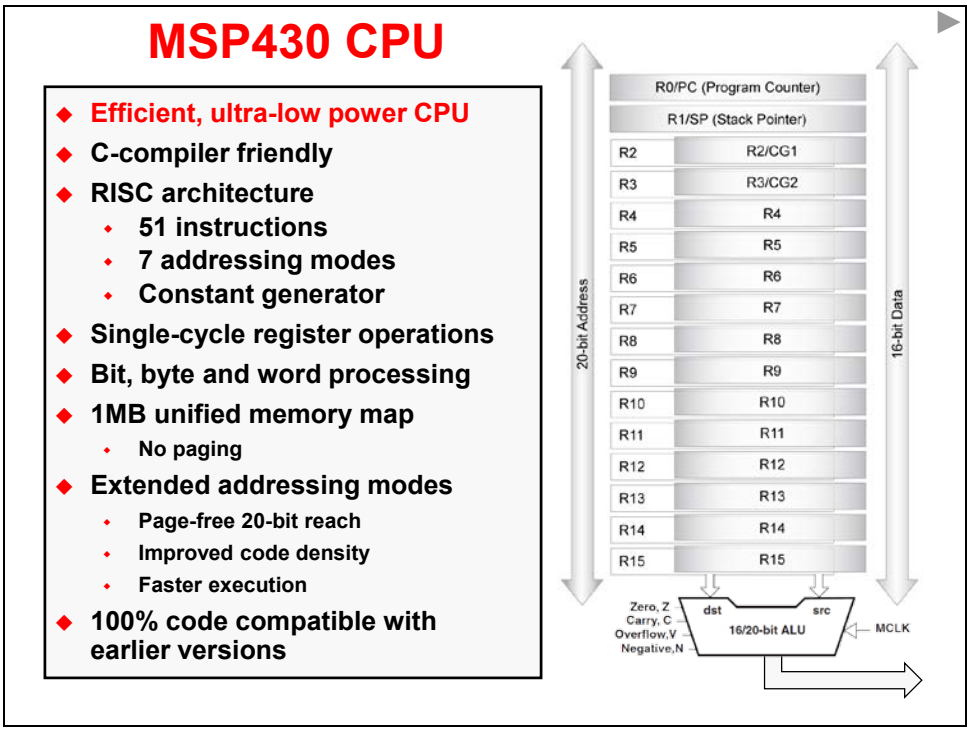
Power & Clocking
- Brownout Reset
- Supply Voltage Supervisor (SVS)
- Low Power Vreg (1.5V LDO)
- External Oscillators: LFXT, HFXT
- Internal Oscillators: VLO, DCO (±2%)



These are three of TI's line-up of MSP430 devices – each featuring highly integrated set of peripherals. We will be exploring quite a bit more about them as we go through this workshop.

MSP430 CPU

As stated earlier, the MSP430 is an efficient, simple 16-bit low power CPU. Its orthogonal architecture and register set make it C-compiler friendly.



The original MSP430 devices were true 16-bit processors. While 16-bits are quite ideal from a data perspective, it's limited from an addressing perspective. With 16-bit addresses, you're limited to only 64K of memory – and that really isn't acceptable in many of today's applications.

As early as the second generation of MSP430 devices, the CPU was expanded to provide full 20-bits of addressing space – which provides 1M of address reach. The new CPU cores that support these enhancements were called CPUX (for eXtended addressing). Thankfully, the extended versions of the CPU maintained backward compatibility with the earlier devices.

In this course, we don't dwell on these CPU features for two reasons:

- This change was made long enough to go that all the processors engineers choose today include the enhanced CPU.
- With the prevalence of C coded applications in world of MSP 430, and embedded processing in general, these variations fall below our radar. The compiler, handily, manages low-level details such as this.

There are many touches to the MSP430 CPU which make it idea for low-power and microcontroller applications, such as the ability to manage bytes, as well as 16-bit words.

Bytes, Words And CPU Registers

```
16-bit addition                                  Code/Cycles
5405            add.w   R4,R5                    ; 1/1
529202000202    add.w   &0200,&0202              ; 3/6

8-bit addition
5445            add.b   R4,R5                    ; 1/1
52D202000202    add.b   &0200,&0202              ; 3/6
```

- Use CPU registers for calculations and dedicated variables
- Same code size for word or byte
- Use word operations when possible

Seven addressing modes …

Note: If you see a 'gray' slide like the one above and below were placed into the workbook, but has been hidden in the slide set, so the instructor may not present it during class.

Seven Addressing Modes

Mode	Example	Notes
Register	`mov.w R10,R11`	Single cycle
Indexed	`mov.w 2(R5),6(R6)`	Table processing
Symbolic	`mov.w EDE,TONI`	Easy to read code, PC relative
Absolute	`mov.w &EDE,&TONI`	Directly access any memory
Indirect Register	`mov.w @R10,0(R11)`	Access memory with pointers
Indirect Autoincrement	`mov.w @R10+,0(R11)`	Table processing
Immediate	`mov.w #45h,&TONI`	Unrestricted constant values

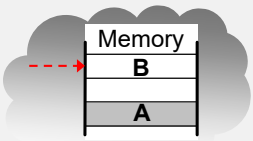
Atomic

Atomic addressing …

A rich set of addressing modes lets the compiler create efficient, small-footprint programs. And, features like ‘atomic’ addressing are critical for real-world embedded processing.

Atomic Addressing

B=B+A




```
; Pure RISC
push    R5
ld      R5,A
add     R5,B
st      B,R5
pop     R5
```

```
; MSP430
 add    A,B
```

- Non-interruptible memory-to-memory operations
- Useable with complete instruction set

Constant generator …

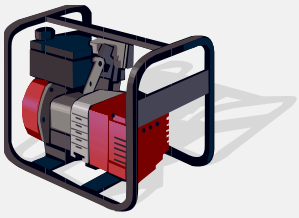
The little bit of genius that is the Constant Generator minimizes code size and runtime cycle count. These ideas save you money while helping to reduce power dissipation.

Constant Generator

```
4314            mov.w   #0002h,R4       ; With CG

40341234        mov.w   #1234h,R4       ; Without CG
```

- Immediate values -1,0,1,2,4,8 generated in hardware
- Reduces code size and cycles
- Completely automatic



A low number of instructions are at the heart of Reduced Instruction Set Computers (RISC). RISC lowers complexity, cost and power … while, surprisingly, maintaining performance.

51 Total Assembly Instructions

Format I Src, Dest	Format II Single Operand	Format III +/- 9bit Offset	Support
add(.b)	**br**	jmp	**clrc**
addc(.b)	call	jc	**setc**
and(.b)	swpb	jnc	**clrz**
bic(.b)	sxt	jeq	**setz**
bis(.b)	push(.b)	jne	**clrn**
bit(.b)	**pop(.b)**	jge	**setn**
cmp(.b)	rra(.b)	jl	**dint**
dadd(.b)	rrc(.b)	jn	**eint**
mov(.b)	**inv(.b)**		**nop**
sub(.b)	**inc(.b)**		**ret**
subc(.b)	**incd(.b)**		reti
xor(.b)	**dec(.b)**		
	decd(.b)		
	adc(.b)		
	sbc(.b)		
	clr(.b)		
	dadc(.b)		
	rla(.b)		
	rlc(.b)		
	tst(.b)		

Bold type denotes emulated instructions

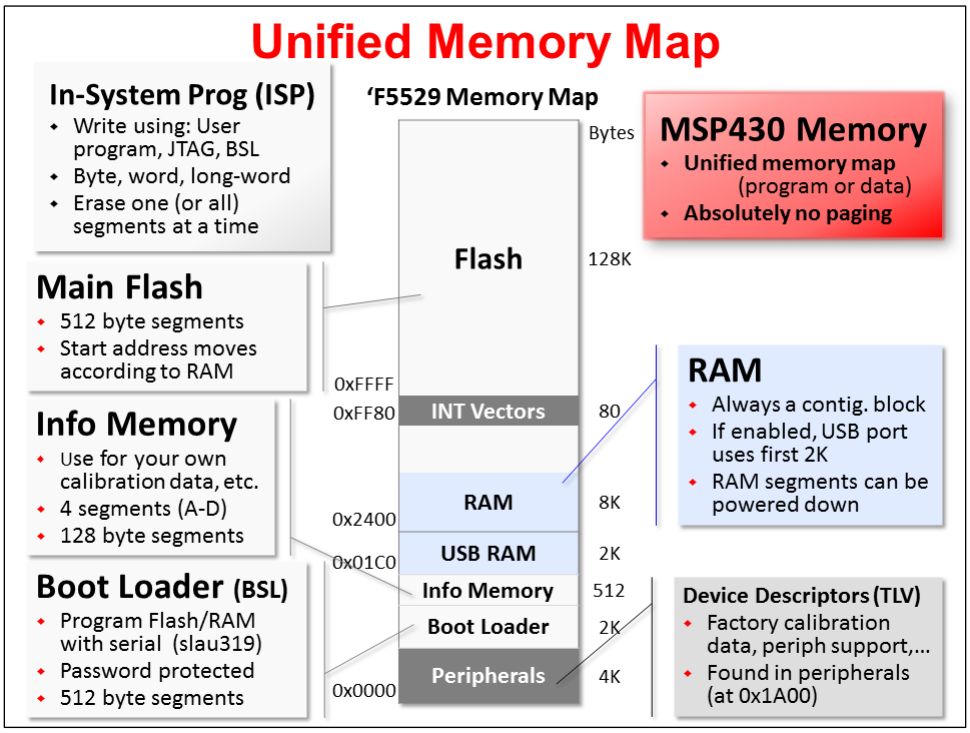
MSP430 Memory

Memory Map

We present the MSP430F5529 memory map as an example of what you find on most MSP430's. It's certainly what we'll see as we work though the lab exercises in this workshop.

A couple of important – and beneficial – points about MSP430's memory map:

- The MSP430 defines a ***unified*** memory map. This means that, technically speaking, data and program code can be located anywhere in the available memory space. *(This doesn't mean it's practical to locate global variables in flash memory, but the architecture does not prevent you from doing so.)*
- The MSP430, as stated earlier (see page 1-14), is implemented using 20-bit addressing; therefore, the MSP430 can directly address the full 1M memory map without resorting to paging schemes. (If you have ever had to deal with paging, we expect you might be cheering at this point.)



Flash

Like most MCU's nowadays, the processor is dominated by non-volatile memory. In this case, Flash technology provides us with the means to store information into the device – which retains its contents, even when power is removed. (As we'll see next, some of the latest MSP430 devices use FRAM technology rather than Flash.)