

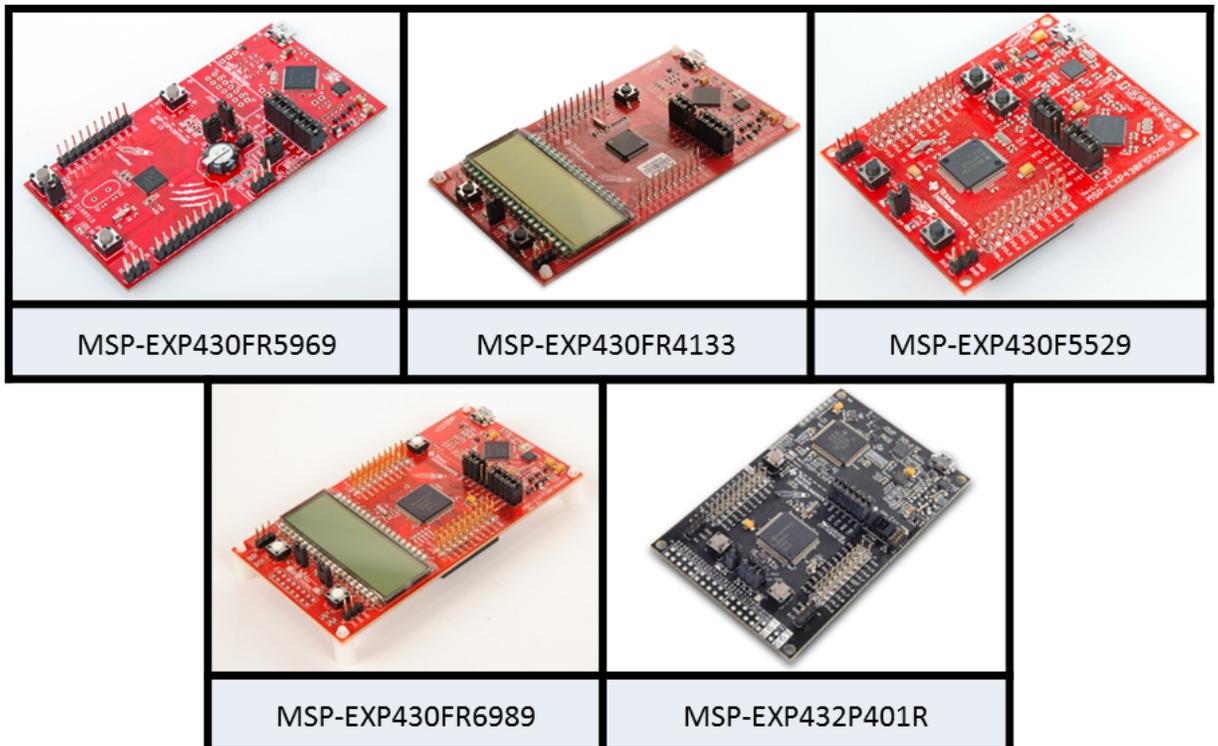
# Introduction to MSP Microcontrollers

---

## Introduction

Welcome to the **MSP Design Workshop**. This workshop covers the fundamental skills needed when designing a system based on the Texas Instruments (TI) MSP™ 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 MSPWare™.

Whether you are a fan of the MSP microcontroller 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.



# Chapter Topics

<b>Introduction to MSP Microcontrollers.....</b>	<b>1-1</b>
<i>Administrative Topics</i> .....	1-3
<i>Workshop Agenda</i> .....	1-4
<i>TI Products</i> .....	1-6
TI's Entire Portfolio.....	1-6
Wireless Products .....	1-7
<i>TI's Embedded Processors</i> .....	1-8
<i>MSP Microcontroller Family</i> .....	1-10
MSP430.....	1-12
MSP432.....	1-13
<i>MSP430 CPU</i> .....	1-15
<i>MSP Microcontroller Memory</i> .....	1-19
Memory Map .....	1-19
FRAM.....	1-22
<i>MSP Peripherals</i> .....	1-25
GPIO .....	1-25
Timers .....	1-26
Clocking and Power Management.....	1-27
Analog.....	1-28
Communications (Serial ports, USB, Radio).....	1-30
Hardware Accelerators.....	1-31
Summary.....	1-32
<i>ULP</i> .....	1-33
Profile Your Activities .....	1-34
<i>Community / Resources</i> .....	1-39
References.....	1-41
<i>LaunchPads</i> .....	1-42
MSP-EXP430F5529LP LaunchPad.....	1-42
MSP-EXP430FR5969 LaunchPad.....	1-44
MSP-EXP430FR4133 LaunchPad.....	1-44
MSP-EXP430FR6989 LaunchPad.....	1-45
MSP-EXP432P401R LaunchPad.....	1-45
<i>Lab 1 – Out-of-Box User Experience Lab</i> .....	1-47

## 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.

## Workshop Agenda

- ▶ **1. Introduction to MSP Microcontrollers**
- 2. Code Composer Studio (CCS)**
- 3. GPIO and MSP430ware**
- 4. Clocking and System Init**
- 5. Interrupts**
- 6. Timers (A/B)**
- 7. Low-Power & EnergyTrace (LPM)**
- 8. Real Time Clocks (RTC)**
- 9. Non-Volatile Memory (FRAM/Flash)**
- 10. Universal Serial Bus (USB)**
- 11. Using Energia (Arduino)**
- 12. Using Segmented Displays (LCD)**

MSP Design Workshop (v4.5) TEXAS INSTRUMENTS

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

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

- Code Composer Studio (CCSv5)
- Target software, such as MSPware 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 MSPWare; specifically, the DriverLib (i.e. driver library) part of MSPWare. 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 MSP. We then progress to examining the rich and robust clocking options provided in MSP Microcontrollers. This is followed by the power management features found on many of the MSP 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 MSP microcontrollers.
- 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:** *Non-Volatile Memory* – provides persistent storage, even when power is removed from the device. Most MSP 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 MSP 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's end-users.

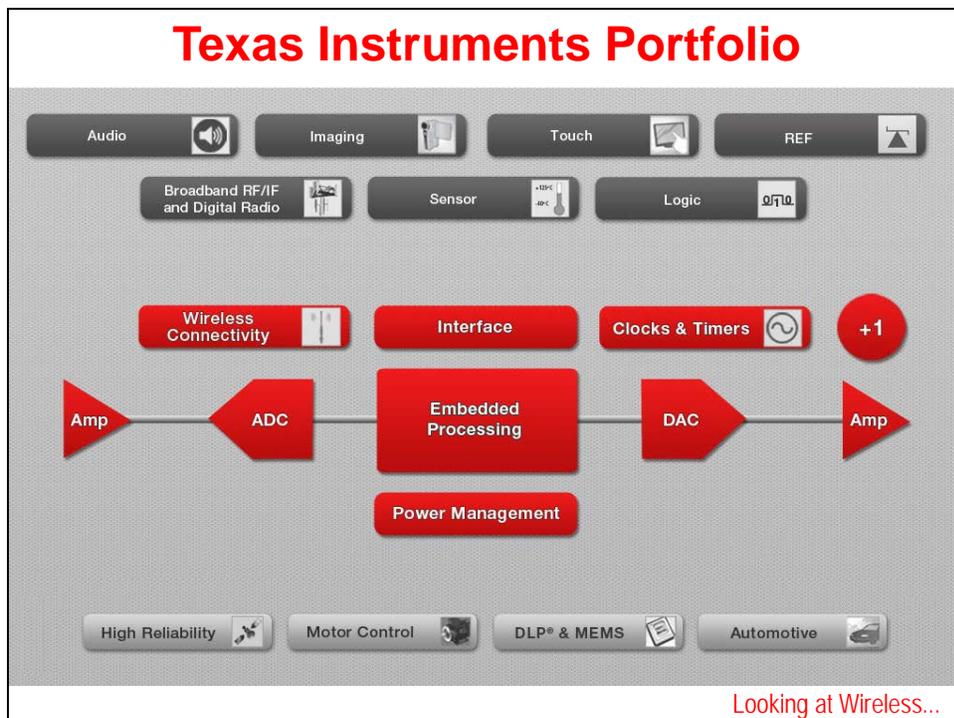
Both of the 'FR4133 and 'FR6989 LaunchPads contain an LCD, which showcase their on-chip LCD peripherals. The 'FR6989 brings forward the excellent LCD\_C controller found on many previous MSP430 processors... but the 'FR4133 stands out by introducing the new LCD\_E peripheral – with the ability to drive a display with the device in "off" mode (LPM3.5), it becomes the lowest power LCD controller available today.

# 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.

The industry's broadest wireless connectivity portfolio					
Supported Standards					
134.2K-13.56MHz	Sub 1GHz	2.4GHz to 5GHz			
RFID, NFC ISO14443A/B ISO15693	SimpliciTI 6LoWPAN W-MBus	SimpliciTI PurePath Wireless	ZigBee® 6LoWPAN RF4CE	Bluetooth® BLE ANT	Wi-Fi
Example Applications					
					
Product Lineup					
TMS37157 TRF796x TRF7970	CC1110 CC1190 CC11xL CC430 CC112X CC120X CC1180	CC2500 CC2543/4/5 CC2590/91 CC8520/21 CC2530/31	CC2530 CC2530ZNP CC2531 CC2533 CC2520	CC2560/4 CC2540/1 CC2570/1	WL1271/3 WL 18xx CC3000 CC3100 CC3200
			Red = SimpleLink family		

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 MSP microcontrollers, 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)		
MSP	C2000	Tiva C	SimpleLink	Sitara	DSP	Keystone
16-bit/32-bit <b>Ultra Low Power &amp; Cost</b>	32-bit <b>Real-time</b>	32-bit <b>Connectivity</b>	32-bit <b>Wireless</b>	32-bit <b>Linux Android</b>	16/32-bit <b>All-around DSP</b>	32-bit <b>Massive Performance</b>
<ul style="list-style-type: none"> <li>• <b>MSP430</b> 16-bit RISC</li> <li>• <b>MSP432</b> 32-bit ARM</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Real-time C28x MCU</b></li> <li>• ARM M3+C28</li> </ul>	<ul style="list-style-type: none"> <li>• <b>ARM Cortex-M4F</b></li> </ul>	<ul style="list-style-type: none"> <li>• <b>ARM Cortex-M3</b></li> <li>• <b>Cortex-M4</b></li> </ul>	<ul style="list-style-type: none"> <li>• <b>ARM Cortex-A8</b></li> <li>• <b>Cortex-A9</b></li> <li>• <b>Cortex-15</b></li> </ul>	<ul style="list-style-type: none"> <li>• <b>DSP C5000</b></li> <li>• <b>C6000</b></li> </ul>	<ul style="list-style-type: none"> <li>• <b>C66 + C66</b></li> <li>• <b>A15 + C66</b></li> <li>• <b>A8 + C64</b></li> <li>• <b>ARM9 + C674</b></li> </ul>
<ul style="list-style-type: none"> <li>• Low Pwr Mode = 250nA (RTC)</li> <li>• 770nA (LCD)</li> <li>• Smart Analog</li> <li>• EnergyTrace++</li> </ul>	<ul style="list-style-type: none"> <li>• Motor Control</li> <li>• Digital Power</li> <li>• Precision Timers/PWM</li> </ul>	<ul style="list-style-type: none"> <li>• 32-bit Float</li> <li>• Massive I/O</li> <li>• Ethernet (MAC+PHY)</li> </ul>	<ul style="list-style-type: none"> <li>• M4 w/ WiFi</li> <li>• M3 w/ 2.4GHz</li> <li>• M3 w/ Sub-1GHz</li> </ul>	<ul style="list-style-type: none"> <li>• \$5 Linux CPU</li> <li>• 3D Graphics</li> <li>• PRU</li> <li>• Industrial I/O</li> </ul>	<ul style="list-style-type: none"> <li>• C5000 Low Power DSP</li> <li>• 32-bit fix/float C6000 DSP</li> </ul>	<ul style="list-style-type: none"> <li>• Fix or Float</li> <li>• Up to 12 cores 4 A15 + 8 C66x</li> <li>• DSP MMACs: 352,000</li> </ul>
<b>TI-RTOS FreeRTOS, RTX</b>	<b>TI-RTOS</b>	<b>TI-RTOS FreeRTOS</b>	<b>TI-RTOS FreeRTOS</b>	<b>Linux, Android, TI-RTOS</b>	<b>C5x: DSP/BIOS</b> <b>C6x: TI-RTOS</b>	<b>Linux TI-RTOS</b>
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 1/3 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 3<sup>rd</sup> 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<sup>®</sup> 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 manufacturers 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 **MSP** microcontroller is not to be outdone on the low end. As the MSP team states, 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 MSP microcontroller 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 MSP’s simple, inexpensive, high-performance RISC engine.

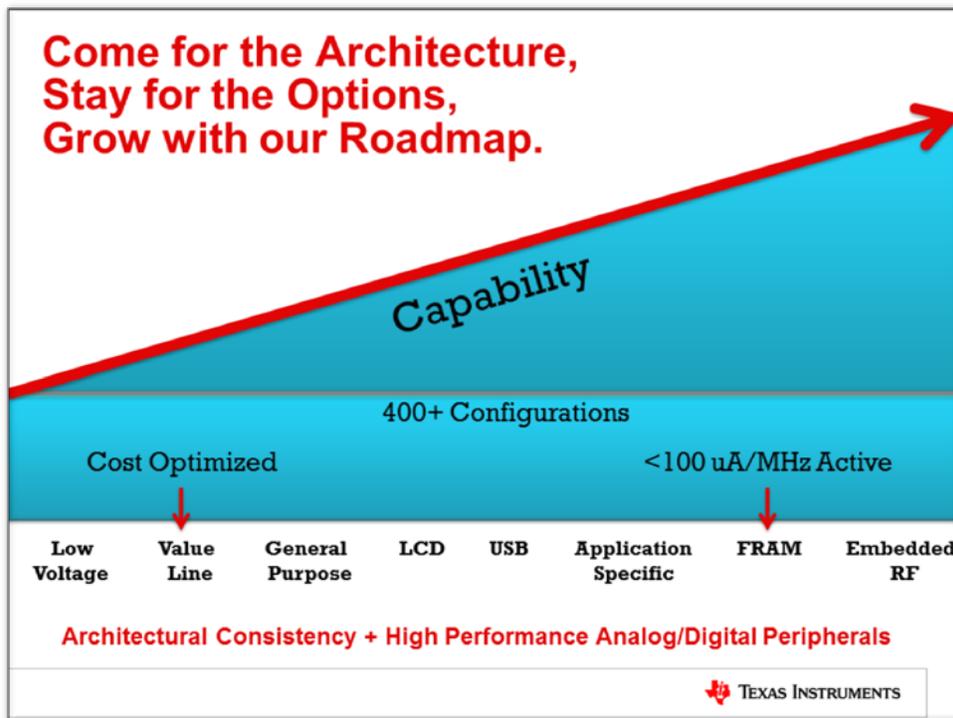
Then again, the MSP432 gives you access to the ARM Cortex-M4F 32-bit performance, while providing the lowest power device in its class.

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.

## MSP Microcontroller Family

As stated, low-power is 'in our DNA'. Though, it's not all the MSP microcontrollers are known for.



One vector of new products has continued to integrate a wide range of low-power peripherals into the MSP 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 MSP FRAM 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.

### MSP430 Families

Series	Ultra-Low Power				Low Power + Performance		Security + Communications	
	L09x Low Voltage	G2x/12x	F1x	F2x/F4x	FRx FRAM	F5x/6x	RF-430	CC430
Part Number								
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	•	•	•	•	•	•	•	•
PC	•	•	•	•	•	•	•	•
SPI	•	•	•	•	•	•	•	•
Capacitive touch		•			•			
Multiplier		•	•	•	•	•		•
DMA			•	•	•	•		•
Op amps			•	•				
LCD				•				•
RTC				•	•	•		•
PMM					•	•	•	•
1.8-V I/O					•	•		•
CRC					•	•	•	•
High-resolution timer						•		
USB						•		
Hardware encryption (AES)					•	•		•
FRAM					•			
RF							On select 13.56 MHz (ISO 15693 or ISO 14443B interface)	Sub-1GHz

Low Power + Performance

Ultra Low Power

Security + Comm

# MSP430

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  $\mu$ A/MIPS
- ◆ 2.5  $\mu$ A standby mode
- ◆ Integrated LDO, BOR, WDT+, RTC
- ◆ 12 MHz @ 1.8V
- ◆ Wake up from standby in <5  $\mu$ s

**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

**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 ( $\pm$ 2%)

## MSP430FR4133

- ◆ Integrated LCD
  - Configurable Pin-out
  - On-chip Charge Pump
  - Operates in LPM3.5 ultra low-power mode
  - Integrated LCD driver offers size and system cost advantages
- ◆ Up to 60 GPIO pins
- ◆ Non-volatile FRAM Technology
  - Flexible use for code or data allocation
  - Ultra-low-power write
  - No external EEPROM needed with write endurance of  $10^{15}$

MSP430™ FR4xx		Temperatures: -40 °C to 85 °C
<b>16-bit RISC</b> Up to 16 MHz  1.8V to 3.6V Operation	<b>Memory</b> Up to 15.5 KB FRAM Up to 2 KB SRAM	<b>Power &amp; Clocking</b> Power on Reset Brownout Reset DCO/FLL (16-MHz) + REFO (32kHz) XT1 (32768Hz crystal oscillator) VLO (10kHz) MODOSC (5MHz)
<b>Debug</b> Real Time JTAG/SBW Embedded emulation Bootstrap Loader	<b>Comms Peripherals</b> 1 × eUSCI_A (UART, SPI, IrDA) 1 × eUSCI_B (SPI, I2C) Up to 60 GPIO Up to 60 Capacitive Touch I/O	<b>Timers</b> 2 × 16-bit TA with 3 CC registers 1 × 16-bit low power counter in LPM3.5 Watchdog Timer
<b>Packages</b> 64-LQFP 56-TSSOP  48 - TSSOP	<b>Analog</b> ADC10 (up to 10ch) 4 × 36 or 8 × 32 LCD segments with charge pump	<b>Data Protection</b> CRC16

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

## MSP432

## MSP432 – Low-power + Performance

**MORE PERFORMANCE FOR MSP430™ DEVELOPERS**

Advance to higher levels of computing and analog performance, while maximizing your ultra-low-power MSP430 MCU investment and expertise

**LOWER POWER FOR ARM® DEVELOPERS**

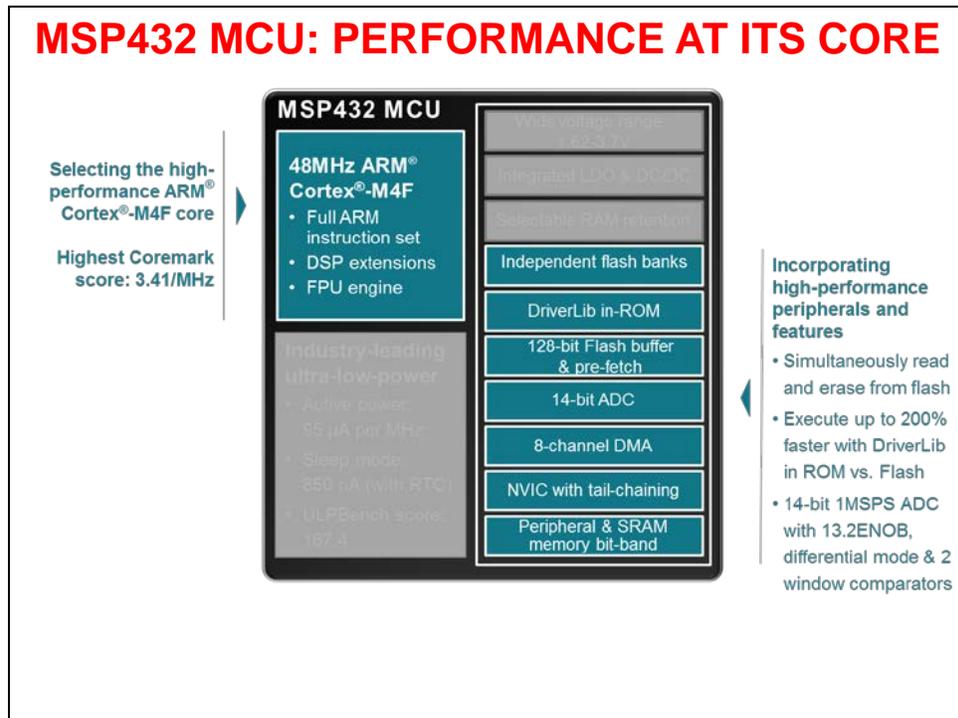
Slash power consumption and boost performance with the world's lowest power Cortex®-M microcontroller

**NO COMPROMISES**

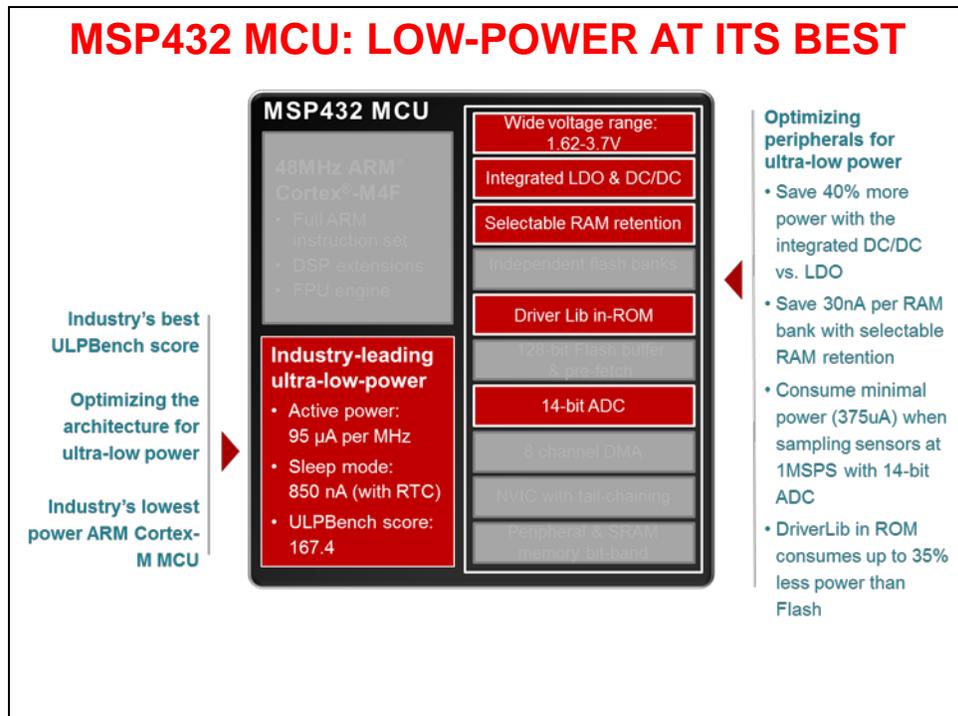
Get low-power and performance with a scalable portfolio of 16-bit and 32-bit MSP microcontrollers in a variety of applications



## MSP432 MCU: PERFORMANCE AT ITS CORE

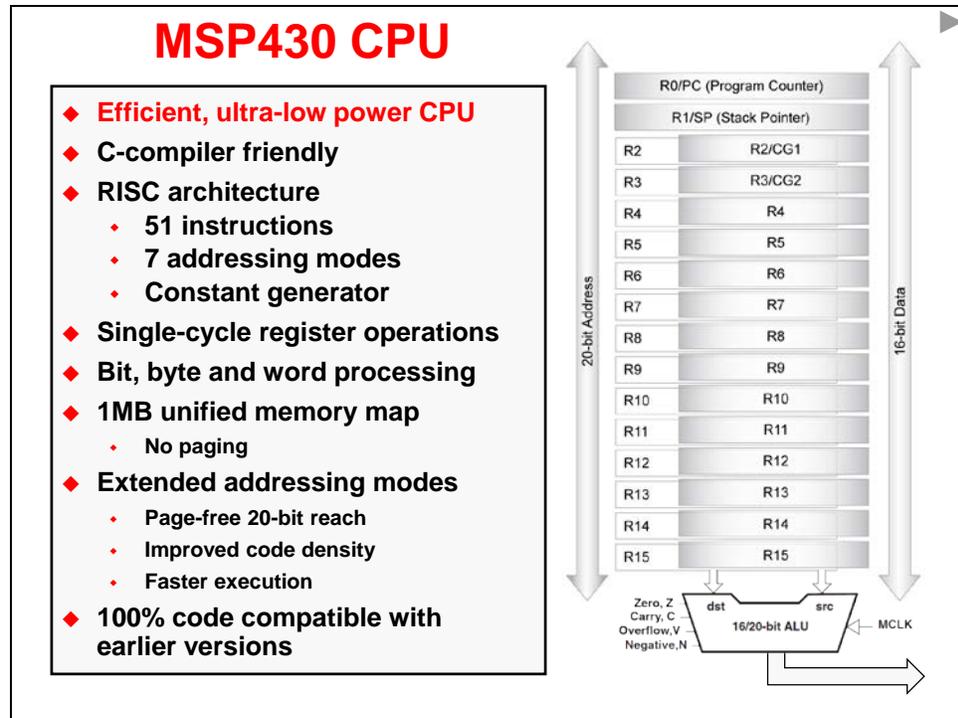


## MSP432 MCU: LOW-POWER AT ITS BEST



## MSP430 CPU

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



The original 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 MSP 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
<b>Register</b>	<code>mov.w R10,R11</code>	Single cycle
<b>Indexed</b>	<code>mov.w 2(R5),6(R6)</code>	Table processing
<b>Symbolic</b>	<code>mov.w EDE,TONI</code>	Easy to read code, PC relative
<b>Absolute</b>	<code>mov.w &amp;EDE,&amp;TONI</code>	Directly access any memory
<b>Indirect Register</b>	<code>mov.w @R10,0(R11)</code>	Access memory with pointers
<b>Indirect Autoincrement</b>	<code>mov.w @R10+,0(R11)</code>	Table processing
<b>Immediate</b>	<code>mov.w #45h,&amp;TONI</code>	Unrestricted constant values

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

$B=B+A$

Memory
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.

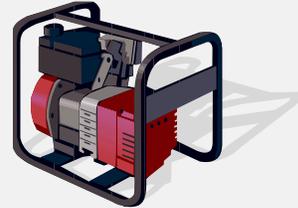
## Constant Generator

```

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)	<b>br</b>	jmp	<b>clrc</b>
addc(.b)	call	jc	<b>setc</b>
and(.b)	swpb	jnc	<b>clrz</b>
bic(.b)	sxt	jeq	<b>setz</b>
bis(.b)	push(.b)	jne	<b>clrn</b>
bit(.b)	<b>pop(.b)</b>	jge	<b>setn</b>
cmp(.b)	rra(.b)	j1	<b>dint</b>
dadd(.b)	rrc(.b)	jn	<b>eint</b>
mov(.b)	<b>inv(.b)</b>		<b>nop</b>
sub(.b)	<b>inc(.b)</b>		<b>ret</b>
subc(.b)	<b>incd(.b)</b>		reti
xor(.b)	<b>dec(.b)</b>		
	<b>decd(.b)</b>		
	<b>adc(.b)</b>		
	<b>sbc(.b)</b>		
	<b>clr(.b)</b>		
	<b>dadc(.b)</b>		
	<b>rla(.b)</b>		
	<b>rlc(.b)</b>		
	<b>tst(.b)</b>		

**Bold type denotes emulated instructions**

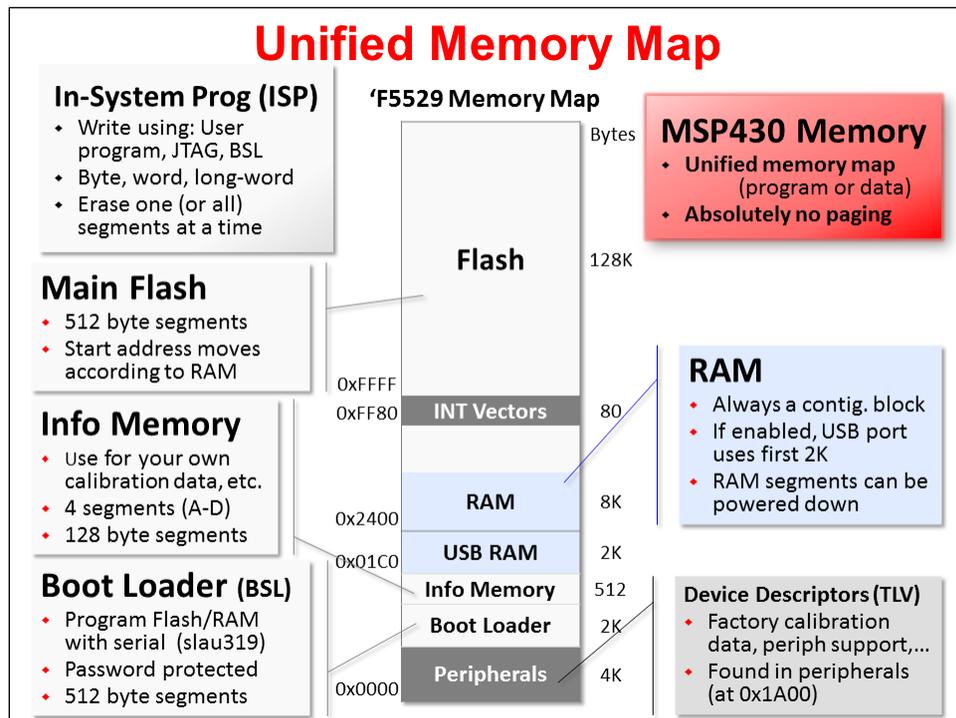
# MSP Microcontroller Memory

## Memory Map

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

A couple of important – and beneficial – points about the MSP memory map:

- MSP devices utilize 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.*)
- MSP microcontrollers, as stated earlier (see page 1-15), is implemented using 20-bit addressing; therefore, the MSP microcontroller 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 MSP devices use FRAM technology rather than Flash.)

The flash memory is In-System Programmable (ISP), which means we can reprogram the memory without taking the chip off of our boards or using difficult bed-of-nails methods. In fact, you can program the flash using:

- An IDE, such as CCS or IAR. These debugging tools utilize the 4-wire JTAG or 2-wire SPI-biwire emulation connections.
- The MSP Boot-Strap Loader supports a variety of connections and options. For example, you can use the serial (or USB) interfaces to reprogram your devices. These interfaces are popular on many manufacturing work flows.
- Finally, you can reprogram all – or part – of the flash memory via your own program running on the device itself. Check out the MSPWare FLASH DriverLib functions.

On the 'F5529, as with most MSP devices, the Flash actually consists of 3 regions.

**Main** consists of the bulk of flash memory. This is where our programs are written to when using the default project settings. Main flash consists of one contiguous memory; although, the Interrupt Vectors are located inside of it at 0xFF80. If your device has more than 64K of flash, then some will exist above and below the vectors – as shown in the diagram for the 'F5529 (which has 128K of flash).

**Info Memory** can be thought of as user data flash. Again, there are not any limitations on what you store here, but these four segments are commonly used to hold calibration data or other non-program items you want to store in non-volatile memory.

**Boot Loader (BSL)** holds the aforementioned boot loader code. This code, in turn, is used to load new programs into Main flash. Please be aware that the BSL is handled differently amongst the various generations of MSP microcontrollers. In some cases, as with the 'F5529, it is stored in its own region of flash memory. On other devices, it may be hard-coded into the device.

## RAM

**RAM** (Static Random Access Memory – SRAM) is found on every MSP device. Like flash, though, the amount of RAM varies from device to device; and the amount of RAM memory is often directly proportional to the cost of the device.

RAM is where most of the data is stored: everything from global variables, to stacks and heaps. It is often thought of as the 'working' memory on the device. Even so, due to the 'unified' nature of the MSP microcontroller architecture, you can also move program code into RAM and run from this space.

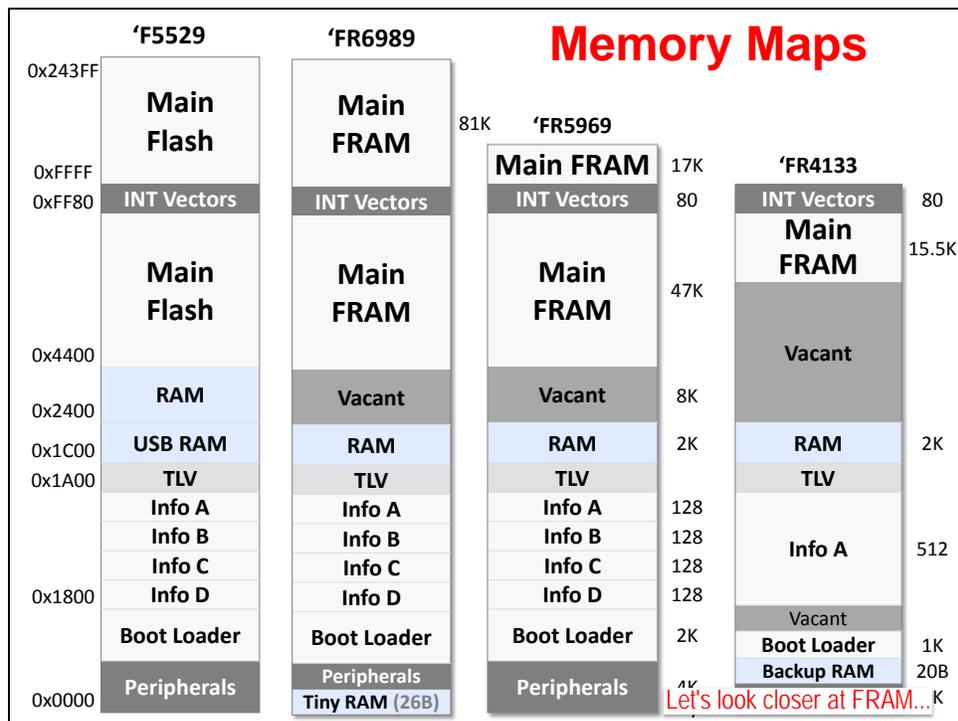
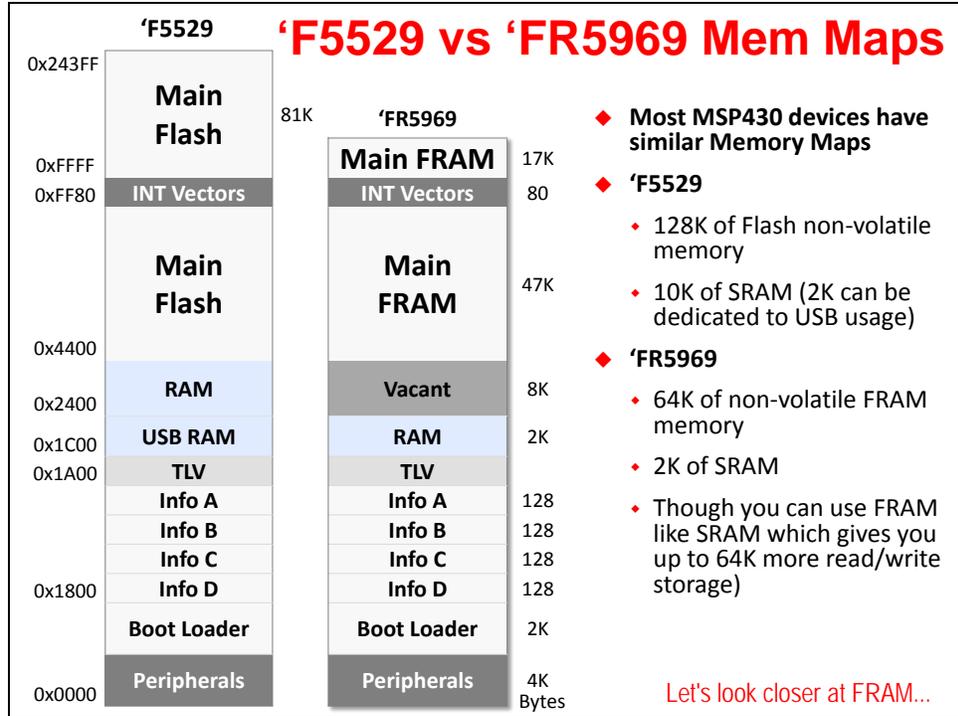
The 'F5529 has one aspect that is common among MSP devices which include the USB peripheral. These devices have an extra 2KB of RAM; this RAM is dedicated to the USB peripheral when it is in use, but available to your programs when the USB port is not being used. Please refer to the USB Developers Package documentation to learn more about how the USB protocol stack uses this RAM.

## TLV

Although not 'memory', the **Device Descriptors (TVL)** does appear within the memory map. This segment contains a tag-length-value (TLV) data structure that comprises a hierarchical description (or on older devices, flat file description) of information such as: the device ID, die revisions, firmware revisions, and other manufacturer and tool related information. Additionally, these descriptors may contain information about the available peripherals, their subtypes and addresses. This info may prove useful if building adaptive hardware drivers for operating systems. (Note that some of the Value Line devices may not contain all of this information; and, their factory supplied calibration data may reside in Info Memory A.)

## Comparing Memory Maps

Most MSP devices have fairly similar Memory Maps; the primary differences ends up coming down to how much memory a specific device contains. Please check the datasheet for the specific details on each device.



The devices shown here have one other major differentiating factor, the 'F5529 uses *Flash* technology while the 'FR5969 uses *FRAM* technology to store its non-volatile information. We briefly compare these two technologies in the next section, though you may want to refer to the *Non-Volatile Memory (Flash & FRAM)* chapter for more details.

## FRAM

Some of the latest MSP microcontroller devices from TI now use FRAM in place of Flash for their non-volatile memory storage. For example, you will find the Wolverine (FR58xx, FR59xx) devices utilize this new technology.

### FRAM: The Future of MCU Memory

- ◆ **Non-volatile, Reliable Storage**
  - ◆ Over 100 Trillion write/read cycles
  - ◆ Write Guarantee in case of power loss
- ◆ **Fast write times like SRAM**
  - ◆ ~50ns per byte or word
  - ◆ 1,000x faster than Flash/EEPROM
- ◆ **Low Power**
  - ◆ Only 1.5v to write & erase
  - ◆ >10-14v for Flash/EEPROM
- ◆ **Universal Memory**

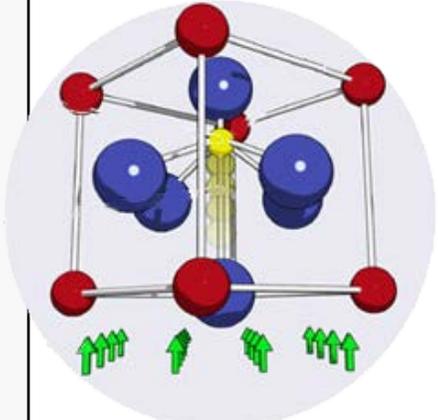


Photo: Ramtron Corporation

Memory Comparison

Actually, FRAM is not a brand new technology. It has been available in stand-alone memory chips for nearly a decade. It is quite new, though, to find it used within micros.

In brief, the MSP microcontroller FRAM provides some exciting new features in our MCUs:

- FRAM memory is a nonvolatile memory that reads and writes like standard SRAM
- It supports Byte or word write access
- A nearly limitless re-write capability – ‘we haven’t worn it out yet’
- Very fast write cycles – much faster than Flash or EEPROM
- Very low power – unlike Flash memory, it only takes 1.5V to write and erase FRAM (really ideal for low-power data logging applications)
- Error Correction Code with bit error correction, extended bit error detection and flag indicators
- Power control for disabling FRAM if it is not used – and due to non-volatile nature, it naturally does not lose its contents in the process of powering down

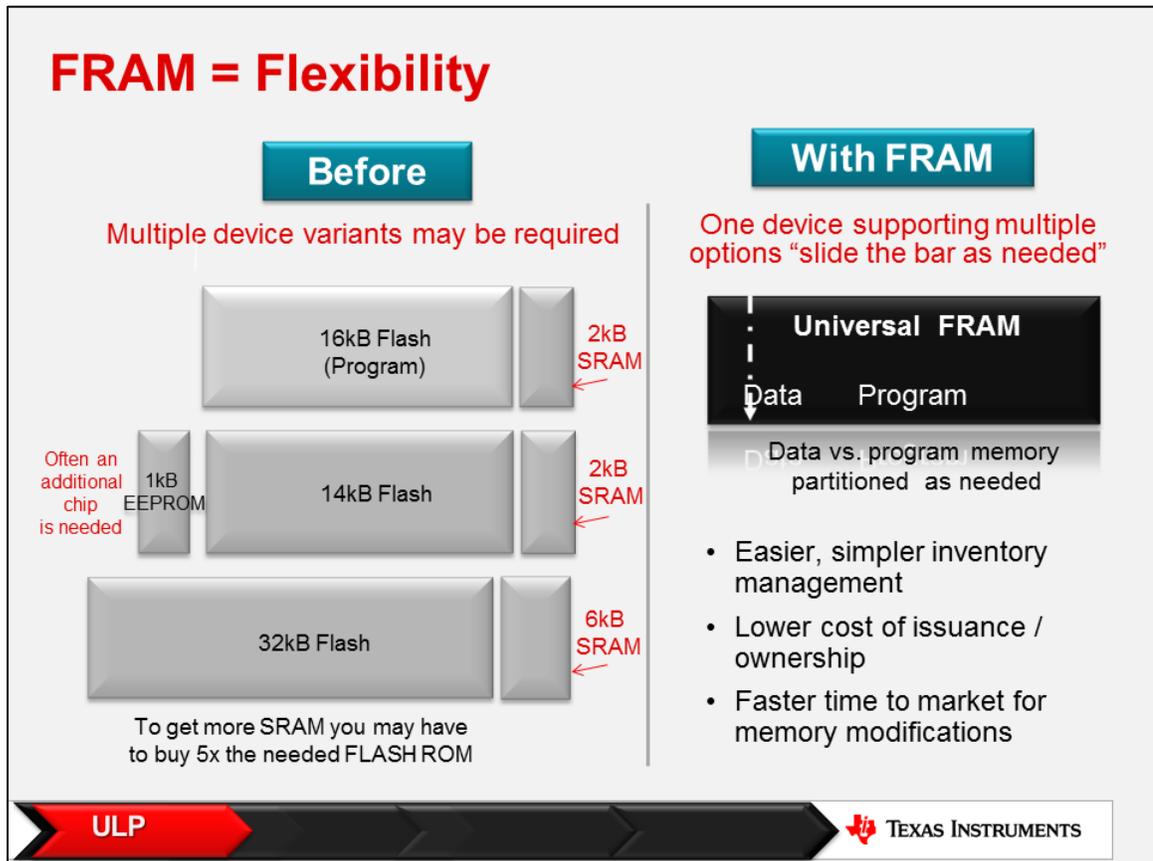
As stated above, FRAM can be read and written in a similar fashion to SRAM and needs no special requirements. This provides a big value in letting you choose how to use your memory; in other words, if your system needs “a little bit more RAM”, this can be accomplished by locating your data in FRAM.

The downside, of course, is that your program could be just as easily overwritten in the same fashion. (We shouldn't have code that writes to program addresses – but accidents occur.) To this end, the FRAM based devices provide a memory protection unit (MPU) that lets you create 1 to 3 segments of FRAM. Often, these segments are set for: Execute only, Read only, and Read/Write.

The other two caveats to FRAM are that reads are a bit slower than Flash and their density is not as great as we can build using flash technology. On the other hand, the benefits are an outstanding fit for many MSP microcontroller types of applications.

<b>FRAM MCU Delivers Max Benefits</b>				
	<b>FRAM</b>	<b>SRAM</b>	<b>EEPROM</b>	<b>Flash</b>
<b>Non-volatile</b> Retains data without power	<b>Yes</b>	<b>No</b>	<b>Yes</b>	<b>Yes</b>
<b>Write speeds</b>	<b>10 ms</b>	<b>&lt;10 ms</b>	<b>2 secs</b>	<b>1 sec</b>
<b>Average active Power</b> ( $\mu\text{A}/\text{MHz}$ )	<b>110</b>	<b>&lt;60</b>	<b>50mA+</b>	<b>230</b>
<b>Write endurance</b>	<b><math>10^{15}</math></b>	<b>Unlimited</b>	<b>100,000</b>	<b>10,000</b>
<b>Dynamic</b> Bit-wise programmable	<b>Yes</b>	<b>Yes</b>	<b>No</b>	<b>No</b>
<b>Unified memory</b> Flexible code/data partitioning	<b>Yes</b>	<b>No</b>	<b>No</b>	<b>No</b>

This graphic speaks to the earlier comment about the trade-offs between Flash and RAM. We have seen users who are forced into purchasing a larger, more expensive MCU just to get a little bit more RAM. The flexibility of FRAM allows your programs to use the non-volatile storage for things like variables and buffers. This flexibility often ends up lowering your overall system costs.



ULP

TEXAS INSTRUMENTS

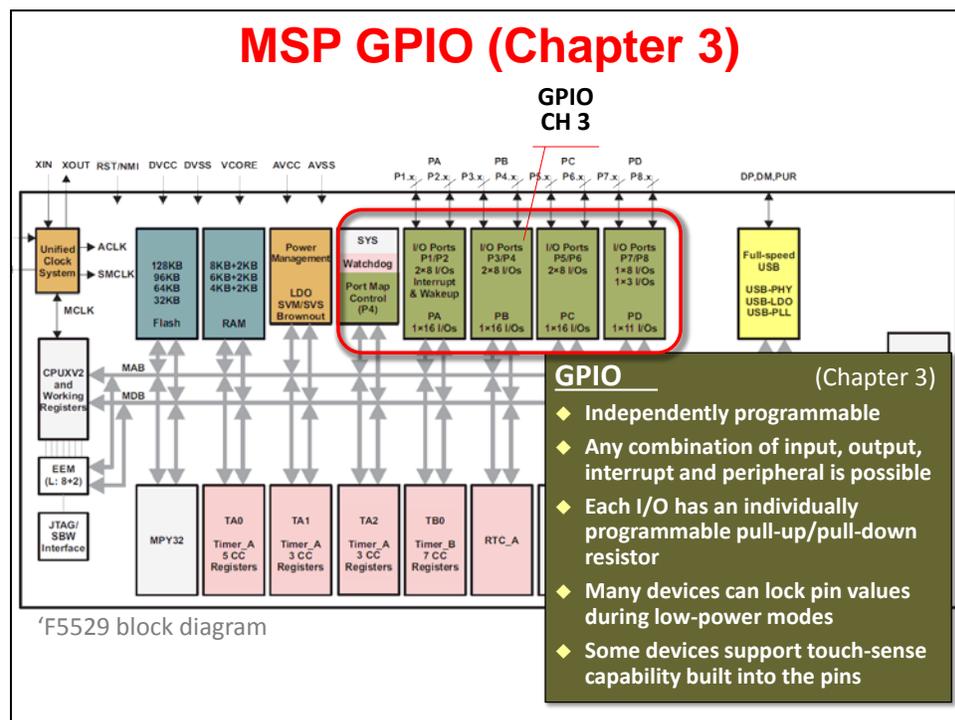
# MSP Peripherals

This section provides a high-level overview of the various categories of MSP peripherals.

## GPIO

MSP devices contain many I/O ports. The largest limitation is usually the package selection – a lesser pin-count package means less General Purpose bit I/O.

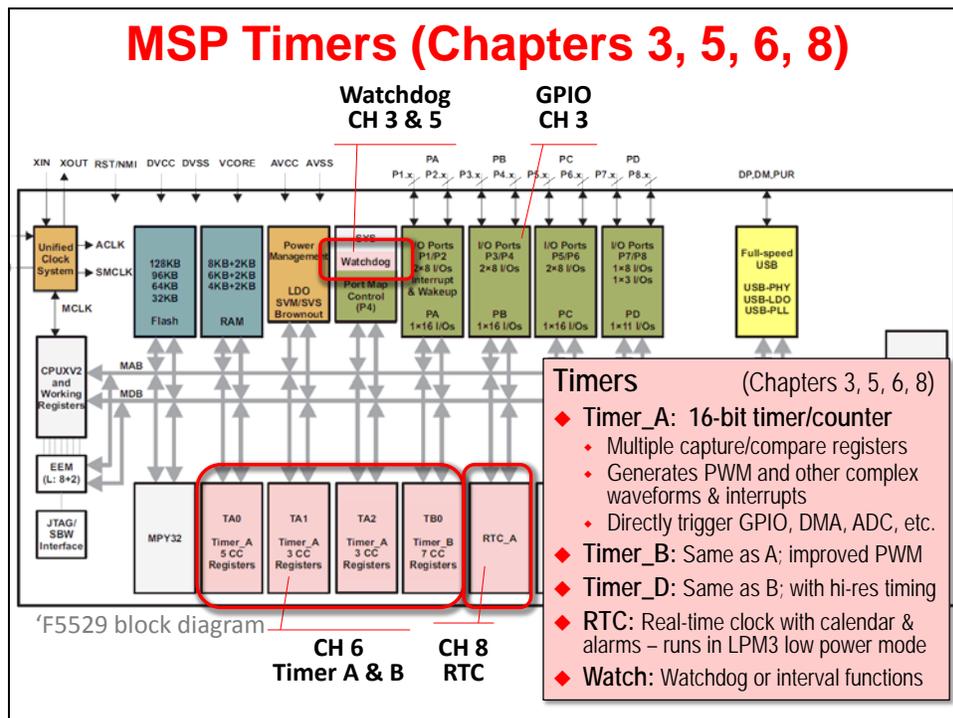
Like most current day microcontrollers, the pins on our devices are heavily multiplexed. That is, you often have one of several choices of signals that can be output to a given pin. The MSP microcontroller makes each signal independently programmable, which affords maximum flexibility.



Other handy GPIO features include:

- I/O ports 1 and 2 can generate interrupts to the CPU. (Some devices support interrupts on additional I/O ports.)
- Pull-up and Pull-down resistors are available as part of the I/O port, simplifying your board design.
- Many devices can lock the state of the pins when going into the lowest power modes, which again saves the effort, power, and cost of adding external transceivers to accomplish this purpose.
- Finally, many I/O ports include 'touch' circuitry. This additional circuitry makes it easy to implement capacitive touch based interfaces in your systems – all without having to add extra hardware.

## Timers



As stated earlier, timers are often thought of as the heartbeat of an embedded system. The MSP microcontroller contains a number of different timers that can assist you with different system needs.

**Timer\_A** (covered in detail in Chapter 6) is the original timer found across all MSP generations. And there is a reason for that, it is quite powerful, as well as flexible.

These 16-bit timers contain anywhere from 2 to 7 capture/compare registers (CCR). Each CCR can capture a time value when triggered (capture mode). Alternatively, each CCR could be used to generate an interrupt or signal (internal or external via a pin) when the timer's counter (TAR) matches the value in the CCR (compare mode). Oh, and each CCR is independently programmable – thus some could be used for capture while others for compare.

Using the CCR feature, it is easy to create a host of complex waveforms – for example, they could be used to generate PWM outputs. (Something we'll explore in Lab 6.)

**Timer\_B** is nearly similar to **Timer\_A**. It provides the ability to use the internal counter in 8/10/12 or 16-bit modes. This affords it a bit more flexibility. Additionally, double-buffered CCR registers, as well as the ability to put the timer outputs into high-impedance, provide a couple of additional advantages when driving H-bridges and such.

**Timer\_D** takes **Timer\_B** and adds a higher resolution capability. (BTW, we're not sure what happened to **Timer\_C**...)

**RTC** (real-time clock) peripherals not only provide a time base, but their calendar and alarm modes make them ideal for clock/calendar types of activities. More importantly, they have been designed to run with extremely low power. This means they can provide a heartbeat while the rest of your system is asleep.

**Watchdog** timers provide two different functions. In their namesake mode, they act as failsafe's for the system. If your code does not reset them before their counter reaches the end, they reset the system. This functionality is ALWAYS enabled at boot. You can also choose to use them as an interval timer.

# Clocking and Power Management

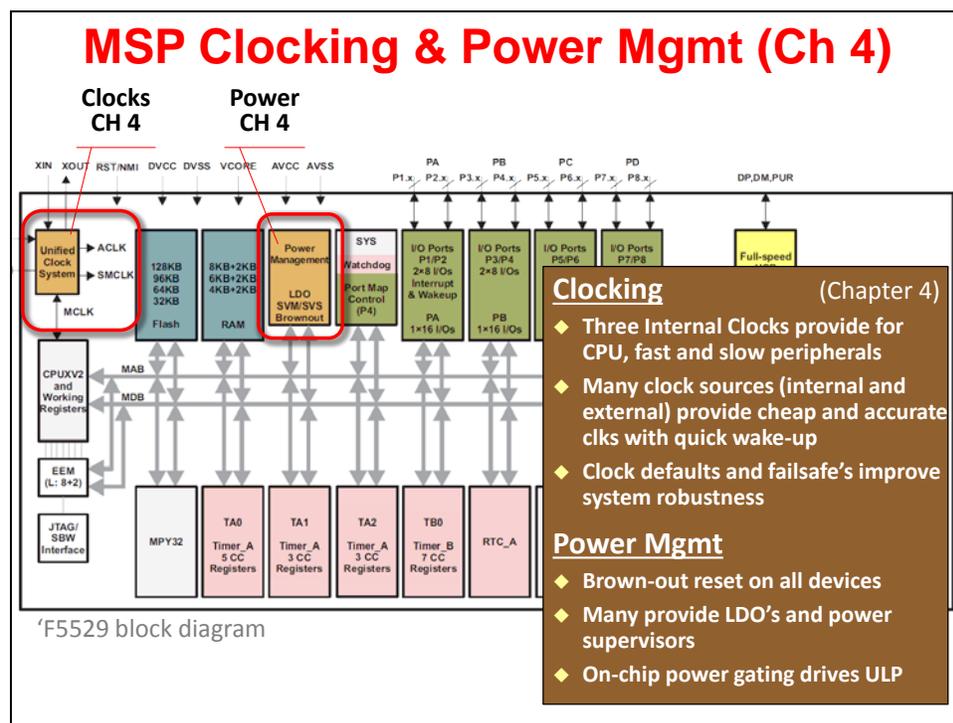
## MSP Clocks (Chapter 4)

The MSP devices provide a rich, robust set of clocking options.

Rich in that they provide a great number of on- and off-chip clock sources. Further, there are three internal clocks routed to the CPU and various peripherals. Why three? Simply, there's a clock for the CPU and two clocks for the peripherals - one fast and the other slow - with goal of providing the user a balance of performance and low power. Of course, some of the devices provide more clock choices than others.

Robust clocking in that there are defaults and failsafe's for all of the various clocks. These failsafe clocks choices can be particularly important for some applications. Imagine a crystal oscillator being forcibly removed from the board - or maybe just broken - when your end-product is accidentally damaged in use. It's nice to know there are internal alternatives that let your product continue working in a well-documented state.

Please turn to the Clocking chapter for further information.



## Power Management

Power is one of those features that every system needs but doesn't often get highlighted. All of the MSP devices provide some level of Power Management. On the most cost-sensitive, it might only be a Brown-Out Reset (BOR) peripheral - which makes sure there is enough power going to the device to assure proper, stable operation. The other notable point is that BOR was designed with extreme sensitivity to low-power system needs.

On other devices you'll find BOR plus an increasing set of power management peripherals. For example, the 'F5529 device adds an LDO (low dropout voltage regulator) which derives a steady CPU voltage from that applied to the device. (Normally, voltage regulation is handled by an extra device in your

system.) The 'F5529 also contains a sophisticated power supervisor to warn (i.e. interrupt) your system when the power is getting close to out-of-spec.

Power gating is another feature found on most of the MSP devices. The basic idea is that we want to power-down anything that is not needed.

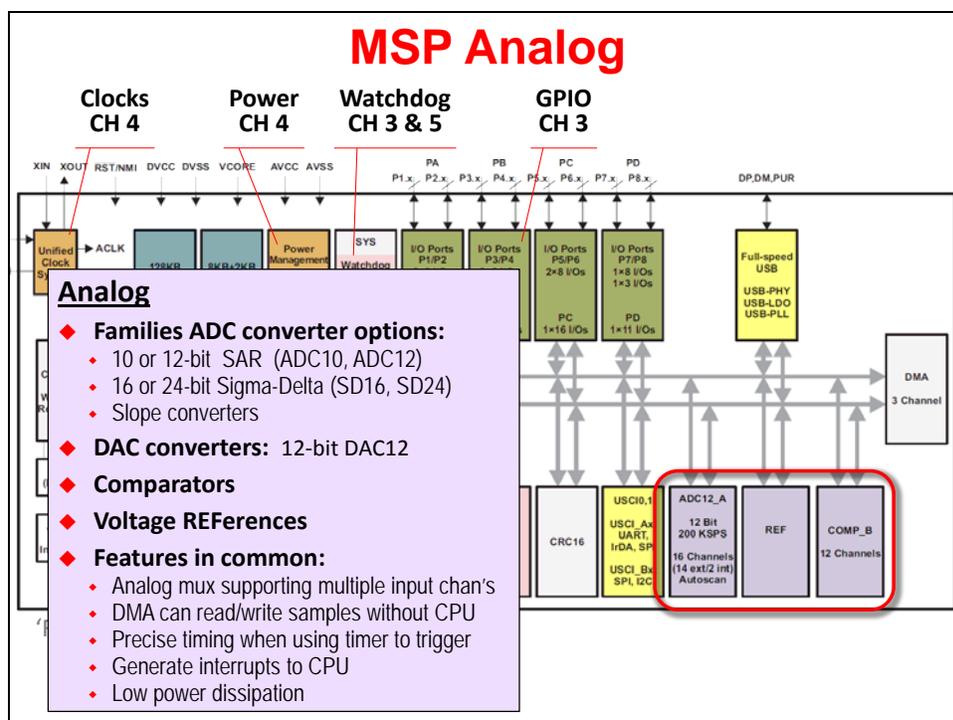
## Analog

Bringing high-quality analog components on-chip was a big selling point of the original MSP devices - and still is today. Besides providing high-quality analog, they've done it with a low-power footprint, too.

MSP analog peripherals cover a wide range of needs. At one end, you'll find most every device contains one or more analog comparators. These signal the processor when an analog input crosses a boundary. (Comparators are often used to build a "poor mans" analog to digital converter.)

In many systems, though, you will want an actual ADC (analog to digital) converter. The MSP microcontroller family provides a wide variety of options. In fact, some designers select their specific MSP microcontroller device based upon which type of converter they want to use.

Almost regardless of the type of analog component, they have a few key features in common. The ability to generate interrupts is fundamental. Also critical are the ability to trigger conversions based on timers; and couple that with using DMA's to transfer the results to memory sans CPU.



The following slide shows a couple of devices which really show off the MSP analog capabilities. The MSP430i2040 provides 4 sigma-delta converters into a low-cost SoC. The MSP430F67791 packs seven (7) sigma-delta converters, along with an additional 10-bit SAR analog to digital convertor.

## Sampling of MSP Analog

### MSP430i2040

- ❖ **4 Sigma-Delta AFE**  
1% accuracy for precise measurements with a 2000:1 dynamic range  $\Sigma\Delta$  converters
- ❖ **Low Cost SoC** – Targets low-end meters with minimal communications (memory) requirements
- ❖ **Internal DCO** – eliminates need for external crystal
- ❖ **Small packages** minimize pin count and cost
- ❖ **Temperature** - -40C to 105C

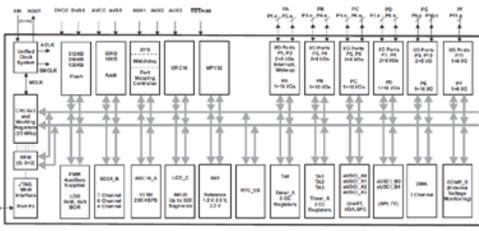
### MSP430i20xx

Temperatures: -40 °C to 105 °C

16-bit MCU 16 MHz	Memory	Power & Clocking
	Upto 39KB Flash Upto 2KB RAM 1KB Flash Information Memory	CLKIN DCO (External/Internal) LDO w/1.8V Regulated Core Supply Voltage
Analog	Comms Peripherals	Timers
4 x 24-bit Delta-Sigma Converters	1 x eUSCI_A, UART, I2C, SPI	SVM SVS POR BOR
System Modules	Debug	2 x 16-bit Timers w/ cap-comp regs Watchdog Timer
16 x 16 Multiplier CRC16 Data Checking Module	Real Time ATC Embedded Emulation Bootstrap Loader	
Packages		
28-pin TSSOP 32-pin QFP		

### MSP430F67791

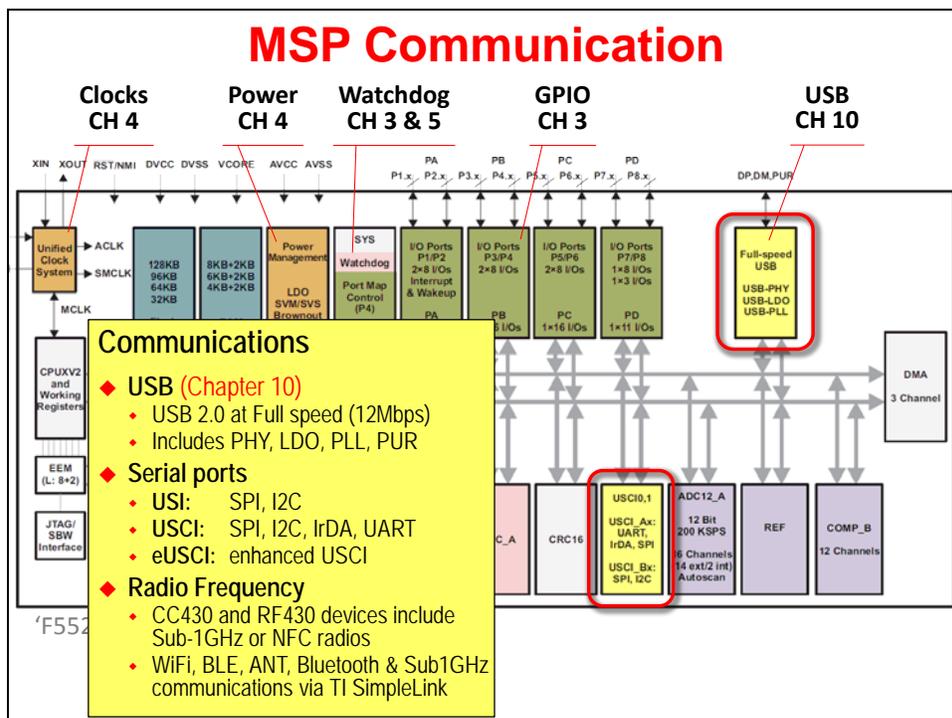
- ❖ **7 Independent Sigma-Delta ADC's** with Differential Inputs and Variable Gain
- ❖ **7 Channel 10-bit SAR ADC (200-kSPS)** Six Channels Plus Supply and Temperature Sensor Measurement
- ❖ **LCD Driver** With Contrast Control for up to 320 segments
- ❖ **Six Enhanced Communications Ports**
- ❖ **512 KB of Flash**
- ❖ **32 KB of SRAM**
- ❖ **MPY and CRC Accelerators**



We've seen folks choose these parts just to get access to their highly integrated analog capabilities. The MSP microcontroller CPU being a big bonus! It's like buying a stand-alone convertor and getting the CPU for free!

## Communications (Serial ports, USB, Radio)

We specifically chose the name "Communications" for this category, rather than the more common "Serial Communications". It's true that most of the communications ports utilize serial connections; this is due to the lower cost and power of using fewer pins. But, in the end, we didn't want to overlook the growing support for wireless communications.



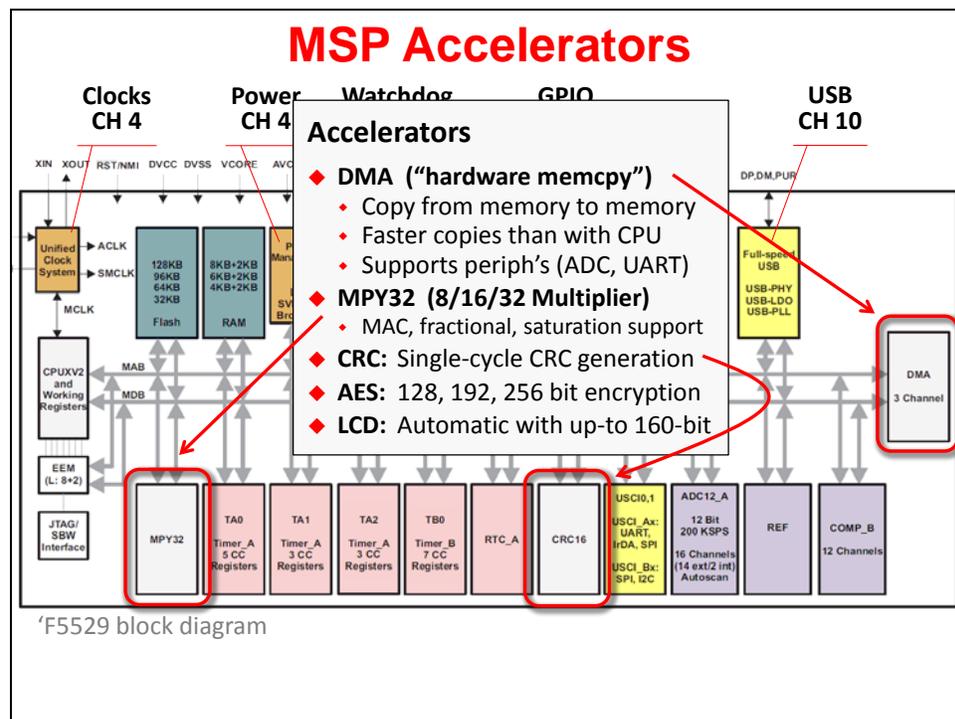
The additional of radios to some MSP devices makes them quite unique in the industry. Beyond that, TI has created wireless chips and modules that can be used from any MSP device. It's really telling when the cheapest Value Line MSP device can actually talk Wi-Fi using TI's CC3000 module. A similar story can be shown across TI's complete portfolio of wireless technologies. In the end, TI is enabling a very low-cost entry point into the "Internet of Things".

Let's not forget the various MSP serial ports. They are the workhorses of communications. There are a variety of serial modules, from UART, to SPI, to I2C.

## Hardware Accelerators

One question that is often asked, "Why would you put dedicated hardware accelerators onto low-cost, low-power processors?"

It's an interesting question ... with a very practical answer. If a specific functionality is required, accelerators are the most efficient implementation. Take for example, the CRC or AES modules; serial (and wireless) communications are often requiring these functions to make the data transmissions robust and secure. To implement these functions in software is possible, but would actually consume a lot more power. Further, the memory footprint for an algorithm (code and data) often ends up greater than the smaller footprint of the hardwired accelerator. Thus, where it makes sense, you'll see TI adding dedicated hardware modules.

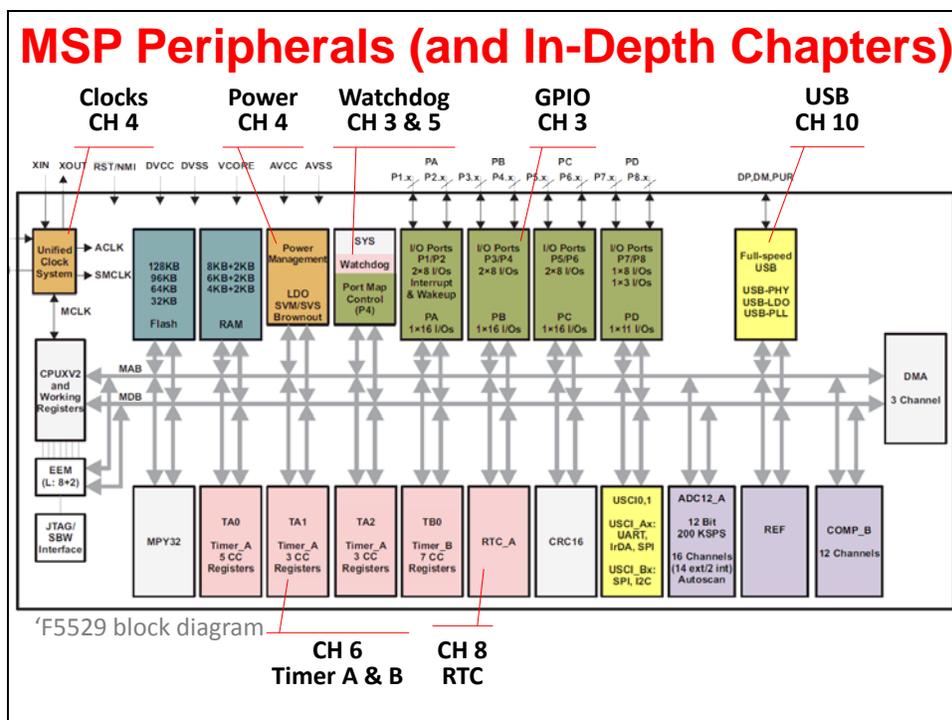


Another example is the multiplier. We can benefit from it without any programming effort, since the compiler automatically uses this hardware, when it's available.

With regards to the Direct Memory Access (DMA) peripheral, we caution you ... if you find yourself using `memcpy()` in your code, you should investigate how the DMA might save you time and power. It also should be utilized in your peripheral driver software whenever and wherever it's available.

## Summary

Many of the peripherals we've just outlined are covered - in detail - within their own chapters. Over time, we'll be adding more chapters to the course to cover additional peripherals.



The following comparison table has not been updated for the latest devices; even so, we included it as a quick comparison between some of the MSP generations.

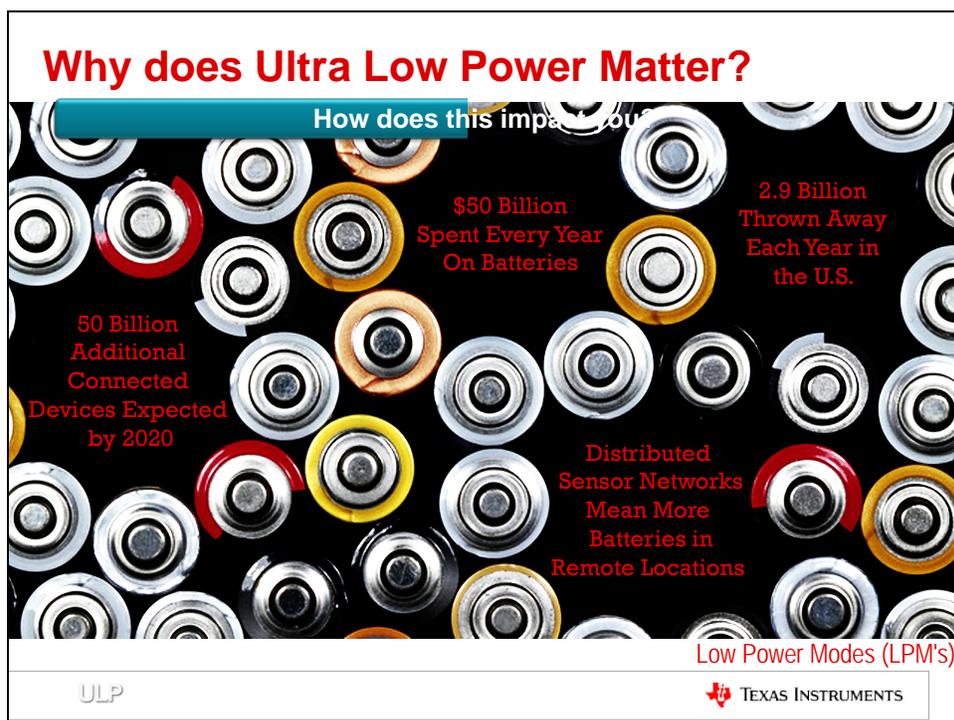
<b>MSP430 Peripheral Overview</b>			
1xx	2xx	4xx	5xx
Basic Clock System	Basic Clock System +	FLL, FLL+	Unified Clock System
Core voltage same as supply voltage (1.8-3.6V)	Core voltage same as supply voltage (1.8-3.6V)	Core voltage same as supply voltage (1.8-3.6V)	Programmable core voltage with integrated PMM (1.8-3.6V)
16-bit CPU	16-bit CPU, CPUX	16-bit CPU, CPUX	16-bit CPUXv2
GPIO	GPIO w/ pull-up and pull-down	GPIO, <u>LCD Controller</u>	GPIO w/pull-up and pull-down, <u>drive strength</u>
N/A	N/A	N/A	<u>CRC16</u>
Software RTC	Software RTC	Software RTC with Basic Timer, Basic Timer + RTC	<u>True 32-bit RTC</u> w/Alarms
USART	USCI, USI	USART, USCI	USCI, <u>USB</u> , <u>RF</u>
DMA up to 3-ch	DMA up to 3-ch	DMA up to 3-ch	DMA up to <u>8-ch</u>
MPY16	MPY16	MPY16, MPY32	MPY32
ADC10,12	ADC10,12, SD16	ADC12, SD16, OPA	ADC12_A
4-wire JTAG	4-wire JTAG, 2-wire Spy Bi-Wire (Some devices)	4-wire JTAG	4-wire JTAG, 2-wire Spy Bi-Wire

## ULP

Does Low Power matter? Our answer is a resounding YES!

Some end-products are only enabled by low-power operation. For example, a wrist watch that cannot make it through a single day would be of little value.

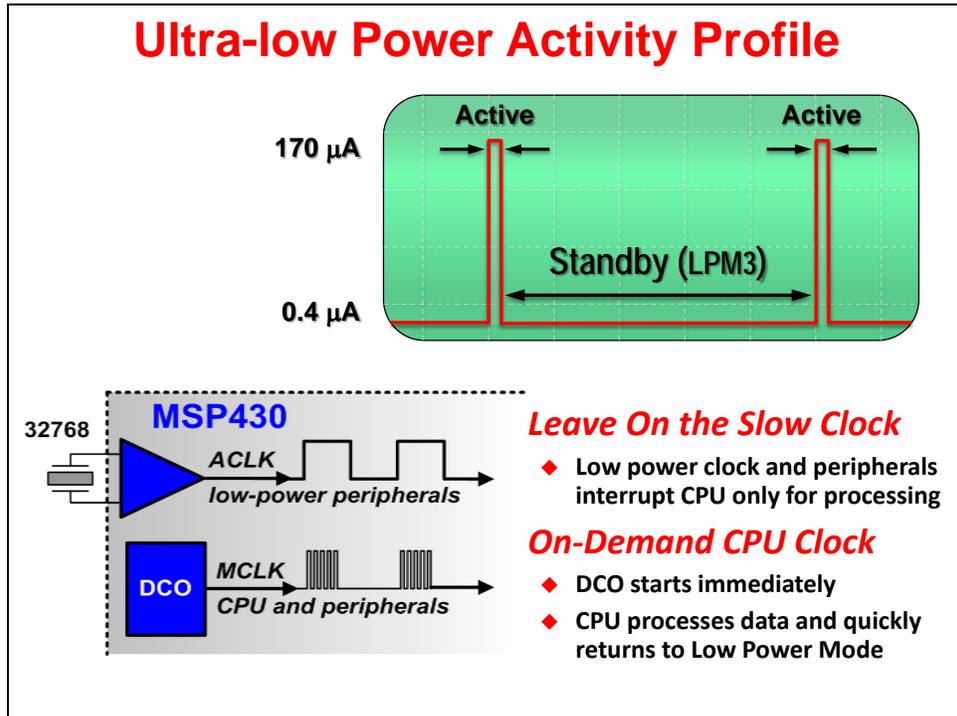
But even when the application does not demand low power, we think it still matters. The trend in electronics over the past few years has been, "Why consume power if you don't have to?" In fact, the MSP microcontroller has found many new applications in the last couple of years where end-users are demanding the reduction of 'phantom load', also known as 'vampire power'. This can be defined as the dissipation of power when electronic products are in standby mode (or even when switched off completely). The MSP is a perfect fit for systems trying to prevent these issues.



## Profile Your Activities

A fundamental precept of low-power systems is: turn on, do something, then turn off.

The following diagram is a good example of this. One of the low-power modes lets you put the fast components of the system to sleep, while retaining the slow clock running a RTC. Then, as needed, the system wakes up, performs one or more tasks, then goes back into low-power mode.



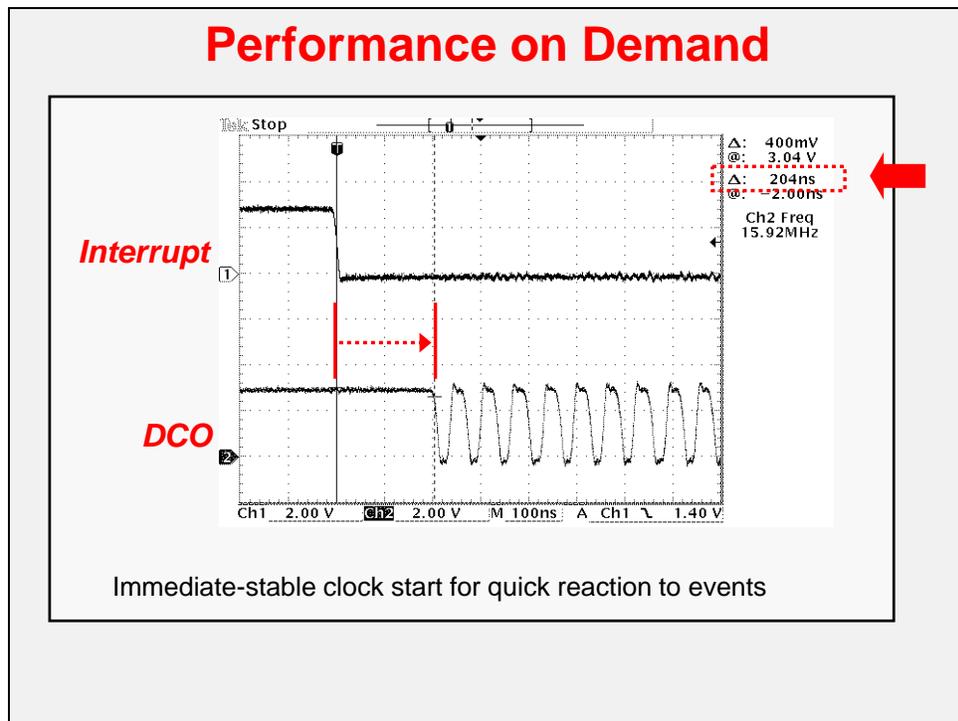
The MSP microcontroller supports this sleep/wake/sleep profile quite well, by providing a variety of low-power modes (LPM). The following chart is an example of the LPM's found on various MSP microcontroller devices, showing which resources are powered down by LP mode. It also broadly indicates what it takes to wake up from a given LPM. (In general, LPM0 and LPM3 are very popular modes.)

### Low-Power Modes

Operating Mode	CPU (MCLK)	SMCLK	ACLK	RAM Retention	BOR	Self Wakeup	Interrupt Sources
Active	☒	☒	☒	☒	☒		
LPM0		☒	☒	☒	☒	☒	Timers, ADC, DMA, WDT, I/O, External Interrupt, COMP, Serial, RTC, other
LPM1		☒	☒	☒	☒	☒	
LPM2			☒	☒	☒	☒	
LPM3			☒	☒	☒	☒	
LPM3.5					☒	☒	External Interrupt, RTC
LPM4				☒	☒		External Interrupt
LPM4.5					☒		External Interrupt

LPM is great, but waking up...

Almost as important is the 430's ability to wake up quickly from a sleep mode as is demonstrated on the next slide. The DCO (digitally controlled oscillator) is one of the on-chip, high-performance clocks available to the MSP. The graphic is powerful statement, showing how quickly the clocks and system can be up-and-running after receiving an interrupt.



This slide shows some of the quantitative data for different LPM's across a few different devices. Please, keep in mind that you should always design your system by referencing the datasheet, but this slide does give us a good comparison between the various MSP generations.

### MSP430™ Series Comparison

Mode		G2xx	F5xx	FR57xx	FR58xx FR59xx
Performance (max)		16 MHz	25 MHz	24 MHz (FRAM at 8MHz)	16 MHz (FRAM at 8MHz)
Flex Unified Memory		No	No	FRAM (16K)	FRAM (64K)
Active	AM	230 µA (1MHz)	180 µA/MHz	100 µA/MHz	<100 µA/MHz
Standby RTC	LPM3	0.7 µA	1.9 µA	6.3 µA	0.7 µA
	LPM3.5		2.1 µA	1.5 µA	0.4 µA
Off	LPM4	0.1 µA	1.1 µA	5.9 µA	0.6 µA
	LPM4.5		0.2 µA	0.3 µA	0.1 µA
Wake-up from	Standby	1.5 µs	3.5 µs or 150 µs	78 µs	<10 µs
	Off	-	2000 µs	310 µs	150 µs

 TEXAS INSTRUMENTS

Much of designing for low-power is common sense; e.g. turn it off when you're not using it. The following slide provides a good set of guidelines (or principles) to use when developing our application.

### Principles For ULP Applications

- ◆ Maximize the time in LPM3
- ◆ Use interrupts to control program flow
- ◆ Replace software with peripherals
- ◆ Power manage external devices
- ◆ Configure unused pins properly
- ◆ Efficient code makes a difference
- ◆ Even wall powered devices can be "greener"
- ◆ **Every unnecessary instruction executed is a portion of the battery wasted that will never return**
- ◆ **Use ULP Advisor to help you minimize power in your system**



 **ULP Advisor** | MSP430™ Ultra-Low Power MCUs

**ULP Advisor - Rule Table**

- ULP 1.1 Ensure LPM usage
- ULP 2.1 Leverage timer module for delay loops
- ULP 3.1 Use ISRs instead of flag polling
- ULP 4.1 Terminate unused GPIOs
- ULP 5.1 Avoid processing-intensive operations: modulo, divide.
- ULP 5.2 Avoid processing-intensive operations: floating point
- ULP 5.3 Avoid processing-intensive operations: (s)printf()
- ULP 6.1 Avoid multiplication on devices without hardware multiplier
- ULP 7.1 Use local instead of global variables where possible
- ULP 8.1 Use "static" & "const" modifiers for local variables
- ULP 9.1 Use pass by reference for large variables
- ULP 10.1 Minimize function callings from within ISRs
- ULP 11.1 Use lower bits for loop program control flow
- ULP 11.2 Use lower bits for port bit-banging
- ULP 12.1 Use DMA for large memory() calls
- ULP 12.1b Use DMA for potentially large memory() calls
- ULP 12.2 Use DMA for repetitive transfer
- ULP 13.1 Count down in loops
- ULP 14.1 Use unsigned int for indexing variables
- ULP 15.1 Use bit-masks instead of bit-fields

Many of these guidelines have been distilled into a static code analysis tool that is part of the TI (and IAR) compiler. This tool can help us learn what techniques to apply - or for the more experienced, help us not overlook something we already know.

## ULP Advisor™ Software: Turning MCU developers into Ultra-Low-Power experts

**ULP Advisor analyzes all MSP430 C code line-by-line.**

- Supports all MSP430 devices and can benefit any application
- Checks all code within a project at build time
- Enabled by default
- Parses code line-by-line



**Checks against a thorough Ultra-Low-Power checklist.**

- List of 15 Ultra-Low-Power best practices
- Compilation of ULP tips & tricks from the well-known to the more obscure
- Combines decades of MSP430 & Ultra-Low-power development experience

- ULP 1.1 Ensure LPM usage
- ULP 2.1 Leverage timer module for delay loops
- ULP 3.1 Use ISRs instead of flag polling
- ULP 4.1 Terminate unused GPIOs
- ULP 5.1 Avoid processing-intensive modulo & d
- ULP 5.2 Avoid processing-intensive floating poin
- ULP 5.3 Avoid processing-intensive (s)printf()
- ULP 6.1 Avoid multiplication when HW multiplier
- ULP 7.1 Use local instead of global variables wh

**Highlights areas of improvement within code.**

- Identify key areas of improvement
- Presented as a “remark” within “Problems” window
- Includes a link to more information





# Community / Resources

## Wiki

The TI Embedded Processor's wiki provides a wealth of information. Highlighted below you'll find the MSP430 and TTO (Technical Training Organization) links found on the main TI wiki page. Of course, most anything else you might be looking for can be easily found from the Google search box, right under the "Main Page" title.

**TTO Workshops: [processors.wiki.ti.com](http://processors.wiki.ti.com)**

The screenshot shows the Texas Instruments Wiki Main Page. The page is titled "Main Page" and features a navigation sidebar on the left. The main content area is divided into several sections:

- Microcontrollers (MCU):** This section lists various microcontroller families, including MSP430, C2000, and MSP430. The "MSP430" link is highlighted with a red circle.
- Development Tools:** This section lists tools such as Code Composer Studio, Code Generation Tools, and XDS Debug Probes.
- Embedded Operating Systems:** This section lists operating systems like Linux, TI-RTOS, and Android.
- Key Links:** This section lists links to "Getting Started Guides" and "Training Homepage". The "Training Homepage" link is highlighted with a blue circle.

A blue arrow points from the "MSP430" link in the "Microcontrollers (MCU)" section to the "Training Homepage" link in the "Key Links" section.

From the TTO wiki page you'll find a link to this workshop. You most likely already found this page when following our download/installation instructions to get ready for the workshop. You may also want to return here often to access updates to [these workshop materials](#).

**This Workshop**

### Hands-On Training for TI Embedded Processors

Hands-On Training for TI Embedded Processors

TI's Technical Training Organization (TTO) conducts hands-on training for TI embedded processors at various sites around the world. This training is organized by specific processor families. You can also enroll in a live workshop using the links below.

#### Workshop Descriptions and Materials

[MSP430™ 16-bit Ultra-Low-Power MCII Training](#)

[Getting Started with the MSP430™ LaunchPad Workshop - online videos provided](#)

[MSP430™ 4xx One-Day Workshop](#)

[MSP430™ 5xx One-Day Workshop - online videos provided](#)

[MSP430™ FRAM Training Workshop](#)

---

#### Getting Started with the MSP430 LaunchPad Workshop

**MSP430 Workshop**  
Version 3.21  
July 2014  
Now supports production version of **MSP430FR5969 FRAM Launchpad!**

**Updated Workshop Features:**

- Adding support for newly updated FR5969 FRAM Launchpad
- Still supports the F5529 USB Launchpad
- New "Low Power Optimization" chapter that includes "Energy Trace"
- New "Real-Time Clock" (RTC) chapter
- Updated to the latest version of MSP430ware DriverLib
- Version 3.21 brings an improved Flash/FRAM lab exercise (Chapter 9)

**Contents**

MSP-EXP430FR5969 (FRAM)      MSP-EXP430F5529 (USB)

**Supported MSP430 Launchpad's**

## Forums

There are a wide ranging set of user-to-user forums. Check them out, when you have a ???

## Engineer-2-Engineer Forums

TEXAS INSTRUMENTS    Products    Applications    Tools & Software    Support & Community    Sample & Buy    About TI

**TI E2E™ Community**    engineer to engineer, solving problems    Join | Sign In with my.TI Login

Support Forums    Blogs    Groups    Videos    简体中文    Search Community

TI Home > TI E2E Community

**Find out if your question has already been answered**

Search through 1,055,110 questions and answers in TI E2E Community    Advanced Search >

Choose a support forum to post a new question

ARM®-based Processors	Amplifiers	DLP® & MEMS	Applications
Digital Signal Processors	Broadband RF/IF & Digital Radio	Interface	Tools & Software
Microcontrollers	Clocks & Timers	Logic	Wireless Connectivity
OMAP™ Applications Processors	Data Converters	Power Management	See all support forums here >

**Recent Forum Activity**    **TI E2E Top Contributors**

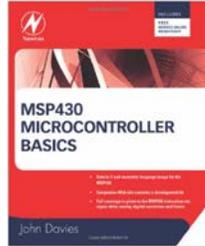
swati arora replied to writing a simple application using cc2540 in Low Power RF Bluetooth® Low Energy & ANT Forum.    Top Contributors    Top TI Contributors

<http://e2e.ti.com>

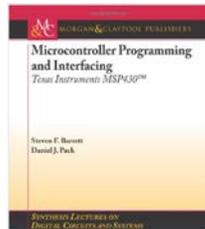
## References

There are many great references for learning more about the MSP microcontroller. Here's two of them that are favored by a number of us in TI's field applications.

### Further Reading...

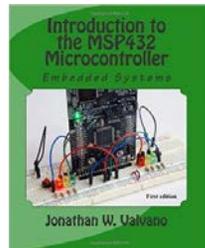


***MSP430 Microcontroller Basics*** by John H. Davies,  
(ISBN-10 0750682760) [Link](#)



***Microcontroller Programming and Interfacing: Texas Instruments MSP430 (Synthesis Lectures on Digital Circuits and Systems)***  
by Steven Barrett and Daniel Pack ,  
(ISBN-10 0750682760) [Link](#)

### Further Reading, 432...



***Embedded Systems: Introduction to the MSP432 Microcontroller (Volume 1)***

by Jonathan W. Valvano  
(ISBN-13: 978-1512185676) [Link](#)

## LaunchPads

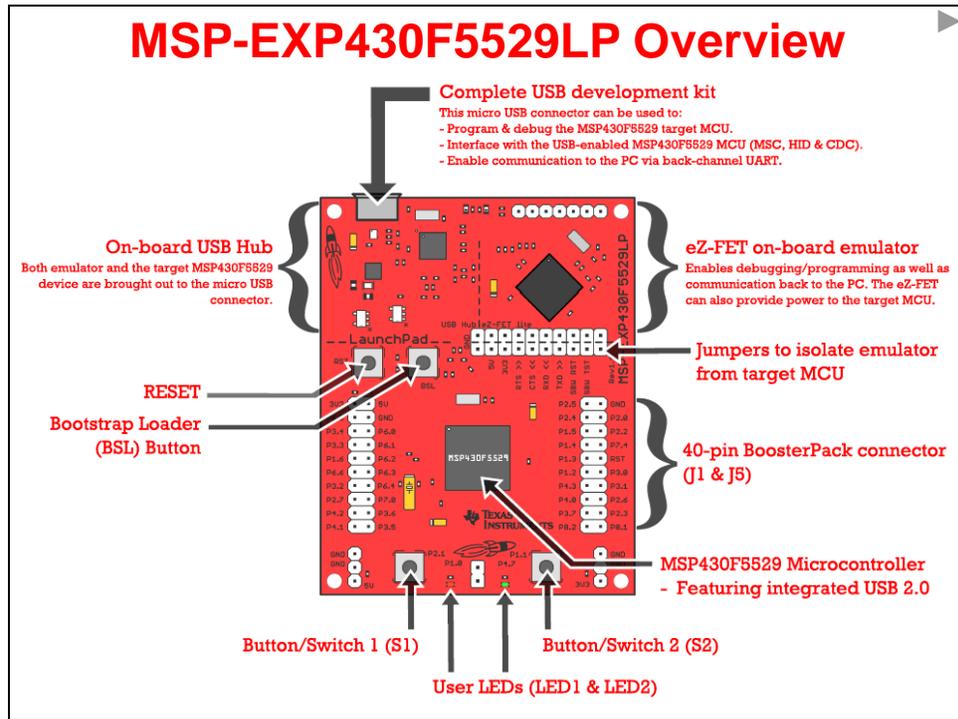
### MSP-EXP430F5529LP LaunchPad

The MSP430F5529 LaunchPad is a powerful, low-cost evaluation (and development) tool.

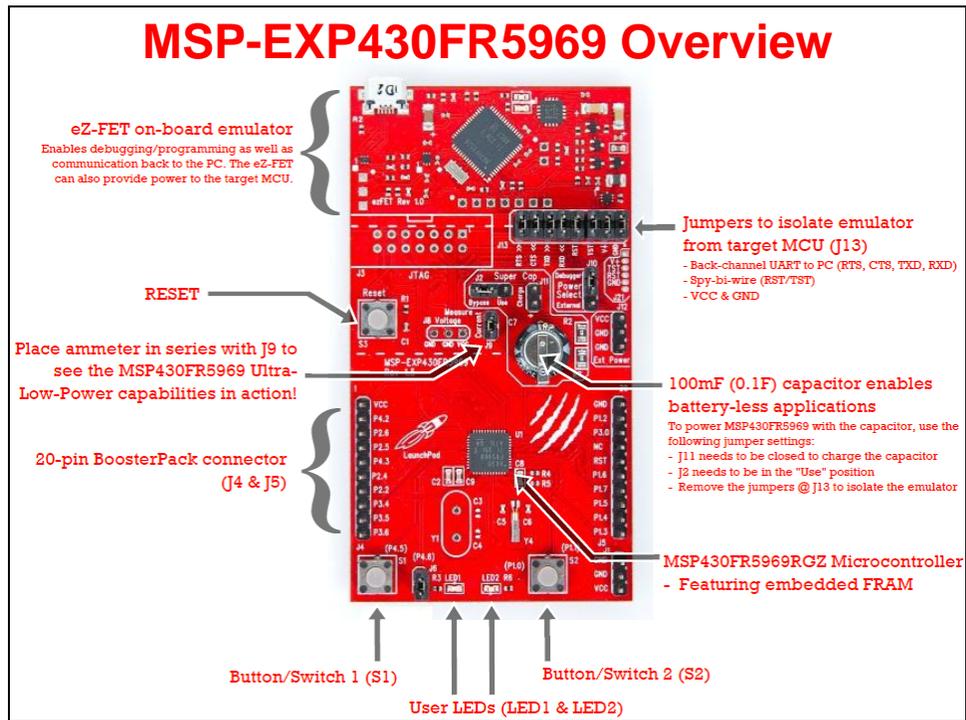


As the diagram shows, the board is really divided into two halves. The top portion (above the ----- line) is an open-source emulator (called eZ-FET lite). This connects our 'target' MSP microcontroller to a PC running a debugging tool, such as Code Composer Studio. You can isolate the emulator from the 'target' processor by pulling the appropriate jumpers (that straddle the dashed line).

The lower portion of the board provides the target of our application programming. There are LED's, pushbuttons, and pins we can use to let our programs interact with the 'real world'.



## MSP-EXP430FR5969 LaunchPad



## MSP-EXP430FR4133 LaunchPad



## MSP-EXP430FR6989 LaunchPad



## MSP-EXP432P401R LaunchPad



**Notes:**

# Introductory Lab Exercise

---

## Introduction

The first lab exercise in this workshop introduces you to the Launchpad you selected. Each one has a unique demonstration preloaded into the non-volatile memory of its MSP microcontroller. (Often times, this preloaded demo is called the *Out-of-the-Box* demo).

**Hint:** Since future lab exercises will over-write the original program, you will see how to restore the original Out-of-the-Box demo in Labs 2c/2d.

## Lab 1: Out-of-Box Demo

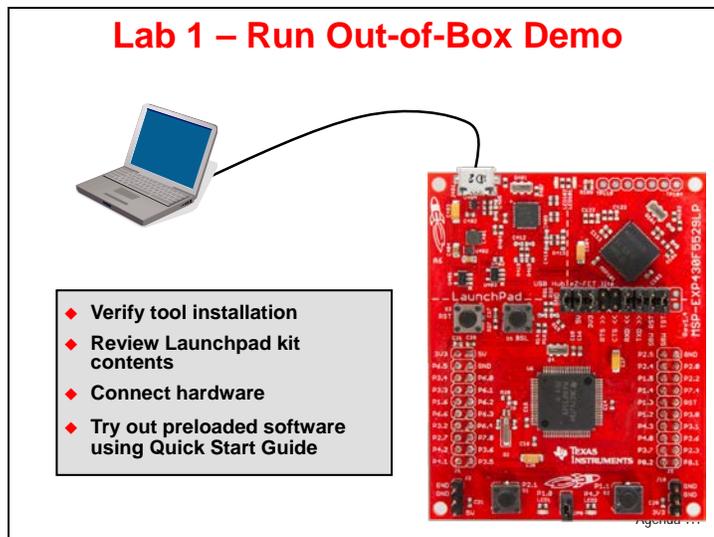
<b>Intro to the MSP Microcontrollers .....</b>	<b>1-47</b>
<i>MSP-EXP430F5529LP User Experience.....</i>	<i>1-48</i>
Examine the LaunchPad Kit Contents .....	1-48
<i>MSP-EXP430FR4133 LaunchPad OOB.....</i>	<i>1-50</i>
<i>MSP-EXP430FR5969 LaunchPad OOB.....</i>	<i>1-51</i>
First Steps – Out-of-Box Experience .....	1-51
('FR5969) Extra Credit .....	1-56
<i>MSP-EXP430FR6989 LaunchPad OOB.....</i>	<i>1-57</i>
<i>MSP-EXP432P401R LaunchPad OOB.....</i>	<i>1-58</i>

## MSP-EXP430F5529LP User Experience

For other LaunchPads, jump to the appropriate section: [F5529](#), [FR4133](#), [FR5969](#), [FR6989](#), [MSP432](#)

This lab simply gives us an opportunity to pull the board out of the box and make sure it runs properly. The board arrives with a USB keyboard/memory application burned into the flash memory on the 'F5529.

You can either follow the quick start directions on the card included with the Launchpad, or follow the directions here. We re-created the directions since some folks have a tough time reading the small print of the quick start card.



### Examine the LaunchPad Kit Contents

#### 1. Open up your MSP430F5529 LaunchPad box. You should find the following:

- The MSP-EXP430F5529LP LaunchPad Board
- USB cable (A-male to micro-B-male)
- “Meet the MSP430F5529 Launchpad Evaluation Kit” card

#### 2. Initial Board Set-Up

Using the included USB cable, connect the USB emulation connector on your evaluation board to a free USB port on your PC.

A PC's USB port is capable of sourcing up to 500 mA for each attached device, which is sufficient for the evaluation board. If connecting the board through a USB hub, it must usually be a powered hub. The drivers should install automatically.

#### 3. Run the User Experience Application

Your LaunchPad Board came pre-programmed with a User Experience application. This software enumerates as a composite USB device.

- HID (Human Interface device): an emulated keyboard
- MSC (Mass Storage class): an emulated hard drive with FAT volume

The contents of the hard drive can be viewed with a file browser such as Windows Explorer.

#### 4. View the contents of the emulated hard drive

Open Windows Explorer and browse to the emulated hard drive. You should see four files there:

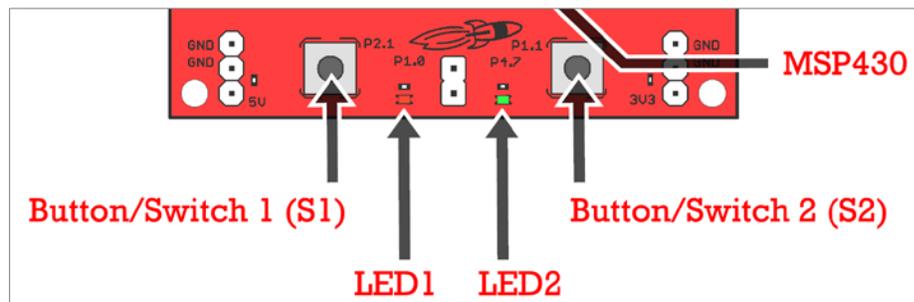
- **Button1.txt** – the contents of this file are "typed out" to the PC, using the emulated keyboard when you press button S1
- **Button2.txt** – the contents of this file are "typed out" to the PC, using the emulated keyboard when you press button S2
- **MSP430 USB LaunchPad.url** – when you double-click, your browser launches the MSP-EXP430F5529LP home page
- **README.txt** – a text file that describes this example

#### 5. Use S1 and S2 buttons to send ASCII strings to the PC

The LaunchPad's buttons S1 and S2 can be used to send ASCII strings to the PC as if they came from a keyboard. These strings that are sent are stored in the files Button1.txt and Button2.txt, respectively; and these files can be modified to change the strings. The text string is limited to 2048 characters, so even though you can make the file contents longer, be aware that the string will be truncated to 2048.

Open Notepad. In the start menu, type "Run", then type "Notepad"

To send the strings to Notepad, press S1.



What do you see? \_\_\_\_\_

Now press S2. What happens now? \_\_\_\_\_

The default ASCII strings stored in the two text files are:

- **Button1.txt:** "Hello world"
- **Button2.txt:** an ASCII-art picture of the LaunchPad rocket

For the rocket picture, please note that the display can be affected by settings of the application receiving the typed characters. On Windows, the basic Notepad.exe is recommended.

---

**Note:** If you have an older version of the 'F5529 Launchpad (prior to "Revision 1.5), then your board must enumerate with a USB host before it can receive power. This means USB batteries – which do not contain a USB host – cannot be used as a power source.

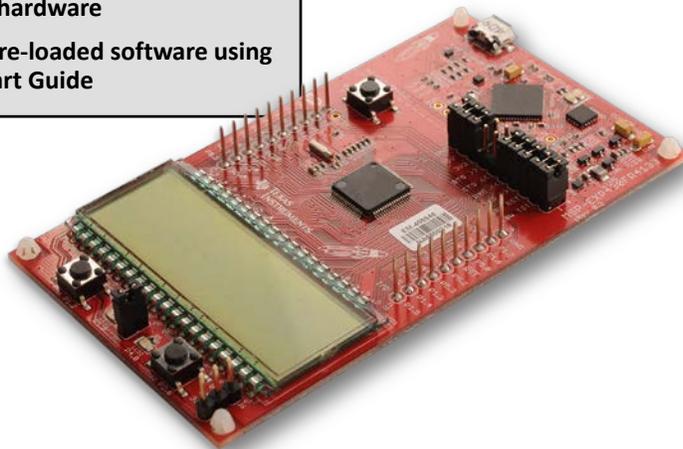
---

# MSP-EXP430FR4133 LaunchPad OOB

For other LaunchPads, jump to the appropriate section: [F5529](#), [FR4133](#), [FR5969](#), [FR6989](#), [MSP432](#)

## Lab 1c – MSP430FR4133 Launchpad

- ◆ Verify tool installation
- ◆ Review Launchpad kit contents
- ◆ Connect hardware
- ◆ Try out pre-loaded software using Quick Start Guide



## Out-of-box Demo

### 1. Connecting to the computer

Connect the LaunchPad using the included USB cable to a computer. A green power LED should illuminate. For proper operation, drivers are needed.

### 2. Running the Out-of-box Demo

When connected to your computer, the LaunchPad will power up and display a greeting message on the LCD. Press and hold the S1 and S2 buttons simultaneously to select a new mode.

#### Stopwatch Mode

This mode provides a simple stopwatch application. It supports split time, where the display freezes while the stopwatch continues running in the background.

##### Timer Stopped:

- S1 - Start time
- S2 - Reset time

##### Timer Running:

- S1 - Stop time
- S2 - Split time (lap time)

#### Temperature Mode

This mode provides a simple thermometer application. Using the on-chip temperature sensor, the temperature is displayed on the LCD.

- S1 - Pause current temperature
- S2 - Toggle temperature between °F/C

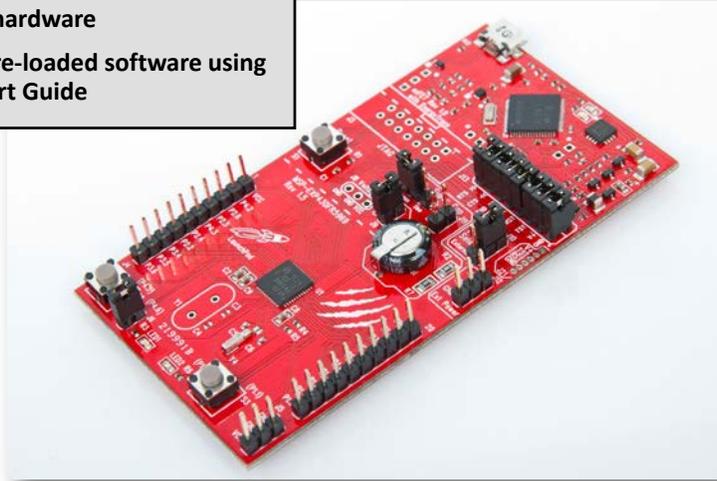
These steps were taken from the MSP-EXP430FR4133 LaunchPad™ Quick Start Guide ( [slau594.pdf](#) )

# MSP-EXP430FR5969 LaunchPad OOB

For other LaunchPads, jump to the appropriate section: [F5529](#), [FR4133](#), [FR5969](#), [FR6989](#), [MSP432](#)

## Lab 1 – MSP430FR5969 Launchpad

- ◆ Verify tool installation
- ◆ Review Launchpad kit contents
- ◆ Connect hardware
- ◆ Try out pre-loaded software using Quick Start Guide



## First Steps – Out-of-Box Experience

These steps were taken from Section 1.4 and 3.0 of the *MSP-EXP430FR5969 LaunchPad™ User's Guide* ([slau535a.pdf](#)).

An easy way to get familiar with the EVM is by using its pre-programmed out-of-box demo code, which demonstrates some key features of the MSP-EXP430FR5969 LaunchPad.

The out-of-box demo showcases MSP430FR5969's ultra-low power FRAM by utilizing the device's internal temperature sensor while running only off of the on-board Super Capacitor.

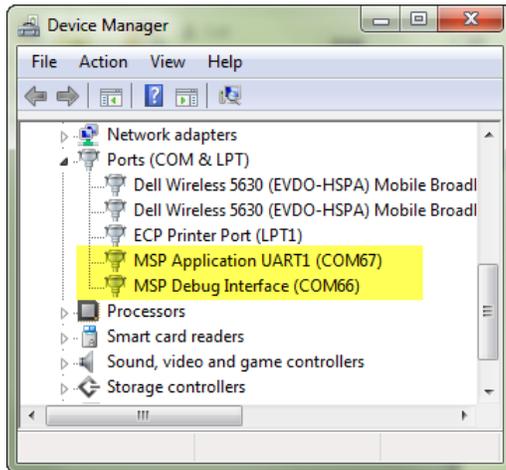
### 1. First step is to connect the LaunchPad to your computer using the included Micro-USB cable.

The RED and GREEN LEDs near the bottom of the LaunchPad toggle a few times to indicate the preprogrammed out-of-box demo is running.

After the LEDs toggle, the MSP430FR5969 CPU enters low-power mode 3 and waits for commands to come from the PC GUI via the backchannel UART. (A backchannel UART is the name given the UART to USB connection where the UART signals on the MSP430 are turned into a USB CDC class protocol by the MSP430 emulator.)

The Out-of-Box GUI is required to connect to the serial port that the LaunchPad's UART communication uses. But, to use the GUI we need to know which COM port our Launchpad was assigned to by Windows.

**2. Open Windows Device Manager and find the two COM ports assigned to the MSP430 Launchpad.**



Write down the two ports listed on your computer.

MSP Application UART1: \_\_\_\_\_

MSP Debug Interface: \_\_\_\_\_

### 3. Start the out-of-box demo GUI.

Using the out-of-box demo GUI, the user can place the LaunchPad into two different modes.

- **Live Temperature Mode**

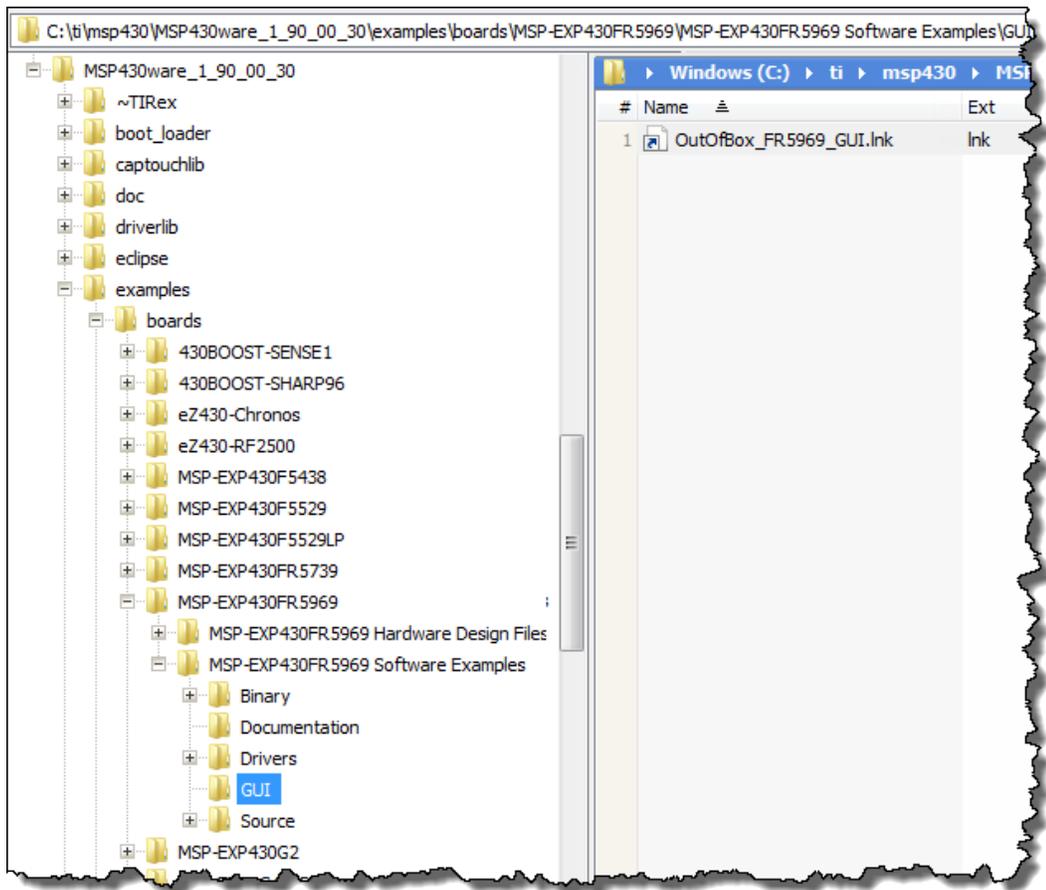
This mode provides live temperature data streaming to the PC GUI. The user is able to influence the temperature of the device and see the changes on the GUI.

- **FRAM Logging Mode**

This mode shows the FRAM data logging capabilities of the MSP430FR5969. After starting this mode, the LaunchPad will wake up every five seconds from sleep mode (indicated by LED blink) to log both temperature and input voltage values. After reconnecting to the GUI, these values can be uploaded and graphed in the GUI.

The easiest way to **start the GUI** is to double-click the link found in the MSP430ware library folder.

```
C:\ti\msp430\MSP430ware_1_97_00_47\examples\boards\MSP-EXP430FR5969\MSP-EXP430FR5969 Software Examples\GUI\OutOfBox_FR5969_GUI.lnk
```



The Out-of-Box example and GUI are included in the latest version of MSP430ware (as we mentioned earlier) as well as the MSP-EXP430FR5969 Software Examples download package (SLAC645).

Here's a snapshot of the GUI.



**4. Connect the GUI to your Launchpad.**

To get it to display data, we first need to connect with it.

Select the "MSP Application UART1" communications port from the list and click the *Connect* button.



5. **Once connected, to enter the live temperature mode, click the "Start" button below "Live Temp Mode" in the GUI's *Application Controls* panel.**

At this point, you should see the graph of temperature data populating the *Incoming Data* panel.

What is 'FR5969 Doing?

It sets up its 12-bit ADC for sampling and converting the signals from its internal temperature sensor. A hardware timer is also configured to trigger the ADC conversion every 0.125 seconds before the device enters low-power mode 3 to conserve power. As soon as the ADC sample and conversion is complete, the raw ADC data is sent through the UART backchannel to the PC GUI.

As the raw ADC data is received by the PC GUI, Celsius and Fahrenheit units are calculated first. The PC GUI keeps a buffer of the most recent 100 temperature measurements, which are graphed against the PC's current time on the *Incoming Data* panel.

A red horizontal line is drawn across the data plot to indicate the moving average of the incoming data.

6. **To exit *Live Temp* mode, click the "Stop" button under "Live Temp Mode". You must exit this mode before starting the FRAM Log Mode.**
7. **To enter the *FRAM Log Mode*, click the "Start" button under "FRAM Log Mode" in the GUI's *Application Controls* panel.**

When the MSP430FR5969 receives the UART command from the GUI, it starts the entry sequence by initializing the Real-Time Clock to trigger an interrupt every 5 seconds. **The red LED blinks three times to indicate successful entry into FRAM Log Mode.**

Unlike in the Live Temperature Mode, the MSP430FR5969 enters low-power mode 3.5 to further decrease power consumption and wakes up every 5 seconds to perform data logging. Because the UART communication module does not retain power in LPM3.5, **the GUI automatically disconnects from the LaunchPad after entry into FRAM Log Mode.**

**Each time the device wakes up, the green LED lights up** to indicate its state to the user. The 12-bit ADC is set up to sample and convert the signals from its internal temperature sensor and battery monitor (Super Cap voltage).

A section of the device's FRAM is allocated to store the raw ADC output data (address 0x9000 to 0xEFFF). This allows the demo to store up to 6144 temperature and voltage data points (5 seconds/sample is approximately 8.5 hours of data).

8. **To exit the *FRAM Log Mode*, press the S2 (right) push button on the LaunchPad.**

The red LED turns on briefly to indicate successful exit.

The LaunchPad returns to the Power up and Idle state and you can reconnect the LaunchPad with the GUI to transfer the logged data from FRAM to the PC.

9. **Make sure the Launchpad is connected to the GUI and click the "Transfer FRAM Data" button in the GUI to begin transfer.**

A progress bar shows progress until the transfer completes, and the temperature and voltage data are plotted in the *Incoming Data* panel.

## (‘FR5969) Extra Credit

Open up the *MSP-EXP430FR5969 LaunchPad™ User’s Guide* ([slau535a.pdf](#)) to section “2.4.5 Super Cap”. Try using the FRAM Log Mode while powered from the Super Cap.

*The FRAM Log Mode also provides the option to log temperature data while powered either through the USB cable or only by the on-board Super Cap. The PC GUI contains step-by-step instructions in its side panel for configuring the jumpers on the LaunchPad to power the device with the Super Cap.*

---

**Hint:** We suggest that you look carefully at the initial jumper locations so that you can easily return the jumpers to their original locations after playing with the Super Cap.

---

# MSP-EXP430FR6989 LaunchPad OOB

For other LaunchPads, jump to the appropriate section: [F5529](#), [FR4133](#), [FR5969](#), [FR6989](#), [MSP432](#)

## Out-of-box Demo

### 1. Connecting to the computer

Connect the LaunchPad using the included USB cable to a computer. A green power LED should illuminate. For proper operation, drivers are needed.

### 2. Running the Out-of-box Demo

When connected to your computer, the LaunchPad will power up and display a greeting message on the LCD. Press and hold the S1 and S2 buttons simultaneously to select a new mode.

#### Stopwatch Mode

This mode provides a simple stopwatch application. It supports split time, where the display freezes while the stopwatch continues running in the background.

##### *Timer Stopped:*

- S1 - Start time
- S2 - Reset time

##### *Timer Running:*

- S1 - Stop time
- S2 - Split time (lap time)

#### Temperature Mode

This mode provides a simple thermometer application. Using the on-chip temperature sensor, the temperature is displayed on the LCD.

- S1 - Pause current temperature
- S2 - Toggle temperature between °F/C

# MSP-EXP432P401R LaunchPad OOB

For other LaunchPads, jump to the appropriate section: [F5529](#), [FR4133](#), [FR5969](#), [FR6989](#), [MSP432](#)

## 1. Connect the LaunchPad to your computer.

Connect the LaunchPad to your computer using the included USB cable. A green power LED should illuminate.

The LaunchPad will power up and the RGB LED (i.e LED2) will toggle during the startup sequence – after which, the LaunchPad will wait for commands from the GUI.

## 2. Open the MSP432 Out-of-Box graphical user interface (GUI).

You can find the MSP432 LaunchPad OOB GUI inside MSPWare. In fact, this GUI is available from MSPWare in the Cloud or on the Desktop.

Let's try out the Cloud version... the online version of MSPWare can be found within TI Resource Explorer Cloud.

### a) Open your web browser to the TI cloud tools webpage.

<http://dev.ti.com>

This opens the TI cloud tools page shown here:

The screenshot shows the TI Cloud Tools webpage. The page has a red header with the Texas Instruments logo and navigation links for Resource Explorer, CCS Cloud, and PinMux. Below the header, there's a main heading 'TI Cloud Tools' and a welcome message. A navigation bar includes links for Applications, Featured LaunchPads, FAQ, Tutorials, Community & Support, and About. The main content area is titled 'Cloud Tools' and features three cards:

- Resource Explorer:** Use Resource Explorer to browse examples, libraries and documentation. Supported Devices: More information > Resource Explorer
- Code Composer Studio Cloud:** Use CCS Cloud to edit, compile and download code to your LaunchPad. Supported Devices: More information > CCS Cloud
- TI PinMux:** Use TI PinMux to configure the device peripherals and pin multiplexing options. Supported Devices: More information > PinMux

A red arrow points from the 'Resource Explorer' card to the instruction below. A blue callout box on the right lists the three cloud tools, and a grey callout box at the bottom right provides a hint.

Currently, we have three cloud tools:

- Resource Explorer
- Code Composer Studio Cloud
- TI PinMux

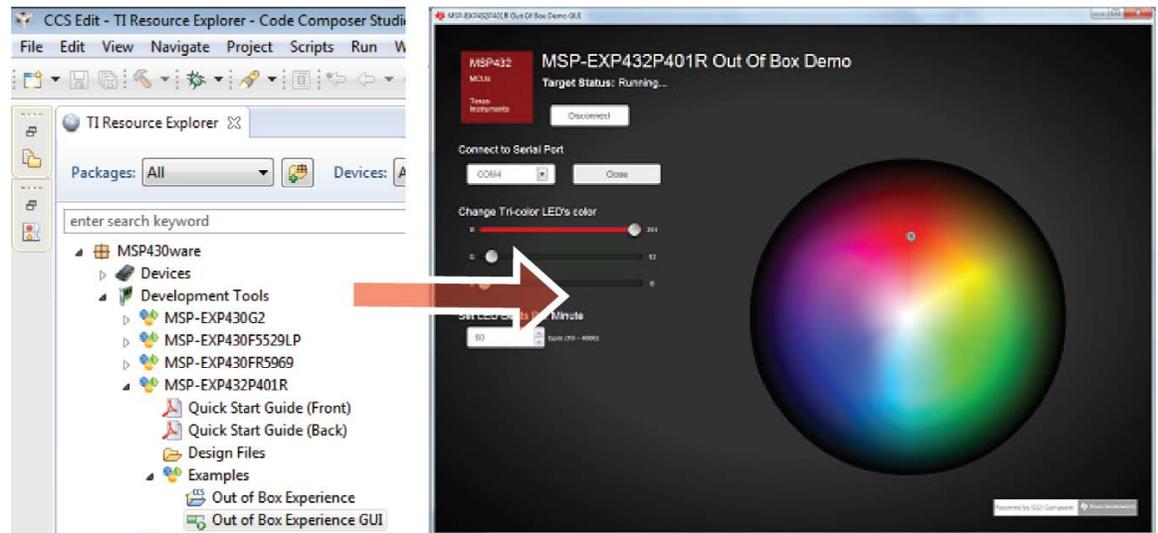
### b) Click on the left button to open TI Resource Explorer in the cloud.

**Hint:**

For best results, we recommend using Firefox or Chrome.

**c) Navigate into MSPWare to find the OOB GUI.**

MSPWare → Development Tools → MSP-EXP432P401R → Examples  
→ Out of Box Experience GUI



**3. Experiment with the LED's RGB color mode.**

The GUI allows you to set the color of the LED2 using a color wheel. Use the sliders to manipulate the channels of Red, Green, and Blue to make any color!

**4. Blink the RGB LED.**

- a) Use switch S1 to set the blink rate of the RGB LED (LED2). The blink speed of the LED matches the rate you are pressing S1. How fast can you blink the LED?
- b) Switch S2 toggles between the colors of the RGB LED, blinking each individual color at different rates. The toggles the LED between:
  - Red
  - Green
  - Blue
  - Random RGB color

**Notes:**