Introduction

In this section we’ll take a look at the MSP430 architecture, instructions, and tools and give you a chance to get some hands-on time with the hardware and software with a lab using the MSP430F2013. We’ll also learn about the I/O and do another lab using the MSP430FG4618/9.

Objectives

- Overview
- TI Embedded Processor Portfolio
- Architecture
- Tools
- Introduction lab
- I/O
- I/O lab
*** This page intentionally left blank ***
MSP430 4xx One Day Workshop 2010

Introduction ............................................................................................................................................... 1-1

MSP430 4xx One Day Workshop 2010 ..................................................................................................... 1-3
TI Microcontroller Portfolio ....................................................................................................................... 1-5
MSP430 Generations ................................................................................................................................. 1-5
MSP430 Peripheral Overview ..................................................................................................................... 1-6
MSP430 Portfolio ...................................................................................................................................... 1-6
LCD Controllers ...................................................................................................................................... 1-7
USB ........................................................................................................................................................ 1-8
CC430 ..................................................................................................................................................... 1-9
Innovative Peripherals ............................................................................................................................... 1-9
FRAM ....................................................................................................................................................... 1-10
No-Power Apps....................................................................................................................................... 1-10
Key Family Features ............................................................................................................................... 1-11
The Nuts and Bolts .................................................................................................................................. 1-13
Embedded Emulation .............................................................................................................................. 1-17
Innovative Tools ....................................................................................................................................... 1-18
eZ430-Chronos ..................................................................................................................................... 1-18
Code Composer Studio V4 ........................................................................................................................ 1-19
IAR Systems........................................................................................................................................... 1-19
IAR Kickstart.......................................................................................................................................... 1-20
Third Party Resources ............................................................................................................................. 1-20
www.ti.com/msp430 ................................................................................................................................ 1-21
Community Support ............................................................................................................................... 1-21
MSP430 Summary ................................................................................................................................... 1-22
Reset Conditions .................................................................................................................................... 1-22
Experimenter’s Board ............................................................................................................................... 1-23

Lab 1 – Flash the LED ............................................................................................................................... 1-25
Hardware list: .......................................................................................................................................... 1-26
Software list: .......................................................................................................................................... 1-26

IAR Kickstart Procedure ............................................................................................................................ 1-27
Install IAR Kickstart................................................................................................................................ 1-27
Hardware Verification ............................................................................................................................... 1-29
Power jumpers ....................................................................................................................................... 1-30
IAR Kickstart ......................................................................................................................................... 1-31
Configuring the Project ............................................................................................................................ 1-31
Create and Add the Source File ............................................................................................................ 1-32
Download and Run the Program ............................................................................................................ 1-33
FLASH Programming Exercise ................................................................................................................ 1-34

CCS 4.1 Procedure ................................................................................................................................... 1-35
Install Code Composer Studio .................................................................................................................. 1-35
Hardware Verification ............................................................................................................................... 1-38
Power jumpers ....................................................................................................................................... 1-39
CCS 4.1 .................................................................................................................................................. 1-40
Configuring the Target ............................................................................................................................ 1-41
Understanding the IDE Display ............................................................................................................. 1-42
Create and Add a Source File .................................................................................................................. 1-43
Download and Run the Program ............................................................................................................. 1-44
FLASH Programming Exercise ................................................................................................................ 1-45

Standard Definitions ............................................................................................................................... 1-47
Controlling GPIO Ports ......................................................................................................................... 1-47
**Lab 2 – I/O Overview.** ................................................................. I-49
  Hardware list: ............................................................................... I-50
  Software list: ................................................................................ I-50

**IAR Kickstart Procedure.** .......................................................... I-51
  New Workspace and Project.......................................................... I-51
  Configure the Project................................................................. I-51
  Add Source File ......................................................................... I-51
  Complete the Code ...................................................................... I-52
  Test Your Code........................................................................... I-54

**CCS 4.1 Procedure** .................................................................. I-57
  Add a Source File ....................................................................... I-58
  Complete the Code ...................................................................... I-58
  Test Your Code .......................................................................... I-60

**Review Questions** ................................................................... I-62
**TI Microcontroller Portfolio**

**TI Embedded Processing Portfolio**

<table>
<thead>
<tr>
<th>Microcontrollers</th>
<th>ARM-Based</th>
<th>DSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-bit</td>
<td>32-bit</td>
<td>32-bit ARM</td>
</tr>
<tr>
<td>MSP430</td>
<td>C2000™</td>
<td>ARM + Cortex A8</td>
</tr>
<tr>
<td>Ultra-Low Power</td>
<td>Fixed &amp; Floating Point</td>
<td>Industry Std Low Power</td>
</tr>
<tr>
<td>Up to 25 MHz</td>
<td>Up to 150 MHz</td>
<td>Up to 100 MHz</td>
</tr>
<tr>
<td>Flash</td>
<td>PWM, ADC, CAN, SPI, I²C, USB, LVDS, ADC, PWM, OCV</td>
<td>USB, LCD, MMC, EMAC</td>
</tr>
<tr>
<td>1KB to 2KB</td>
<td>32KB to 512KB</td>
<td>8KB to 256KB</td>
</tr>
<tr>
<td>Analog I/O, ADC, LCD, USB, RF</td>
<td>Motor Control, Digital Power, Lighting</td>
<td>Host Control</td>
</tr>
<tr>
<td>Measurement, Sensing, General Purpose</td>
<td>$1.00 to $20.00</td>
<td>$2.00 to $8.00</td>
</tr>
</tbody>
</table>

**MSP430 Generations**

<table>
<thead>
<tr>
<th>CPU Clock (Max)</th>
<th>2xx</th>
<th>4xx</th>
<th>5xx</th>
</tr>
</thead>
<tbody>
<tr>
<td>16MHz</td>
<td>16MHz</td>
<td>8 &amp; 16 MHz</td>
<td>25 MHz</td>
</tr>
<tr>
<td>Flash/RAM (Largest comparable device)</td>
<td>120KB / 4KB (F24xx)</td>
<td>120KB / 4KB (FG4xxx)</td>
<td>256KB / 16KB (F54xx)</td>
</tr>
<tr>
<td>Active Current (3.3V) µA/µIPS</td>
<td>515 µA</td>
<td>600 µA/µIPS</td>
<td>220 µA/µIPS</td>
</tr>
<tr>
<td>1MB</td>
<td>525 µA/µIPS</td>
<td>600 µA/µIPS</td>
<td>165 µA/µIPS</td>
</tr>
<tr>
<td>8MB</td>
<td>569 µA/µIPS</td>
<td>N/A</td>
<td>188 µA/µIPS</td>
</tr>
<tr>
<td>16MB</td>
<td>N/A</td>
<td>N/A</td>
<td>224 µA/µIPS</td>
</tr>
<tr>
<td>256MB</td>
<td>N/A</td>
<td>N/A</td>
<td>224 µA/µIPS</td>
</tr>
<tr>
<td>Standby Current (LPM3)</td>
<td>0.3 – 1.1µA</td>
<td>0.7 – 1.3µA</td>
<td>2.6µA (w/ active true RTC)</td>
</tr>
<tr>
<td>Power Down Current (LPM4/5)</td>
<td>0.1µA</td>
<td>0.1µA</td>
<td>1.6µA (LPM4) / 0.1µA (LPM5)</td>
</tr>
<tr>
<td>Wake-up Time From LPM3</td>
<td>1µs</td>
<td>6µs</td>
<td>5µs</td>
</tr>
<tr>
<td>Flash ISP Minimum DVcc</td>
<td>2.2V</td>
<td>2.7V</td>
<td>1.8V</td>
</tr>
<tr>
<td>Port I/O Interrupt Capability</td>
<td>P1/P2</td>
<td>P1/P2</td>
<td>P1/P2 (F5438)</td>
</tr>
<tr>
<td>Prog. Port Pin Drive Strength</td>
<td>N/A</td>
<td>N/A</td>
<td>All port pins</td>
</tr>
<tr>
<td>Prog. Pull-ups/downs</td>
<td>All port pins</td>
<td>N/A</td>
<td>All port pins</td>
</tr>
<tr>
<td>Available MCLK Sources</td>
<td>DCO, VLO, LXIT, XT2</td>
<td>FLL, LXIT, XT2</td>
<td>FLL, VLO, REFO, XT1, XT2</td>
</tr>
<tr>
<td>FLL Reference Clocks</td>
<td>N/A</td>
<td>LXIT</td>
<td>REFO, XT1, XT2</td>
</tr>
</tbody>
</table>

**MSP430 Peripheral Overview ...**
### MSP430 Peripheral Overview

#### MSP430 Peripheral Overview

<table>
<thead>
<tr>
<th></th>
<th>1xx</th>
<th>2xx</th>
<th>4xx</th>
<th>5xx</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Clock System</td>
<td>Basic Clock System</td>
<td>FLL, FLL+</td>
<td>Unified Clock System</td>
<td></td>
</tr>
<tr>
<td>Core voltage same as supply voltage (1.8-3.6V)</td>
<td>Core voltage same as supply voltage (1.8-3.6V)</td>
<td>Core voltage same as supply voltage (1.8-3.6V)</td>
<td>Programmed core voltage with integrated PMM (1.8-3.6V)</td>
<td></td>
</tr>
<tr>
<td>16-bit CPU</td>
<td>16-bit CPU, CPUX</td>
<td>16-bit CPU, CPUX</td>
<td>16-bit CPUXx2</td>
<td></td>
</tr>
<tr>
<td>GPIO</td>
<td>GPIO w/ pull-up and pull-down</td>
<td>GPIO, LCD Controller</td>
<td>GPIO w/ pull-up and pull-down, drive strength</td>
<td></td>
</tr>
<tr>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>CRC16</td>
<td></td>
</tr>
<tr>
<td>Software RTC</td>
<td>Software RTC</td>
<td>Software RTC with Basic Timer, Basic Timer + RTC</td>
<td>True 32-bit RTC with A/D</td>
<td></td>
</tr>
<tr>
<td>USART</td>
<td>USCI, USI</td>
<td>USART, USCI</td>
<td>USCI, USB, RF</td>
<td></td>
</tr>
<tr>
<td>DMA up to 3-ch</td>
<td>DMA up to 3-ch</td>
<td>DMA up to 3-ch</td>
<td>DMA up to 8-ch</td>
<td></td>
</tr>
<tr>
<td>MPY16</td>
<td>MPY16</td>
<td>MPY16, MPY32</td>
<td>MPY32</td>
<td></td>
</tr>
<tr>
<td>ADC10,12</td>
<td>ADC10,12, SD16</td>
<td>ADC12, SD16, OPA</td>
<td>ADC12_A</td>
<td></td>
</tr>
<tr>
<td>4-wire JTAG</td>
<td>4-wire JTAG, 2-wire Spy Bi-Wire (Some devices)</td>
<td>4-wire JTAG, 2-wire Spy Bi-Wire</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### MSP430 Portfolio

#### MSP430 Portfolio

- **1xx-Catalog**
  - 16 MIPS
  - 8 kB RAM
  - 500 nA Standby
  - 1.8 – 3.6V

- **2xx-Catalog**
  - 16 MIPS
  - 120 kB Flash
  - 6 kB RAM
  - 500 nA Standby
  - 1.8 – 3.6V

- **4xx: LCD**
  - 16 MIPS
  - 120 kB Flash
  - 6 kB RAM
  - LCD Controller, 160 segments
  - 1.8 – 3.6V

- **5xx-6xx**
  - 25MIPS
  - 256 kB Flash
  - 16 kB RAM
  - FRAM, USB, RF
  - 6xx: LCD Controller
  - 160 uA/MIPS

- **F23x0**
  - The New Generation

- **F20xx**
  - F/C11xx

- **F21x1**
  - F13x-F14x

- **F22xx**
  - F15x-F16x

- **F23x-F24x**
  - F17x-F18x

- **F25x**
  - F19x-F20x

- **F261x**
  - F241x

- **F27x**
  - F22xx

- **FR1000**
  - F/C21xx

- **F30x**
  - F44x

- **F31xx**
  - F/C41x

- **F34x**
  - F/C42x

- **F36x**
  - F/C43x

- **F33x-F34x**
  - F/C46x

- **F35x**
  - F/C47x

- **F40x**
  - F/C48x

- **F41x**
  - F/C49x

- **F42x**
  - F/C50x

- **F43x**
  - F/C51x

- **F45x**
  - F/C52x

- **F46x**
  - F/C53x

- **F47xx**
  - F/C54x

- **F50x**
  - F/C55x

- **F541x**
  - F/C56x

- **F55xx**
  - CC430 RF

- **F56xx**
  - USB

- **Fx42x**
  - Fx43x

- **Fx47x**
  - Fx48x

- **Fx471xx**
  - Fx49x

- **Fx472xx**
  - Fx50x

- **Fx473xx**
  - Fx51x

- **Fx474xx**
  - Fx52x

- **Fx475xx**
  - Fx53x

- **Fx476xx**
  - Fx54x

- **Fx477xx**
  - Fx55x

- **Fx478xx**
  - Fx56x

- **Fx479xx**
  - Fx57x

- **Fx480xx**
  - Fx58x

- **Fx481xx**
  - Fx59x

- **Fx482xx**
  - Fx60x

- **Fx483xx**
  - Fx61x

- **Fx484xx**
  - Fx62x

- **Fx485xx**
  - Fx63x

- **Fx486xx**
  - Fx64x

- **Fx487xx**
  - Fx65x

- **Fx488xx**
  - Fx66x

- **Fx489xx**
  - Fx67x

- **Fx490xx**
  - Fx68x

- **Fx491xx**
  - Fx69x

- **Fx492xx**
  - Fx70x

- **Fx493xx**
  - Fx71x

- **Fx494xx**
  - Fx72x

- **Fx495xx**
  - Fx73x

- **Fx496xx**
  - Fx74x

- **Fx497xx**
  - Fx75x

- **Fx498xx**
  - Fx76x

- **Fx499xx**
  - Fx77x

- **Fx500xx**
  - Fx78x

- **Fx501xx**
  - Fx79x

- **Fx502xx**
  - Fx80x

- **Fx503xx**
  - Fx81x

- **Fx504xx**
  - Fx82x

- **Fx505xx**
  - Fx83x

- **Fx506xx**
  - Fx84x

- **Fx507xx**
  - Fx85x

- **Fx508xx**
  - Fx86x

- **Fx509xx**
  - Fx87x

- **Fx510xx**
  - Fx88x

- **Fx511xx**
  - Fx89x

- **Fx512xx**
  - Fx90x

- **Fx513xx**
  - Fx91x

- **Fx514xx**
  - Fx92x

- **Fx515xx**
  - Fx93x

- **Fx516xx**
  - Fx94x

- **Fx517xx**
  - Fx95x

- **Fx518xx**
  - Fx96x

- **Fx519xx**
  - Fx97x

- **Fx520xx**
  - Fx98x

- **Fx521xx**
  - Fx99x

- **Fx522xx**
  - 100 devices

- **Fx523xx**
  - 160 devices

- **Fx524xx**
  - 200 devices
LCD Controllers

- Ultra-low power
- Fully automatic
- 4/3/2/1 mux
- Up to 160-bit display
- Internal regulated voltage generator
- Internal or external bias generation
- Contrast control
- 1/2 bias for 3 or 4 mux
- Internal clock generation
- Auto segment blinking
USB

Enabling You with Full Speed USB

**Ultra-low power MCUs + USB for smarter connectivity**

- Embedded full-speed USB 2.0 (12 Mbps)
- High flexibility with configurable 2K data buffers that can be used as RAM
- Unused USB interface pins can function as high-current I/O (5v tolerant)

**Analog and peripheral integration reduces system cost**

- Multiple analog options with 10 or 12-bit ADC, DAC, comparator
- Integrated 3.3V LDO for use with 5V USB bus power
- Uses low-cost crystal for USB clock, with flexible, integrated PLL

**44 New USB devices within next 12 months**

- Wide range of memory configurations and package options, 8k-128k flash
- Diverse peripheral mix in the MSP430F55xx family
- Pricing as low as $0.96 in volume

USB made easy ...

USB Made Easy

- USB Bootstrap Loader (USB)
  - Supporting device programming
  - Field Firmware updates
- USB Descriptor Tool
  - Configures stack functions via GUI
- Free USB stacks available:
  - Communication Device Class (CDC)
  - Human Interface Device (HID)
  - Mass Storage Class (MSC)
- Additional stacks available from third parties

FREE Vendor ID/ Product ID sharing program

MSP430F5529 Sample Kit

FREE CC430 ...

USB made easy...
CC430

CC430: Enabling You With RF

Lowest Power Monolithic RF SoC

The Best of Both Worlds

MSP430 MCU

• Market’s lowest power MCU
• High analog performance
• High level of integration
• Ease of development
• Sensor interface

Low Power RF Transceiver

• High sensitivity
• Low current consumption
• Excellent blocking performance
• Flexible data rate & modulation format
• Backwards compatible

Innovative peripherals ...

Innovative Peripherals

CC430: Innovative Peripherals

LCD_B

• Blinking of individual segments, Programmable frame frequency,
  Software-driver contrast control
• Regulated charge pump
• Integrated drivers

AES 128

• Encryption and decryption according to AES FIPS PUB 197
  with 128-bit keys
• Key expansion for en- and decryption
• Off-line key generation for decryption

Comparator_B

• Selectable reference voltage & voltage hysteresis generator
• High-speed, normal, and ultra-low power 100nA modes
• Internal output to Timer A capture
• Selectable RC filter for comparator output

FRAM ...
FRAM

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**

No-power apps …

No-Power Apps

MSP430 Enables No-Power Apps

- **Energy harvesting** is the process by which energy is captured and stored
- Can substitute batteries that are costly to maintain and can extend system uptime
- Only possible with ultra-low power components
- Solar, kinetic, thermal, RF, salinity gradients, pH difference and other ambient sources available

F2xx key features …
Key Family Features

### F2xx Key Features

- <1µA standby LPM3
- <1µs 0-16MHz
- Zero-power BOR
- Failsafe oscillator
- Enhanced watchdog
- Pull-up / down resistors
- Hack proof boot loader
- 2.2V Flash ISP
- Extended temp 105°C
- Same instruction set architecture

### F4xx Key Features

- <1µA standby LPM3
- <1µs 0-16MHz
- 4-120 KB Flash
- Built-in LCD Driver
- Zero-power BOR
- Pull-up / down resistors
- 2.7V Flash ISP
- Same instruction set architecture

### F5xx Key Features ...

F4xx key features ...

F5xx key features ...
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 docking 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
The Nuts and Bolts

16-bit RISC CPU

- Efficient, ultra-low power CPU
- C-compiler friendly
- RISC architecture
  - 27 core instructions
  - 24 emulated instructions
  - 7 addressing modes
  - Constant generator
- Single-cycle register operations
- Memory-to-memory atomic addressing
- Bit, byte and word processing
- 20-bit addressing on MSP430X for Flash >64KB

Bytes, Words And CPU Registers

<table>
<thead>
<tr>
<th>16-bit addition</th>
<th>Code/Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>add.w R4,R5</td>
<td>1/1</td>
</tr>
<tr>
<td>5405</td>
<td>3/6</td>
</tr>
<tr>
<td>add.w &amp;0200,&amp;0202</td>
<td></td>
</tr>
<tr>
<td>529202000202</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8-bit addition</th>
<th>Code/Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>add.b R4,R5</td>
<td>1/1</td>
</tr>
<tr>
<td>5445</td>
<td>3/6</td>
</tr>
<tr>
<td>add.b &amp;0200,&amp;0202</td>
<td></td>
</tr>
<tr>
<td>52D202000202</td>
<td></td>
</tr>
</tbody>
</table>

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

Seven addressing modes ...
Seven Addressing Modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>Example 1</th>
<th>Example 2</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Mode</td>
<td><code>mov.w R10,R11</code></td>
<td></td>
<td>Single cycle</td>
</tr>
<tr>
<td>Indexed Mode</td>
<td><code>mov.w 2(R5),6(R6)</code></td>
<td></td>
<td>Table processing</td>
</tr>
<tr>
<td>Symbolic Mode</td>
<td><code>mov.w EDE,TONI</code></td>
<td></td>
<td>Easy to read code, PC relative</td>
</tr>
<tr>
<td>Absolute Mode</td>
<td><code>mov.w &amp;EDE,&amp;TONI</code></td>
<td></td>
<td>Directly access any memory</td>
</tr>
<tr>
<td>Indirect Register Mode</td>
<td><code>mov.w @(R10),0(R11)</code></td>
<td></td>
<td>Access memory with pointers</td>
</tr>
<tr>
<td>Indirect Autoincrement</td>
<td><code>mov.w @(R10),0(R11)</code></td>
<td></td>
<td>Table processing</td>
</tr>
<tr>
<td>Immediate Mode</td>
<td><code>mov.w #45h,&amp;TONI</code></td>
<td></td>
<td>Unrestricted constant values</td>
</tr>
</tbody>
</table>

Atomic Addressing

B=B+A

<table>
<thead>
<tr>
<th>Pure RISC</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>push R5</code></td>
</tr>
<tr>
<td><code>ld R5,A</code></td>
</tr>
<tr>
<td><code>add R5,B</code></td>
</tr>
<tr>
<td><code>st B,R5</code></td>
</tr>
<tr>
<td><code>pop R5</code></td>
</tr>
</tbody>
</table>

MSP430

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>add A,B</code></td>
</tr>
</tbody>
</table>

- Non-interruptible memory-to-memory operations
- Useable with complete instruction set
**Constant Generator**

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

```
4314 mov.w #0002h,R4 ; With CG
40341234 mov.w #1234h,R4 ; Without CG
```

**24 Emulated Instructions**

- Easier to understand - no code size or speed penalty
- Replaced by assembler with core instructions
- Completely automatic

```
4130 ret ; Return (emulated)
4130 mov.w @SP+,PC ; Core instruction
```

Emulated instructions ...

Assembly instruction formats ...
Three Assembly Instruction Formats

**Format I**
Source and Destination

- `add.w R4,R5` ; R4+R5=R5 `xxxx`
- `add.b R4,R5` ; R4+R5=R5 `00xx`

**Format II**
Destination Only

- `rlc.w R4`
- `rlc.b R4`

**Format III**
8(Un)conditional Jumps

- `jmp Loop_1` ; Goto Loop_1

51 Total Assembly Instructions

<table>
<thead>
<tr>
<th>Format I Source, Destination</th>
<th>Format II Single Operand</th>
<th>Format III +/- 9bit Offset</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>add(.b)</code></td>
<td><code>br</code></td>
<td><code>jmp</code></td>
<td><code>clrc</code></td>
</tr>
<tr>
<td><code>adcc(.b)</code></td>
<td><code>call</code></td>
<td><code>jc</code></td>
<td><code>setc</code></td>
</tr>
<tr>
<td><code>and(.b)</code></td>
<td><code>swpb</code></td>
<td><code>jnc</code></td>
<td><code>clrz</code></td>
</tr>
<tr>
<td><code>bic(.b)</code></td>
<td><code>sxt</code></td>
<td><code>jsq</code></td>
<td><code>setz</code></td>
</tr>
<tr>
<td><code>bit(.b)</code></td>
<td><code>push(.b)</code></td>
<td><code>jse</code></td>
<td><code>clrn</code></td>
</tr>
<tr>
<td><code>bit(.b)</code></td>
<td><code>pop(.b)</code></td>
<td><code>jge</code></td>
<td><code>setn</code></td>
</tr>
<tr>
<td><code>cmp(.b)</code></td>
<td><code>rxa(.b)</code></td>
<td><code>jl</code></td>
<td><code>dint</code></td>
</tr>
<tr>
<td><code>dadd(.b)</code></td>
<td><code>rrc(.b)</code></td>
<td><code>ja</code></td>
<td><code>eint</code></td>
</tr>
<tr>
<td><code>mov(.b)</code></td>
<td><code>inv(.b)</code></td>
<td><code>nop</code></td>
<td></td>
</tr>
<tr>
<td><code>sub(.b)</code></td>
<td><code>inc(.b)</code></td>
<td><code>ret</code></td>
<td></td>
</tr>
<tr>
<td><code>subc(.b)</code></td>
<td><code>incd(.b)</code></td>
<td><code>reti</code></td>
<td></td>
</tr>
<tr>
<td><code>xor(.b)</code></td>
<td><code>dec(.b)</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>dadc(.b)</code></td>
<td><code>adc(.b)</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>sbcc(.b)</code></td>
<td><code>clr(.b)</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>sadc(.b)</code></td>
<td><code>dadc(.b)</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>rl(.b)</code></td>
<td><code>rlc(.b)</code></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Bold type denotes emulated instructions**

Unified memory map...
Unified Memory Map

- Absolutely no paging
- Supports code agility
- In System Programmable (ISP) Flash
  - Self programming
  - JTAG
  - Bootloader

```
// Flash In System Programming
FCTL3 = FWKEY; // Unlock
FCTL1 = FWKEY | WRT; // Enable
*(unsigned int *)0xFC00 = 0x1234;
```

Embedded Emulation

- Real-time, in-system debug
  - No application resources used
  - Full speed execution
  - H/W breakpoints
  - Single stepping
  - Complex triggering
  - Trace capability
- Powerful, easy to use tools
- Spy Bi-Wire
  - 2-wire debug interface
  - No pin function impact
- Only 1 tool required for all devices
Innovative Tools

Easy To Use, Innovative Tools

Flash Emulation Tools
• Compatible with all devices
• Target boards available
• $99 ($149 with target board)
• Target boards available without FET
• Free IDEs included

MSP430 Experimenter Boards
• Fully featured prototyping system
• Available for FG4618 & F5438
• Starting at $99

eZ430 Tools
• Complete development system in USB stick
• Available for wireless and energy harvesting
• Starting at $20

eZ430-Chronos

eZ430-Chronos: CC430 Dev Tool

• CC430-based wireless development tool in a watch
• 915/868/433 MHz versions available
• Custom LCD driven directly by CC430
• Features:
  • 3-axis accelerometer
  • Altimeter
  • Temperature sensor
  • Buzzer

USB RF access point

Updated eZ430 emulator for programming

CCS v4 ...
**Code Composer Studio V4**

**CCE is now Code Composer Studio v4**

- Code Composer Studio v4: A single development platform for all TI processors
- CCE users will feel at home
- Enhancements since CCE:
  - Speed
  - Code size improvements
  - Auto-updating
  - License manager
  - Support for all TI MCUs
- Only $495 for MCU Edition
- FREE 16KB-limited edition

---

**IAR Systems**

**TI and IAR Systems: Deep and Evolving Partnership**

*1990’s*  
TI and IAR Systems partners on MSP430

*2005*  
TI and IAR partners on ARM MCUs

*2006*  
TI acquires Chipcon, partner to IAR Systems

*2009*  
TI acquires Luminary Micro, partner to IAR Systems

**TI and IAR Systems Product Integration and Support**

- IAR Embedded Workbench C/C++ compiler and debugger tool set
- IAR PowerPac RTOS
- USB Device Stack
- IAR visualSTATE design, test and verification tools
- IAR KickStart Kits

---

**IAR kickstart ...**
IAR Kickstart

IAR Kickstart IDE

- 4kB Compiler
  (8kB for >60k Flash devices)
- Assembler/Linker
- Editor
- Debugger

Third Party Resources

Third Party Development Resources

- Rowley CrossWorks
  - Complete IDE solution
  - High code density
  - Simulator
  - Windows, Linux, Mac
  - www.rowley.co.uk

- Elprotronic
  - MSP430, CC Chipcon, C2000 Programmers
  - Fastest download speed
  - Production programmers

- RTOS Options
  - µC/OS-II™
  - CMX-Tiny+™
  - embOS
  - FreeRTOS™
  - IAR PowerPac
  - QP™
  - Salvo™
  - TinyOS

- MSPGCC Tool Chain
  - Free
  - Open Source
  - GNU C Compiler, Assembler/Linker, GDB Debugger
  - Windows, Linux, Unix
  - http://mspgcc.sourceforge.net

- Amber Wireless
  - Drop in wireless modules
  - <1GHz eZ430-RF target boards
  - CC430 Development boards

- USB Stacks
  - IAR
  - HCC

- Third parties...
www.ti.com/msp430

Community Support

**Extensive Community Support**

**E2E Community**
- Videos, Blogs, Forums
- Extensive community support and idea exchange
- Global customer support
- [http://e2e.ti.com](http://e2e.ti.com)

**Processor Wiki**
- Growing collection of technical wiki articles
- Tips & tricks, common pitfalls, and design ideas
- [http://wiki.msp430.com](http://wiki.msp430.com)
MSP430 Summary

- Ultra-low Power
- Broad portfolio
  - Access for size and cost constraints
  - Performance for precision and speed
- Enabling Technologies
  - FRAM, USB, RF, energy harvesting
- Ease of Use
  - HW and SW Tools
  - Community

Reset Conditions

- RST/NMI configured in the reset mode
- All I/O pins are switched to input
- Watchdog timer powers up as active watchdog
- Other peripheral modules are disabled
- Status register (SR) is reset
- Program counter (PC) is loaded with (0FFFEh)
- Always refer to the user guide for information specific to your device
Experimenter’s Board

Two MSP430 devices:
- MSP430FG4618 or MSP430FG4619
- MSP430F2013

Interface for ChipCon RF transceiver EMK boards
Lab 1 – Flash the LED

Let’s familiarize ourselves with the lab equipment and then move on to performing a simple task: flashing the LED using the F2013.

There are two sets of instructions for the labs; one using the IAR Kickstart IDE and the other using TI’s Code Composer Studio 4.1. Decide which IDE you’d like to use and then team up with a partner using the same IDE.
Lab 1 – Flash the LED

Hardware list:
- WinXP PC
- MSP-FET430UIF
- USB cable
- JTAG ribbon cable
- MSP430FG461x/F28xx Experimenter’s Board
- Jumpers

Software list:
- IAR Kickstart for MSP430 version 4.21B
- Code Composer Studio 4.1
- Labs
- Additional pdf documentation
- Adobe™ Reader
IAR Kickstart Procedure

In this lab, you will verify that the hardware/software has been set up properly. We’ll also familiarize ourselves with the tools we’ll be using for the rest of the workshop via a short program running on the F2013.

Install IAR Kickstart

1. **Disconnect** any evaluation board that you have connected to your PCs USB port(s). **Insert** the Workshop Installation Flash Drive into a free USB port.

2. Using **Windows Explorer**, find and double-click on the file named **EW430-KS-web-4212.exe**.

3. Follow the steps in the IAR installation program. When you reach the **Enter User Information** window, use Windows Explorer to find and open the **IAR License.txt** file on the installation flash drive. **Copy/paste** the license number as shown below and click **Next**.

![IAR Embedded Workbench Kickstart for MSP430 4.21](image)
4. In the same way, **copy/paste** the **License Key** into the next window and click **Next**.

![Image of the IAR Embedded Workbench Kickstart for MSP430 4.21 window](image)

Select a **Complete** installation and click **Next**. Install the tools into the **default folder**, if possible. The installation should take less than 10 minutes to complete.

5. **Driver Installation**

Using Windows Explorer, look on the workshop flash drive and double-click on `swrc094e setup`. Follow the wizard steps until it completes. Again using Windows Explorer, navigate to `C:\Program Files\Texas Instruments Inc\TUSB3410 Single Driver Installer\DISK1` and double-click on `setup`. Follow the wizard steps until it completes.

6. **Lab Files Installation**

Using Windows Explorer, look on the workshop flash drive and double-click on `all_labs.exe`. Leave the unzip directory as `C:\` and click **Unzip**. When the process completes, click **Close**. The labs have been placed in `C:\MSP430ODW`.

If you’ve been tasked with installing IAR Kickstart, the drivers and labs only, please stop here and ask your instructor for further directions.
Hardware Verification

1. Check out the hardware

Make sure that the MSP430 USB FET is connected to the USB cable and that the other end of the cable is connected to the PC’s USB port. The ribbon cable should be connected to the debug interface at one end to the port marked Target and to the lower of the two debug ports on the MSP430FG461x/F28xx Experimenter’s Board (the MSP430F2013 emulation port).

2. Software driver

If you are prompted to load the driver when you connect the FET to the PC, don’t search the web for the driver and don’t load the driver automatically. You can locate the driver in the C:\Program Files\IAR Systems\Embedded Workbench 5.4 Kickstart\430\drivers\TIUSBFET folder.
Power jumpers

3. The board has several jumpers that control power to the board …

Make sure the jumpers are set as follows:

PWR1 controls power to the MSP430FG4619 (ON)

PWR2 controls power to the MSP430F2013 (ON)

JP2 isolates the LED from the touch pad (ON)

BATT controls power from the AAA batteries and can be used to measure current (OFF)

VCC_1 and VCC_2 control whether the microcontrollers are powered by the emulator (FET) or the batteries (LCL). Since we’ll be powering from the board from the emulator, place both jumpers over the rightmost two pins as shown:
IAR Kickstart

4. Start up the IDE

On the desktop of your PC you should see a shortcut that looks like:

Double-click the shortcut to start IAR Kickstart. The IAR Information Center window will appear on top of the IAR tool. Click the X in the upper right to close the window.

5. Create a New Workspace

Click File ⇒ New ⇒ Workspace on the menu bar to create a new workspace.

6. Create a New Project

On the menu bar, click Project ⇒ Create New Project. When the Create New Project dialogue appears, click OK. The Save As dialogue will appear; name your project Lab1 in the C:\MSP430ODW\IAR Labs\Lab1 folder and click Save.

Configuring the Project

7. Set the Project Options

From the IAR Embedded Workbench menu bar, select Project ⇒ Options.

Under the Target tab, note the Device selection box. Click the drop-menu to the right of this box and select MSP430x2xx Family, then MSP430F2013 from the list.

Still under the Target tab, click Assembler-only project.

In the Category list to the left, click Debugger. Under the Setup tab, select FET Debugger from the Driver drop-down menu.

Select the Plugins tab, and uncheck the box next to Stack.

In the Category list to the left, click FET Debugger. Under the Setup tab, select Texas Instrument USB-IF from the Connection drop-down menu.

Click OK.