

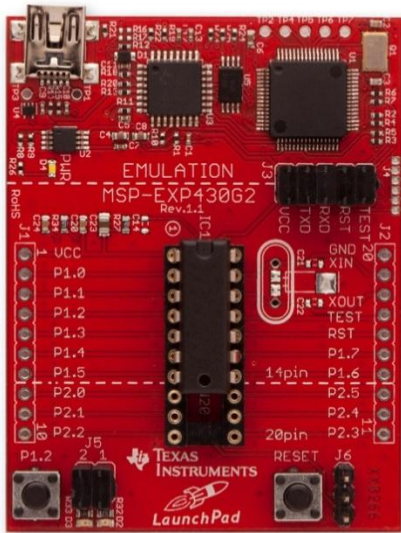
MSP 430 Tutorial

Gabe Cohn

TI MSP 430

- Ultra-low-power!
- Widely used in low-power research
 - Power harvesting
 - Ultra-low-power sensor networks
- More complicated than AVR (Atmega)
- Not used much in industry (yet...)
- Very low cost evaluation/dev kits

MSP430 Eval/Dev Kits



MSP430 LaunchPad

\$4.30



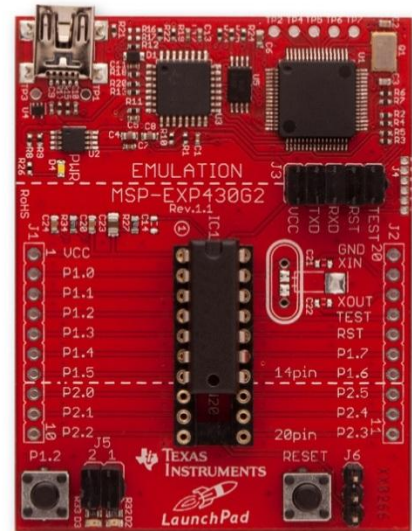
eZ430



eZ430-Chronos

MSP430 Launch Pad Dev. Kit

- Very low cost!
- Simple MSP430
- USB programmer / debugger
- 1 PB-switch
- 2 LEDs (red and green)
- All I/O pins exposed
- **Only \$4.30!**



eZ430 Dev. Kit

- USB thumb-drive form-factor
- Simple MSP430
- USB programmer / debugger
- Removable target board
- All I/O pins exposed
- RF versions available
(e.g. eZ430-RF2500)



eZ430-Chronos Dev. Kit

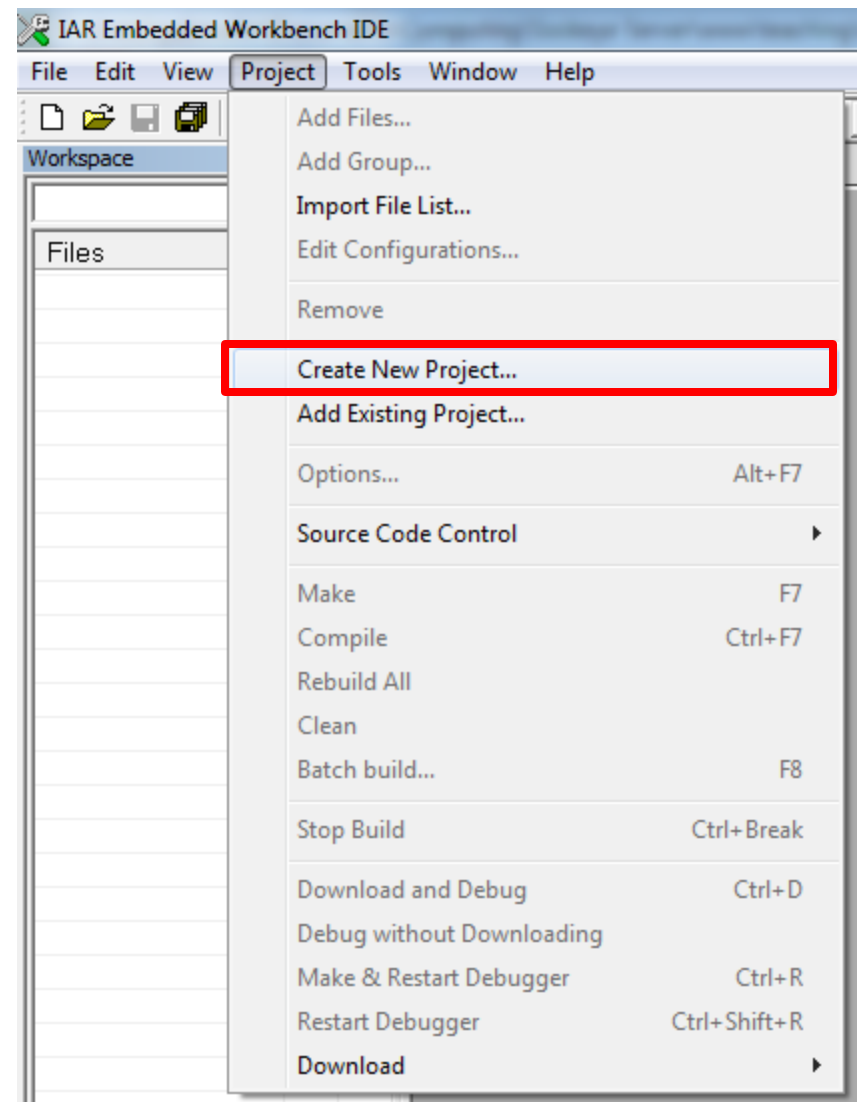
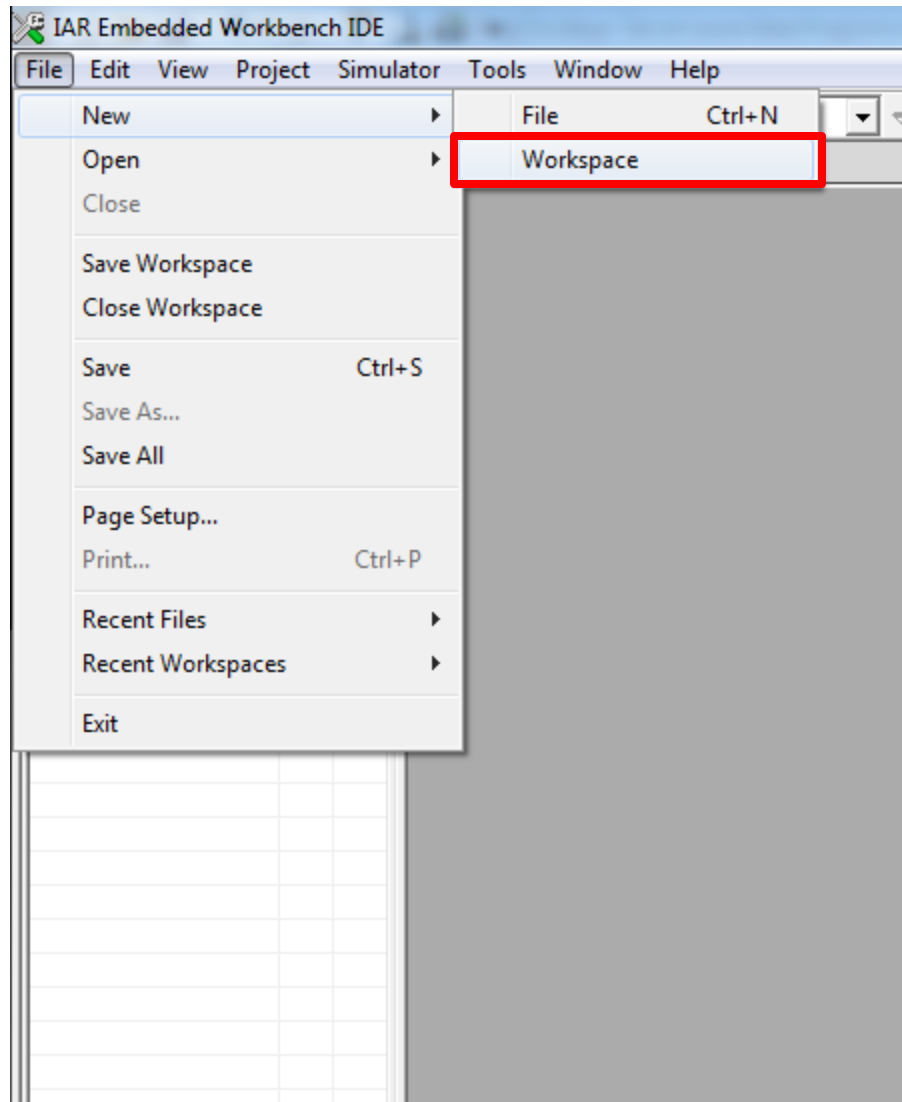
- Watch form-factor!
- Wireless programmer!
- USB programmer / debugger
- 3-axis accelerometer
- Barometric pressure sensor
- Temperature sensor
- Battery/Voltage sensor
- BlueRobin protocol (heart-rate)



Software Environment (IDE)

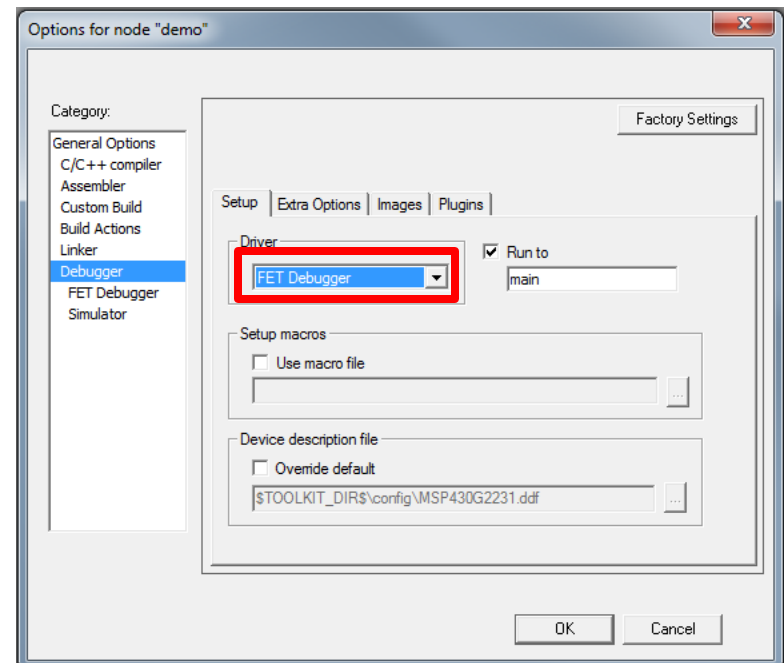
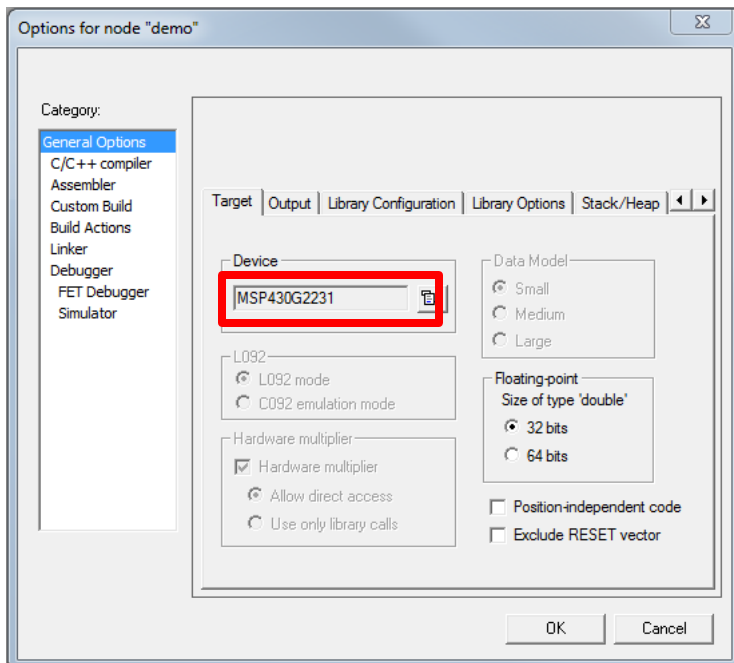
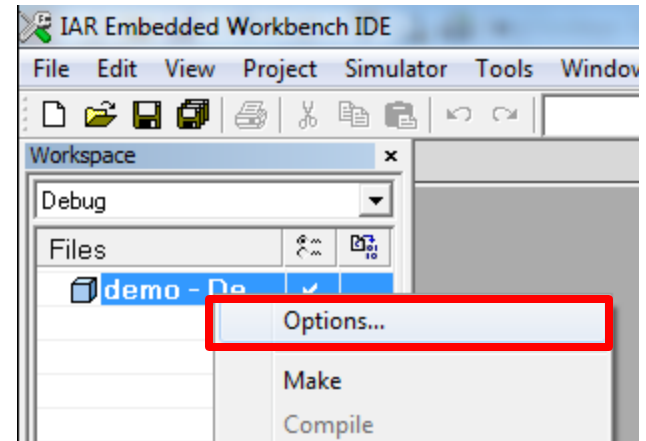
- IAR Embedded Workbench (IAR)
 - C/C++ compiler
 - simulator and debugger
 - Free version with 4 KB code size limit
 - easy to use and understand
- Code Composer Studio (CCS)
 - Eclipse
 - Free version with 16 KB code size limit
 - recommended for larger (RF) projects
 - complicated and buggy!

Create IAR Workspace and Project



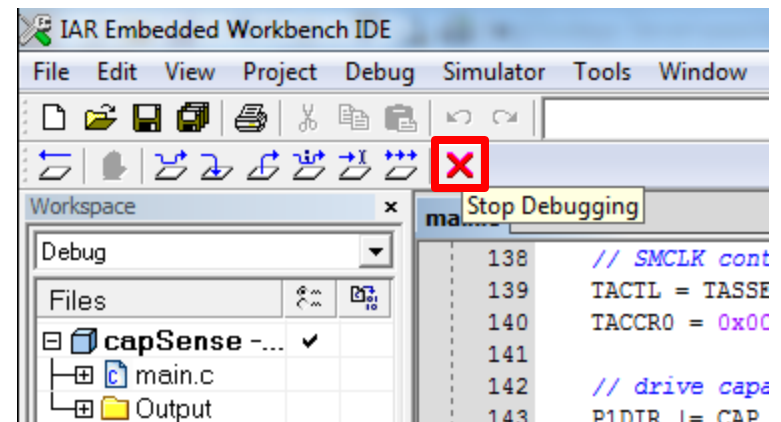
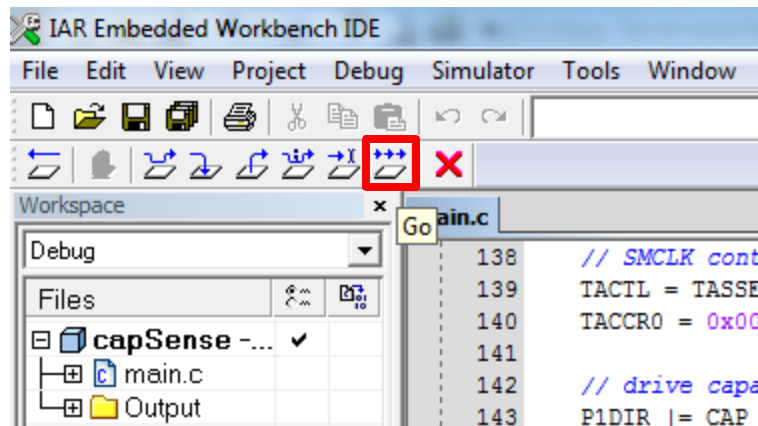
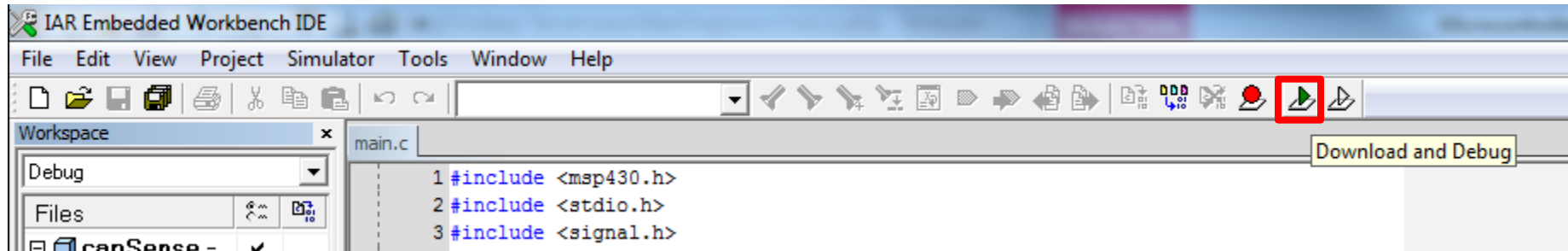
Set Project Options

- Device: MSP430G2231
- Debugger Driver: FET Debugger



Program and Run the Code

- Download and Run code on MSP 430



MSP 430 Code (Hello World)

Contains all definitions for specific device

```
#include "msp430.h" /* include MSP430 definitions */

/* **** definitions **** */
#define LED_TOGGLE_CNT 0x7FFF /* loop cycles between LED toggles */

/* pinout */
#define LED1 BIT0 /* LED1 is on P1.0 */

/** mainloop */
void main(void) {

    unsigned int cnt; /* counter variable */

    /* initialize system */
    WDCTL = WDTPW | WDTHOLD; /* disable WDT */

    /* configure LED1 as a digital output */
    P1REN &= ~LED1; /* disable pull-up/down */
    P1DIR |= LED1; /* configure as output */

    /* run mainloop */
    cnt = 0;
    while (1) { /* mainloop should never return */
        if (cnt++ == LED_TOGGLE_CNT) {
            cnt = 0;
            P1OUT ^= LED1; /* toggle LED1 */
        }
    }
}
```

MSP 430 Code (Hello World)

```
#include "msp430.h"      Constants      /* include MSP430 definitions */

/* **** definitions **** */
#define LED_TOGGLE_CNT  0x7FFF          /* loop cycles between LED toggles */

/* pinout */
#define LED1            BIT0           /* LED1 is on P1.0 */

/** mainloop */
void main(void) {

    unsigned int cnt;                  /* counter variable */

    /* initialize system */
    WDCTL = WDTPW | WDTHOLD;          /* disable WDT */

    /* configure LED1 as a digital output */
    P1REN &= ~LED1;                   /* disable pull-up/down */
    P1DIR |= LED1;                     /* configure as output */

    /* run mainloop */
    cnt = 0;
    while (1) {                        /* mainloop should never return */
        if (cnt++ == LED_TOGGLE_CNT) {
            cnt = 0;
            P1OUT ^= LED1;             /* toggle LED1 */
        }
    }
}
```

MSP 430 Code (Hello World)

```
#include "msp430.h"                /* include MSP430 definitions */

/* **** definitions **** */
#define LED_TOGGLE_CNT  0x7FFF     /* loop cycles between LED toggles */

/* pinout */
#define LED1            BIT0       /* LED1 is on P1.0 */

/** mainloop */
void main(void) {

    unsigned int cnt; Initialization /* counter variable */

    /* initialize system */
    WDICTL = WDTPW | WDTX0;        /* disable WDT */

    /* configure LED1 as a digital output */
    P1REN &= ~LED1;                /* disable pull-up/down */
    P1DIR |= LED1;                 /* configure as output */

    /* run mainloop */
    cnt = 0;

    while (1) {                    /* mainloop should never return */
        if (cnt++ == LED_TOGGLE_CNT) {
            cnt = 0;
            P1OUT ^= LED1;         /* toggle LED1 */
        }
    }
}
```

MSP 430 Code (Hello World)

```
#include "msp430.h"                /* include MSP430 definitions */

/* **** definitions **** */
#define LED_TOGGLE_CNT  0x7FFF     /* loop cycles between LED toggles */

/* pinout */
#define LED1             BIT0      /* LED1 is on P1.0 */

/** mainloop */
void main(void) {

    unsigned int cnt;              /* counter variable */

    /* initialize system */
    WDCTL = WDTPW | WDTHOLD;      /* disable WDT */

    /* configure LED1 as a digital output */
    P1REN &= ~LED1;               /* disable pull-up/down */
    P1DIR |= LED1;                /* configure as output */

    /* run mainloop */
    cnt = 0;

    while (1) {                    /* mainloop should never return */
        if (cnt++ == LED_TOGGLE_CNT) {
            cnt = 0;
            P1OUT ^= LED1;        /* toggle LED1 */
        }
    }
}
```

Mainloop – loops forever

MSP 430 vs. Arduino Code

Constant Definitions

```
#include "msp430.h"                /* include MSP430 definitions */

/* **** definitions **** */
#define LED_TOGGLE_CNT 0x7FFF      /* loop cycles between LED toggles */

/* pinout */
#define LED1 BIT0                 /* LED1 is on P1.0 */

/** mainloop */
void main(void) {

    unsigned int cnt;              /* counter variable */

    /* initialize system */
    WDCTL = WDTPW | WDTHOLD;      /* disable WDT */

    /* configure LED1 as a digital output */
    P1REN &= ~LED1;               /* disable pull-up/down */
    P1DIR |= LED1;                 /* configure as output */

    /* run mainloop */
    cnt = 0;
    while (1) {                    /* mainloop should never return */
        if (cnt++ == LED_TOGGLE_CNT) {
            cnt = 0;
            P1OUT ^= LED1;         /* toggle LED1 */
        }
    }
}
```

```
/* constants */
#define BLINK_DELAY 500           // number of milliseconds between LED toggles

/* pin definitions */
#define LED 13                    // LED is on pin 13

/* initialization code */
void setup() {
    pinMode(LED, OUTPUT);        // set LED pin as an output
}

/* mainloop - runs forever */
void loop() {
    digitalWrite(LED, HIGH);     // turn LED on
    delay(BLINK_DELAY);          // wait before turning it off
    digitalWrite(LED, LOW);      // turn LED off
    delay(BLINK_DELAY);          // wait before turning it back on
    // now return to the top of the loop
}
```

MSP 430 vs. Arduino Code

Initialization Code (run once at startup)

```
#include "msp430.h"                /* include MSP430 definitions */

/* **** definitions **** */
#define LED_TOGGLE_CNT  0x7FFF      /* loop cycles between LED toggles */

/* pinout */
#define LED1            BIT0        /* LED1 is on P1.0 */

/** mainloop */
void main(void) {

    unsigned int cnt;                /* counter variable */

    /* initialize system */
    WDTCTL = WDTPW | WDTHOLD;        /* disable WDT */

    /* configure LED1 as a digital output */
    P1REN &= ~LED1;                 /* disable pull-up/down */
    P1DIR |= LED1;                   /* configure as output */

    /* run mainloop */
    cnt = 0;

    while (1) {                      /* mainloop should never return */
        if (cnt++ == LED_TOGGLE_CNT) {
            cnt = 0;
            P1OUT ^= LED1;           /* toggle LED1 */
        }
    }
}
```

```
/* constants */
#define BLINK_DELAY     500         // number of milliseconds between LED toggles

/* pin definitions */
#define LED             13          // LED is on pin 13

/* initialization code */
void setup() {
    pinMode(LED, OUTPUT);          // set LED pin as an output
}

/* mainloop - runs forever */
void loop() {
    digitalWrite(LED, HIGH);       // turn LED on
    delay(BLINK_DELAY);            // wait before turning it off
    digitalWrite(LED, LOW);        // turn LED off
    delay(BLINK_DELAY);            // wait before turning it back on
    // now return to the top of the loop
}
```


MSP 430 vs. Arduino Code

Mainloop (runs in a loop forever)

```
#include "msp430.h"                /* include MSP430 definitions */

/* **** definitions **** */
#define LED_TOGGLE_CNT  0x7FFF     /* loop cycles between LED toggles */

/* pinout */
#define LED1            BIT0       /* LED1 is on P1.0 */

/** mainloop */
void main(void) {

    unsigned int cnt;              /* counter variable */

    /* initialize system */
    WDTCTL = WDTPW | WDTHOLD;     /* disable WDT */

    /* configure LED1 as a digital output */
    P1REN &= ~LED1;               /* disable pull-up/down */
    P1DIR |= LED1;                /* configure as output */

    /* run mainloop */
    while (1) {                    /* mainloop should never return */
        if (cnt++ == LED_TOGGLE_CNT) {
            cnt = 0;
            P1OUT ^= LED1;         /* toggle LED1 */
        }
    }
}
```

```
/* constants */
#define BLINK_DELAY     500        // number of milliseconds between LED toggles

/* pin definitions */
#define LED             13         // LED is on pin 13

/* initialization code */
void setup() {
    pinMode(LED, OUTPUT);         // set LED pin as an output
}

/* mainloop - runs forever */
void loop() {
    digitalWrite(LED, HIGH);     // turn LED on
    delay(BLINK_DELAY);          // wait before turning it off
    digitalWrite(LED, LOW);      // turn LED off
    delay(BLINK_DELAY);          // wait before turning it back on
    // now return to the top of the loop
}
```

IAR Compiler Syntax

- Must include msp430.h

```
#include <msp430.h>
```

- To specify an interrupt routine:

```
#pragma vector=WDT_VECTOR
```

```
__interrupt void WDT_ISR(void)
```

- To enable global interrupts:

```
__enable_interrupt ();
```

MSP 430 LaunchPad Demos

- **Hello World**

Blinks an LED

- **Interrupts**

Toggles one LED using timer interrupts and toggles other LED using user interrupts (when user presses a switch)

- **PWM**

LED brightness changes continuously using PWM

- **ADC**

Periodically samples voltage across light sensor and outputs brightness level using LEDs

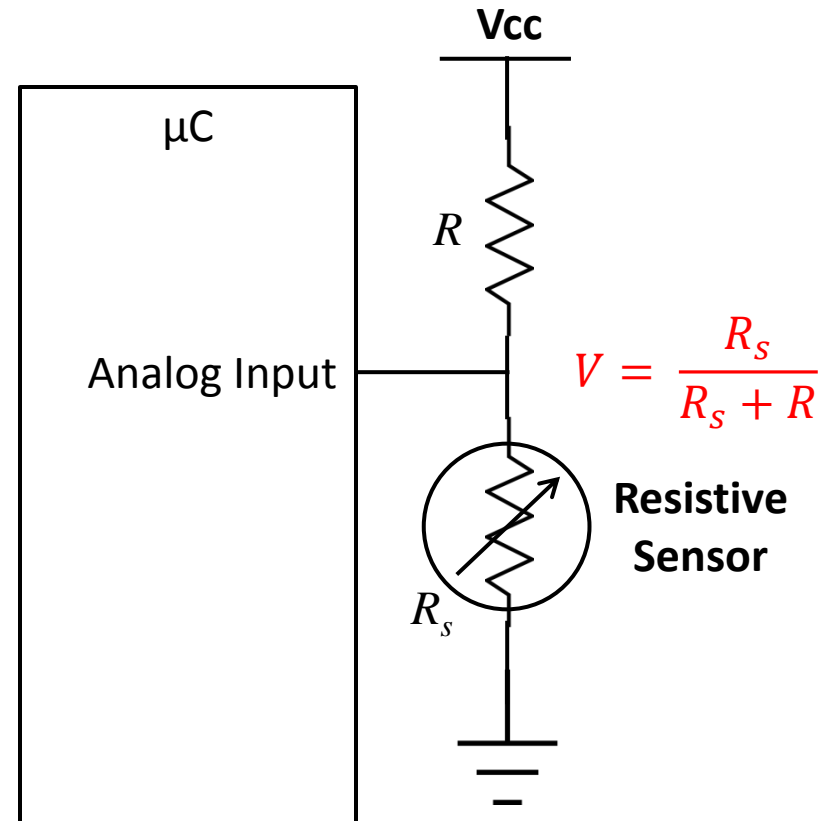
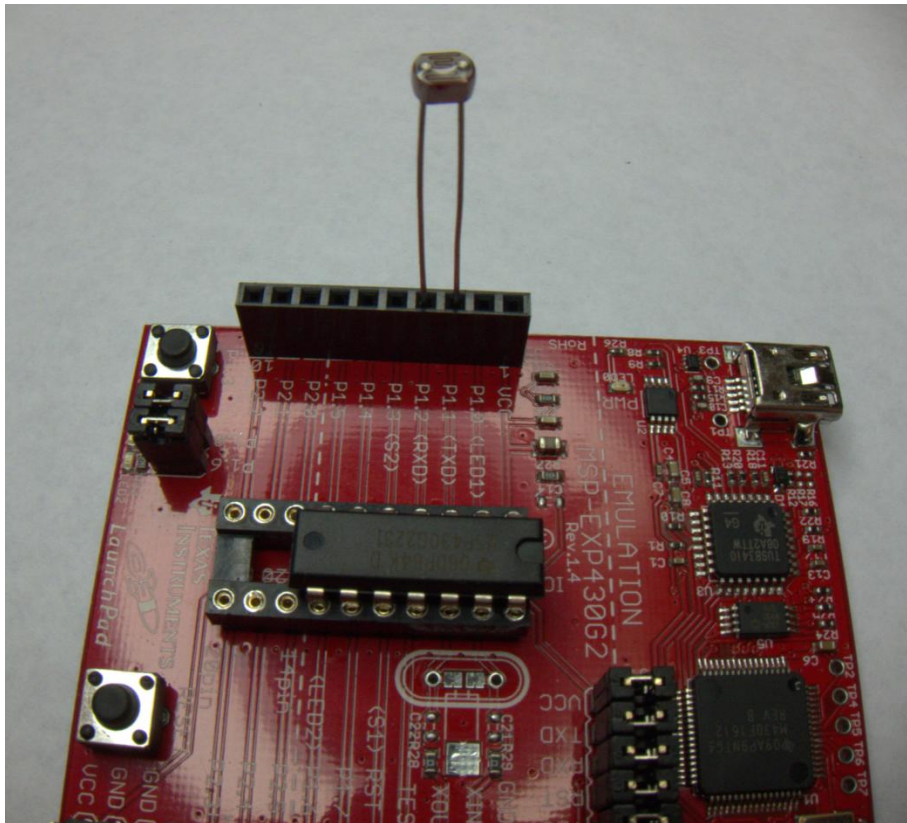
- **Capacitive Sensing**

Senses capacitance using AI foil and outputs user proximity on LED

- Code Available at: www.gabeacohn.com/teaching/micro

MSP430 LaunchPad ADC Demo

- Need to connect photo-resistor between P1.1 and P1.2



Capacitive Sensing Demo

- Capacitive Sensing in under \$5!
- Parts:
 - MSP430 LaunchPad
 - 1 M Ω resistor
 - 47 pF ceramic capacitor
 - sheet of aluminum foil
 - 1 alligator clip
 - code:
<http://blog.hodgepig.org/2010/09/16/launchpad-capacitive-sensing/>