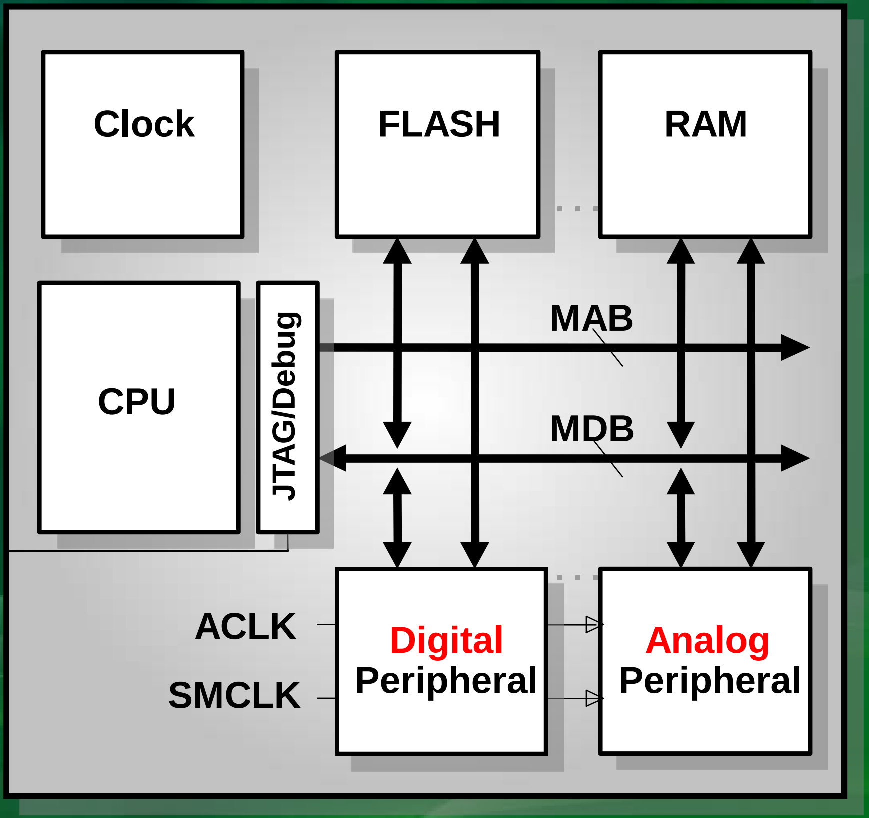
Embedded Systems

Introduction to Microcontrollers
TI MSP430 G2553

UNC CHARLOTTE

What is an MCU?

- A microcontroller is a computer:
- Has a CPU, memory, inputs/outputs
- Embedded
- Dedicated to one task & run one specific program
- Low power
- Small & low cost



UNC CHARLOTTE

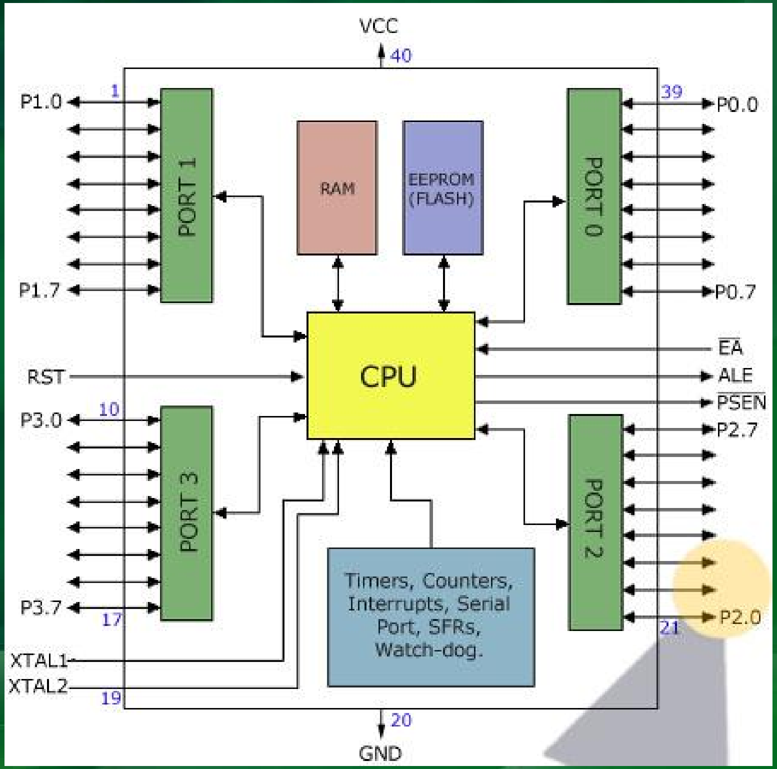
MCU vs General Purpose Computer

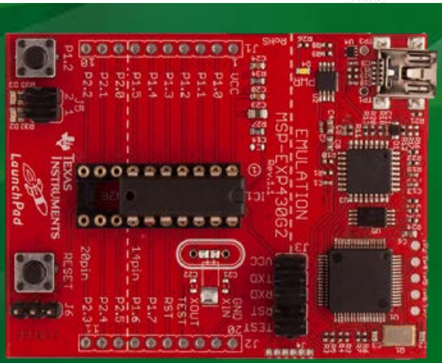
- MCU
  - Memory (RAM and ROM) all on chip
  - Peripherals (Serial Data Controllers, Analog to Digital Converters, Timers)
  - Specific Purpose (Low Power, Size Contraints, Processing Constraints)

- CPU
  - Cache Memory, RAM and ROM External
  - General Purpose

The MSP430 G2553

- 16 MHz Clock
- 16 KB Flash
- Peripherals:
  - 24 General-Purpose I/O
  - USCI_A
    - UART/LIN/IrDA/SPI
  - USCI_B
    - I2C & SPI
  - 10 bit ADC, 8 Channels
  - 2 16 bit Timers
  - Temperature Sensor
  - BootStrap Loader

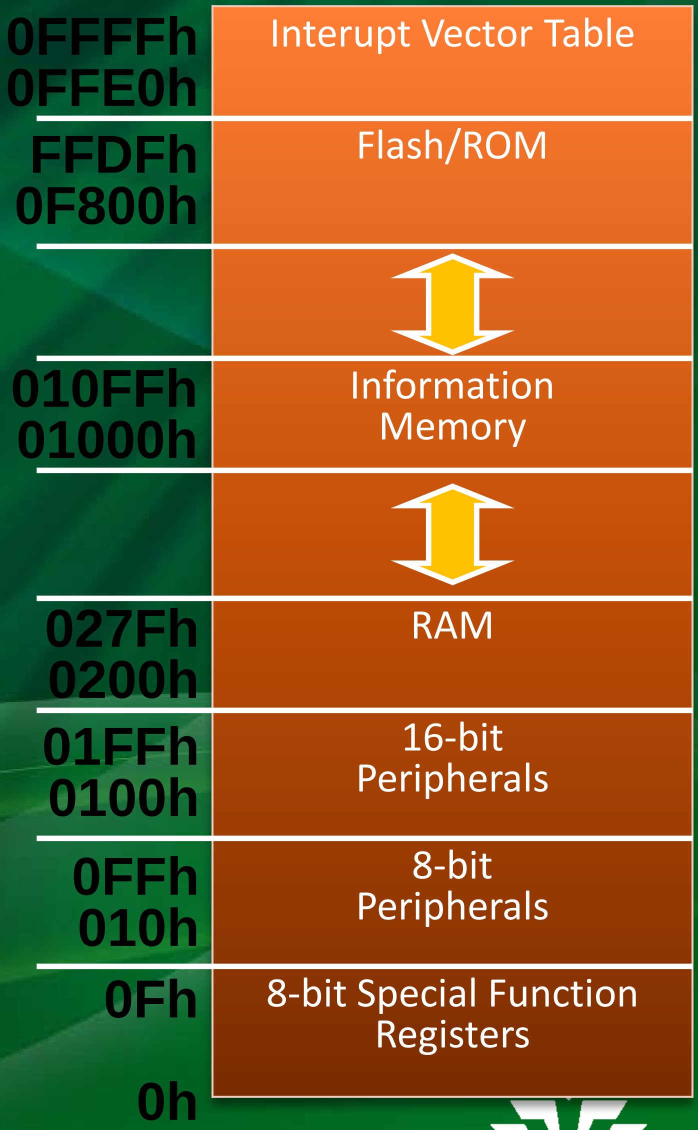




Memory Map

- Flash programmable via JTAG or In-System (ISP)
- ISP down to 2.2V. Single-byte or Word
- Interruptible ISP/Erase
- Main memory: 512 byte segments (0-n). Erasable individually or all
- Information memory: 64 byte segments (A-D)
  - Section A contains device-specific calibration data and is lockable
- Programmable Flash Memory Timing Generator

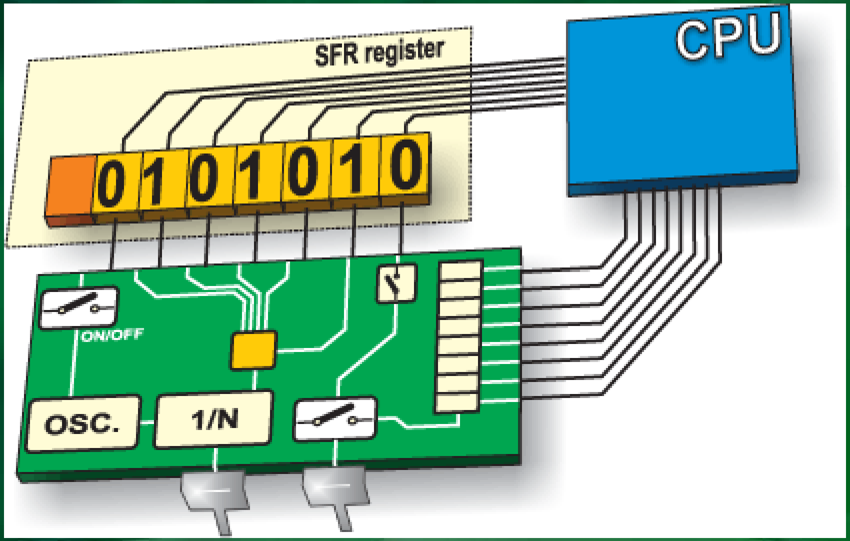
x2231 shown



0FFFFh 0FFE0h	Interupt Vector Table
FFDFh 0F800h	Flash/ROM
010FFh 01000h	Information Memory
027Fh 0200h	RAM
01FFh 0100h	16-bit Peripherals
0FFh 010h	8-bit Peripherals
0Fh 0h	8-bit Special Function Registers

Peripherals and Special Function Registers

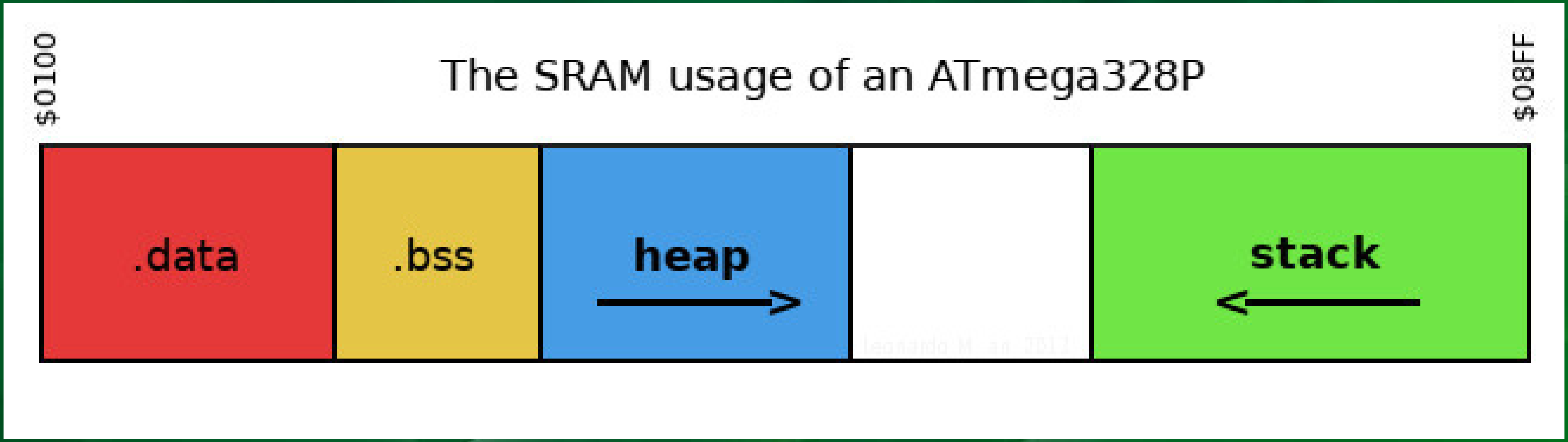
- Peripherals are configured and initialized through memory locations known as “special function registers”.
- Special Function registers may also contain the locations of the stack pointers/interrupt pointers/ program counter



UNC CHARLOTTE

RAM

- Begins at address 0x0200
- Ends at 0x027F
- Contains the stack, heap, .bss, and .data



Information Memory

- The information memory is often used to store calibrated application parameters which can be updated without affecting the code stored in the main flash memory
- Linker can be modified to use this memory as code space if needed

Flash/ROM

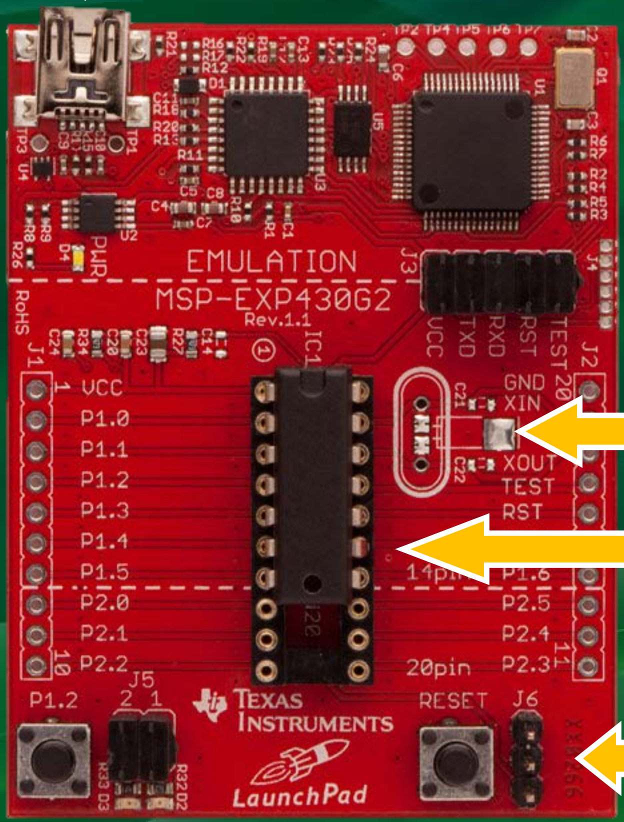
- Program Memory is stored here
- Non-Volatile Memory
  - Does not erase when power is disconnected
  - RAM is volatile memory

Interrupt Vector Table

- Used to store locations of interrupt service routines
- Called when an “event” happens to execute certain code
    - After an analog voltage is read, stop what the processor is doing, and analyze the ADC value

The LaunchPad Development Board

USB Emulator Connection



Embedded Emulation

6-pin eZ430 Connector

Crystal Pads

Chip Pinouts

Part and Socket

P1.3 Button

Power Connector

LEDs and Jumpers

P1.0 & P1.6

Reset Button

LaunchPad Features

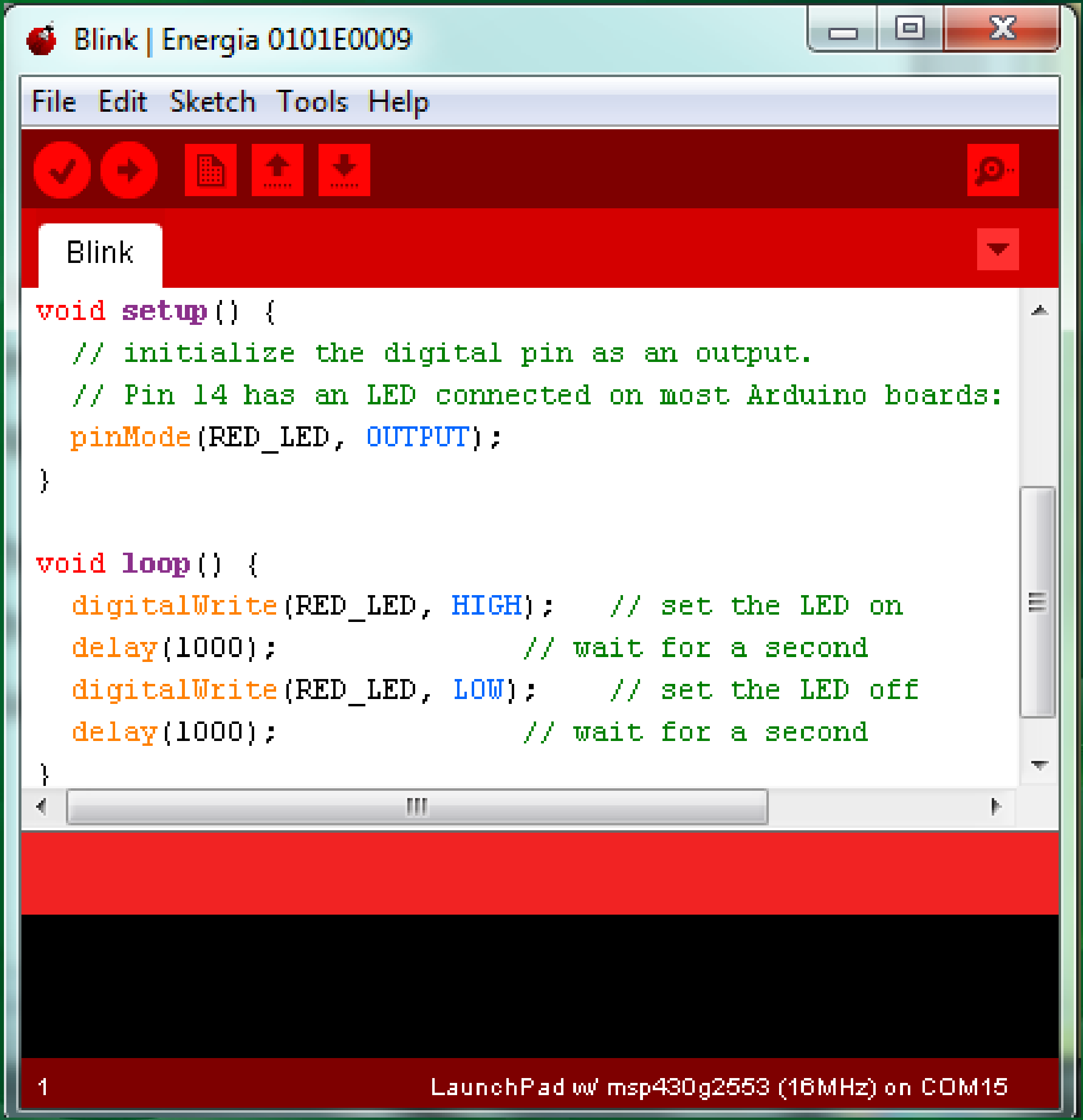
- On-Board Emulator/Debugger
  - Allows user to step through code line-by-line
- I/O Pinouts
  - Allow for breadboarding or connecting TI BoosterPacks (Sensorboards)
- LEDs, Button, and Jumpers for general purpose use
- Reset Switch
- Jumpers to bypass debugger for serial-to-USB communication

UNC CHARLOTTE

Programming the LaunchPad

- LaunchPad can be programmed using IAR, Code Composer Studio, and Energia
- IAR is a professional embedded integrated development environment (IDE) for many different microcontroller architectures
- Energia is an Arduino based IDE for the LaunchPad Boards
  - Easy to program
  - Abstracts many features from user
  - No debugger access

Energia



Blink | Energia 0101E0009
File Edit Sketch Tools Help
Blink

```
void setup() {
  // initialize the digital pin as an output.
  // Pin 14 has an LED connected on most Arduino boards:
  pinMode(RED_LED, OUTPUT);
}

void loop() {
  digitalWrite(RED_LED, HIGH);   // set the LED on
  delay(1000);                  // wait for a second
  digitalWrite(RED_LED, LOW);    // set the LED off
  delay(1000);                  // wait for a second
}
```

1
LaunchPad w/ msp430g2553 (16MHz) on COM15

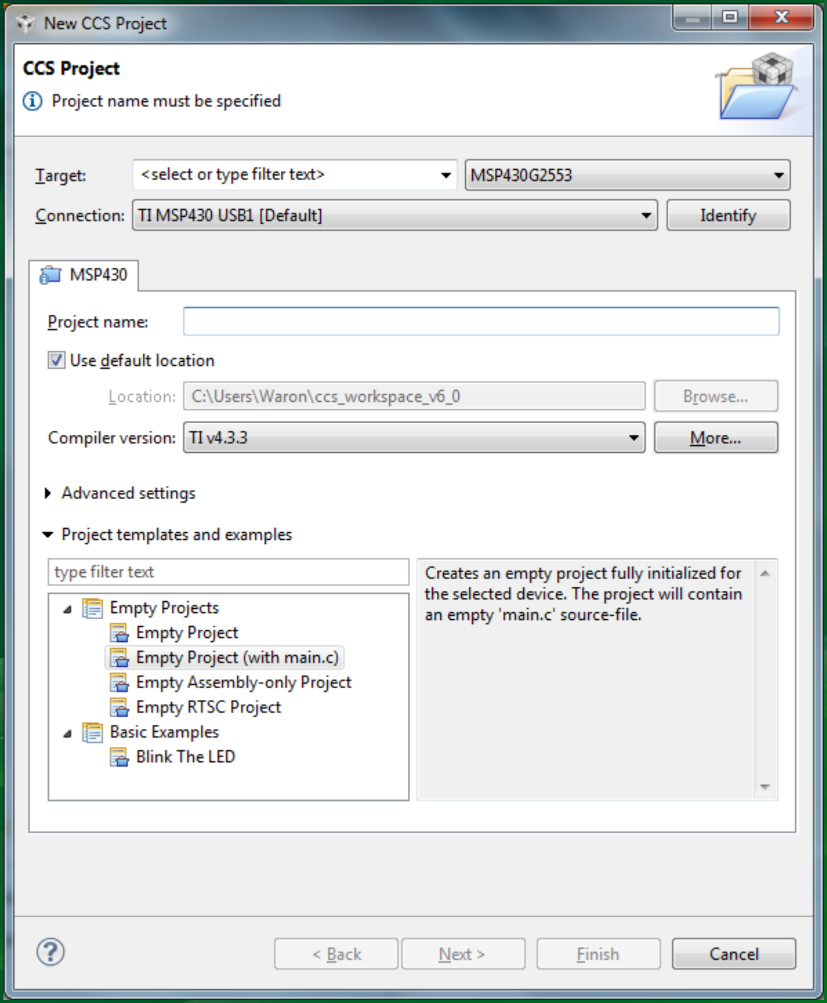
UNC CHARLOTTE

Programming the LaunchPad

- Code Composer Studio
  - TI’s custom environment
  - TI offers many software libraries and tools supported by code composer
  - More industry standard IDE unlike energia
  - Can be used to import Energia code

Code Composer Studio

- Creating a new project:
  - File -> new -> CCS Project
  - Select MSP430 G2553 in the target list
  - Name your Project
  - Empty Project (with main.c) provides a good empty code space to start, including code to disable the watchdog timer



New CCS Project

CCS Project

ⓘ Project name must be specified

Target: <select or type filter text> MSP430G2553

Connection: TI MSP430 USB1 [Default]

MSP430

Project name:

☑ Use default location

Location: C:\Users\Waron\ccs_workspace_v6_0

Compiler version: TI v4.3.3

▸ Advanced settings

▾ Project templates and examples

type filter text

- Empty Projects
  - Empty Project
  - Empty Project (with main.c)
  - Empty Assembly-only Project
  - Empty RTSC Project
- Basic Examples
  - Blink The LED

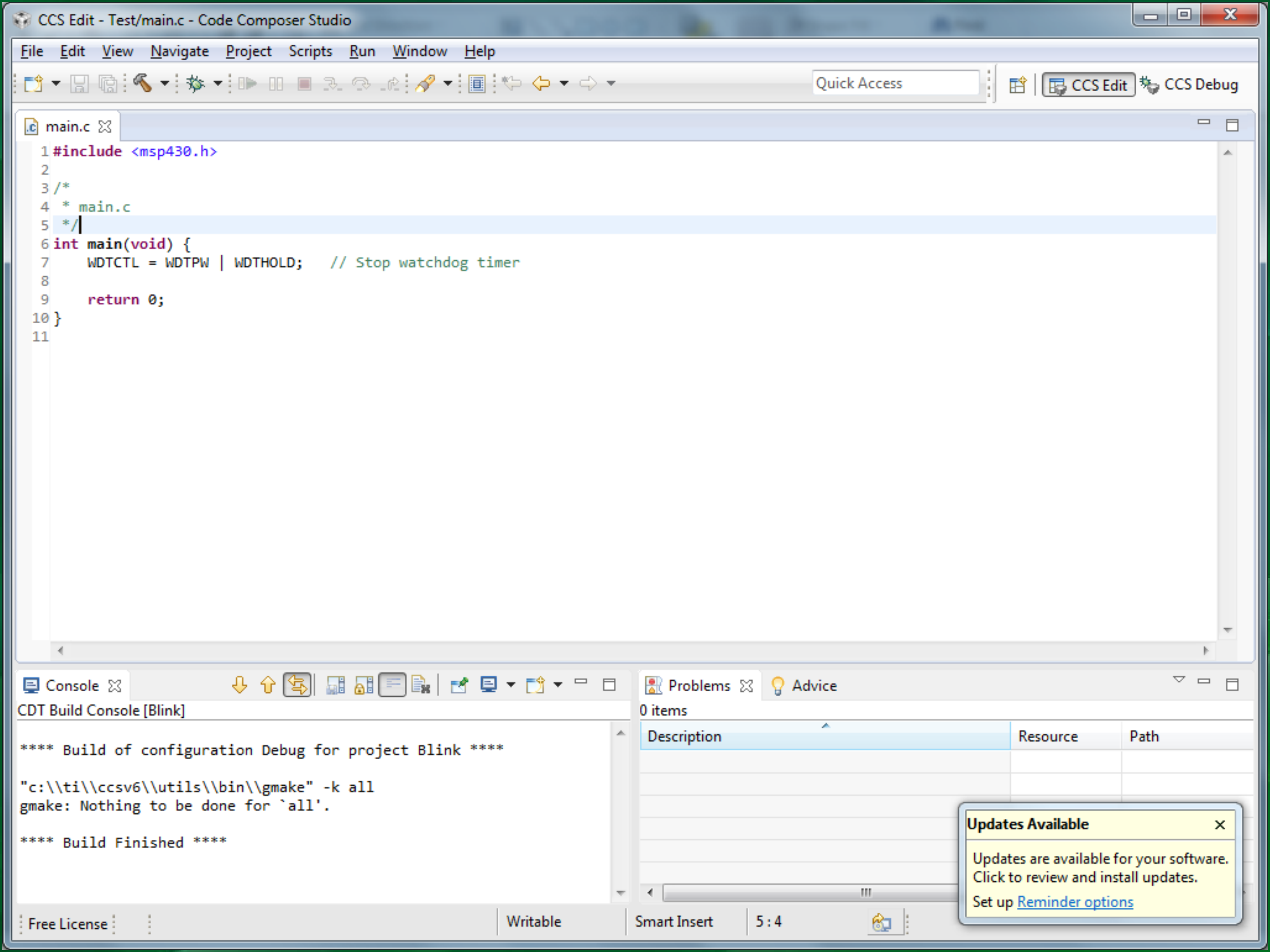
Creates an empty project fully initialized for the selected device. The project will contain an empty 'main.c' source-file.

UNC CHARLOTTE

Code Composer Studio

- Build Projects by clicking on the “hammer” icon
  - Compiles code, generates hex file
- Load programs by clicking the “bug” icon
  - Loads the program and sets the IDE into debug mode
  - Hit the green arrow “play” button to run the code

Code Composer Studio



CCS Edit - Test/main.c - Code Composer Studio
File Edit View Navigate Project Scripts Run Window Help
Quick Access
CCS Edit
CCS Debug
main.c
Console
Problems
Advice

```c
#include <msp430.h>

/*
 * main.c
 */
int main(void) {
    WDTCTL = WDTPW | WDTHOLD;    // Stop watchdog timer

    return 0;
}
```

CDT Build Console [Blink]

```
**** Build of configuration Debug for project Blink ****

"c:\\ti\\ccsv6\\utils\\bin\\gmake" -k all
gmake: Nothing to be done for `all'.

**** Build Finished ****
```

0 items

Description	Resource	Path

Updates Available

Updates are available for your software. Click to review and install updates.

Set up Reminder options

Free License | Writable | Smart Insert | 5 : 4

Lab 1 - Blink

- Assignment is found at Dr. Conrad’s webpage
- Program the MSP430 to blink the two on-board LEDs
  - You will need to find which pin are attached to each LED and button
- Deadline: September 4, 2015, 5:00 pm
  - Lab TA will hold office hours on lab checkout days from 2-4:00PM
- What to turn in?
  - Lab check-off sheet – No lab report!