Embedded Systems

Introduction to Microcontrollers
TI MSP430 G2553
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
MCU

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

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0FFFFh - 0FFE0h</td>
<td>Interrupt Vector Table</td>
</tr>
<tr>
<td>FFDFh - 0F800h</td>
<td>Flash/ROM</td>
</tr>
<tr>
<td>010FFh - 01000h</td>
<td>Information Memory</td>
</tr>
<tr>
<td>027Fh - 0200h</td>
<td>RAM</td>
</tr>
<tr>
<td>01FFh - 0100h</td>
<td>16-bit Peripherals</td>
</tr>
<tr>
<td>0FFh - 010h</td>
<td>8-bit Peripherals</td>
</tr>
<tr>
<td>0Fh - 0h</td>
<td>8-bit Special Function Registers</td>
</tr>
</tbody>
</table>
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
RAM

- Begins at address 0x0200
- Ends at 0x027F
- Contains the stack, heap, .bss, and .data
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
- Chip Pinouts
- Crystal Pads
- Part and Socket
- Power Connector
- Reset Button
- LEDs and Jumpers
  - P1.0 & P1.6
- P1.3 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
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
```cpp
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
}
```
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
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
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
```c
#include <msp430.h>

int main(void) {
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer
    return 0;
}
```
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!