Getting Started with the EasyWeb2 Hardware Platform

by Alex Milenkovich, milenkovic@computer.org

Objectives: This tutorial will help you get started with the MSP30 IAR Assembly program development on the hardware platform EasyWeb2. You will learn the following topics:

- Basic clock system (clock configuration)
- Interfacing buttons.
- Program downloading on a real platform (Olimex EasyWeb2).

Note: Required are all previous tutorials.

1. Blink a LED Using Software Delay

This section defines the problem that will be solved by the “Blink a LED” application. Your task is to write a C program that will repeatedly blink the status LED on the EasyWeb2 board every second, i.e., the LED1 will be on and off for about 0.5 seconds each.

Step 1: Analyze the assignment.
In order to better understand the assignment we will first study schematics of the board. This board includes a TI’s MSP430 microcontroller (MSP430F149), Ethernet controller (CS8900A), serial RS232 port, a status led (LED6), a buzzer, four buttons (B1-B4), and several extension slots that allow an easy access to all microcontroller ports. Detailed schematic of the board is provided in the following document: http://www.ece.uah.edu/~milenka/cpe323-08F/docs/easyweb2-sch.gif.
Open the easyweb2-sch.gif file, zoom in and study the schematic. Please locate the status LED output port. It is connected to port P2.1.

Step 2. Develop a plan.
From configuration it is clear that if we want LED6 on, we should provide a logical ‘0’ at the output port of the microcontroller (port P2.1), and a logical ‘0,’ if we want LED6 to be off. We could take several approaches to solving this problem. The simplest one is to toggle the port P2.1 and have 0.5 seconds delay in software.

After initializing the microcontroller, our program will spend all its time in an infinite loop (LED6 should be repeatedly blinked). Inside a loop we will toggle the port P2.1 and then wait for approximately 0.5 seconds.

2. Blink a LED Project: C Code

1. Figure 1 shows a C program implementing blinking the status LED using software delay.
2. Notes: Upon power-on reset the MSP430F149 clocks MCLK and SMCLK are supplied internally from the DCO. The default clock frequency is approximately 800 KHz.
#include <msp430x14x.h>

void main(void)
{
    WDTCTL = WDTPW + WDTHOLD;         // Stop watchdog timer
    P2DIR |= 0x02;                     // Set P2.1 to output direction (0000_0010)
    for (;;)
    {
        unsigned int i;
        P2OUT ^= 0x02;                   // Toggle P2.1 using exclusive-OR
        i = 50000;                       // Delay
        do { i--; }
        while (i != 0);
    }
}
3. Assignments

Assignment #1: Blinking the Status LED.
Write a C program that toggles the status LED on the EasyWeb2 development board with frequency of 0.5 Hz (approx. 1s is on, and 1s is off). Use the high frequency crystal for MCLK (8 MHz) and ACLK (1 MHz), and watchdog timer in the interval-timer mode. The watchdog timer is clocked by ACLK. Change parameters to increase toggle frequency to 2Hz and to decrease it to 0.25Hz.

Assignment #2: Interfacing buttons on the MSP430 EasyWeb2
Write a C program that scans the buttons of the EasyWeb2 board and responds as described below. The buttons B1, B2, B3, and B4 are connected to P4.4 – P4.7.
- If the yellow button (B1) is pressed (port P4.4), blinks the status LED (port P2.1) with frequency of 4 Hz.
- If the red button (B2) is pressed (port P4.5), toggle Relay 2 (connected to the port P1.6).
- If the white button (B3) is pressed (port P4.6), toggle Relay 1 every 5 seconds. Relay 1 is connected to the port P1.5.