

Lab 2 - For-Loops, While-Loops, If-Then Branching, Subroutines, and Time Delays

# Preparation

* Make a copy of your Lab 1 and rename it Lab 2
* [Watch the related video(s) here](https://www.youtube.com/playlist?list=PLxYT_0WQMg35FiE5EDnCwW-1p7-E3gNtJ)

# Purpose

The purpose of this lab is to learn simple programming structures in assembly. You will also learn how to estimate how long it takes to run the software, and use this estimation to create a time delay function. You will learn how to use the clock profiling feature in CCS to estimate the time delay. Assembly software skills you will learn include masking, toggling, if-then, subroutines, and looping.

# System Requirements

The system has two input switches (a.k.a., pushbuttons) and two output LEDs (see Figure 1). Figure 2 shows the port signals when switch 2 is pressed and released (NOTE: for this lab, we will use SW2 and LED1). A negative logic switch means the P1.4 signal will be 1 (high, 3.3V) if the switch is not pressed, and the P1.4 signal will be 0 (low, 0V) if the switch is pressed. A positive logic red LED interface means if the software outputs a 1 to P1.0 (high, 3.3V) the LED will turn ON, and if the software outputs a 0 to P1.0 (low, 0V) the red LED will be OFF. In this lab, you will first debug on the real board using the switch and LED on the LaunchPad itself, but no external hardware will be required. The overall functionality of this system is described in the following rules:

1. Make P1.0 an output and make P1.4 an input (enable pull-up for P1.4)
2. The system starts with the LED ON (make P1.0 =1)
3. Delay for about 100 ms (using a loop(s))
4. If the switch is pressed (P1.4 is 0), then toggle the LED once, else turn the LED ON
5. Repeat steps 3 and 4 over and over

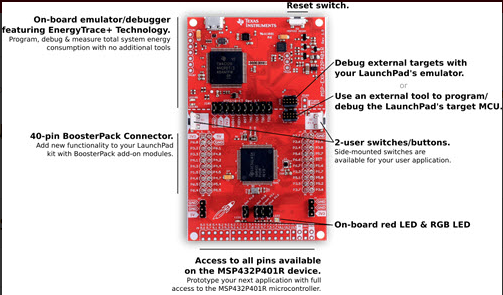


Figure 1: MSP432 LaunchPad.

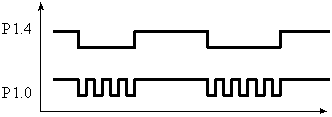


Figure : The output oscillates when the switch is pressed. The output is high when the switch is released.

Time is very important in embedded systems. One of the simplest ways in which we manage time is determining how long it takes to run our software. There are two ways to determine how long each instruction takes to execute. The first method uses the ARM data sheet. For example, the following is a page from the Cortex-M4 Technical Reference Manual (e.g., see pages 3-6 to 3-10 of <http://users.ece.utexas.edu/~valvano/EE345L/Labs/Fall2011/CortexM4_TRM_r0p1.pdf>).

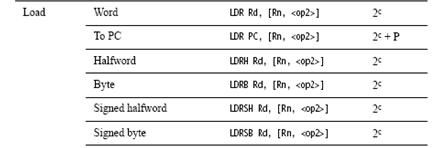


Figure : From the Technical Reference Manual page 36.

On the MSP432 the default bus clock is 3 MHz ±0.5%. Later we will use the clock system, and the bus clock will be exactly 48 MHz. For now, however, we will run at about 3 MHz. The following is a portion of a listing file with a simple delay loop. The SUBS and BNE are executed 900 times. The SUBS takes 1 cycle, and the BNE takes 1 to 3 (a branch takes 0 to 3 cycles to refill the pipeline). The minimum time to execute this code is 900\*(1+1)/3 μs = 600 μs. The maximum time to execute this code is 900\*(1+4)/3 μs = 1500 μs. Since it is impossible to get an accurate time value using the cycle counting method, we will need another way to estimate execution speed. We will use the clock profiling feature of CCS to estimate the clock cycles (see the “Profiling the Clock” slides in Lecture 5).

mov r0,#900

wait subs r0,r0,#0x01

bne wait

# Procedure

The basic approach to this lab will be to develop and debug your system using a negative switch (P1.4) and a positive logic LED (P1.0) on the MSP432 LaunchPad board.

1. Design a subroutine that delays about 100 ms. First, draw a flowchart, and then write pseudo code. You will call the subroutine with a BL instruction and return with a BX LR instruction. Any delay from 8 to 12 ms is OK. To implement a delay, you could set a register to a large number, and then count it down to zero. With a bus clock of 3 MHz, there are 3,000 bus clock cycles in 1 ms. You need to know how long it takes to execute the loop once, then determine the number of times you need to execute the loop to create the 100 ms delay.
2. Write a main program in assembly that implements the input/output system. The pseudo code and flowchart are shown in Figure 4, illustrating the basic steps for the system. I recommend, at this early stage of your design career, that you access the entire I/O port using P1OUT. After you fully understand how I/O works, then you can use bit-banded addressing to access I/O ports (we will discuss how to do this later).

main Initialize Port 1:

Set the Port 1 direction register, so P1.4 is an input and P1.0 is an output

Set bit 4 of P1REN to enable the internal register for P1.4

Specify this resistor as a pull up by setting bit 4 of P1OUT

Set P1.0, so the LED is ON

loop Delay about 100 ms

Read the switch and test if the switch is pressed

If P1.4=0 (the switch is pressed), toggle P1.0 (i.e., flip from 0 to 1, or vice versa)

If P1.4=1 (the switch is not pressed), set P1.0, so the LED is ON

Go to loop

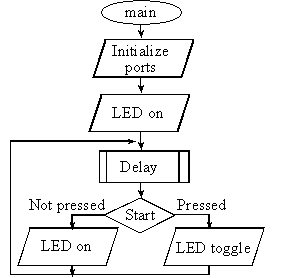


Figure : Flowchart of Part b.

1. Use the clock profiling feature in CCS (see Lecture 5) to measure the delay function.
   1. **NOTE: the black MSP432 LaunchPad boards cannot be used to clock profile. If you don’t have a red board, you will need to borrow one for this part.**
2. Load your software onto the board and test it. If the P1.4 switch is not pressed, then the red LED is on. If the P1.4 switch is pressed, then the red LED toggles causing it to blink. [Here is a video of how it should function](https://www.youtube.com/watch?v=d88LRcatZb0&list=PLxYT_0WQMg35FiE5EDnCwW-1p7-E3gNtJ&index=4).

# Demonstration

You will show the instructor your program operation on the real board. Be prepared to explain how the delay function works. How would it be different if the delay were 1 ms instead of 100 ms? The instructor will pick an instruction from your code and ask you which addressing mode it uses. Execute the program step-by-step and run to cursor.

# Lab Report

The lab report is how I will grade you on your labs. Usually, the report is due 1 week following the completion of the lab (see TITANium for due dates). However, due to unforeseen circumstances, due dates may change. I will try my best to keep everyone informed of any changes. With this said, it is your responsibility to turn the report in during the scheduled due date. **There is a 10% penalty for late reports**. **Your report must be in MS Word Doc format. Submitting the report in another format will result in a 10% penalty.** Your lab report must include the cover sheet from the lab report template available on TITANium. **Not including the cover sheet will result in a 10% penalty.** The template contains instructions for the report and the rubric used for grading the labs. Please, read it thoroughly. **Don’t forget to include pertinent information such as code, flowcharts, waveform output, etc. in your report**. Please, no “spaghetti” code, keep your code clean and use comments. Remember, your code will affect your lab grade. If I can’t understand it, then I will assume it’s incorrect.