# ECE2049 A-Term 2022

# Lab #0 -- Sign-off Sheet

# 

***Report due***: **Wednesday 09/07/2022**

**Student 1:** Cristobal Rincon Rogers

**Student 2:** Lili Loughlin

**Board #:** 61

| ***Task Max points*** | ***Max***  ***points*** | ***TA’s assessment*** |
| --- | --- | --- |
| Build and demonstrate blink.c tutorial | 10 |  |
| Download, build and demonstrate MSP430 development board demo | 10 |  |
| Demonstrate modifieddemo project | 10 |  |
| Answer to TA Questions | 5 | Student 1    Student 2 |
| Report  (Include answers to all questions and code submitted on-line) | 15 |  |
| ***Total points*** | **50** |  |

1. **Introduction**

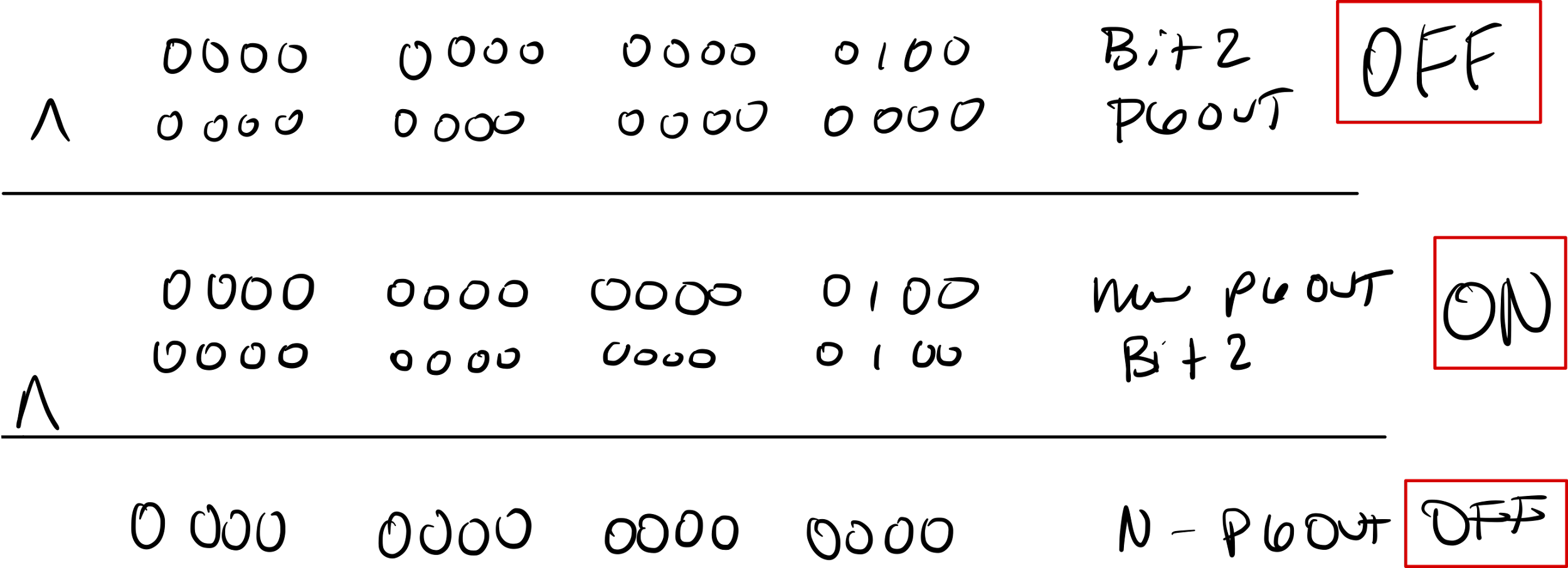
Our objective in this lab was to program and debug our custom MSP430F5529 Launchpad-based Lab Board using Code Composer Studio (CCS) 10.4. In this lab, we tested a pre-written C program that controlled various LEDs on the board. To run this program titled *blink.c*, we had to create a new project, import, build, and run in debug mode. CCS allows you to write to various processor architectures, so we needed to select the correct processor.

Secondly, we learned how to manipulate output using different bitwise operators. Lastly, we learned how to write to an LCD screen and collect input using the keypad onboard.

1. **Discussion/Questions**

Questions to answer:

Blink Portion of project

* Explain how and why this works in your report.
  + 
  + The xor (^) inverts the output of Port 6, pin2.
* Document which logic level lights LED1 in your report.
  + A logic level of 1 in the 3rd-bit lights LED1.

Keypad Demo

1. Document what controls the position of text on the screen in your report.
   1. The LCD has a set pixel dimension. Function Graphics\_drawStringCentered() determines the position of the text. When we alter the pixel coordinates, text translates vertically and or horizontally.
2. Write your name to the LCD. Do you need to include a NULL terminator (‘\n’) as the last element of your name array or not?
   1. No, a NULL terminator isn’t needed as the last element of the name array. A NULL terminator is used to indicate a new line, which isn’t necessary in this one-line case.

3. Answer the following:

* 1. What size does the Code Composer MSP430 Compiler use for the following types?
     1. float → 32 bits
     2. int → 16 bits
     3. long int → 32 bits
     4. char → 8 bits
  2. What value is stored in myGrade (what is the ASCII code for ‘A’?)
     1. The ASCII code value for ‘A’ is 65.
  3. What is the new value of test?
     1. The new value of the variable test is 1536 (0x0600 in decimal).

4. When does the buzzer sound? More importantly, what turns it off?

* 1. The buzzer sounds when you press the asterisk (\*) and turns off when you press the pound (#) symbol.

1. What is the relationship between the keypad and the 4 colored LEDs? Does the keypad function return ASCII or integer values? Explain in your report.
   1. The keypad returns ASCII values. To convert from ASCII representation of decimal numbers, we subtract by 0x30. So for instance, if we press 0 on the keypad, the microcontroller receives a value of 0x30. So if we subtract these two, we get zero. Once we get the binary representation of the number pressed on the keyboard, we turn on the LEDs so that they represent the number pressed in binary. As a second example, say the user pressed the value of 9. The controller would receive a value of 0x39. After subtracting by 0x30, we have the value as 0x09. The binary representation of 0x09 is 0000 1001. Therefore, the placement of logic 1s indicates which LEDs are lit up. So we know that the fourth LED is lit and the first LED is lit.

**Conclusion**

In conclusion, in this lab, we learned how to program and debug our custom MSP430F5529 using Code Composer Studio. The first program we ran on the controller was the *blink.c* file. In this file, we controlled the output of a pin that was connected to an LED using bitwise operations. We experimented with three different bitwise operations. The first was using the xor (^) operation, where we xor’ed the output of pin6 with BIT2, which is a constant defined in MSP430F5529 as 0x0004, 0000 0000 0000 0100b. The output of port 6 at startup is 0000 0000 0000 0000. When we xor this value, we get a new output equal to BIT2, 0000 0000 0000 0100b, turning the LED on. Therefore, the third bit controlled the output of port 6.

The other bitwise operators (| and &) resulted in either the LED being permanently off or permanently on. The code including the | (OR) operation, resulted in the LED being permanently ON, whereas the code containing the & operation resulted in the LED being permanently OFF.

Lastly, we learned about the conversion from ASCII numbers to their integer representations. When pressing a number on the keypad, we would subtract 0x30 from the pressed number. This works because 0x30 is the hex value for the ASCII number of zero. So subtracting (0x30 - 0x30) results in zero.