Jessie Acfalle

HW6

ECE 2534

10/25/18

Problem 1. HW6_PWM

1) Read the entire starter code that is not "gray" in the CCS. This is the active part of your code chosen by the compiler using the #ifdef directive. This code uses Timer32 and a simple FSM to generate a PWM signal on the blue LED on the BoosterPack board. Compile and debug (or flash) your program. You should see a light that is mostly on and only blinks briefly. Answer the following questions:

   a. What is the period of the PWM in seconds? **1s**

   b. What is the period of the PWM in counter cycles? Show the calculations based on a source clock of 3MHz and a prescaler of 1. **3000000/1 = 3000000**
   c. What is the duty cycle in percentage? **90%**
   d. What is the duty cycle in counter cycles? **0.9**
   e. What is the length of time (in seconds) that the blue LED is off during each PWM period? **0.1s**

2) Change the DUTY_CYCLE_FRACTION macro to 0.1. Recompile and debug (or flash) your code again. Answer the below questions:
   a. Describe the change in blue LEDs behavior in a few sentences.
      The blue LED appeared to be on less when we changed the duty cycle fraction to 0.1 instead of 0.9.
   b. What is the period of the PWM in seconds? **1s**
   c. What is the duty cycle in percentage? **10%**
   d. What is the duty cycle in counter cycles? **300,000**
   e. What is the length of time (in seconds) that the blue LED is off during each PWM period? **0.9s**

3) Change the BLOCKING macro to 1 and keep the DUTY_CYCLE_FRACTION macro set to 0.1. Recompile and debug (or flash) your code again. Answer the following questions:
   a) Do you observe any change in the blue LED's behavior? If yes, describe it in a few sentences.
      The blue LED stays on longer when the changes are implemented.
   b) In a few sentences, explain the reason behind your answer to part a.
      Those changes occurred because when blocking is set to 1, the "ON" state becomes longer and disturbs the correct duty cycle.

4) Put // in front of the TIMER32 macro definition, while keeping other macros set as in the previous step. This will make the entire region of the code using Timer32 gray and activate the region that uses Timer_A. Recompile and debug (or flash) again. Answer the following questions:
   a. What is the period of the PWM in seconds? **1s**
   b. What is the period of the PWM in counter cycles? Show the calculations based on a source clock of 3MHz and a prescaler of 64.
   3000000/64 = 46875
   c. What is the duty cycle in percentage? **90%**
   d. What is the duty cycle in counter cycles? **2,700,000**
   e. What is the length of time (in seconds) that the blue LED is off during each PWM period? **0.1s**

5)) Change the DUTY_CYCLE_FRACTION macro to 0.1, again keeping macros set as in the previous step. Recompile and debug (or flash) your code again. Answer the below questions:
   a. Describe the change in blue LEDs behavior in a few sentences.
      The blue LED is now ON for a shorter amount of time.
   b. What is the period of the PWM in seconds? **1s**
   c. What is the duty cycle in percentage? **10%**
   d. What is the duty cycle in counter cycles? **4687.5**
   e. What is the length of time (in seconds) that the blue LED is off during each PWM period? **0.9s**

6) Change the BLOCKING macro to 1. Recompile and debug (or flash) your code again. Answer the below questions:
   a. Do you observe any change in the blue LED's behavior? If yes, describe it in a few sentences. **No, I didn't observe a difference in the blue LED.**
   b. In a few sentences, explain the reason behind your answer to part a. In other words, answer one of these two questions: 1) If you observe any change, what is causing it? or 2) Why you see no changes? **There was no difference since Timer_A automatically handles the pulse and the processor is hands free.**

Problem 2. HW6_RGB

1) In this project, you will work with Timer_A again and learn more about PWM. Read the entire code and answer the following questions (type your answers in the same HW6.* file you started for problem 1.

a. What is the prescaler value used for Timer_A? **64**

b. Which Timer_A does the blue LED? 0, 1, 2, or 3? **A2**

c. Which channel (register) in Timer_A is the blue LED is connected to? 0, 1, 2, 3, 4, 5, or 6? **Register 1**

d. Which port and pin is the blue LED connected to? **port 2 pin 1**

e. What is the frequency of the PWM signal driving the blue LED? **1/3000**

f. What is the load value used in Timer_A? **0x04**

g. You might notice the light is very faint. What can you do to intensify the light? Change your code, compile and run and see if it intensifies the light. Copy paste the name of the macro and the value you changed it to and include that in your solution.

#define ONE_MILLISEC_COUNT 3000 (macro needed to be altered)
If we change it to 3,000,000 the light will intensify.

2) Update the starter code such that the Boosterpack red LED is turned on using PWM at 50% intensity. Furthermore, decrease the intensity of the blue LED to 20%. If you are eyes are sensitive to colors, you should be able to see a shade of magenta. Changing the intensity of these two lights gives you different mix or "red" and "blue". Stage and push the changes to GitHub at this point.

Problem 3. HW6_ADC

1. Push the left button. What color is the LED? What happened to the circle?
   When you initially push the left button, the ball moves from start towards the left of the screen, the number of moves increases by one, and the color switches to GREEN.
2. Push the left button again. What color is the LED? What happened to the circle?
   When you push the button again, the LED changes to BLUE, and the circle moves from the current position towards the left of the screen.
3. Push the right button. What color is the LED? What happened to the circle?
   When the right button is pressed, the LED changes back to GREEN and moves from the current location towards the right of the screen.

4. Reset the program (it can be done using the reset button or using CCS), push the joystick to the left and keep it there. What happens to the circle? What happens to the LED lights? How do you explain the color of the LED? (Get help from your work on Problem 2).

When you hold the joystick in the left position the ball moves all the way to the left edge of the screen and the LED changes between red, green, and blue rapidly causing a whiteish color to flash. The LED changes color rapidly due to the fact the x position of the joystick held in the left position will always be less than the

threshold macro, which makes joyStickPushedtoLeft true, thus modifying the LED color at a rapid speed.

5. Push the joystick to the right. What happens to the LED? What happens to the light? (The answer is nothing!). Change the code such that pushing the joystick to the right moves the circle to the right like the behavior in step 4 for pushing the joystick to the left.

Stage and push the changes to GitHub at this point. Also, make sure to put your HW6.pdf (consisting of all your answers) in HW6_PWM folder and staging and pushing it on CCS from HW6_PWM project.