Milestone 1 PWM Questions

Matthew Bass

16. After working through this process, you will need to address the following questions:  
  
1. What does pwmled2 set the PWM period to?

It sets it to a period of 3000 ms.  
  
2. Which PWM\_xxx() function sets the PWM period.

PWM\_setPeriod() sets the PWM period.  
  
3. Which PWM\_xxx() function sets the PWM duty cycle.

PWM\_setDuty() sets the PWM duty cycle.  
  
4. What is the purpose of the while(1) loop in pwmled2?  
  
 The purpose of the while(1) loop is to continuously update the duty cycle of the PWM signals and   
control the LEDs connected to those PWM outputs. The loop performs the following actions repeatedly:

1. Sets the duty cycle of pwm1 and pwm2 using PWM\_setDuty().  
2. Increments the duty value by dutyInc.  
3. Checks if the duty has reached the maximum value (pwmPeriod) or becomes zero.   
 If so, it negates the dutyInc to change the direction of duty cycle increment/decrement.  
4. Sleeps for time microseconds using usleep() to control the rate at which the duty cycle changes.  
  
By using an infinite loop, the code ensures that these steps are repeated continuously,   
resulting in the LEDs exhibiting a pulsating or fading effect as the duty cycle of the PWM signals is adjusted.  
The presence of while(1) loop in this code implies that the application is intended to run indefinitely until it is manually terminated.  
  
5. What is the purpose of usleep() in the while(1) loop?  
  
The usleep() function is used to introduce a delay or pause in the execution of the program for a specified number of microseconds. In the context of the while(1) loop in this code, the usleep() function is responsible for controlling the rate at which the duty cycle of the PWM signals changes. After updating the duty cycle of the PWM signals and determining the next duty value, the program sleeps for time microseconds using usleep(time) before repeating the loop. This sleep period introduces a pause in the execution,   
causing a delay between each iteration of the loop. By adjusting the value of time, you can control the duration of each duty cycle before it changes again. A smaller value of time will result in a faster change in duty cycle and thus faster pulsation or fading effect of the LEDs, while a larger value will slow down the change. In summary, the purpose of usleep() in the while(1) loop is to control the timing and create a desired time interval between each iteration, effectively controlling the speed of the LED pulsation or fading effect.