ME 333
Assignment 7 and 8 - PI Control of LED/Phototransistor Pair

Overview

For this assignment, you will be controlling the light emitted from and received by an LED/phototransistor pair. There are many steps in the process, so we will be breaking it up into 4 main parts:

1. Implement a PWM output circuit that will drive the LED at a given frequency.
2. Implement a timer that will cycle through an array and update a current reference value of desired light level.
3. Set up a phototransistor circuit to read analog voltage levels and then convert these levels to digital values.
4. Implement a proportional-integral controller that will take feedback from the phototransistor circuit, calculate an error from the current reference value, and adjust the LED PWM accordingly.

Step 1 - In class 2/11

- Wire the LED in series with digital output D0 and a 330 ohm resistor.
    - The long leg of the LED will go into D0, the short leg will go to the 330 ohm resistor, and the 330 ohm resistor will go to ground.
- Setup Timer2 on the NU32 to interrupt every 0.1 seconds. To set the timer to interrupt you must:
    - Set the correct timer prescaler (TCKPS bits in T2CON)
    - Set the input to PBCLK (TCS bit in T2CON)
    - Turn the timer on (ON bit in T2CON)
    - Set the period register so the timer rolls over every 0.1 seconds (PR2)
    - Reset the Timer2 count (TMR2) to 0
    - Set the Timer2 interrupt priority and subpriority bits
    - Clear the Timer2 interrupt flag
    - Enable Timer2 interrupt

- Within the interrupt, invert the current value of D0 (This will toggle the output LED).
    - This should create a very simple 5 Hz PWM signal that has a 50% duty cycle. That means that it spends half of its time high, the other half low, and that it repeats itself once per cycle.

- Hook up your nScope on channel A to verify your timing is correct.

Exercises:

1. Using this timer based PWM, generate a 50% duty cycle PWM signal running at 100Hz, 1kHz, and 10kHz. Verify these times on the nScope and take a snapshot of each.

2. What are the values of the timer prescaler, N, and period register, PR2, you used in question 1?

3. What is the slowest 50% duty cycle PWM signal you can generate using this method, and what prescaler would you use to create it?