

ENSC 254 LAB 4 Report

Effects of Duty Cycle & Frequency on a LED

Authors: Johnston Yang & Luke Gair

Date: June 18th 2018

Data Collected on June 14th & 15th

Abstract:

The Purpose of this lab was to explore and familiarize ourselves with the Xilinx FPGA and SDK (Field-Programmable Gate Array & its Software Development Kit) through programming in ARM and observing the relative effects that our program had on the board. There were several tasks that we completed throughout the lab. We created a program so that the LED flashes at a 50% Duty Cycle with a period of two seconds (0.5 Hz). We also experimented with other duty cycles e.g. 5% at 0.1 Hz & 80% at 0.2 Hz.

At the end of this lab we should be able to understand the effects that duty cycle has on what frequency is needed to display a continuous light on the LED, & how it effects the LED's brightness at higher frequencies.

Materials:

XILINX SDK

XILINX FPGA (ZedBoard)

ARM 7.0

Code Used for the Experiment:

```
.global asm_main
```

```
 ;@ Duty cycle is ratio of time on to time off ON/OFF
```

```
asm_main:
```

```
    ldr r0, =0x41210000      ;@ r0 is address of LED's
    ldr r5, =0x41220000      ;@ r5 is switch addresses
    ldr r1, =0                ;@ r1 is value of LED we are turning on
    ldr r2, =0                ;@ r2 is the count
    ldr r3, =#10500           ;@ OnCycle Count 1050000 is approx 1s
    ldr r4, =#15750           ;@ OffCycle Count
```

```
OnCycle:
```

```
    ldr r1, =0b10000000
    str r1, [r0]
    add r2, r2, #1
    cmp r2, r3
    beq MidCycle1
    b OnCycle
```

```
MidCycle1:
```

```
    mov r2, #0
    b OffCycle
```

```
MidCycle2:
```

```
    mov r2, #0
    b OnCycle
```

```
OffCycle:
```

```
    ldr r1, =0b01000000      ;@ write to LED 6 to compare with L7
    str r1, [r0]              ;@ reset counter
    add r2, r2, #1
    cmp r2, r4
    beq MidCycle2
    b OffCycle
```

Data:

PART 6 - Effects of Changing Duty Cycle and the Resulting Frequency

Duty Cycle	Frequency (Hz)
99%	5
97%	10
95%	14
90%	22
88%	24
85%	26
80%	28
75%	30
70%	32
65%	34
63%	35
60%	36
55%	37
50%	38
45%	39
40%	39
37%	39.5
35%	39.5
30%	39.7
25%	39.8
20%	39.8
15%	39.85
12%	39.85
10%	39.9
5%	39.9
3%	39.95
1%	40

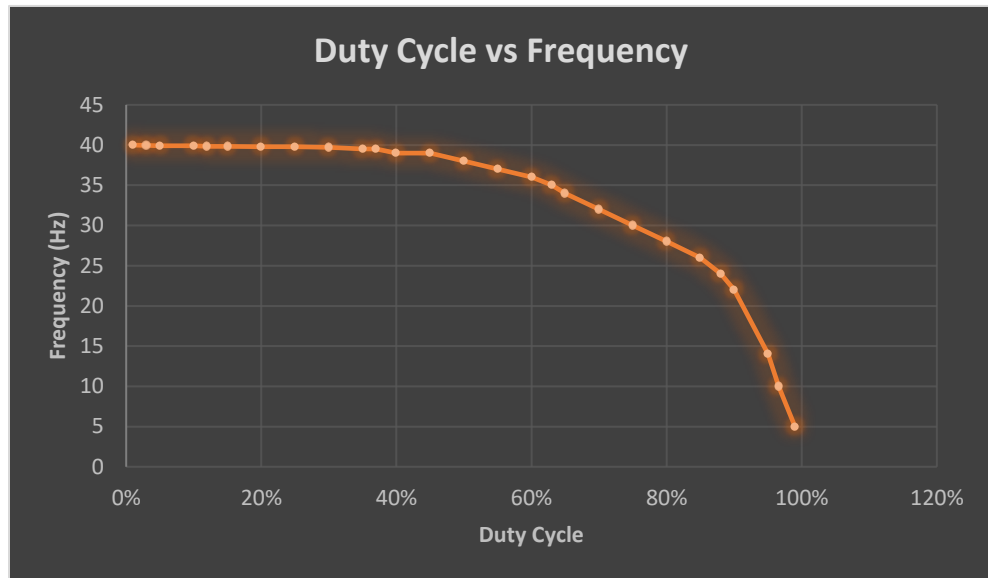


Figure 1. A graphical representation of duty cycle versus frequency.

Part 7 - Effects on Light Intensity when Changing the Duty Cycle at a Set Frequency

We found that at any Frequency greater than 40 Hz no blinking was distinguishable at any Duty Cycle.

Duty Cycle	Brightness (0-1)
100.00%	1.000
80.00%	1.000
60.00%	1.000
50.00%	1.000
45.00%	0.975
40.00%	0.950
30.00%	0.870
25.00%	0.800
20.00%	0.650
15.00%	0.500
12.50%	0.450
10.00%	0.380
7.50%	0.300
5.00%	0.250
3.33%	0.200
2.00%	0.125
1.00%	0.100
0.50%	0.075
0.25%	0.060
0.10%	0.050
0.01%	0.009

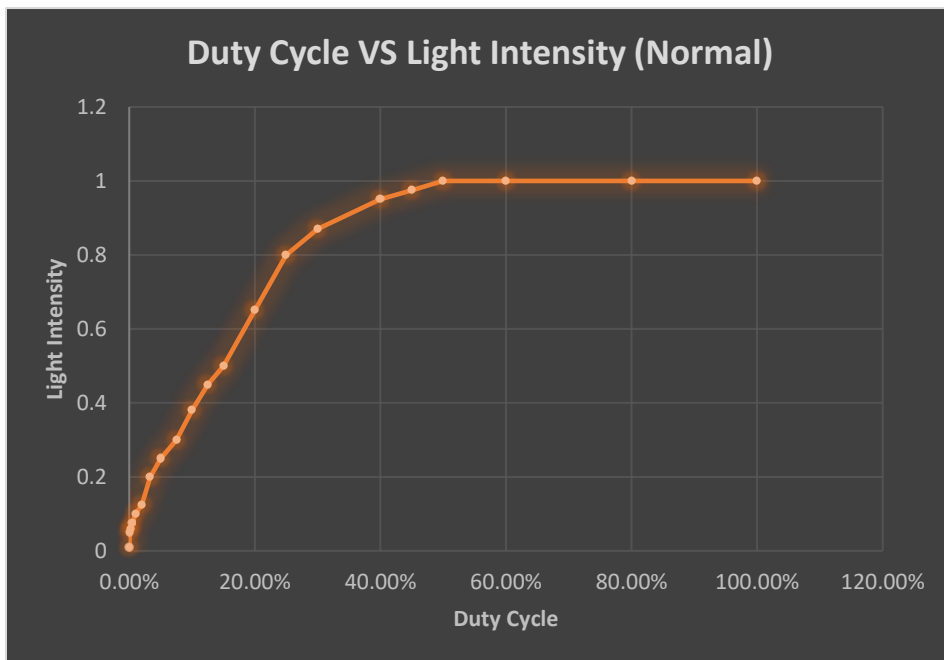


Figure 2. Graphical representation comparing the duty cycle with the intensity of the LED.

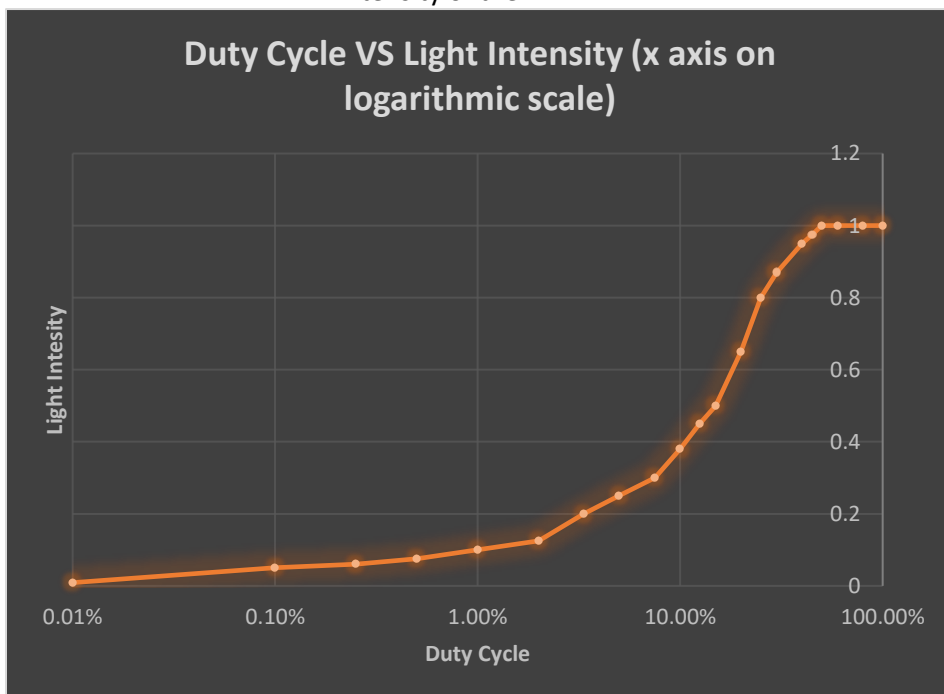


Figure 3. Graphical representation comparing the duty cycle with the intensity of the LED on a logarithmic scale

Results & Discussion:

Part 6:

For our code, we took several measurements and found that 1,050,000 operations was about one second. Using this, we based all our calculations to find the other relative duty cycles. Our formula for calculating the counts for both the on and off loops is as follows:

$$\frac{1}{\text{Frequency}} * (\text{Duty Cycle} * 1,050,000) = \text{On Count}$$

$$\frac{1}{\text{Frequency}} * (1 - \text{Duty Cycle}) * (1,050,000) = \text{Off Count}$$

We collected 28 points of data for this part of the experiment. For each point we calculated the number of iterations that was needed to produce the desired duty cycle at a frequency which we thought would not display any visible blinking. For example, when producing our data for a duty cycle of 99% we tested 1Hz first, which produced a slight blinking result. Upon finding that a frequency does not work, we move up in frequency until we find one that works.

After graphing all our data (Figure 1). It is easy to see that as we increase the duty cycle, the frequency needed to induce a perceived continuous light without flashing decreases. The data follows a linear distribution at first but then changes to a curved distribution when it approaches higher duty cycles.

Part 7:

Using the same method to calculate our loop cycles as in part 6, we collected 20 points of data. In this part of the lab we fixed our frequency at 40 Hz and observed the change of the state of the LED when we adjusted the duty cycle. Theoretically the light intensity should follow a linear pattern. As an example, if we have a 90% duty cycle at 40 Hz The LED is still only on 90% of the time. This should result in a 0.9 intensity on our scale of 0-1.

However, as observed by our data plots (Figure 2 & Figure 3) this is not the case. We observed that any duty cycle above 50% was indistinguishable from a duty cycle of 100%. Therefore, we were prompted to test duty cycles in the lower range. As to why this is, we believe it is because since we chose a frequency that does not result in any blinking to the human eye, that the time the LED spends off at this frequency (40Hz is approximately 25ms) is undetected by the human eye as well. Even though technically the intensity of light should follow a linear distribution, our observed data differs because of this.

Possible Sources of Errors:

- Everyone's eyes are different, a frequency at which one person might observe blinking, someone else might see continuous light.
- Our measurement of the intensity of light was purely qualitative, we had no instruments to take proper readings.
- Our measurement of the number of operations to make up one second in our code was approximate and not exact.

Conclusion

Overall this lab helped us get used to the XILINX FPGA & SDK, which in of itself is great. We found that changing the duty cycle of a discrete operation can have a great effect on the frequency at which it needs to operate to be observed as continuous. We also found out at which frequency's the human eye can and cannot distinguish. We also found that modifying the duty cycle can influence the intensity of an operation. Most of what we have looked at here can be applied to other systems, like the frequency and duty cycle of a car engine in operation. Overall modifying frequency or duty cycle provides you the ability to fine tune systems.