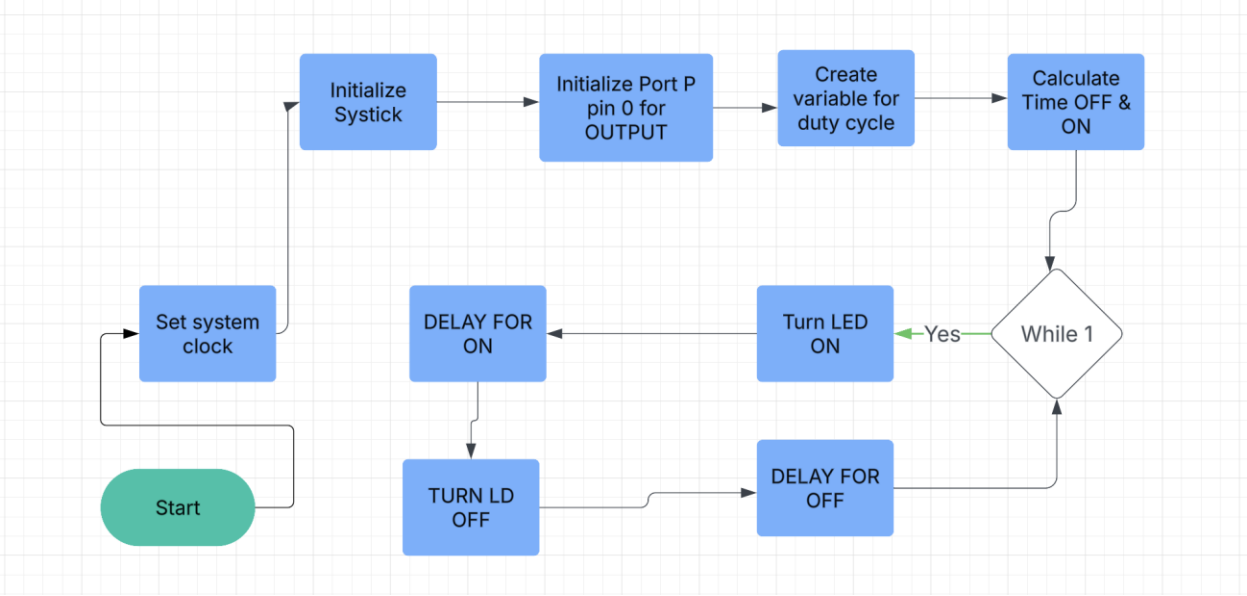
Lab 4 - Duty Cycle and Pulse Timing – Prelab Report

Sunday, February 15th, 2025

Joey McIntyre (400520473), Devin Gao (400508489), Minhazur Rakin (400511143)

As a future member of the engineering profession, the student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is our own and adheres to the Academic Integrity Policy of McMaster University and the Code of Conduct of the Professional Engineers of Ontario.

1. Draw a flowchart for an ARM microcontroller program to flash an LED with a 50% duty-cycle. Set the period of one cycle as a variable to be later defined.[5 marks]




2. Based upon your flowchart, write a C program (and provide it in your prelab document) to flash the onboard LED. Assuming a 50% duty-cycle, the period of one cycle is defined based upon the least-significant-digit (LSD) of your student number. Include structured and commented source code (must have your name and student number) in your prelab document. [10 marks]

```
1  //Minhazur Rakin
2  //400511143
3
4  #include <stdint.h>
5  #include "tm4c1294ncpdt.h"
6  #include "PLL.h"
7  #include "SysTick.h"
8
9
10 void PortN_Init(void){
11   //Use PortN onboard LED
12   SYSCTL_RCGCGPIO_R |= SYSCTL_RCGCGPIO_R12;       // activate clock for Port N
13   while((SYSCTL_PRGPIO_R&SYSCTL_PRGPIO_R12) == 0){};  // allow time for clock to stabilize
14   GPIO_PORTN_DIR_R |= 0x01;                       // make PN0 out (PN0 built-in LED1)
15   GPIO_PORTN_AFSEL_R &= ~0x01;                    // disable alt funct on PN0
16   GPIO_PORTN_DEN_R |= 0x01;                       // enable digital I/O on PN0
17                                                   // configure PN1 as GPIO
18   //GPIO_PORTN_PCTL_R = (GPIO_PORTN_PCTL_R&0xFFFFFF0F)+0x00000000;
19   GPIO_PORTN_AMSEL_R &= ~0x01;                    // disable analog functionality on PN0
20
21   GPIO_PORTN_DATA_R ^= 0b00000001;                //hello world!
22   SysTick_Wait10ms(10);                           //.1s delay
23   GPIO_PORTN_DATA_R ^= 0b00000001;
24   return;
25 }
26
27
28
29 void createDelayON(float cyclepercentage){
30   float period = 300.0f; //student number ends in 3 so we do 300 ms
31   uint32_t  delay = (uint32_t)(period*cyclepercentage); // creating delay percentage
32   SysTick_Wait10ms(delay); // calling systick wait to create delay•
33 }
34
```

```
36 int main(void){
37
38
39   //initializing everything
40   PLL_Init();
41   SysTick_Init();
42   PortN_Init();
43   GPIO_PORTN_DATA_R *= 0b00000001;
44
45   while(1){                              // run program for fixed number of steps/samples
46     createDelayON(0.5f);                          // create delay
47     GPIO_PORTN_DATA_R ^= 0b00000001;              // toggle LED
48     createDelayON(0.5f);                          // create delay
49     GPIO_PORTN_DATA_R ^= 0b00000001;              // toggle LED
50   }
51
52 }
```

3. If a stepper motor were to have 36 steps per rotation, how fast would the motor turn with a 10ms delay per step? Show calculations and answer in RPM. You may assume memory access time etc. are negligible. [5 marks]

Steps per rotation = 36 steps

Delay per step = 10ms = 0.01s

Time per rotation = 36 x 0.01 = 0.36s

RPS = 1/time per rotation = 1/0.36 = 2.78rps

RPM = RPS x 60 = 2.78 x 60 = 166.67rpm

4. If a stepper motor were to have 200 steps per rotation, calculate the delay per step for the motor to spin at 60 RPM. You may assume memory access time etc. are negligible. [5 marks]

Steps per rotation: 200 steps

Speed = 60 RPM

RPS = RPM/60 = 60/60 = 1 RPS

Time per rotation = 1/RPS = 1/1 = 1 second

Delay per step = time per rotation/steps per rotation = 1/200 = 5ms