2DX4: Microprocessor System
PreLab 4

Instructor: Drs.Boursalie, Doyle, Haddara

Lab TAs: Fatemeh Derakshani (derakhsf), Han Zhou (zhouh115)

Junbo Wang – L01 – wangj430

Yichen Lu – L01 – luy191

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. Submitted by **[Junbo Wang wangj430 400249823]**

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. Submitted by **[Yichen Lu luy191 400247938]**

1. **DrawMilestone2 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.**

1. Initialize the data

2. Turn on the LED

3. Toggle the LED

4. Waiting for 1 second (the delay)

2. Prelab2

Appendix

Prelab 2

```
// 2DX4_Knowledge_Thread_3_Session 0
// This program illustrates the use of SysTick in the C language.
// Note the library headers asscoaited are PLL.h and SysTick.h,
// which define functions and variables used in PLL.c and SysTick.c.
//

//
// Name: Yichen Lu  Junbo Wang
// Student id: 400247938
// Date: Feb, 14th, 2022


#include <stdint.h>
#include "tm4c1294ncpdt.h"
#include "PLL.h"
#include "SysTick.h"


void PortN_Init(void){
    //Use PortN onboard LED
    SYSCTL_RCGCGPIO_R |= SYSCTL_RCGCGPIO_R12;               // activate clock
for Port N
    while((SYSCTL_PRGPIO_R&SYSCTL_PRGPIO_R12) == 0){};  // allow time for clock
to stabilize
    GPIO_PORTN_DIR_R |= 0x05;                                        // make PN0
out (PN0 built-in LED1)
  GPIO_PORTN_AFSEL_R &= ~0x05;                                    // disable alt
funct on PN0
  GPIO_PORTN_DEN_R |= 0x05;                                       // enable
digital I/O on PN0

// configure PN1 as GPIO
  //GPIO_PORTN_PCTL_R = (GPIO_PORTN_PCTL_R&0xFFFFFF0F)+0x00000000;
  GPIO_PORTN_AMSEL_R &= ~0x05;                                    // disable
analog functionality on PN0

    GPIO_PORTN_DATA_R ^= 0b00000001;                                  //hello
world!
    SysTick_Wait10ms(10);
//.1s delay
    GPIO_PORTN_DATA_R ^= 0b00000001;
    return;
}


void DutyCycle_Percent(uint8_t duty){
```

```
        float percent;
        percent = ((float)duty*1000)/(255);
        int percent_int;
        percent_int = (int)percent;
        GPIO_PORTN_DATA_R ^= 0b00000100;
        SysTick_Wait10ms(percent_int);  //SysTick was changed to 0.01 ms in
order for this to work
        GPIO_PORTN_DATA_R ^= 0b00000100;
        SysTick_Wait10ms(1000-percent_int);
}


int main(void){

    PLL_Init();
// Default Set System Clock to 120MHz
    SysTick_Init();
// Initialize SysTick configuration
    PortN_Init();
// Initialize Port N

    uint8_t duty = 128;
    while(1){
        DutyCycle_Percent(duty);
    }
}
```