

Alan Rieger and Ayden Dauenhauer

Prof. Wolfe

ELEN 121L Tuesday 2:15 p.m.

9 April 2024

Lab 1 - Writing and debugging some simple C programs

Lab Procedure:

Part 1:

```
#include "stm32l476xx.h"  
#include "SysClock.h"  
#include <string.h>
```

```
int fiblist[25];  
char string1[15] = "Hello";  
char string2[15] = "World";  
char string3[255] = "\0";  
int main(void){
```

```
    System_Clock_Init(); // Switch System Clock = 80 MHz  
    // Fib Stuff  
    int i=1;  
    int j=0;  
    fiblist[0]=0;  
    fiblist[1]=1;  
    for(i=1; i<24; i++){  
        fiblist[i+1]=fiblist[i]+fiblist[j];  
        j++;  
    }  
    // Hello World Stuff  
    strcat(strcat(string1, " "), string2);  
    strcat(string3, string1);
```

```

while(1) {

}

}

```

Screenshots:

The screenshot shows two memory dump windows. The top window is titled 'Address: fiblist' and displays binary data starting at address 0x20000020. The bottom window is titled 'Address: string3' and displays the string "Hello World....." starting at address 0x200014E8.

```

Address: fiblist
0x20000020: 0000000000 0000000001 0000000001 0000000002 0000000003 0000000005 0000000008 0000000013 0000000021
0x20000044: 0000000034 0000000055 0000000089 0000000144 0000000233 0000000377 0000000610 0000000987 0000001597
0x20000068: 000002584 000004181 000006765 000010946 000017711 000028657 000046368 000000000 0000000000

Address: string3
0x200014E8: Hello World.....
0x20001550: .....
0x200015B8: .....
0x20001620: .....
0x20001688: .....
0x200016F0: .....
0x20001758: .....
0x200017C0: .....
0x20001828: .....
0x20001890: .....
0x200018F8: .....
0x20001960: .....
0x200019C8: .....

```

Part 2:

```

#include "stm32l476xx.h"
#include "SysClock.h"
#include <string.h>
int clock(){
    RCC->AHB2ENR |= RCC_AHB2ENR_GPIOBEN;
    RCC->AHB2ENR |= RCC_AHB2ENR_GPIOEEN;
}

int PUPDR(){
    GPIOB->PUPDR &= ~GPIO_PUPDR_PUPDR2;
    GPIOE->PUPDR &= ~GPIO_PUPDR_PUPDR8;
}

int MODER(){
    GPIOB->MODER &= GPIO_MODER_MODER2_0;
    GPIOE->MODER &= GPIO_MODER_MODER8_0;
}

int Red_LED_On(){

```

```

GPIOB->ODR |= GPIO_ODR_ODR_2;
}

int Red_LED_Off(){
    GPIOB->ODR &= ~GPIO_ODR_ODR_2;
}

int Red_LED_Toggle(){
    GPIOB->ODR ^= GPIO_ODR_ODR_2;
}

int Green_LED_On(){
    GPIOE->ODR |= GPIO_ODR_ODR_8;
}

int Green_LED_Off(){
    GPIOE->ODR &= ~GPIO_ODR_ODR_8;
}

int Green_LED_Toggle(){
    GPIOE->ODR ^= GPIO_ODR_ODR_8;
}

int delaymicro(unsigned int x){
    int i=0;
    for(i=0; i<x*4; i++){
        i=i+1;
    }
}

int main(void){

    System_Clock_Init(); // Switch System Clock = 80 MHz
}

```

```
clock();
PUPDR();
MODER();

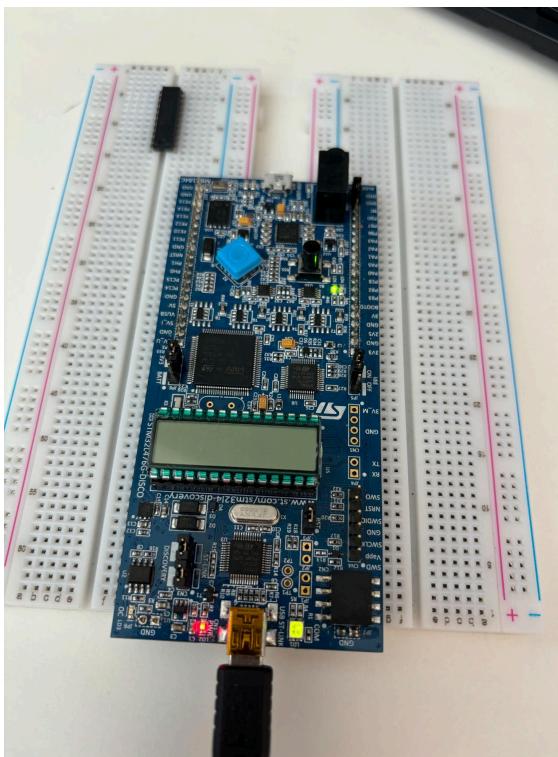
Red_LED_On();
Red_LED_Off();

Green_LED_On();
Green_LED_Off();

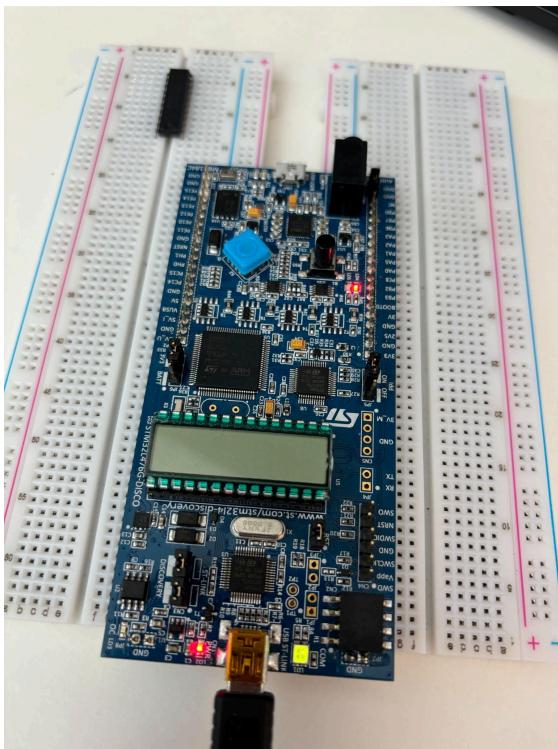
while(1) {
    Red_LED_Toggle();
    delaymicro(10000000);
    Red_LED_Toggle();
    delaymicro(10000000);
    Green_LED_Toggle();
    delaymicro(10000000);
    Green_LED_Toggle();
    delaymicro(10000000);
}

}
```

Red LED On:



Green LED On:



In your lab report, explain how you determined the timing of the delaymicro() function.

We calculated the delay by first noticing that the clock speed is 80 MHz. Then we added a 40 million delay value to turn the led on, then an additional 40 million value to turn it back off, thus filling the entire 80 million.