SECTION A

LED PORTS

PORT B

LED REGISTERS

GPIOB_MODER

GPIOB_OTYPER

GPIOB_OSPEEDR

GPIOB_PUPDR

GPIOB_IDR

GPIOB_ODR

GPIOB_BSRR

GPIOB_LCKR

GPIOB_AFLR

GPIOB_AFHR

GPIOB_BRR

SECTION B

NORMALLY OPEN PUSHBUTTON PORTS

PORT A

NORMALLY OPEN PUSHBUTTON REGISTERS

GPIOA_MODER

GPIOA_OTYPER

GPIOA_OSPEEDR

GPIOA_PUPDR

GPIOA_IDR

GPIOA_ODR

GPIOA_BSRR

GPIOA_LCKR

GPIOA_AFLR

GPIOA_AFHR

GPIOA_BRR

SECTION E

```
// ENABLE LED

        RCC->AHBENR |= RCC_AHBENR_GPIOBEN;
        // SET LED AS OUTPUT
        GPIOB->MODER |=          (    GPIO_MODER_MODER0_0|
                                             GPIO_MODER_MODER1_0 |
                                             GPIO_MODER_MODER2_0|
                                             GPIO_MODER_MODER3_0|
                                             GPIO_MODER_MODER4_0|
                                             GPIO_MODER_MODER5_0|
                                             GPIO_MODER_MODER6_0|
                                             GPIO_MODER_MODER7_0|
                                             GPIO_MODER_MODER10_0|
                                             GPIO_MODER_MODER11_0 );
```

SECTION D

```
        RCC->AHBENR |= RCC_AHBENR_GPIOAEN;
        // SET PUSHBUTTONS AS INPUT
        GPIOA->MODER &= ~ ( GPIO_MODER_MODER0|
                                             GPIO_MODER_MODER1|
                                             GPIO_MODER_MODER2|
                                             GPIO_MODER_MODER3 );
        // SET PUSHBUTTONS PULL UP AND PULL DOWN RESISTORS
        GPIOA->PUPDR |=          ( GPIO_PUPDR_PUPDR0_0|
                                             GPIO_PUPDR_PUPDR1_0|
                                             GPIO_PUPDR_PUPDR2_0|
                                             GPIO_PUPDR_PUPDR3_0    );
```

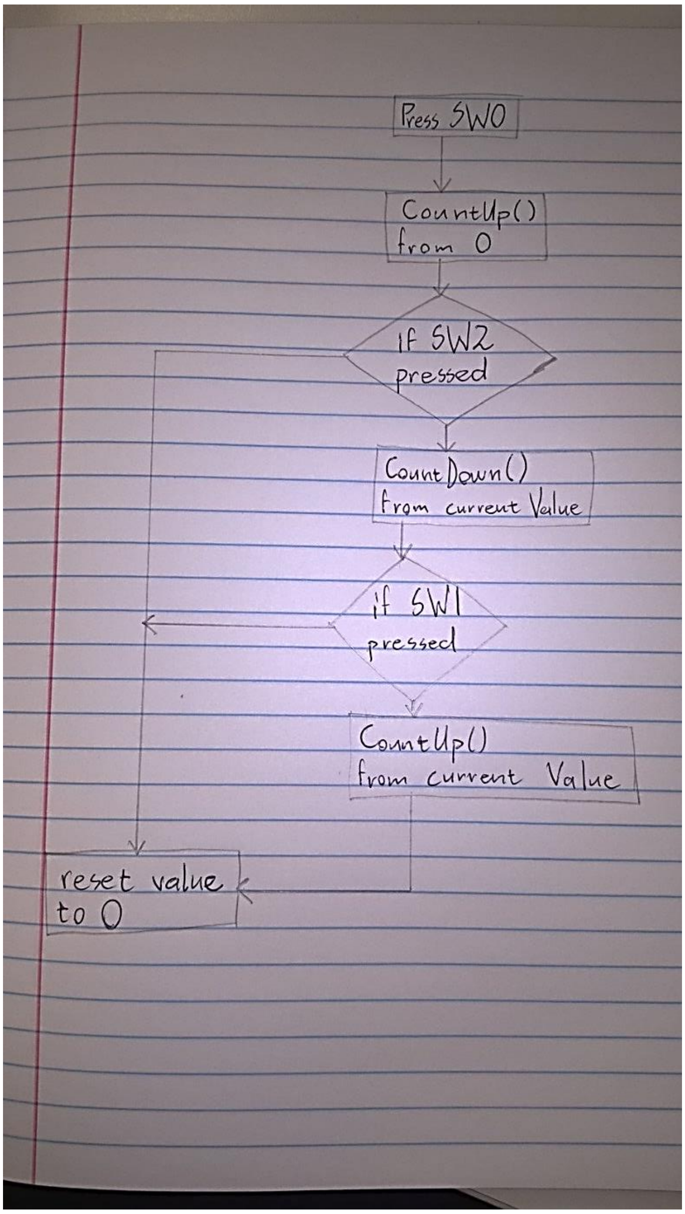
SECTION E

```
void InitPorts()
{
        // ENABLE PUSHBUTTONS
        RCC->AHBENR |= RCC_AHBENR_GPIOAEN;
        // SET PUSHBUTTONS AS INPUT
        GPIOA->MODER &= ~ ( GPIO_MODER_MODER0|
                                             GPIO_MODER_MODER1|
                                             GPIO_MODER_MODER2|
                                             GPIO_MODER_MODER3 );
        // SET PUSHBUTTONS PULL UP AND PULL DOWN RESISTORS
        GPIOA->PUPDR |=          ( GPIO_PUPDR_PUPDR0_0|
                                             GPIO_PUPDR_PUPDR1_0|
                                             GPIO_PUPDR_PUPDR2_0|
                                             GPIO_PUPDR_PUPDR3_0    );
        // ENABLE LED
        RCC->AHBENR |= RCC_AHBENR_GPIOBEN;
        // SET LED AS OUTPUT
        GPIOB->MODER |=          (    GPIO_MODER_MODER0_0|
                                             GPIO_MODER_MODER1_0 |
                                             GPIO_MODER_MODER2_0|
                                             GPIO_MODER_MODER3_0|
                                             GPIO_MODER_MODER4_0|
                                             GPIO_MODER_MODER5_0|
                                             GPIO_MODER_MODER6_0|
                                             GPIO_MODER_MODER7_0|
                                             GPIO_MODER_MODER10_0|
                                             GPIO_MODER_MODER11_0 );
}
```

SECTION F

```
        for(;;) //INFINITE LOOP
        {
                if((GPIOA->IDR & SW0) == 0)
                {
                        GPIOB->ODR = 0b00000001;
                        value = 1;
                }
        }
```

SECTION G



SECTION H

```
void Delay(void)
{
	// INITIALIZE I AND J
	int i, j;
	for (i = 0; i < DELAY1; i++)
		for (j = 0; j < DELAY2; j++); // DELAY FOR 1 SECOND
}
```

SECTION I

```
char CountUp(char value)
{
	// SET LED TO BITPATTERN
	GPIOB->ODR = value;
	// INCREASE VALUE BY 1
	value++;
	// RETURN VALUE
	return value;
}
```

SECTION J

```
char CountDown(char value)
{
        GPIOB->ODR = value;
        // DECREASE VALUE BY 1
        value--;
        return value;
}
```

SECTION K

```
//********************************************************************
//*                    EEE2046F C main                               *
//*==================================================================*
//* WRITTEN BY: Tumelo Lephadi                                   *
//* DATE CREATED: 13 May 2017                                        *
//* MODIFIED: 19 May 2017                                            *
//*==================================================================*
//* PROGRAMMED IN: Eclipse Luna Service Release 1 (4.4.1)            *
//* DEV. BOARD:    UCT STM32 Development Board                       *
//* TARGET:       STMicroelectronics STM32F051C6                    *
//*==================================================================*
//* DESCRIPTION: ENABLES LED'S TO COUNT FROM 0 TO 256 OR VICE VERSA  *
//*             IN BINARY                                            *
//********************************************************************
// INCLUDE FILES
//====================================================================
#include "lcd_stm32f0.h"
#include "stm32f0xx.h"
//====================================================================
// SYMBOLIC CONSTANTS
//====================================================================
#define SW0 GPIO_IDR_0
#define SW1 GPIO_IDR_1
#define SW2 GPIO_IDR_2
#define SW3 GPIO_IDR_3
#define DELAY1 1092
#define DELAY2 1092
//====================================================================
// GLOBAL VARIABLES
//====================================================================
unsigned int bitpattern = 0b0000000000000001; //PATTERN TO TURN D0 ON
uint16_t value = 0b0; //VALUE IN COUNT DOWN AND UP FUNCTION
//====================================================================
// FUNCTION DECLARATIONS
//====================================================================
void InitPorts(void);
char CountUp(char value);
char CountDown(char value);
void Delay(void);
//====================================================================
// MAIN FUNCTION
//====================================================================
void main (void)
{
        init_LCD();    // Initialise lcd
        InitPorts(); // Initialise LED and PushButtons
        lcd_putstring("EEE2050F PRAC2B");// Display string on line 1
        lcd_command(LINE_TWO);// Move cursor to line 2
        lcd_putstring("***LPHTUM003***");// Display string on line 2

        for(;;) // INFINITE LOOP
        {
                while ((GPIOA->IDR & SW0) == 0) // WAIT FOR SW0 TO BE PRESSED
                {
                        for(;;) //INFINITE LOOP
                        {
                                while ((GPIOA->IDR & SW2) != 0) //COUNT UP IF SW0 OR SW1 IS
PRESSED
                                {
                                        if(value <= 0b11111111)
                                        {
                                                Delay();
                                                value = CountUp(value);
                                        }
                                        else
                                                GPIOB->ODR = 0;
                                                value = 0;
                                }
                                while ((GPIOA->IDR & SW1) != 0) //COUNT DOWN IF SW1 IS PRESSED
                                {
                                        if(value > 0b0)
                                        {
                                                Delay();
                                                value = CountDown(value);
                                        }
                                        else
                                                GPIOB->ODR = 0;
                                                value = 0;
                                }
                        }
                }
        }
}// End of main

//======================================================================
// FUNCTION DEFINITIONS
//======================================================================
void InitPorts()
{
        // ENABLE PUSHBUTTONS
        RCC->AHBENR |= RCC_AHBENR_GPIOAEN;
        // SET PUSHBUTTONS AS INPUT
        GPIOA->MODER &= ~ ( GPIO_MODER_MODER0|
                                                GPIO_MODER_MODER1|
                                                GPIO_MODER_MODER2|
                                                GPIO_MODER_MODER3 );
        // SET PUSHBUTTONS PULL UP AND PULL DOWN RESISTORS
        GPIOA->PUPDR |=          ( GPIO_PUPDR_PUPDR0_0|
                                                GPIO_PUPDR_PUPDR1_0|
                                                GPIO_PUPDR_PUPDR2_0|
                                                GPIO_PUPDR_PUPDR3_0    );
        // ENABLE LED
        RCC->AHBENR |= RCC_AHBENR_GPIOBEN;
        // SET LED AS OUTPUT
        GPIOB->MODER |=          (    GPIO_MODER_MODER0_0|
                                                GPIO_MODER_MODER1_0 |
                                                GPIO_MODER_MODER2_0|
                                                GPIO_MODER_MODER3_0|
                                                GPIO_MODER_MODER4_0|
                                                GPIO_MODER_MODER5_0|
                                                GPIO_MODER_MODER6_0|
                                                GPIO_MODER_MODER7_0|
                                                GPIO_MODER_MODER10_0|
                                                GPIO_MODER_MODER11_0 );
}

char CountUp(char value)
{
        // SET LED TO BITPATTERN
        GPIOB->ODR = value;
        // INCREASE VALUE BY 1
        value++;
        // RETURN VALUE
        return value;
}

char CountDown(char value)
{
        GPIOB->ODR = value;
        // DECREASE VALUE BY 1
        value--;
        return value;
}

void Delay(void)
{
        // INITIALIZE I AND J
        int i, j;
        for (i = 1; i < DELAY1; i++)
                for (j = 1; j < DELAY2; j++) // DELAY FOR 1 SECOND
                {};
}
//*********************************************************************
// END OF PROGRAM
//*********************************************************************
```