PRAC3B_LPHTUM003

SECTION A

```
void main (void)
{
	init_LCD();                                     // Initialise lcd
	lcd_putstring("EEE2050F PRAC3B");               // Display string on line 1
	lcd_command(LINE_TWO);                          // Move cursor to line 2
	lcd_putstring("***LPHTUM003***");                    // Display string on line 2
	for(;;);                                        // Loop forever
}                                                       // End of main
```

SECTION B

```
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);
}
```

SECTION C

```
	for(;;)// Loop forever
	{
		if ((GPIOA->IDR & SW0) == 0)
		{
			init_LCD();// initialise lcd
			lcd_putstring("Weather Station");// Display string on line 1
			lcd_command(LINE_TWO); // Move cursor to line 2
			lcd_putstring("Press SW2");// Display string on line 2
		}
	}
```

SECTION D

```
		if ((GPIOA->IDR & SW1) == 0)
		{
			Delay();// DELAY FOR +/- 1 SECOND
			count = 1 + count; // INCREASE COUNT BY 1
			GPIOB->ODR = count; // WRITE COUNT VALUE ONTO LED
			init_LCD(); // INITIALISE LCD
			lcd_putstring("Rain bucket tip"); // WRITE ON SCREEN
		}
```

SECTION E

```
void ConverttoBCD(void)
{
        thousands = (count / 1000) ; // FIND THOUSANDS
        hundreds = ((count - thousands * 1000) / 100); // FIND HUNDREDS
        tens = (count - thousands * 1000 - hundreds * 100) / 10; // FIND TENS
        units = (count - thousands * 1000 - hundreds * 100 - tens * 10) / 1; // FIND UNITS
        remainder = count - thousands * 1000 - hundreds * 100 - tens * 10 - units;
        decimal[0] = thousands + 48;
        decimal[1] = hundreds + 48;
        decimal[2] = tens + 48;
        decimal[3] = units + 48;
        decimal[4] = 46; // DECIMAL POINT
        decimal[5] = remainder + 48;
}
```

SECTION F

```
                if ((GPIOA->IDR & SW2) == 0)
                {
                        lcd_command(CLEAR); // CLEAR SCREEN
                        ConverttoBCD(); // CONVERT TO BCD
                        lcd_putstring("Count:"); // WRITE ON SCREEN
                        lcd_command(LINE_TWO); // GO TO LINE 2
                        lcd_putstring(decimal); // WRITE DECIMAL NUMBER ON SCREEN
                }
```

SECTION G

```
//*****************************************************************
//*            EEE2050F C MAIN                                    *
//*================================================================*
//* WRITTEN BY: TUMELO LEPHADI                                     *
//* DATE CREATED: 09/06/2017                                       *
//* MODIFIED: 09/06/2017                                           *
//*================================================================*
//* PROGRAMMED IN: Eclipse Luna Service Release 1 (4.4.1)          *
//* DEV. BOARD:    UCT STM32 Development Board                     *
//* TARGET:                STMicroelectronics STM32F051C6          *
//*================================================================*
//* DESCRIPTION:                                                   *
//* PROGRAM THAT MEASURES RAINFALL IN MILIMETRES                   *
//*****************************************************************
// 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
//=================================================================
int count;
char thousands, hundreds, tens, units, remainder;
char decimal[10];
//=================================================================
// FUNCTION DECLARATIONS
//=================================================================
void InitPorts(void);
void ConverttoBCD(void);
void Delay(void);
//=================================================================
// MAIN FUNCTION
//=================================================================
```

```
void main (void)
{
	init_LCD(); // Initialise lcd
	InitPorts(); // INITIALISE PORTS
	lcd_putstring("EEE2046F PRAC3B"); // Display string on line 1
	lcd_command(LINE_TWO); // Move cursor to line 2
	lcd_putstring("**LPHTUM003**");		// Display string on line 2
	for(;;) // Loop forever
	{
		if ((GPIOA->IDR & SW0) == 0)
		{
			lcd_command(CLEAR);
			lcd_putstring("Weather Station"); // Display string on line 1
			lcd_command(LINE_TWO); // Move cursor to line 2
			lcd_putstring("Press SW2"); // Display string on line 2
		}
		if ((GPIOA->IDR & SW1) == 0)
		{
			Delay();// DELAY FOR +/- 1 SECOND
			count = 1 + count; // INCREASE COUNT BY 1
			GPIOB->ODR = count; // WRITE COUNT VALUE ONTO LED
			lcd_command(CLEAR); // CLEAR SCREEN
			lcd_putstring("Rain bucket tip"); // WRITE ON SCREEN
		}
		if ((GPIOA->IDR & SW2) == 0)
		{
			lcd_command(CLEAR); // CLEAR SCREEN
			ConverttoBCD(); // CONVERT TO BCD
			lcd_putstring("Rainfall:"); // WRITE ON SCREEN
			lcd_command(LINE_TWO); // GO TO LINE 2
			lcd_putstring(decimal); // WRITE DECIMAL NUMBER ON SCREEN
			lcd_putstring(" mm");
		}
	}
}                                                                               // 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);
}
void ConverttoBCD(void)
{
	double rain = count;
	rain = rain * 0.2; // 0.2 mm PER BUCKET
	thousands = (rain / 1000) ; // FIND THOUSANDS
	hundreds = ((rain - thousands * 1000) / 100); // FIND HUNDREDS
	tens = (rain - thousands * 1000 - hundreds * 100) / 10; // FIND TENS
```

```
		units = (rain - thousands * 1000 - hundreds * 100 - tens * 10) / 1; // FIND UNITS
		remainder = 10 * (rain - thousands * 1000 - hundreds * 100 - tens * 10 - units); // FIND DECIMAL NUMBERS
		decimal[0] = thousands + 48; // INSERT ASCII NUMBERS INTO ARRAY
		decimal[1] = hundreds + 48;
		decimal[2] = tens + 48;
		decimal[3] = units + 48;
		decimal[4] = 46; // INSERT ASCII VALUE FOR DECIMAL POINT
		decimal[5] = remainder + 48;
}
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
//*********************************************************************
```