Practical 4B: ADC & EXTERNAL INTERRUPTS

Name: Mehta, Ronak
Student No: MHTRON001
Practical Group: Group 2
Practical Venue: Red Lab
Date: 16th May 2018

Plagiarism Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.

2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this practical report from the work(s) of other people has been attributed and has been cited and referenced.

3. This practical report (including circuit diagrams and code) is my own work.

4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work.

5. I acknowledge that copying someone else's code, schematics or report, or part of it, is wrong, and declare that this is my own work.

RMehta

16th May 2018

Signature Date

(a)

```
void Init_GPIO(void)
{
	RCC  ->AHBENR |= 0b00000000000001100000000000000000;
	GPIOA->MODER |=  0b00101000000000000001100000000000 ;  //sets pins to
                                input for the switches and analogue for the pot0
	RCC -> APB2ENR |=RCC_APB2ENR_ADCEN;
	GPIOB->MODER  |= 0x00505555;
	GPIOB->ODR     |= 0b00000000;
	GPIOA ->PUPDR |= 0b01010101;           //sets the register to pull down
}


void Init_ADC(void)
{
	ADC1 -> CR &= ~ADC_CR_ADSTART;
	ADC1 -> CFGR1 |= ADC_CFGR1_RES_1;      // 8 bit resolution
	ADC1 -> CFGR1 &= ~ADC_CFGR1_ALIGN;     // right align
	ADC1 -> CFGR1 |= ADC_CFGR1_CONT;
	ADC1 -> CHSELR |= ADC_CHSELR_CHSEL5;   // Channel 5 = PA5
	ADC1 -> CR |= ADC_CR_ADEN;             // sets ADEN to ‘1’ in ADC_CR register
	while((ADC1->ISR & ADC_ISR_ADRDY)==0); // exits loop when ADRDY == 1
}
```

(b)

$V_{IN} = ADC_{OUT}\ (\frac{\text{VRANGE}}{(2^N)-1})$

$ADC_{OUT} = V_{IN}\ /\ (\frac{\text{VRANGE}}{(2^N)-1})$

$ADC_{OUT} = 14\ /\ (\frac{24}{(2^8)-1})$

$ADC_{OUT} = 14\ /\ (\frac{24}{255})$

$ADC_{OUT} = 148.75$

$\mathbf{ADC_{OUT} = 149}$

(c)

```
void check_battery(void)
{
        init_LCD();
        lcd_putstring("a");                                 // Displays on line 1

        Delay();
        ADC1 -> CR |= ADC_CR_ADSTART;
        init_LCD();
        lcd_putstring("B");                                 // Displays on line 1

        while((ADC1->ISR & ADC_ISR_EOC) == 0);
        init_LCD();
        lcd_putstring("c");                                 // Displays on line 1

        Delay();

        int ADC_value = ADC1->DR;
        if (149 >= ADC_value )
        {
                GPIOB->ODR     |= 1<<11;
        }

        battery_voltage = ADC_value*(24/((1<<8)-1));
}
```

(d)

```
void display(void)
{
        while((GPIOA->IDR & SW0)!=0){
        }
        init_LCD();
        lcd_putstring("EEE2046F PRAC4B");                   // Displays on line 1
        lcd_command(LINE_TWO);                              // Moves to line 2
        lcd_putstring("RONAK MEHTA");                       // Displays on line 2

        while ((GPIOA->IDR & SW1) != 0);
        init_LCD();
        lcd_putstring("Battery Monitor");                   // Displays on line 1
        lcd_command(LINE_TWO);                              // Moves to line 2
        lcd_putstring("Press SW2");                         // Displays on line 2

        while ((GPIOA->IDR & SW2) != 0) {
        }
        check_battery();
        init_LCD();
        lcd_putstring("Battery:");                          // Displays on line 1
        lcd_command(LINE_TWO);                              // Moves to line 2
        lcd_putstring(battery_voltage +" V");               // Displays on line 2
}
```

(e)

```
// INCLUDE FILES
//====================================================================
#include "lcd_stm32f0.h"
#include "stm32f0xx.h"
//====================================================================
// GLOBAL VARIABLES
//====================================================================
#define SW0 GPIO_IDR_0
#define SW1 GPIO_IDR_1
#define SW2 GPIO_IDR_2
#define DELAY1 1000
#define DELAY2 4000
int battery_voltage;
//====================================================================
// FUNCTION DECLARATIONS
//====================================================================
void Init_GPIO(void);
void Init_ADC(void);
void check_battery(void);
void Delay(void);
void display(void);
//====================================================================
// MAIN FUNCTION
//====================================================================
void main(void)
{

       init_LCD();                            // Initialise lcd
       lcd_putstring("RONAK MEHTA");          // Display string on line 1
       lcd_command(LINE_TWO);                 // Move cursor to line 2
       lcd_putstring("MHTRON001");            // Display string on line 2
       Init_GPIO();
       void Init_ADC();
       for(;;)
       {
             display();
       }                                      // Loop forever
}                                             // End of main

//====================================================================
// FUNCTION DEFINITIONS
//====================================================================
void Init_GPIO(void)
{
       RCC  ->AHBENR |= 0b00000000000001100000000000000000;
       GPIOA->MODER |=  0b00101000000000000110000000000000 ;  //sets pins to
                                    input for the switches and analogue for the pot0
       RCC -> APB2ENR |=RCC_APB2ENR_ADCEN;
       GPIOB->MODER  |= 0x00505555;
       GPIOB->ODR     |= 0b00000000;
       GPIOA ->PUPDR |= 0b01010101;           //sets the register to pull down

}
```

```
void Init_ADC(void)
{
       ADC1 -> CR &= ~ADC_CR_ADSTART;
       ADC1 -> CFGR1 |= ADC_CFGR1_RES_1;      // 8 bit resolution
       ADC1 -> CFGR1 &= ~ADC_CFGR1_ALIGN;     // right align
       ADC1 -> CFGR1 |= ADC_CFGR1_CONT;
       ADC1 -> CHSELR |= ADC_CHSELR_CHSEL5;   // Channel 5 = PA5
       ADC1 -> CR |= ADC_CR_ADEN;             // sets ADEN to ‘1’ in ADC_CR register
       while((ADC1->ISR & ADC_ISR_ADRDY)==0); // exits loop when ADRDY == 1

}

void check_battery(void)
{
       init_LCD();
       lcd_putstring("a");                          // Displays on line 1

       Delay();
       ADC1 -> CR |= ADC_CR_ADSTART;
       init_LCD();
       lcd_putstring("b");                          // Displays on line 1

       while((ADC1->ISR & ADC_ISR_EOC) == 0);
       init_LCD();
       lcd_putstring("c");                          // Displays on line 1
       Delay();

       int ADC_value = ADC1->DR;
       if (149 >= ADC_value )
       {
              GPIOB->ODR     |= 1<<11;
       }

       battery_voltage = ADC_value*(24/((1<<8)-1));
}

void display(void)
{
       while((GPIOA->IDR & SW0)!=0){
       }
       init_LCD();
       lcd_putstring("EEE2046F PRAC4B");                  // Displays on line 1
       lcd_command(LINE_TWO);                             // Moves to line 2
       lcd_putstring("RONAK MEHTA");                      // Displays on line 2

       while ((GPIOA->IDR & SW1) != 0);
       init_LCD();
       lcd_putstring("Battery Monitor");                  // Displays on line 1
       lcd_command(LINE_TWO);                             // Moves to line 2
       lcd_putstring("Press SW2");                        // Displays on line 2
       while ((GPIOA->IDR & SW2) != 0) {
       }
       check_battery();
       init_LCD();
       lcd_putstring("Battery:");                        // Displays on line 1
       lcd_command(LINE_TWO);                            // Moves to line 2
       lcd_putstring(battery_voltage +" V");             // Displays on line 2
}

void Delay(void)                                           // Creates a Delay
{
        int i,j;
        for (i=0;i<=DELAY1;i++)
              for(j=0;j<=DELAY2;j++);
}


//********************************************************************
// END OF PROGRAM
//********************************************************************
```