***Practical 3B: GPIO Inputs***

**Name:** Mehta, Ronak

**Student No:** MHTRON001

**Practical Group:** Group 2

**Practical Venue:** Red Lab

**Date:** 02nd 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.

02nd May 2018

Signature Date

a) The LEDs are connected to port B. The registers needed to control their operation are

Output Data Register (ODR), Mode register (MODER) and AHBENR for Clock enable.

b) The Normally Open (N.O.) push buttons are connected to port A. The registers needed to

control their operation are Mode Register (MODER) and Pull Up Pull Down Register

(PUPDR) and AHBENR for Clock enable.

c) To get digital values to the LEDs, we must configure the mode of each of the pins to the

value 01. I.e:

// Enable clock for LEDs

RCC->AHBENR |= RCC\_AHBENR\_GPIOBEN;

// set pins B0-B7, B10 and B11 to GPIO Outputs

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);

//To turn the LEDs off at the start of the program, we set the ODR to 0

GPIOB->ODR = 0b00000000;

d) To use SW0, SW1 and SW2 as digital inputs, we set the mode of each of the pins to 00 and enable pull up resistors.

// enable clock for push buttons

RCC->AHBENR |= RCC\_AHBENR\_GPIOAEN;

// set pins A0-A3 to GPIO inputs

GPIOA->MODER &= ~(GPIO\_MODER\_MODER0|

GPIO\_MODER\_MODER1|

GPIO\_MODER\_MODER2);

// enable pull up resistors

GPIOA->PUPDR |= (GPIO\_PUPDR\_PUPDR0\_0|

GPIO\_PUPDR\_PUPDR1\_0|

GPIO\_PUPDR\_PUPDR2\_0);

e)

**void** **InitPorts**(**void**){

// enable clock for push buttons

RCC->AHBENR |= RCC\_AHBENR\_GPIOAEN;

// set pins A0-A3 to GPIO inputs

GPIOA->MODER &= ~(GPIO\_MODER\_MODER0|

GPIO\_MODER\_MODER1|

GPIO\_MODER\_MODER2);

// enable pull up resistors

GPIOA->PUPDR |= (GPIO\_PUPDR\_PUPDR0\_0|

GPIO\_PUPDR\_PUPDR1\_0|

GPIO\_PUPDR\_PUPDR2\_0);

// Enable clock for LEDs

RCC->AHBENR |= RCC\_AHBENR\_GPIOBEN;

// set pins B0-B7, B10 and B11 to GPIO Outputs

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);

//To turn the LEDs off at the start of the program, we set the ODR to 0

GPIOB->ODR = 0b00000000;

}

f)

**while** ((GPIOA->IDR & SWO) != 0){

GPIOB->ODR = 0b00000001;

}

g)

**char** **CountUp**(**char** value){

**if** (value < 256){

GPIOB->ODR = value; // displays value

value++; // increments the value by 1 each time

}

**else** {

value = 0; // resets to 0 if value greater than 256

GPIOB->ODR = value;

}

**return**(value);

Delay();

}

h)

**char** **CountDown**(**char** value){

**if** (value != 0){

GPIOB->ODR = value; // displays value

value--; // decrements the value by 1 each time

}

**else** {

value = 255; // gives back value 255 if less than 0

GPIOB->ODR = value;

}

**return**(value);

Delay();

}

i)

// INCLUDE FILES

//====================================================================

**#include** "lcd\_stm32f0.h"

**#include** "stm32f0xx\_conf.h"

**#include** <stdint.h>

**#include** <stdio.h>

//====================================================================

// SYMBOLIC CONSTANTS

//====================================================================

**#define** SW0 GPIO\_IDR\_0

**#define** SW1 GPIO\_IDR\_1

**#define** SW2 GPIO\_IDR\_2

**#define** SW3 GPIO\_IDR\_3

**int** DELAY1 = 1000;

**int** DELAY2 = 3000;

uint16\_t bitpattern = 0b00000000;

//====================================================================

// GLOBAL VARIABLES

//====================================================================

**char** value1;

**int** toggle = 1;

**char** flag;

//====================================================================

// FUNCTION DECLARATIONS

//====================================================================

**void** **InitPorts**(**void**);

**void** **Delay**(**void**);

**void** **CountUp**(**char** value);

**void** **CountDown**(**char** value);

//====================================================================

// 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

**while** ((GPIOA->IDR & SW0) != 0){

value1 = 0;

}

flag = 1;

**do** {

**if**((GPIOA->IDR & SW1) == 0){

flag=1;

}

**else** **if** ((GPIOA->IDR & SW2) == 0){

flag=2;

}

**if**(flag=1){

GPIOB->ODR = CountUp(value1);

value1++;

}

**else** **if** (flag=2){

GPIOB->ODR = CountDown(value1);

value1--;

}

Delay();

}

**while** (toggle = 1); // Loop Forever

} // End of main

//====================================================================

// FUNCTION DEFINITIONS

//====================================================================

**void** **Delay**(**void**){

**for**(**int** i=0; i<DELAY1; i++){

**for** (**int** j = 0; j < DELAY2; ++j) {

}

}

}

**void** **InitPorts**(**void**){

// enable clock for push buttons

RCC->AHBENR |= RCC\_AHBENR\_GPIOAEN;

// set pins A0-A3 to GPIO inputs

GPIOA->MODER &= ~(GPIO\_MODER\_MODER0|

GPIO\_MODER\_MODER1|

GPIO\_MODER\_MODER2);

// enable pull up resistors

GPIOA->PUPDR |= (GPIO\_PUPDR\_PUPDR0\_0|

GPIO\_PUPDR\_PUPDR1\_0|

GPIO\_PUPDR\_PUPDR2\_0);

// Enable clock for LEDs

RCC->AHBENR |= RCC\_AHBENR\_GPIOBEN;

// set pins B0-B7, B10 and B11 to GPIO Outputs

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);

//To turn the LEDs off at the start of the program, we set the ODR to 0

GPIOB->ODR = bitpattern;

}

**char** **CountUp**(**char** value){

**if** (value < 256){

GPIOB->ODR = value; // displays value

value++; // increments the value by 1 each time

}

**else** {

value = 0; // resets to 0 if value greater than 256

GPIOB->ODR = value;

}

**return**(value);

Delay();

}

**char** **CountDown**(**char** value){

**if** (value != 0){

GPIOB->ODR = value; // displays value

value--; // decrements the value by 1 each time

}

**else** {

value = 255; // gives back value 255 if less than 0

GPIOB->ODR = value;

}

**return**(value);

Delay();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// END OF PROGRAM

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*