

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.



Signature

02nd May 2018

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 & SW0) != 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
//*****

```