Ramzi El-abdallah

Lab 4 Submission

**Exercise 1:**

/\*----------------------------------------------------------------------------

LAB EXERCISE 8.1 - DIGITAL INPUTS AND OUTPUTS

PROGRAMMING USING MBED API

----------------------------------------

In this exercise you need to use the mbed API functions to:

1) Define BusIn, BusOut interfaces for inputs and outputs

2) The RGB LED is controlled by the buttons:

+ Button 1 - light RED

+ Button 2 - light BLUE

+ Button 3 - light GREEN

+ Button 4 - light WHITE (RED, GREEN and BLUE at the same time)

GOOD LUCK!

\*----------------------------------------------------------------------------\*/

//Ramzi El-abdallah

#include "mbed.h"

#include "pindef.h"

//Define input bus

BusIn SWITCHES(Din3, Din2, Din1, Din0);

//Define output bus for the RGB LED

BusOut LEDS (Dout0, Dout1, Dout2);

//Function Prototypes, Defined after main

void toggle\_r(void);

void toggle\_g(void);

void toggle\_b(void);

void toggle\_all(void);

/\*----------------------------------------------------------------------------

MAIN function

\*----------------------------------------------------------------------------\*/

int main(){

while(1){

//Check which button was pressed and light up the corresponding LEDs

//Indicates if button 4 is pressed, then toggles all LEDS

if (SWITCHES == 0xE){

toggle\_all();

wait(.2);

}

//Indicates if button 2 is pressed, then toggles blue LED

if (SWITCHES == 0xD){

toggle\_b();

wait(.2);

}

//Indicates if button 3 is pressed, then toggles green LED

if (SWITCHES == 0xB){

toggle\_g();

wait(.2);

}

//Indicates if button 1 is pressed, then toggles red LED

if (SWITCHES == 0x7){

toggle\_r();

wait(.2);

}

}

}

//Each function uses mask to toggle each LED by changing the value of the corresponding bit in LEDS bus

void toggle\_r(void){

LEDS = LEDS xor 1;

}

void toggle\_g(void){

LEDS = LEDS xor 2;

}

void toggle\_b(void){

LEDS = LEDS xor 4;

}

void toggle\_all(void){

LEDS = LEDS xor 7;

}

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ARM University Program Copyright (c) ARM Ltd 2014\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Exercise 2:**

/\*----------------------------------------------------------------------------

LAB EXERCISE 8.1 - DIGITAL INPUTS AND OUTPUTS

PROGRAMMING USING MBED API

----------------------------------------

In this exercise you need to use the mbed API functions to:

1) Define BusIn, BusOut interfaces for inputs and outputs

2) The RGB LED is controlled by the buttons:

+ Button 1 - light RED

+ Button 2 - light BLUE

+ Button 3 - light GREEN

+ Button 4 - light WHITE (RED, GREEN and BLUE at the same time)

GOOD LUCK!

\*----------------------------------------------------------------------------\*/

//Ramzi El-abdallah

#include "mbed.h"

#include "pindef.h"

int count\_r = 0;

int count\_b = 0;

int count\_g = 0;

int count\_all = 0;

//Define input bus

BusIn SWITCHES(Din3, Din2, Din1, Din0);

//Define output bus for the RGB LED

BusOut LEDS (Dout0, Dout1, Dout2);

//Function Prototypes, Defined after main

void toggle\_r(void);

void toggle\_g(void);

void toggle\_b(void);

void toggle\_all(void);

/\*----------------------------------------------------------------------------

MAIN function

\*----------------------------------------------------------------------------\*/

int main(){

while(1){

//Check which button was pressed and light up the corresponding LEDs

//Indicates if button 4 is pressed, then toggles all LEDS

if (SWITCHES == 0xE){

toggle\_all();

wait(.2);

}

//Indicates if button 2 is pressed, then toggles blue LED

if (SWITCHES == 0xD){

toggle\_b();

wait(.2);

}

//Indicates if button 3 is pressed, then toggles green LED

if (SWITCHES == 0xB){

toggle\_g();

wait(.2);

}

//Indicates if button 1 is pressed, then toggles red LED

if (SWITCHES == 0x7){

toggle\_r();

wait(.2);

}

}

}

//Each function uses mask to toggle each LED by changing the value of the corresponding bit in LEDS bus

void toggle\_r(void){

count\_r++; //increment count each time the button is pushed

if (count\_r > 9){ //once count has been pushed 10 times it will toggle the LED on or off

LEDS = LEDS xor 1;

count\_r = 0; //resets the count

}

}

void toggle\_g(void){

count\_g++;

if (count\_g > 9){

LEDS = LEDS xor 2;

count\_g = 0;

}

}

void toggle\_b(void){

count\_b++;

if (count\_b > 9){

LEDS = LEDS xor 4;

count\_b = 0;

}

}

void toggle\_all(void){

count\_all++;

if (count\_all >9){

LEDS = LEDS xor 7;

count\_all = 0;

}

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ARM University Program Copyright (c) ARM Ltd 2014\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*