**Khalifa University of Science, Technology and Research**

**Microprocessor Systems LaboratoryELCE332**

**hcs12 input and output ports**

Laboratory Experiment No. 4

**Lab Partners**

Mohammad Jassem 100037002

Khalid waleedAlzarouni 100035499

Khalid Alhosani 100037003

**Date Experiment Performed**: 18-2-2015

**Date Lab Report Submitted**: 25-2-2015

**Lab Instructor: Mahmoud Khonji**

**Mohammed Alzaabi**

**Spring 2015**

Table of Contents

List of Figures and Tables 3

Summary 4

1. Introduction 5

1.1 Aim 5

1.2 Objectives 6

2. Design and Results 7

3. Assignment Questions 13

5. Conclusions and Recommendations 16

List of Figures and Tables

List of figures:

Figure1: Reading DIP Switches and Writing them to LEDs 7

Figure 2: Task 1 Board Testing 8 Figure 3: Reading Key Pad and Writing it to LEDs 9

Figure 4: Writing to LCD 10

Figure 5: Reading DIP Switches and Writing them to LED’s and LCD 11

Figure 6: Interfacing LCD with Key Pad and Push Buttons 12

Summary

In this lab report, we will explainwho to read data from an input and write data from output ports of the microcontrollerby taking the time delay into consideration. . We used the CodeWarrior program to write the C code, and we used the Dragon Pulse Trainer board to run and execute the code in most of thetasksof the lab. We performed five tasks in this lab. The first task is about reading the DIP switches and writing them to LEDs. The second taskis about reading input data from the Key Pad and writing it to LEDs.The third task is about writing strings in the LCD of the microcontroller. Moreover, the fourth task is a combination of the first task and the third task in which we will read from the DIP switches and write the data to the LEDs and LCD. Finally,the last task is about doing a division operation using Key Pad and Push Buttons as inputs, and the result should be displayed on the LCD.

1. Introduction

One unique feature of the microcontroller is using number of input/output pins used for connection with peripherals. Most microcontrollers will have at least 8 digital I/O lines. Somehave 12, 32 and 96 I/O lines. A common way to describe a grouping of I/O lines is as a port. Not all ports are equal, and the term I/O port doesn't really do justice to reality. There are different ports that can have different functions such as the following:

* PORTA is split between input and output.
* PORTB is output only.
* PORTC is called a bi-directional port, and can be programmed as input or output.
* PORTD is also programmable as input or output, but only 6 of its bits are useful to the programmer.
* PORTE (PORTE) is input only.

In this laboratory experiment, a C language program is written to display various patterns on the eight individual LEDs and LCD on the Dragon12-Plus board. The displayed pattern depends on the state of 8-bits of the board DIP switch and the Key Pad.

1.1 Aim

The main aim of this experiment is to understand the read and write data from input and output ports and introduce the time delay implementation using loops.

1.2 Objectives

1. Learn how a C program can access I/O registers.
2. Develop simple programs for an embedded system.
3. Use the CodeWarrior Integrated Development Environment for the

development of HCS12 microcontroller C programs.

1. Writing simple C language programs for interfacing DIP switches

and Key Pad with LCD.

1. To assemble, download and run a C program using CodeWarrior C

compiler and Dragon12 Plus Trainer board

1. Download, run, and test code on a Dragon12+ Board.

**2.** **Design and Results**

**2.1 Task-1: Reading DIP Switches and Writing them to LEDs**

In the first task, we were required to use the HCS12 microcontroller ports to drive 8 outputs to display the LEDs either on or off. Those 8 LEDs are connected to PORTB of the microcontroller. The figure below shows how to write a program to switch the LEDs on or off.

#include <hidef.h> /\* common defines and macros \*/

#include "derivative.h" /\* derivative-specific definitions \*/

void main(void)

{

int num = 0;

int count = 0;

EnableInterrupts;

DDRH = 0x00;

DDRB = 0xff;

DDRJ = 0xff;

DDRP = 0xff;

PTJ = 0x00;

PTP = 0x0f;

while(1) {

num = key\_pad();

PORTB = PTH;

}

Figure1: Reading DIP Switches and Writing them to LEDs

After writing the program, it should be downloaded to the microcontroller to verify the results.

The following figure shows the testing with board.

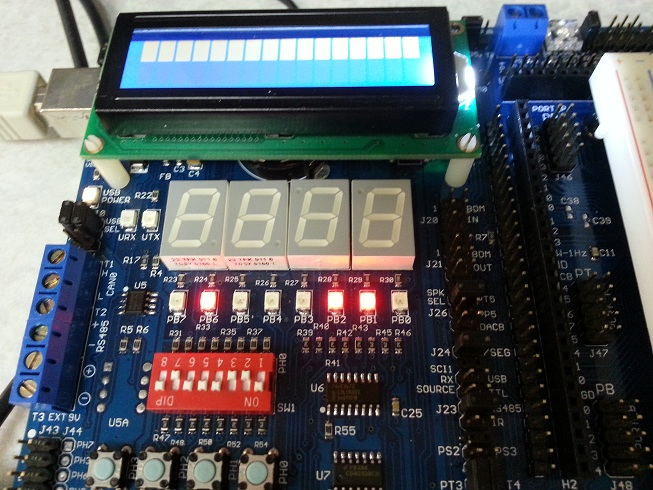
****

Figure2: Task 1 Board Testing

**2.2 Task-2: Reading Key Pad and Writing it to LEDs**

We were asked in this task to rewrite the program, which was written in the first task to read from the DIP switchesthat are connected to PTH and write those data in LEDs which are connected to PORTB. The program as shown below should read the state of the DIP switches on the dragon board and display them continuously to the LEDs.

#include <hidef.h> /\* common defines and macros \*/

#include "derivative.h" /\* derivative-specific definitions \*/

void main(void)

{

int num = 0;

int count = 0;

EnableInterrupts;

DDRH = 0x00;

DDRB = 0xff;

DDRJ = 0xff;

DDRP = 0xff;

PTJ = 0x00;

PTP = 0x0f;

DDRA =0x0f;

PUCR = 0x01;

while(1) {

PORTB = key\_pad();

}

int key\_pad(void)

{

int X;

while(1)

{

PORTA = 0XFE;

X = PORTA;

if (X == 0xEE)return 0x01;

if (X == 0xDE)return 0x04;

if (X == 0xBE)return 0x07;

if (X == 0x7E)return 0x0E;

PORTA = 0XFD;

X = PORTA;

if (X == 0xED)return 0x02;

if (X == 0xDD)return 0x05;

if (X == 0xBD)return 0x08;

if (X == 0x7D)return 0x00;

PORTA = 0XFB;

X = PORTA;

if (X == 0xEB)return 0x03;

if (X == 0xDB)return 0x06;

if (X == 0xBB)return 0x09;

if (X == 0x7B)return 0x0F;

PORTA = 0XF7;

X = PORTA;

if (X == 0xE7)return 0x0A;

if (X == 0xD7)return 0x0B;

if (X == 0xB7)return 0x0C;

if (X == 0x77)return 0x0D;

}

}

**Figure 3: Reading Key Pad and Writing it to LEDs**

Then the written program should be downloaded and verified using the Dragon Plus Trainer.

**2.3 Task-3: Writing to LCD**

In the third task, it is supposed to write a C language program using the LCD function calls to print the text message "ELCE 333 Lab" on the first line of Dragon 12 LCD and "Microprocessor" on the secondline. The two messages should be displayed for 0.5 second and cleared for another 0.5 second and so on as a flashing text. The flashing should be done alternatively so that when the first line appears, the second disappear and vice versa. In order to do that, the following code has been written:

#include <hidef.h> /\* common defines and macros \*/

#include "derivative.h" /\* derivative-specific definitions \*/

#include "lcd.h"

void main(void)

{

EnableInterrupts;

LCD\_Init();

while(1) {

delay(250);

delay(250);

LCDWriteLine(1, "ELCE 333 Lab");

delay(250);

delay(250);

LCD\_clear\_line(1);

delay(250);

delay(250);

LCDWriteLine(2, "Microprocessor");

delay(250);

delay(250);

LCD\_clear\_line(2);

}

}

**Figure 4: Writing to LCD**

This program should be downloaded to the Dragon Plus board to ensure the correction of the written code.

**2.4 Task 4: Reading DIP Switches and Writing them to LED’s and LCD**

In this task, we are asked to Re-examine the program of Task-1 and modify it to display the DIP Switches value in the LEDs as well as the LCD. In order to do the following task, the following code has been written:

#include <string.h>

#include "lcd.h"

void main(void)

{

int num = 0;

int count = 0;

EnableInterrupts;

DDRH = 0x00;

DDRB = 0xff;

DDRJ = 0xff;

DDRP = 0xff;

PTJ = 0x00;

PTP = 0x0f;

DDRA =0x0f;

PUCR = 0x01;

LCD\_Init();

LCDWriteLine(1, "\*Enter Number\*");

while(1) {

PORTB = PTH;

LCD\_clear\_line(2);

LCDWriteLine(1, "RESULT = ");

LCDWriteLine(2, "");

LCDWriteInt(PORTB);

delay(255);

delay(255);

LCD\_clear\_line(2);

}

}

Figure 5: Reading DIP Switches and Writing them to LED’s and LCD

Finally the code should be implemented using the Dragon Plus Trainer board to verify the correction of the full program.

**2.5 Task 5: Interfacing LCD with Key Pad and Push Buttons**

In the final task, we are supposed to write a C language program that prompts the user to enter a number.

The number should be displayed on the LCD display second line if SW5 is not pressed and should be divided by 2 if SW5 is pressed, which is “PTH\_PTH0” asshown in the code. In order to do so we defined “\_*FLOAT\_*IS\_IEEE32\_” in order to use integer values in a float variable. By this we were able to solve the problem of parsing and do the operation successfully. The code of this program is shown in the figure below:

#include <hidef.h> /\* common defines and macros \*/

#include "derivative.h" /\* derivative-specific definitions \*/

#define \_FLOAT\_IS\_IEEE32\_

#include "lcd.h"

void main(void)

{

int num = 0;

float result= 0;

EnableInterrupts;

DDRH = 0x00;

DDRB = 0xff;

DDRJ = 0xff;

DDRP = 0xff;

PTJ = 0x00;

PTP = 0x0f;

DDRA =0x0f;

PUCR = 0x01;

LCD\_Init();

LCDWriteLine(1, "\*Enter Number\*");

while(1) {

num = key\_pad();

LCD\_clear\_line(2);

LCDWriteLine(2, "");

LCDWriteInt(num);

if(PTH\_PTH0 == 0){

result = (num\*1.0) /2

LCDWriteFloat(result);

}

else LCDWriteInt(num);

delay(255);

delay(255);

delay(255);

delay(255);

}

}

int key\_pad(void)

{

int X;

while(1)

{

PORTA = 0XFE;

X = PORTA;

if (X == 0xEE)return 0x01;

if (X == 0xDE)return 0x04;

if (X == 0xBE)return 0x07;

if (X == 0x7E)return 0x0E;

PORTA = 0XFD;

X = PORTA;

if (X == 0xED)return 0x02;

if (X == 0xDD)return 0x05;

if (X == 0xBD)return 0x08;

if (X == 0x7D)return 0x00;

PORTA = 0XFB;

X = PORTA;

if (X == 0xEB)return 0x03;

if (X == 0xDB)return 0x06;

if (X == 0xBB)return 0x09;

if (X == 0x7B)return 0x0F;

PORTA = 0XF7;

X = PORTA;

if (X == 0xE7)return 0x0A;

if (X == 0xD7)return 0x0B;

if (X == 0xB7)return 0x0C;

if (X == 0x77)return 0x0D;

}

}

**Figure 6: Interfacing LCD with Key Pad and Push Buttons**

**3. Assignment Questions**

**Q1)** Create a C language program that checks the state of PH2 (while all DIP switches at their high position) and flashes all the eight LEDs of Port B from left to right if PH2 is high, and flashes them from right to left (like they do when the board is first powered on) if PH2 was low. Hint: Use the keyword >> to shift right or << to shift left.



**Q2)**

****

****

****

5. Conclusions and Recommendations

To sum up, we were able to understand the usage of dipswitches, LEDs, and LCD of the microcontroller. In the first task, we understand of the usage of PORTB in how to control the LEDs. Also, the second task, we used the function of the KeyPad, whichidentify which Button is pressed. In the third task, the code shows the way of displaying a text on LCD with some delay between the first line of the LCD and the second line. Moreover, the fourth task combinesthe first task and third task by reading the input from the DIP switches and writing the output to LEDs and LCD.Finally,we were required to do a program to perform a division operation that is controlled by pressing on the SW5.

The tasks of the fourthlab are successfully fulfilled. It is obvious that the objectives are successfully reached. The ability ofUnderstanding the input/ output lines as ports is achieved easily. Reading and writing the data from IO ports have been done. Also, we were able to download and test the code on the dragon 12 +. Furthermore, we were able to know the effect of the time delay on the LEDs and other operations. The instructions manual of dragon trainer help us in dealing with the hardware implementation. There were not any problems or risks faced in this practical session. Finally, it seemed easier to write a code program and check it by the hardware implementation because of the serious practice andpaying attention to the lab instructor which helped to avoid errors.