Microprocessor Systems Laboratory (ELCE333)

Laboratory Experiment No.4

MARKINGSHEET

Student Names and ID No's: **Amal AlQasimi 100036830**
Shaikha AlBesher 100037064

Laboratory Section: **Wednesday** **Experiment Date:** **February 18, 2015**

No.	**Criteria**	**Description**	**Weight %**	**Mark**	**Comments**
1	Pre-lab	A mark will be allocated to each student that reflects his preparations for the lab.	20		
2	Performance In the lab	A mark will be allocated to each student individually that reflects his performance in the lab	10		
3	Results and Analysis	Documentation and analysis of the results for each task performed in the lab	30		
4	Summary/ Conclusions	Conclusions for each task performed in the lab	10		
5	Assignment Questions	Answers to assignment questions	20		
6	Report Presentation	Overall presentation of the report including proper layout and clarity of figures, tables, and graphs. Correct use of English language.	10		
	Total		**100**		

KHALIFA UNIVERSITY

ELCE 333: Microprocessor Systems Laboratory

Lab Report- Experiment No.4

Experiment Title: HCS12 Input and Output Ports

Completed by:

Amal Al Qasimi (100036830)

Shaikha Al Besher (100037064)

Lab instructors:

Mahmoud Khonji

Mohammed Al Zaabi

Spring 2015

Table of Contents

Summary

The following lab report will include the details of the fourth Microprocessor System’s Laboratory lab session. It will first start off by stating the aims and objectives of the tasks assigned to us throughout the lab session. Following the aims and objectives, the report will discuss the purpose of each task separately with the results obtained upon the completion of the task. Based on the observations of the tasks completed, the analysis and interpretation section will include the explanation of why we obtained the results we did. Finally the report will end with a brief conclusion. The assignment questions given to us as part of the lab are answered after the lab repot.

1. Introduction

Lab 4 can be described as somehow another version as well as a continuation of the previous lab. The variation of this lab is that instead of programming using assembly language we used C language. Advantages of C include that its syntax is a lot easier to learn than Assembler syntax. In addition, C is easier to use for making more complex programs. Nonetheless, assembly is a lower level programming language than C,so this makes it a good for programming directly to hardware. Is a lot more flexible alluding you to work with memory, interrupts, micro-registers ,etc. Regardless of these differences, this lab experiment mainly focuses on becoming more familiar and utilizing I/O interference such as the Keypad and the LCD screen.

Key Pad: The HCS12 board has a 4 x 4 keypad which is connected to PORT A. PORT A has to be configured as a bidirectional register so that PA0-PA3 are output columns and PA4-PA7 are input rows, as a result DDRA should equal 0x0F. Moreover, the pull up resistors for the pins associated with the core of PORT A should be on by setting PUCR to 0x01.

LCD screen: The LCD is a screen on the HCS12 board and it is connected to PORT K. It contains 2 lines; each line has the capacity for 16 characters. In order to initialize and access the LCD some pre-written C library functions can be used.

2. Aims and Objectives:

The following are the aims and objectives of the laboratory experiment:

Aim: The aim of this laboratory experiment is to make students familiar with the layout and structure of C language programs and their formats

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.

4- Writing simple C language programs for interfacing DIP switches and Key Pad with LCD.

5- To assemble, download and run a C program using CodeWarrior C compiler and Dragon12 Plus Trainer board.

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

3. Lab tasks

TASK-1: Reading DIP Switches and Writing them to LEDs

In this task, we are asked to write a program in C that reads the input from the DIP switches and outputs the result in the LEDS. This task is similar to a task completed in the previously lab with the only difference that the program is written in C. Below is the C code to perform this function:

```
void main(void) {
  /* put your own code here */

DDRB = 0xFF;
DDRJ = 0xFF;
DDRP = 0xFF;
PTJ  = 0x00;
PTP  = 0x0F;
DDRH = 0x00;
//key_pad();

              EnableInterrupts;


  for(;;) {  PORTB = PTH;
   // _FEED_COP(); /* feeds the dog */
  } /* loop forever */
  /* please make sure that you never leave main */
}
```

The code starts off by setting the LEDs as output (B & J = output) and disabling the 7-segments to use the LEDs. In C language, the value of the ports is simply assigned using the symbol "=" unlike in the assembly language. Furthermore, the inputs are the set to be the switches. Finally, the switches states are mirrored to the LEDs by assigning port H to Port B.

TASK-2 Reading Key Pad and Writing it to LEDs

In this task the code had to be changed in a manner where the input was now the Key Pad. Here is the code after some adjustments:

```
void main(void) {
 /* put your own code here */

   DDRB=0xFF;
   DDRJ=0xFF;
   DDRP=0xFF;
   PTJ=0x00;
   PTP=0x0F;
      DDRA=0x0F;
   PUCR=0x01;
               EnableInterrupts;

  for(;;) {
  PORTB=key_pad();


    _FEED_COP(); /* feeds the dog */
  } /* loop forever */
 /* please make sure that you never leave main */
}

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;
}
}
```

A subroutine was used to assign the LEDs to the key pad, Port A is used for the key pad, which is enabled at the begging of the coding. After that, the if statements are used to set the key pad coding as described in Dragon12 manual. The board worked as expected and the LEDs mirrored the value of the keypad button.

TASK-3: Writing to LCD

The objective of the third task was to write a C language code using the LCD function calls to print in the LCD of the Dragon12+ board "ELCE 333 Lab" on the first line and "Microprocessor" on the second line. Both messages must be demonstrated for 0.5 second and cleared for another 0.5 second and so on as a flashing text. The process of the flashing text is when the first line appears, the second one disappears and vice versa. The code used to process the task is shown below.

```
#include <hidef.h>      /* common defines and macros */
#include "derivative.h"      /* derivative-specific definitions */
#include "lcd.h"


void main(void) {
 /* put your own code here */


LCD_Init();

  for (;;) {
     LCDWriteLine(1, "ELCE 333 Lab");
     delay(500);
     LCD_clear_disp( );
     delay(500);
     LCDWriteLine(2, "Microprocessor");
     delay(500);
     LCD_clear_disp( );
     delay(500);

   // _FEED_COP(); /* feeds the dog */
  } /* loop forever */
 /* please make sure that you never leave main */
}
```

TASK-4: Reading DIP Switches and Writing them to LED's and LCD

In this task, the program of Task-1 was re-examined and modified to display the DIP Switches value in the LEDs as well as the LCD. Below is the C code for this task.

```
void main(void) {
  /* put your own code here */
LCD_Init();

DDRB = 0xFF;
DDRJ = 0xFF;
DDRP = 0xFF;
PTJ  = 0x00;
PTP  = 0x0F;
DDRH = 0x00;
//key_pad();

            EnableInterrupts;


  for(;;) {  PORTB = PTH;
          LCDWriteInt(PORTB);
          LCDWriteLine(1, " ");
          delay(1000);
   // _FEED_COP(); /* feeds the dog */
  } /* loop forever */
  /* please make sure that you never leave main */
}
```

TASK 5- Interfacing LCD with Key Pad and Push Buttons

Task 5 required to write a C program that read number from the board and displayed it on the LCD. A condition set what when SW5 was off, the number is to be displayed normally on the second line of the LCD and if it was on, the number will be divided by 2 and displayed as well on the second line. Below is the C code for this task.

```
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;
}
}
void main(void) {
int num;
float half;

  DDRA=0x0F;
  PUCR=0x01;
              EnableInterrupts;

LCD_Init();

for(;;) {

LCDWriteLine(1,"*Entetr Number*");
num=key_pad();
half=(num)2.0;

delay(50);

LCD_clear_disp();

if(PTH_PTH0==1) {

LCDWriteLine(2,"Number= ");
LCDWriteInt(num);
   } else{
LCDWriteLine(2,"Number/2=");
LCDWriteFloat(half);
   }
     _FEED_COP(); /* feeds the dog */
}
```

Issues with task 5:

When we tried to display the number directly without using the SW5 condition it worked perfectly however upon adding another float variable and this condition we received an error message stating that there was problem with the float variable. This error was not in the actual initialization but when we started using the variable later on like in the if statement.

4. Analysis and Interpretation

Laboratory experiment 4 is about functioning C language in CodeWarrior, and it consists of five tasks. Task 1 was about converting the assembly code of Task 3 in the previous laboratory into a C code. In Task 2 the input was read from the Key pad and the output was displayed on the LEDs. To use the Key pad some extra functions has been added to the code. Task 3 used LCD. We were able to write two lines and clear them. Task 4 the input was inserted using the DIP switches and the output was displayed on the LEDs and the LCD. Task 5 required to write a C language program that prompts the user to enter a number. The number should be displayed as is on the LCD display second line if SW5 is not pressed and should be divided by 2 if SW5 is pressed. The code included if-else statements as the other tasks did not.

5. Conclusions and Recommendations

This laboratory encompassed five different tasks each of them determines a specific skill to achieve in order to understand how to use the CodeWarrior program and the Dragon12 board with C language. It showed how to convert assembly code into a C code, as in the First Task. On the screen of LCD, we learned how to display words, numbers, symbols, etc. DIP switches were used as inputs where the output was displayed on the LCD. Also, we learnt how to use switches to do simple calculations, dividing number by 2. Some coding was needed to do the calculations. The concept of learning this type of knowledge can help on accomplish various kinds of projects in the future.

6. Assignment Questions

1)

```
Void main(void) {
EnableInterrupts:

  DDRB=0xFF           ; Setting Port B as output (LEDS)
  DDRJ=0xFF           ; Setting Port j as output to set the LEDs
  DDRP=0xFF           ; disabling the 7-segments
  PTP=0x0F            ; disabling the 7-segments

  //PORT H
  DDRH=0x00           ;Set Port H as input
```

```
For(;;){
While (PTH == 251)
  If (PTH_PTH@ == 1)
    {
        PORTB = 128;
         For(int=1; i<8; i++)
         PORTB = PORTB-1;
      }
  Else
{
PORTB=1;
     For(int i=1; i<8 : i++)
     PORTB=PORTB-1;
   }
```

2)

```
Float num1, num2, results;
Char op;
DDRH = 0x00;
PUCR = 0xFF;
EnableInterrupts;
LCD_Init () ;
For(;;) {

if (PTH_PTH4 == 0 && PTH_PTHS ==0 && PTH_PTH6 ==0 && PTH_PTH7 == 0) num1= 0;
if (PTH_PTH4 == 1 && PTH_PTHS ==0 && PTH_PTH6 ==0 && PTH_PTH7 == 0) num1= 1;
if (PTH_PTH4 == 0 && PTH_PTHS ==1 && PTH_PTH6 ==0 && PTH_PTH7 == 0) num1= 2;
if (PTH_PTH4 == 1 && PTH_PTHS ==1 && PTH_PTH6 ==0 && PTH_PTH7 == 0) num1= 3;
if (PTH_PTH4 == 0 && PTH_PTHS ==0 && PTH_PTH6 ==1 && PTH_PTH7 == 0) num1= 4;
if (PTH_PTH4 == 1 && PTH_PTHS ==0 && PTH_PTH6 ==1 && PTH_PTH7 == 0) num1= 5;
if (PTH_PTH4 == 0 && PTH_PTHS ==1 && PTH_PTH6 ==1 && PTH_PTH7 == 0) num1= 6;
if (PTH_PTH4 == 1 && PTH_PTHS ==1 && PTH_PTH6 ==1 && PTH_PTH7 == 0) num1= 7;
if (PTH_PTH4 == 0 && PTH_PTHS ==0 && PTH_PTH6 ==0 && PTH_PTH7 == 1) num1= 8;
if (PTH_PTH4 == 1 && PTH_PTHS ==0 && PTH_PTH6 ==0 && PTH_PTH7 == 1) num1= 9;
if (PTH_PTH4 == 0 && PTH_PTHS ==1 && PTH_PTH6 ==0 && PTH_PTH7 == 1) num1= 10;
if (PTH_PTH4 == 1 && PTH_PTHS ==1 && PTH_PTH6 ==0 && PTH_PTH7 == 1) num1= 11;
if (PTH_PTH4 == 0 && PTH_PTHS ==0 && PTH_PTH6 ==1 && PTH_PTH7 == 1) num1= 12;
if (PTH_PTH4 == 1 && PTH_PTHS ==0 && PTH_PTH6 ==1 && PTH_PTH7 == 1) num1= 13;
if (PTH_PTH4 == 0 && PTH_PTHS ==1 && PTH_PTH6 ==1 && PTH_PTH7 == 1) num1= 14;
if (PTH_PTH4 == 1 && PTH_PTHS ==1 && PTH_PTH6 ==1 && PTH_PTH7 == 1) num1= 15;
if (PTH_PTH3 ==1) op= ‘/’; result= num1/num2;
if (PTH_PTH2 ==1) op= ‘/’; result= num1*num2;
if (PTH_PTH1 ==1) op= ‘/’; result= num1-num2;
if (PTH_PTH0 ==1) op= ‘/’; result= num1+num2;
LCDWriteLine(1, “No.1=”);
LCDWriteFloat(num1);
num2= key_pad();
LCDWriteLine(1, “No.2=”);
LCDWriteFloat(num2);
LCDWriteLine(2, “op=”);
LCDWriteFloat(op);
LCDWriteLine(2, “res=”);
result=
LCDWriteFloat(result);
   _FEED_COP(); /* feeds the dog */
  } /* loop forever */
  /* please make sure that you never leave main */
}

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;
}
}
```

References

1. CourseTextBook
2. http://www.evbplus.com/
3. MC9S12DP256User’sManual