Kuwait University
College of Engineering and
Petroleum Computer
Engineering Department

CpE-<Course No. 0612364>: <EMBEDDED SYSTEMS LAB>

Semester: <Fall 2021>

Section <No.11A>

Assignment No. < EMBEDDED SYSTEMS LAB PROJECT>

Student Name: <Yassine Serrid, Sulaiman Almuhaitib >

Student Id: <2181156439, 2181146376>

Instructor Name: <Dr. Mohammad Alfailakawi>

TA Name: < Eng. Dia Buhamra >

Date: <17th Of January 2022>

Contents

Table Of Figures

Kuwait University

Computer Engineering Department
CpE-364 Embedded Systems Laboratory

Fall 2021-2022
GPA Calculator
Project Description

Learning Objectives:

- To enhance programming skills in assembly language programming.
- To interface 8051-microcontroller with I/O devices such as, switches, keyboard and LCD display.
- To gain experience with modern CAD tools such as MicroTRAK/51-C2 Kit to model and simulate microcontroller based systems.
- To design and implement a simple microcontroller based application including hardware and software.
- To develop skills necessary for teamwork to accomplish a project.

Title: *Design and implementation of an 8051-microcontroller based system to implement a GPA Calculator*.

1. Specifications:

This semester project is designing and implementing an 8051-microcontroller based system to implement **GPA Calculator**. The system will get its input from the keypad and display the output on LCD.

The project consists of four different phases which when integrated together (with minor modification), can be used to realize the complete system.

1.1 How the System Work:

The system will display a welcome message on LCD and wait for an external interrupt pressed by the user.

- If no interrupt is not detected, then the program will not continue.
- If the user presses an interrupt button, a menu will be displayed on LCD as shown below.
  (MAIN MENU: 1)CAL GPA 2)GPA ANALYSIS 3)EXIT)
- Select the process to calculate or analyze the GPA or exit the program by entering its corresponding number using the keypad.
- If the user selected the first option it will enter the number of courses(ENTER THE NUMBER OF CURCES), it will iterate every course credit(ENTER THE CURCES CREDET 4)MJ 3)MR 2)LAB) aas shown in table 1 and grade(ENTER THE GRAID 4)A 3)B 2)C 1)D 0)F) according to table 2, it will iterate according to the number of courses.
- It will calculate the GPA according to the input from the user. (YEUR GPA IS:)
- If the user selected the second option it will ask the user if he wants to analyze the saved GPA from option 1 ((YOU HAVE A SEVE GPA USE IT? 1)YES 2)NO) or enter another GPA selected from the user(ENTER YOUR GPA 0 TO 4).
- It will analyze the GPA and then it will display the best recommendation for the user as shown in Figure1.
- If the user selected the third option, the program will quit after displaying a closing message.

It will check if the user enters correct inputs, then displays the output on the LCD according to the option selected.

Example operation 2:
If the user wanted to analyze his GPA calculated from option 1
1
The program will fetch the GPA from the memory (e.g 3) :
The program will display (V.Good) and wait for 3 seconds then display recommendation (Great,but work more /n you can do it)
The program will return to the main menu.

1.2 System Limitation:

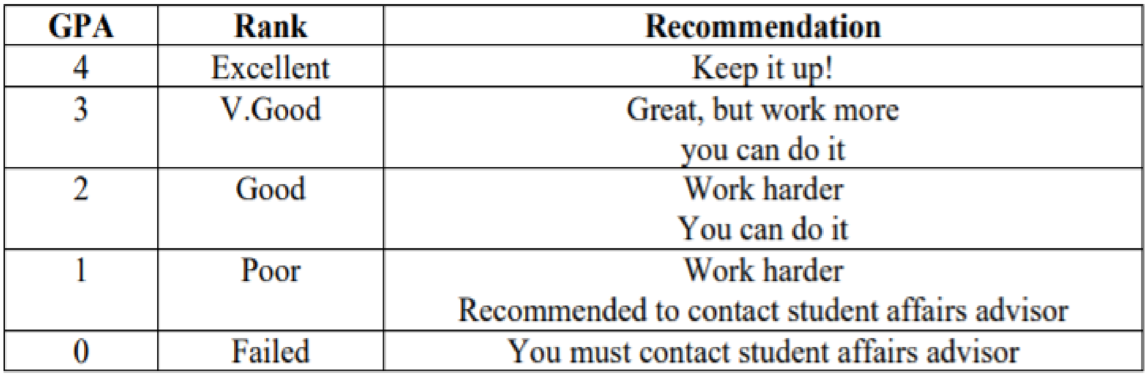
In your implementation assume that the credits for each course and grade points will be counted based on the following tables as following:

Table 1: Credits of each course

Course	Course Credit
General Course	3
Major Course	4
Lab	2

Table 2: Credits of each grade

Grade	Grade Points
A	4
B	3
C	2
D	1
F	0



GPA	Rank	Recommendation
4	Excellent	Keep it up!
3	V.Good	Great, but work more you can do it
2	Good	Work harder You can do it
1	Poor	Work harder Recommended to contact student affairs advisor
0	Failed	You must contact student affairs advisor

Figure 1. Recommendation and rank according to the GPA

******** Note ********

- **Warning: All calculations required should be done in the program and no lookup tables (PREVIOUSLY CALCULATED FUNCTIONS VALUES) is strictly forbidden.**

2. Project Implementation:

The system must be implemented using assembly language utilizing the following resources:

- MicroTRAK Carrier Board
- MINI-MAX/51-C2 Microcontroller board
- 8051 I/O Module.
- 4X4 Keypad board, LCD242-BK LCD board.
- +5VDC Power supply & RS232 Serial Cable.

First of all, you should choose your design team partners (two students per group). The design teams are required to accomplish the following goals:

- o Develop the user-interface for the LCD such as what you want to display when you turn-on the power etc.? What message you want to display in case of error?
- o Develop the program to implement GPA Calculator. You must provide a detailed analysis of your program based on time and storage requirements. You should convert each digit to ASCII in order to be displayed on the LCD.
- o Integrate and simulate all the modules to test the functionality of your complete design.

3. System Operation:

The system must be designed to operate in the following manner:

1. The system will display a welcome message.
2. The system should use one push button to enable interrupt to enter the program.
3. If the push button is pressed the system will take the user to GPA Calculator main menu and begin executing the following:
4. The system will display a selection menu on the LCD and ask the user to choose one of the classes (1)CAL GPA 2)GPA ANALYSIS 3)EXIT)

5. The user will enter one choice using the keypad.
   5.1 If the user enter wrong choice, an error message will be displayed and **GOTO step 4.**
   5.2 If the user enter vailed choice the program will proceed according to the desird option.
6. If the user selected the first option.
   6.1 It will ask the user to enter the number of courses.
   6.2 It will ask the user to enter the course credit.
   6.3 It will ask the user to enter the course grade.
   6.4 The program will repeat steps 6.2 and 6.3 based on the number selected on step 6.1.
   6.5 The program will calculate the GPA with three decimal numbers.
   6.6 It will display the GPA on LCD.
   6.7 Save the GPA calculated on the RAM.
7. If the user selected the second option:
   7.1 Ask the user if he want to used the stored GPA in the RAM or input a GPA.
      7.1.1. If the user selected to analyz based on the GPA that will be loaded for the RAM.
      7.1.2. If the user selected to analyz by inputing a GPA selected by the user.
   7.2 It will analyz the GPA value that is loaded.
   7.3 Display the rank and best recommendation for the user.
8. If the user selected the third option:
   8.1 The program will display closing message.
   8.2 The program will freez on the closing message.

4. Project Phases and Grading:

The timeline for the different phases of the project are given in Table 3. For every phase of the project except the last phase (complete system), each team is required to:

- Submit a flow chart that represents the algorithmic implementation of the phase. The flow chart must be typed using appropriate program (not hand written).
- Submit the source code in OCS for the phase with appropriate comments.
- Demonstrate the correct operation of the program to lab engineer. Note that all the actions should be clearly visible.

Table 3: Project Timeline

Phase	Part to be implemented	Timeline
1	Implement the GPA Calculator by entering the values using registers and memory locations.	16/12/2021
2	Flashing indicator for some time and display messages on the LCD.	23/12/2021
3	Use Keypad to enter the input values.	06/01/2022
4	Using Interrupt to enter the program GPA calculator.	20/01/2022
5	Submit final work and final report	20/01/2022

4.1 Phase 1 Specification:

In this phase of the project, you must give **two alternative designs** for the coding part but implement only one. You should design a GPA Calculator system using assembly language.

First Solution	Second Solution
1- The user will input the total number of credits. 2- The user will enter the grade and credit for each course. 3- The program will check if the total remining number of credits equal zero or the carry flag becomes 'one'. 3.1- If the total remining number of credits doesn't equal zero the carry flag will not be equal one and it will repeat step number 2. 3.2- If the total remining number of credits equal zero the program will procced to step 4. 3.3- If the total remining number of credits doesn't equal zero the carry flag will be equal one and it will display error message and will return to the main menu. 4- The program will calculate and display the GPA with 3 decimal places.	1- The user will input the total of courses 2- The user will input the total number of credits. 3- The user will input the multiplication of each course (course x grade). 4- The program will calculate the summation of (course x grade). 5- The program will divide the value that calculated in step 4 over the total number of course entered by the user in step 2. 6- The program will calculate and display the GPA with 3 decimal places.

Implementation for the first solution:

```
E1:
        ; LCD ASKS THE USER ABOUT THE NUMBER OF  COURSES
        ;

        MOV 18H,#0H
        MOV 19H,#0H
        MOV 1AH,#0H
        MOV 1BH,#0H

        MOV DPTR,#E1STR1
        LCALL LCDL1


        MOV DPTR,#E1STR2
        LCALL LCDL2




        ;
        ACALL KEYPAD
        ANL A,#00001111B
        MOV 26H,A
        ACALL KEYPAD
        ANL A,#00001111B
```

```
MOV 27H,A
MOV SP,#3FH
MOV R0,#0H
MOV A,26H
MOV B,#10
MUL AB
ADD A,27H
MOV 26H,A
MOV 27H,26H

E1CD:
;LCD ASKS THE USER ABOUT THE COURSE CREDIT
;LCD 4 MAJOR 3 MINOR 2 LAB
;
MOV DPTR,#E1STR3
LCALL LCDL1



MOV DPTR,#E1STR4
LCALL LCDL2




;
ACALL KEYPAD
ANL A,#00001111B
PUSH 0E0H
;LCD ASKS THE USER ABOUT THE COURSE GRADE
;;;

MOV DPTR,#E1STR5
LCALL LCDL1

MOV DPTR,#E1STR6
LCALL LCDL2



;;;
ACALL KEYPAD
ANL A,#00001111B
PUSH 0E0H

TOTALNUMBEROFCREDITS:
MOV A,27H
DEC SP
```

```
POP 28H
INC SP
INC SP
SUBB A,28H
MOV 27H,A
JNC NOERROR
MOV DPTR,#ERRORSTR
LCALL LCDL1
MOV DPTR,#ERRORSTR1
LCALL LCDL2
LJMP MM
NOERROR:
CJNE 27H,#0,E1CD




MOV R7,SP
INC R7
CLR A
MOV 37H,26H


MOV R0,#40H
ECG:
MOV A,@R0
CJNE A,#0,E1ALTER
SJMP ALTER2
E1ALTER:
INC R0
MOV B,@R0
INC R0
MUL AB
MOV 20H,A
PUSH 20H
SJMP ECG
ALTER2:
MOV A,R7
MOV R1,A
CLR A
ECC:
ADD A,@R1
INC R1
DJNZ 33H,ECC
MOV 39H,A
MOV A,39H
MOV B,37H
DIV AB
```

```
        MOV 18H,A
        MOV R1,#19H
        MOV R6,B
        CJNE R6,#0H,LOOP
        SJMP EXIT
        LOOP:
        MOV 20H,#1H
        BNZ:
        MOV A,B
        MOV B,#0AH
        MUL AB
        MOV B,37H
        DIV AB
        MOV @R1,A
        INC R1
        INC 20H
        JB 02H, EXIT
        MOV R6,B
        CJNE R6,#0H,BNZ
        EXIT:



        ;LCD DISPLAY
        ;DISPLAY 18H
        ;DISPLAY .
        ;DISPLAY 19H
        ;DISPLAY 1AH
        ;DISPLAY 1BH
        ;;;


        MOV DPTR,#E1STR7
        LCALL LCDL1


    MOV A,#0CH
    ACALL CMD
    MOV A,#0000B
        LCALL CMD
        LCALL LDELAY
        MOV A,18H
        ADD A,#30H
        LCALL WCHR

        LCALL LDELAY

        MOV A,#'.'
```

```
            LCALL WCHR

            LCALL LDELAY
            MOV A,19H
            ADD A,#30H
            LCALL WCHR


            LCALL LDELAY
            MOV A,1AH
            ADD A,#30H
            LCALL WCHR

            LCALL LDELAY
            MOV A,1BH
            ADD A,#30H
            LCALL WCHR



            LCALL DELAY3SEC
            LCALL DELAY3SEC

            ;;;
            LJMP MM

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

In this phase the code must function properly and with comments. We will check the system functionality using the debugger of Micro-IDE. Submit a flow chart that represents the algorithmic implementation of the phase. Even though not required for this phase, all appropriate calculation needed to implement delay must be documented in the final report.

4.2 Phase 2 Specifications:

In this phase of the project, you are required to write a program that will display the following sentences on the LCD:

Screen 1:

```
HELLO & WELLCOME TO
YOUR GPA CAL & ANALYSIS
```

Screen 2:

```
MAIN MENU: 1)CAL GPA
2)GPA ANALYSIS 3)EXIT
```

Screen 3:

```
ENTER THE NUMBER OF
CURCES
```

Screen 4:

```
ENTER THE CURCES CREDET
4)MJ  3)MR  2)LAB
```

Screen 5:

```
ENTER THE GRAID:-
4)A 3)B 2)C 1)D 0)F
```

Screen 6:

```
YEUR GPA IS:
(GPA CALCULATED)
```

Screen 7:

```
MAIN MENU: 1)CAL GPA
2)GPA ANALYSIS 3)EXIT
```

Screen 8:

```
YOU HAVE A SEVE GPA
USE IT? 1)YES 2)NO (SELECTED NO)
```

Screen 9:

```
ENTER YOUR GPA 0 TO 4 (e.g THE USER ENTERED 3)
```

Screen 10:

```
V.Good
```

Screen 11:

```
Great, but work more
you can do it
```

Screen 12:

```
MAIN MENU: 1)CAL GPA
2)GPA ANALYSIS 3)EXIT (SELECTED 5)
```

Screen 13:

```
ERROR-ERROR:
INPUT IS OUT OF BUND
```

Screen 14:

```
MAIN MENU: 1)CAL GPA
2)GPA ANALYSIS 3)EXIT (SELECTED 2)
```

Screen 15:

```
YOU HAVE A SEVE GPA
USE IT? 1)YES 2)NO (e.g. SELECTED YES, GPA CALCULTED FROM OPTION 1 = 2.2)
```

Screen 17:

```
Good
```

Screen 18:

```
Work harder
You can do it
```

Screen 19:

```
MAIN MENU: 1)CAL GPA
2)GPA ANALYSIS 3)EXIT (SELECTED 3)
```

Screen 20:

```
THANK YOU FOR USING
OUR APPLICATION
```

The results you will display is already stored in the program from phase 1.

Submit a flow chart that represents the algorithmic implementation of the phase and the source code with comments.

4.3 Phase 3 Specifications:

In this phase of the project, you are required to utilize the keypad to select the process, number of courses, credit, and the grade for each course. Then displays the GPA on LCD as from phase 2. Your design must use the keypad available on the MINI-MAX kit.

Submit a flow chart that represents the algorithmic implementation of the phase and the source code with comments.

4.4 Phase 4 Specifications:

In this phase, you need to utilize the previous phases of this project (modified appropriately) with proper extensions to create the complete system as specified in section 3 of this document (System Operation). You are required to use the external interrupt in this phase to run the system.

Your design must use an external interrupt available on the MINI-MAX kit. Moreover, you need to utilize the previous phases of this project (modified appropriately) with proper extensions to create the complete system as specified at the beginning of this document and submit the final report.

Appendix:

The program flow chart:

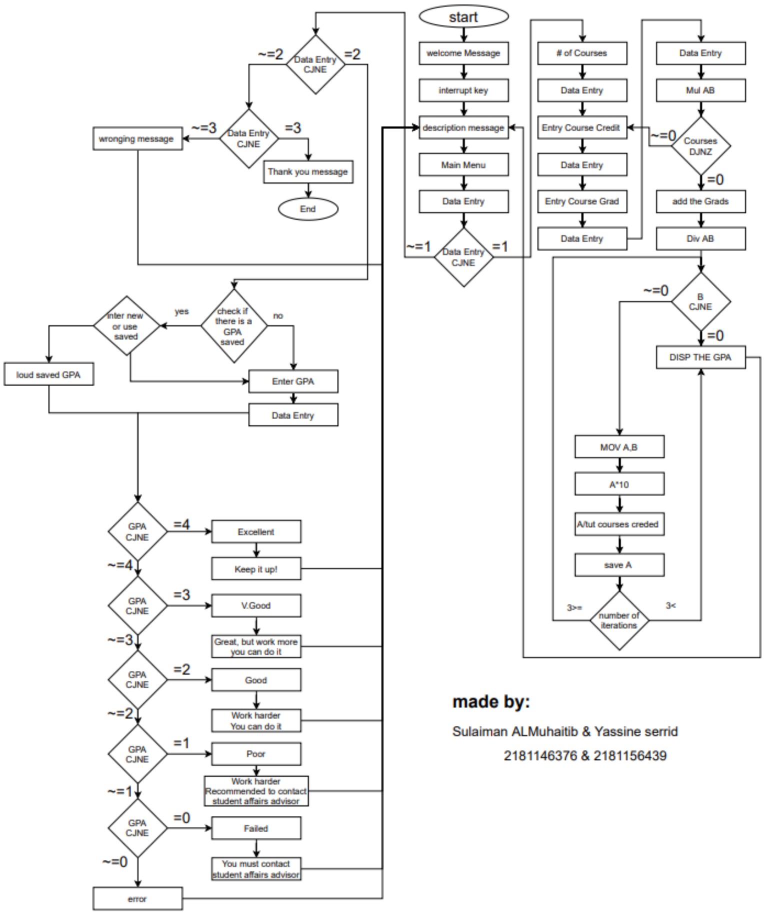


Figure 2. The Program Flowchart

The source code of the program:

```
;Project Phase 2
;Name: SULAIMAN ALMUHAITIB & YASSINE SERRID
;ID; 2181146376 & 2181156439

;The system is simply created to design and implement an 8051-microcontroller based system
;that allows the user to calculate Grade Point Average (GPA).

LDATA EQU P0
EN EQU P0.2
RS EQU P0.0
RW EQU P0.1

ORG 0H

        LJMP MAIN
        ORG 0003H
                LCALL EX0SIR
                ORG 0030H

        ;LCD(WELOCME PAGE)
        ;INTERORPY KEY
        ;;

MAIN:

        ;LCD(WELOCME PAGE)
        ;INTERORPY KEY
        MOV DPTR,#BEGSTR1
        LCALL LCDL1



        MOV IE,#10001001B
        SETB IE0
        SJMP $
                EX0SIR:
                INC R7
                MOV DPTR,#BEGSTR2
        LCALL LCDL2
        LCALL DELAY3SEC
MOV 18H,#0FFH
                MM:
        ;LCD(MAIN MENUE)
        ;;
        MOV DPTR,#MMSTR
        LCALL LCDL1
```

```
MOV DPTR,#MMSTR2
LCALL LCDL2




;;

LCALL KEYPAD


CJNE A,#31H,SKIP1
SJMP E1
SKIP1:
CJNE A,#32H,SKIP2
LJMP E2
SKIP2:
CJNE A,#33H,SKIP3
LJMP E3
SKIP3:
;LCD(ERROR) IF IT'S NOT 1,2 OR 3
;;

MOV DPTR,#ERRORSTR
LCALL LCDL1

MOV DPTR,#ERRORSTR1
LCALL LCDL2

LCALL DELAY3SEC



;;
SJMP MM

E1:
; LCD ASKS THE USER ABOUT THE NUMBER OF  COURSES
;

MOV 18H,#0H
MOV 19H,#0H
MOV 1AH,#0H
MOV 1BH,#0H

MOV DPTR,#E1STR1
LCALL LCDL1
```

```
MOV DPTR,#E1STR2
LCALL LCDL2




;
ACALL KEYPAD
ANL A,#00001111B
MOV 30H,A
MOV SP,#3FH
MOV R0,#0H

MOV 35H,30H
MOV 36H,30H
MOV 34H,30H
MOV 33H,30H

E1CD:
;LCD ASKS THE USER ABOUT THE COURSE CREDIT
;LCD 4 MAJOR 3 MINOR 2 LAB
;
MOV DPTR,#E1STR3
LCALL LCDL1



MOV DPTR,#E1STR4
LCALL LCDL2




;
ACALL KEYPAD
ANL A,#00001111B
PUSH 0E0H
;LCD ASKS THE USER ABOUT THE COURSE GRADE
;;;

MOV DPTR,#E1STR5
LCALL LCDL1

MOV DPTR,#E1STR6
LCALL LCDL2
```

```
;;;
ACALL KEYPAD
ANL A,#00001111B
PUSH 0E0H
DJNZ 30H,E1CD

MOV R7,SP
INC R7
CLR A
MOV R1,#40H
ADDCC:
ADD A,@R1
INC R1
INC R1
DJNZ 35H,ADDCC
MOV 37H,A

MOV R0,#40H
ECG:
MOV A,@R0
INC R0
MOV B,@R0
INC R0
MUL AB
MOV 20H,A
PUSH 20H
DJNZ 34H,ECG


MOV A,R7
MOV R1,A
CLR A
ECC:
ADD A,@R1
INC R1
DJNZ 33H,ECC
MOV 39H,A
MOV A,39H
MOV B,37H
DIV AB
MOV 18H,A
MOV R1,#19H
MOV R6,B
CJNE R6,#0H,LOOP
SJMP EXIT
LOOP:
        MOV 20H,#1H
        BNZ:
        MOV A,B
        MOV B,#0AH
        MUL AB
        MOV B,37H
        DIV AB
        MOV @R1,A
        INC R1
        INC 20H
        JB 02H, EXIT
        MOV R6,B
        CJNE R6,#0H,BNZ
        EXIT:



        ;LCD DISPLAY
        ;DISPLAY 18H
        ;DISPLAY .
        ;DISPLAY 19H
        ;DISPLAY 1AH
        ;DISPLAY 1BH
        ;;;


        MOV DPTR,#E1STR7
        LCALL LCDL1


MOV A,#0CH
ACALL CMD
MOV A,#0000B
        LCALL CMD
        LCALL LDELAY
        MOV A,18H
        ADD A,#30H
        LCALL WCHR

        LCALL LDELAY

        MOV A,#'.'
        LCALL WCHR

        LCALL LDELAY
        MOV A,19H
        ADD A,#30H
        LCALL WCHR
        LCALL LDELAY
        MOV A,1AH
        ADD A,#30H
        LCALL WCHR

        LCALL LDELAY
        MOV A,1BH
        ADD A,#30H
        LCALL WCHR



        LCALL DELAY3SEC
        LCALL DELAY3SEC

        ;;;
        LJMP MM

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;



E2:

MOV R4,18H
        CJNE R4,#0FFH,INORSEV



        NEWGRADE:
        ;DISPLAY ENTER YOUR GPA GRADE
        ;;;


        MOV DPTR,#E2STR1
        LCALL LCDL1




        ;;;

        LCALL KEYPAD
        ANL A,#00001111B
        MOV R4,A
        LJMP IMCALC
```

```
INORSEV:
;LCD (YOU HAVE SAVED GREAD)
;LCD (1 USE SAVE 2 INTER NEW GREAD)
;;;



MOV DPTR,#E2STR2
LCALL LCDL1


MOV DPTR,#E2STR3
LCALL LCDL2




;;;
CLR A
ACALL KEYPAD
CJNE A,#31H,SKIPIN1
SJMP IMCALC
SKIPIN1:
CJNE A,#32H,SKIPIN2
LJMP NEWGRADE
SKIPIN2:
;ERROR IF IT'S NOT 1 OR 2 (LCD DISPLAY)
;;;

MOV DPTR,#ERRORSTR
LCALL LCDL1



MOV DPTR,#ERRORSTR1
LCALL LCDL2

LCALL DELAY3SEC


;;;
LJMP INORSEV


IMCALC:
CJNE R4,#4H,SKIPAS4
SJMP AS4
```

```
SKIPAS4:
CJNE R4,#3H,SKIPAS3
LJMP AS3
SKIPAS3:
CJNE R4,#2H,SKIPAS2
LJMP AS2
SKIPAS2:
CJNE R4,#1H,SKIPAS1
LJMP AS1
SKIPAS1:
CJNE R4,#0H,SKIPAS0
LJMP AS0
SKIPAS0:

;LCD ERROR IF IT'S NOT 4,3,2,1 OR 0
;;;;;;


MOV DPTR,#ERRORSTR
LCALL LCDL1

MOV DPTR,#ERRORSTR1
LCALL LCDL2


LCALL DELAY3SEC




;;;;;;
LJMP MM
AS4:
;LCD "EXCELENT" /N "KEEP IT UP"
;;;


MOV DPTR,#E2STR4
LCALL LCDL1


MOV DPTR,#E2STR5
LCALL LCDL2

LCALL DELAY3SEC

;;;
LJMP MM
```

```
AS3:
;LCD "V.GOOD" /N "GREAT BUT WORK MORE YOU CAN DO IT"
;;;


MOV DPTR,#E2STR6
LCALL LCDL1


LCALL DELAY3SEC



MOV DPTR,#E2STR7
LCALL LCDL1


MOV DPTR,#E2STR8
LCALL LCDL2

LCALL DELAY3SEC

;;;
LJMP MM
AS2:
;LCD "GOOD" /N "WORK HARD YOU CAN DO IT"
;;;


MOV DPTR,#E2STR9
LCALL LCDL1

LCALL DELAY3SEC


MOV DPTR,#E2STR10
LCALL LCDL1



MOV DPTR,#E2STR11
LCALL LCDL2

LCALL DELAY3SEC

;;;
LJMP MM
AS1:
	;LCD "POOR" /N "WORK HARDER RECCOMENDED TO CNTACT STUDENT AFFAIRS ADIVSOR"
	;;;



	MOV DPTR,#E2STR12
	LCALL LCDL1


	LCALL DELAY3SEC


	MOV DPTR,#E2STR13
	LCALL LCDL1



	MOV DPTR,#E2STR14
	LCALL LCDL2

	LCALL DELAY3SEC


	;;;
	LJMP MM
	AS0:
	;LCD "FAILED" /N "YOU MUST CONTACT STUDENT AFFAIRS ADVISOR"
	;;;


	MOV DPTR,#E2STR15
	LCALL LCDL1


	LCALL DELAY3SEC


	MOV DPTR,#E2STR16
	LCALL LCDL1


	MOV DPTR,#E2STR17
	LCALL LCDL2

	LCALL DELAY3SEC

	;;;
```

```
LJMP MM

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


E3:
        ;LCD THANK YOU FOR USING OUR APPLICATION
        ;;;


        MOV DPTR,#E3STR1
        LCALL LCDL1


        MOV DPTR,#E3STR2
        LCALL LCDL2

LCALL DELAY3SEC
        ;;;


FF:     SJMP FF







        LCDL1:
        PUSH 0E0H
        LCALL INLCD
        MOV A,#8H
        LCALL CMD
        MOV A,#0000B
        LCALL CMD
        LCALL LDELAY
        LCALL WSTR
        POP 0E0H
        RET

        LCDL2:
        PUSH 0E0H
        MOV A,#0CH
        LCALL CMD
        MOV A,#0000B
        LCALL CMD
        LCALL LDELAY
        LCALL WSTR
        POP 0E0H
        RET




        ;-- LCD Initialization Procedure starts here -----
INLCD:
PUSH 7
MOV R7,#20
WAIT:
ACALL LDELAY ; Step 1
DJNZ R7,WAIT
MOV P0,#00000111B ; Initialise 3 control signals=1
MOV A,#0011B ; Step 2
ACALL CMD
ACALL LDELAY ; Step 3
MOV A,#0011B ; Step 4
ACALL CMD
MOV A,#0011B ; Step 5
ACALL CMD
MOV A,#0010B ; Step 6
ACALL CMD
MOV A,#0010B ; Step 7 – send high nibble
ACALL CMD
MOV A,#1000B ; send low nibble
ACALL CMD
MOV A,#0000B ; Step 8 – Turn off display – send high nibble
ACALL CMD
MOV A,#1000B ; send low nibble
ACALL CMD
MOV A,#0 ; Step 9 - Clear Display – send high nibble
ACALL CMD
MOV A,#0001B ; send low nibble
ACALL CMD
ACALL LDELAY ; 4.1 msec required for this command
MOV A,#0000B ; Step 10 - Set cursor Move RIGHT- send high nibble
ACALL CMD
MOV A,#0110B ; send low nibble
ACALL CMD
MOV A,#0000B ; Step 11 –send high nibble
ACALL CMD ; Turn ON Display, Cursor ON, Blink Cursor
MOV A,#1111B ; send low nibble
ACALL CMD
POP 7
RET
```

```
;--- End of LCD initialization -----
;---- Subroutine to write COMMAND in A to the LCD -------
CMD:
CLR RS ; RS = 0command write
ACALL COMMON
RET
;---- Subroutine to write character in A to the LCD -------
WCHR:
PUSH ACC
PUSH B
SETB RS ; RS = 1 data write
MOV B,A
SWAP A ; Move higher nibble to lower nibble
ACALL COMMON ; write operati
MOV A,B
ACALL COMMON
POP B
POP ACC
RET
;----- Common operation for CHAR write and COMMAND write
COMMON:
CLR RW
SWAP A ; Move Lower nibble to higher nibble
ANL A,#11110000B
ANL P0,#00000111B
ORL P0,A
SETB EN
CLR EN
ACALL LDELAY
RET
;---- Subroutine to write A STRING character by character -------
WSTR:
PUSH ACC
CONT1:
CLR A
MOVC A,@A+DPTR ; move character to A
JZ EXIT1
ACALL WCHR ; call procedure to write a CHAR
INC DPTR ; get next character
AJMP CONT1 ; go to CONT1
EXIT1:
POP ACC ; restore A
RET
;------------ ˜ 5.4msec DELAY -----------------------------------------
LDELAY:
PUSH 0
PUSH 1 ; save register1.
MOV R1,#20 ; move 12 to register R1
```

```
CON4:
MOV R0,#250
DJNZ R0,$
DJNZ R1,CON4 ; decrease R1, if R1 !=0, go to CON4
POP 1
POP 0
RET
;;;;;;;;;;;;;;;;;;;;;;;;

        DELAY:
        MOV 7FH,R5
        MOV 7EH,R4
        MOV 7DH,R3
        MOV R5,#20
        X3:MOV R4,#200
        X2:MOV R3,#250
        X1:DJNZ R3,X1
        DJNZ R4,X2
        DJNZ R5,X3
        MOV R5,7FH
        MOV R4,7EH
        MOV R3,7DH
        RET

        DELAY3SEC:
        MOV R7,#3
        LXDELAY:LCALL DELAY
        DJNZ R7,LXDELAY
        RET

KEYPAD:
MOV P2,#11110000B ;make bits P2.4-P2.7 input (columns)
K1: MOV P2,#11110000B ;ground all rows
MOV A,P2 ;read all col. (ensure all keys open)
ANL A,#11110000B ;masked unused bits
CJNE A,#11110000B,K1 ;check till all keys released
K2: ACALL TMSEC
MOV A,P2 ;see if any key is pressed
ANL A,#11110000B ;mask unused bits
CJNE A,#11110000B,OVER ;key pressed, await closure
SJMP K2 ;check if key pressed
OVER:
ACALL TMSEC
MOV A,P2 ;check key closure
ANL A,#11110000B ;mask unused bits
CJNE A,#11110000B,OVER1 ;key pressed, find row
SJMP K2 ;if none keep polling
OVER1:
```

```
MOV P2,#11111110B ;ground row 0
MOV A,P2 ;read all columns
ANL A,#11110000B ;mask unused bits
CJNE A,#11110000B,ROW_0 ;key row 0, find the col.
MOV P2,#11111101B ;ground row 1
MOV A,P2 ;read all columns
ANL A,#11110000B ;mask unused bits
CJNE A,#11110000B,ROW_1 ;key row 1, find the col.
 MOV P2,#11111011B ;ground row 2
 MOV A,P2 ;read all columns
ANL A,#11110000B ;mask unused bits
 CJNE A,#11110000B,ROW_2 ;key row 2, find the col.
 MOV P2,#11110111B ;ground row 3
 MOV A,P2 ;read all columns
 ANL A,#11110000B ;mask unused bits
 CJNE A,#11110000B,ROW_3 ;key row 3, find the col.
LJMP K2 ;if none, false input, repeat
ROW_0: MOV DPTR,#KCODE0 ;set DPTR=start of row 0
SJMP FIND ;find col. key belongs to
ROW_1: MOV DPTR,#KCODE1 ;set DPTR=start of row 1
SJMP FIND ;find col. key belongs to
            ROW_2: MOV DPTR,#KCODE2 ;set DPTR=start of row 2
SJMP FIND ;find col. key belongs to
ROW_3: MOV DPTR,#KCODE3 ;set DPTR=start of row 3
FIND:
SWAP A ;exchange low and high nibble
FIND1:
RRC A ;see if any CY bit low
JNC MATCH ;if zero, get the ASCII code
INC DPTR ;point to next col. address
SJMP FIND1 ;keep searching
MATCH:
CLR A ;set A=0 (match is found)
MOVC A,@A+DPTR ;get ASCII code from table
RET
; Timer0, 30 msec delay
TMSEC:
MOV TMOD,#00000001B
MOV TL0,#0CAH
MOV TH0,#27H
SETB TR0
BACK: JNB TF0,BACK
CLR TR0
CLR TF0
RET

                        BEGSTR1:
            DB 'HELLO & WELLCOME TO' ,0
```

```
        BEGSTR2:
        DB 'YOUR GPA CAL & ANALYSIS',0
        MMSTR:
        DB 'MAIN MENU: 1)CAL GPA',0
        ERRORSTR:
        DB 'ERROR-ERROR:',0
        ERRORSTR1:
        DB 'INPUT IS OUT OF BUND',0
        MMSTR2:
        DB '2)GPA ANALYSIS 3)EXIT',0
E1STR1:
DB 'ENTER THE NUMBER OF',0
E1STR2:
DB 'CURCES',0

E1STR3:
DB 'ENTER THE CURCES CREDET',0

E1STR4:
DB '4)MJ  3)MR  2)LAB',0

E1STR5:
DB 'ENTER THE GRAID:-',0
E1STR6:
DB '4)A 3)B 2)C 1)D 0)F',0
E1STR7:
DB 'YEUR GPA IS:',0
E2STR1:
DB 'ENTER YOUR GPA 0 TO 4',0
E2STR2:
DB 'YOU HAVE A SEVE GPA',0
E2STR3:
DB 'USE IT? 1)YES 2)NO',0
E2STR4:
DB 'EXCELENT',0
E2STR5:
DB 'KEEP IT UP',0
E2STR6:
DB 'V.GOOD',0
E2STR7:
DB 'GREAT BUT WORK MORE',0
E2STR8:
DB 'YOU CAN DO IT',0
E2STR9:
DB 'GOOD',0
E2STR10:
DB 'WORK HARD',0
E2STR11:
```

```
DB 'YOU CAN DO IT',0
E2STR12:
DB 'POOR',0
E2STR13:
DB 'WORK HARDER CNTACT',0
E2STR14:
DB 'STUDENT AFFAIRS ADIVSOR',0
E2STR15:
DB 'FAILED',0
E2STR16:
DB 'YOU MUST CONTACT',0
E2STR17:
DB 'STUDENT AFFAIRS ADVISOR',0
E3STR1:
DB 'THANK YOU FOR USING',0
E3STR2:
DB 'OUR APPLICATION',0

KCODE0: DB '1','2','3','A' ;Row0 ASCII codes
KCODE1: DB '4','5','6','B' ;Row1
KCODE2: DB '7','8','9','C' ;Row2
KCODE3: DB '*','0','#','D' ;Row3

        EXITPROGRAM:

        END
```

Grading policy:

The grading policy for the project is as shown in Table 3. All team members must be present at the time of project demos whether it is a weekly or final one. Missing any demo will result in a grade of "0" in the complete project. Medical excuses will only be accepted in extreme cases and must be approved by the course coordinator.

Table 3: Grading Policy

Weekly Project Demo (phase 1 to 3)	30
Full Project Demo (include phase 4)	20
Final report	20
Practical project examination	30

Report Guidelines & Grading policy:

The Report should include the following:

a) Cover page: Includes the following: Kuwait University, Department of Computer Engineering, course title and number, student names and IDs, name of the supervising engineer.

b) Table of contents: Includes report topic with their page numbers.

c) Abstract: A brief paragraph explaining the target of the project and/or what does this project do.

d) Introduction: An overall idea of the problem (problem statement, brief problem description, instructions used, Specs of the microcontroller and the kit).

e) Software Part: Detailed problem description, algorithm, flow chart.

f) Hardware Part: list all hardware being used (including the kit), with a short description on each.

g) Testing: Different test cases as being done (a step by step testing), and attaching clearly the output of each step.

h) Conclusion: Discussion of the output results and comment on the troubles faced with the project (if any), and how to improve the design (if possible).

i) Appendix: Consists of the code and the simulation output. Also a **team work evaluation sheet** should be attached (i.e. A template of that sheet is available on the website).

j) References: Any additional references used

Grade	5	3	2	**Weight**
Organization:				
Sequence of contents	Topics are in correct sequence	Partial sequence	Sequence is jumbled	1
Mechanics (Grammar/Spelling)	No spelling mistakes and grammatical errors	Few spelling mistakes and/or grammatical errors	Too many spelling mistakes and/or grammatical errors	1
			Total	**10**

Grade	5	3	1	
Individual Report contents:				
Content	Content fully explained	Content is 70% explained	Design is 50% explained	**Weight**
Cover Page				0.5
Table of contents				0.5
Abstract				1
Introduction				2
Software Part				4
Hardware Part				2
Testing				4
Conclusion				2
Appendix				1
References				1
			Total	**90**

Total points for Report = 100