The University of Jordan

Embedded Systems Lab

Fall 2022/2023

Project: Two-Player Knowledge Game

Group Size	Grading	Due Date
3-4 Students	20 Points	8/1/2023 2:00 PM

Project Details and Description

Overview

The two-player knowledge game is a competition between two players to answer simple math questions. The game has 5 questions; four math questions and one bonus question. The math questions are multiple-choice questions with three choices per question. Each math question is used one time in the game; hence each player will answer two math questions in addition to the bonus question. The bonus question is basically a random value that is generated and added to the player score. The bonus value is an integer value between [-N,N], where N is a number of your choice, but it is not greater than 10.

In this project, you will be using a PIC16F877A® microcontroller to design a simple control system that mimics the process of saving the questions, displaying the questions, managing the game and tracking the results. The general block diagram for the system is illustrated in Figure 1.

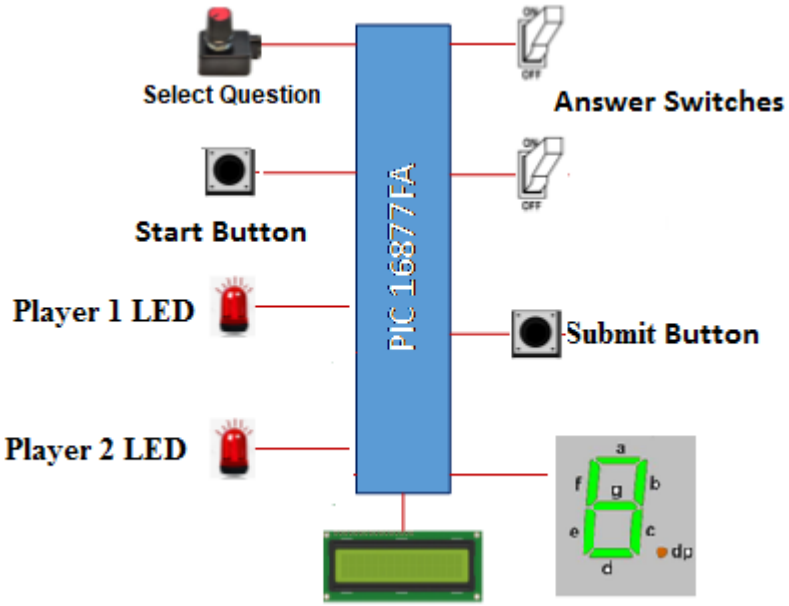

Figure 1. Block diagram for µC-based system.

Before playing the game using the system, the system should have the four questions, the three answer choices, and the correct answer saved in its memory. The text of each question may have a maximum of six characters and two characters for each of the answer choices.

The game starts when the Start Button is pressed. Afterwards, Player1's LED is turned on while Player2's is turned off to indicate that it is Player1's turn. Next, Player1 can select the question using the Select Question knob (potentiometer). The voltage read from the knob is to

be converted to digital format by the PIC in order to determine the question selected by the user. The relation between the read voltage and the number of the question is given in Table 1.

Table 1. Voltage Ranges for Different Questions

Voltage	Question to Use
$0 \le V_{Pot} \le 20\%*Vref+$	Question 0
$20\%*Vref+ < V_{Pot} \le 40\%*Vref+$	Question 1
$40\%*Vref+ < V_{Pot} \le 60\%*Vref+$	Question 2
$60\%*Vref+ < V_{Pot} \le 80\%*Vref+$	Question 3
$80\%*Vref+ < V_{Pot} \le 100\%*Vref+$	Bonus Points

Based on the converted value, the system displays the question text and its choices on the LCD as shown in Figure 2. Once the question is displayed on the LCD, the system starts a 9-second count-down counter and shows it on the 7-segment display. The user has to use the two Answer Switches to select the correct answer and to press the Submit Button within 9 seconds. When the player presses the button or the timer is 0, the answer is read and compared with the stored correct answer of that question. If the answer is correct, then Player1 gets **ten points**; otherwise he gets 0. If the player selects the bonus question, the system generates the random value and show it on the LCD as shown on Figure 3. The bonus point(s) are added to the player's score.

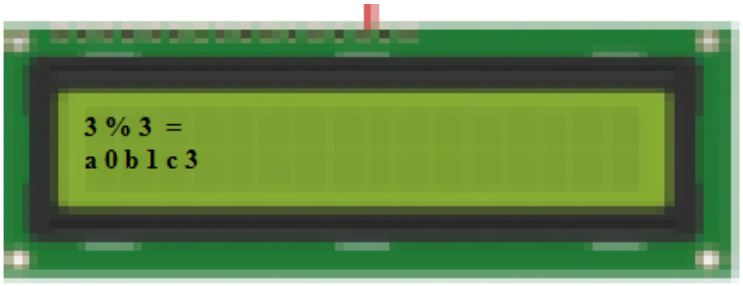

Figure 2. Displaying a question and its answer choices on LCD.

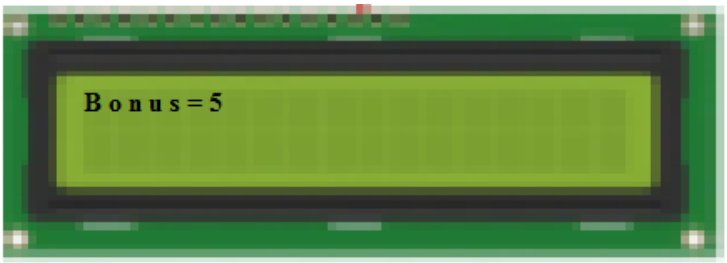

Figure 3. Displaying bonus points on LCD.

Once Player1 submits his answer or the time is over, the system will turn off Player1's LED and turn on Player2's LED to indicate that it is Player2's turn. Player2 will repeat the same steps to select and answer the question. However, the system should not allow the same math

question to be used again. In other words, if the player selects a math question that was used before, the system should show some message of the LCD and ask the player to select a different question. This is applicable on the four math questions. As for the bonus question, it can be selected by both players; however, each player can select it once.

The game continues until the question bank (4 math questions) is exhausted and both players have tried the bonus question. At this stage, the system should end the game and show the name of the winner on the LCD based on scored points as shown in Figure 4.

Figure 4. Showing winning player on LCD.

Project Programming Language and Simulation

This project has a simulation part and a hardware part. For the simulation part you are required to use Proteus® simulation software. If you do not have Proteus® simulation software, please contact us immediately. In Proteus, you can find all parts that you need. Please review the PDF file we posted earlier on how to use Proteus.

You are required to submit an **ASSEMBLY** code for this project. Any C submissions will be given a straight **zero** right away even if it is fully working. Any assembly code derived form a C compilation will be easily detected and given **a grade of zero**.

Project Groups

You must form groups of three to four students in order to work on this project. As you can see, the project can be divided into at least three major parts, each group member can take on one part:

- Saving questions, answers and other messages.
- The A/D configuration and programming to determine the stage.
- Switches, buttons, 7 segment and timer system integration
- LCD, system integration (putting everything together), and testing in Proteus.

We prefer that you form groups from within your own lab session, but if you cannot find a partner, then you can form a group with colleagues from the other sessions. Please fill in the names of your team members using the following Google form by **December 25th, 2022.**

[Google Form Link]

WE ARE NOT RESPONSIBLE IF YOU CANNOT FIND PARTNERS, OR IF YOUR PARTNER IS NOT GIVING ENOUGH EFFORT, OR DROPS THE COURSE!

Grading for Code Efficiency and Generality

It is of the most importance to write optimal codes which are minimum in size and execute quickly. We ask you to use subroutines, modular design and functional reuse whenever possible. In some cases, the use of indirect addressing (FSR and INDF) can be helpful in reducing code size and increasing speed**. It is important not to forget to use functional comments.**

During the discussion, we will ask you to show us on MPLAB or Proteus how your code works under different conditions so be prepared for the generic case.

The instructors and engineers are in no way responsible for helping you form groups or solve issues if your partner(s) drop(s) out from the course or is not working at all on the project. Choose your partner wisely. You might end up doing the project by yourself. **We will accept no excuses.**

Important Notes

- Start as early as possible on your project, though the project description sounds simple, there is inherent complexity in both hardware and software aspects, so do not underestimate the time it needs, you will have many problems along the way which you will have to resolve!
- Never think of buying a model or commissioning someone to do it for you, not only will you get a zero in the project, but also your act will be considered as a direct violation to JU laws and your actions shall be reported as cheating in the final exam!
- Code sharing between groups is NOT allowed and leads to 0 points.
- If you acquire a ***part*** of your software from a book, website, etc, then kindly reference it properly, else it will be considered as plagiarism.
- You are only allowed to base your project on PIC16877A.
- Your submitted work must be professional:
    - Software: your work should be fully documented, all inputs/outputs should be listed, and each subroutine/macro should be fully documented! Use functional comments! Refer to the last section in Experiment 2 regarding documentation.
- Divide the work such that each student is responsible for a specific task, **YET EVERY** student is required to answer for **ANY QUESTIONS** in relation to any submitted work of the project.

Report Guidelines

You should submit a hard copy of your report and it should contain the following parts:

• **Introduction.** In this section, you give a brief description of the overall project in your own technical language.

• **System Description.** Here you give a detailed description of the system design and how it was decomposed into subsystems.

• **Hardware System.** Here you must explain and elaborate on your system hardware design, its inputs and outputs. Present a clear and professional circuit design schematic. **Justify why you configured and initialized your system and modules in this way.**

• **System Testing and Results.** Present the complete methodology which you have undertaken to perform unit testing, system integration, and the final overall test. You must present the test cases which you have used to test your system for correct functionality. You must justify the choices for your test cases and whether or not they cover all possible regions of operation of your system. Provide technical discussion of the results or any abnormal operation you have witnessed during testing. **See the grading sheet for more details**

• **Conclusion.** Give a short summary of the project, your work, and the steps you have undertaken during the design process. Furthermore, you must clearly state the contribution of each student in the project. Discuss the major obstacles that you faced during the design process.

Submission

You will need to submit all the files that you use in your design including the MPLAB project, Proteus project and the report in one compressed file through MS Teams by 2:00 PM of January 8th, 2023.