

# Practical assignment of Digital Systems and Microprocessors

**Course 2021-2022**

## Practice 2B

### LSSudoku

Students	Login	Nom

Delivery	Board	Report	Grade

Date	
------	--

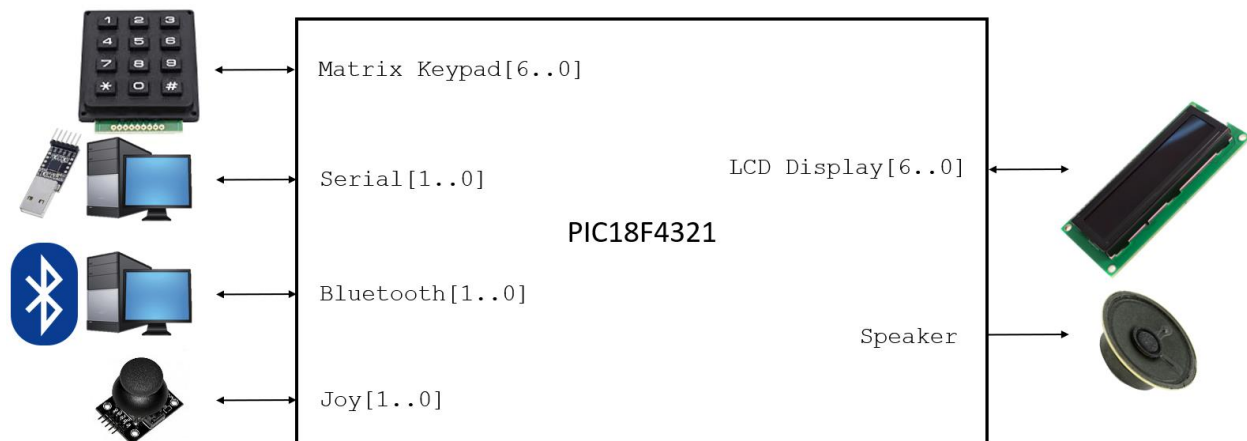
## Introduction and general considerations:

The Mathematics teachers suffer for the logical and mathematical thinking of their students, which is why, after seeing the success of the LSXylophone, together with the Role & Play club, they have asked the students of Digital Systems and Microprocessors to design an application that emulates the famous game of Sudoku. This game develops different factors, such as great ability to concentrate, cognitive skills or logical reasoning, where all these advantages are key to be able to aspire to the desired grade in the subjects they take.



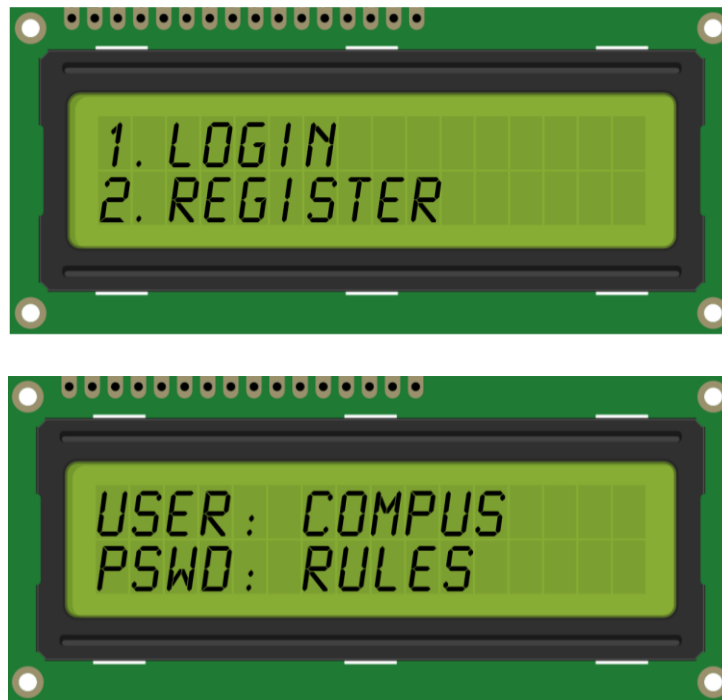
Specifically, this game will be designed as a cooperative system implemented in the PIC18F4321 microcontroller. The logic of the game (the numbers that appear and their verification) will be implemented with a program written in Java. The microcontroller will be the responsible for interacting with the user, and therefore, must communicate with the computer in order to control the movement of the player and the numbers entered, along with the evolution of the game.

As it can be seen, the microcontroller will have a digital joystick, a 3x4 matrix keypad, and a LCD display with 2 rows and 16 columns, along with two different serial connections, EUSART and Bluetooth, to different devices (Java and Terminal).



## Functioning of the system

When the microcontroller is turned on, an access menu should be displayed in order to initialize the user who wants to play. In this menu the user must be able to register or login with their pre-registered account. It will be needed to select a '1' to Login or a '2' to Register. Therefore, if the user presses the number 1, they will have to enter a user and a password of an existing user, otherwise the entry will be rejected, and redirected to the access menu. If the user presses a 2, the user will need to enter a username and a password again, but this time to save the user, notice this must not already exist. If the registration is correct, it will be redirected to the login option (so the username and the password will have to be entered again), otherwise it will be redirected to the access menu.



Once the user has logged in, the following menu will be displayed, this being the main menu of the game:

1. Play a Game
2. Modify Time
3. Show General Top 5 Scores
4. Logout
5. Show All Personal Scores
6. Show Personal Top 5 Scores
7. Show All Users
8. Show Time

This menu has some mandatory parts (1-4) and some optional parts (5-8). The mandatory functionalities, as its name implies, must be solved in any case, whereas for the optional ones, there is only the need to implement, at least, 2 tokens.

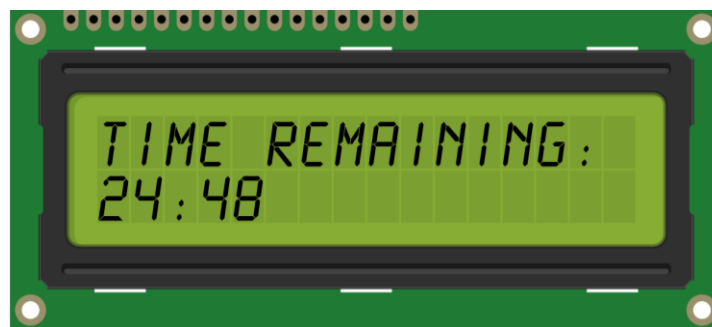
The LCD display has only two rows, thus, only two parts of the menu will be displayed at the same time. To move around it, the joystick will be used vertically, where each time it

moves down, the next option will be shown, and up, the previous one. It is important to emphasize that in order to go through two options, the joystick must be lowered twice. In other words, to go to option 3 from the beginning, the joystick must be lowered, moved to the central position, and lowered again. To select an option, the '#' button will be pressed. Once this happens, the selected option will be the one in the first row. If the first option is in the front row, if the joystick moves up in position, there should be no change in the LCD, just as if the last option is in the first row, it could not go further down.

It is important to note that if any part of the menu is not fully visible due to its length, it should be shown using a marquee.

### 1. Play a Game

This mode consists of the functioning of the game, implemented through the serial connection to the computer, done by either the EUSART of the microcontroller connected to a USB-to-TTL modem or a Bluetooth one. Firstly, the PIC will send each character of the user sequentially, from the PIC to the Java application, with a last '\0' to indicate the end of the user. Once this last character has been sent, and the computer has processed it, the Java interface will answer with a 'K', and the game will start with a limited time. If the user exceeds this time, they will return to the main menu. The limited time, in mm:ss format, will be implemented exclusively in the microcontroller as a countdown, which can be changed in the second option of the menu. Additionally, while playing, this time must be displayed on the LCD, updated in real time:



The Sudoku works in the same way as a conventional one (Appendix 1), where the blanks must be filled with the possible numbers. Both joystick and the serial device selected (either USB-to-TLL modem or Bluetooth modem) will be used to move through the Sudoku interface. The pointer will always start in an empty cell, and each time the joystick moves vertically or horizontally, through the serial channel, a signal must be sent in order to notify the Java interface. In order to detect 2 movements in the same direction, the user must go firstly through the center position of the joystick. In order to fill the sudoku with numbers, the matrix keypad will be used, once a number is pressed, this will be sent to the Java application, occupying the position where the pointer is located.

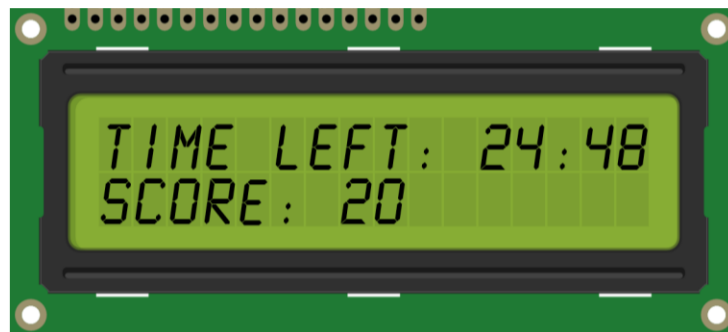
The characters to send to the Java application in order to play are the numbers (in ASCII) to fill the sudoku and the 'W', 'A', 'S', 'D' letters (North, West, South, East, respectively) to move around the interface.

A			6	2			4	8	
B					7	8		5	6
C	7		5	4		9			
D			1		3		8	6	
E		9	4			6	7	2	1
F	6	7			2	4	3	9	5
G			9	6			5		2
H				5			6		9
I	5		3	7	9		1		
	1	2	3	4	5	6	7	8	9

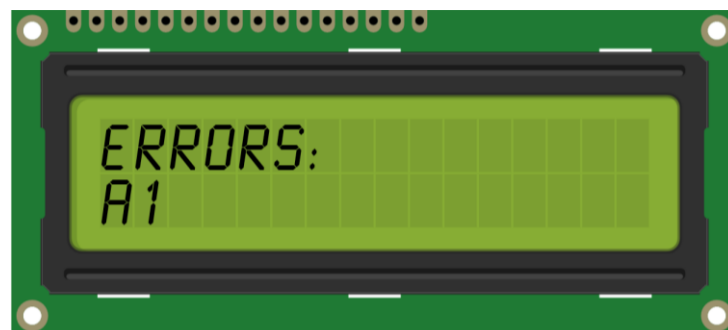
To notify the end of the Sudoku game, the '\*' button of the matrix keypad must be pressed, sending the 'F' character in order to notify the Java interface. This will reply with the score and errors. Depending on the result of the sudoku, two things can occur:

- The sudoku solution is correct: In such case the PIC will receive the score from the Java application and will wait 3 seconds to check if any error arrives, after this time, the remaining time and the score will be shown in the LCD screen until the '#' is pressed, redirecting the user to the main menu afterwards.

- The sudoku solution is incorrect: In such case the Java interface will send the score first, followed by all the positions where there was an error, with a delay of 1 second each. The PIC will show them in the LCD one by one as they are received. For instance, if the user has entered an incorrect number in the positions C4, D5, E8 and F2, and has achieved a punctuation of 15, the Java program will send a 15, a 'C', and a '4', will wait for 1 second, it will send a 'D', and a '5', will wait for 1 second again, and so on. You can consider you will not receive more errors if nothing else is received 3 seconds after the last error. In such case, the score and remaining time will be displayed until '#' key is pressed.



As it has been mentioned, the LCD should display each second the new error on the LCD, overwriting the previous one in its second row, while in the first one it must be written 'ERRORS:'. The next Figure illustrates what it must be shown in case an error is received in position 'A1':



### Music control

While the game is being played, for each minute played, a background melody should be generated for 5 seconds, using a speaker connected to the board. This melody should contain at least 5 different notes.

### Time control

While there is a game in course, the game time remaining should be sent to one of the serial channels (concretely, the one not used to communicate with the Java program). This time should be sent each second and displayed clearly. Therefore, "25:0024:5924:58" will not be accepted. This time has to be sent to a computer terminal (PuTTY) or a mobile terminal, but not to the Java program provided. If the student wishes to implement a new Java program, different from the one provided is also accepted.

To implement this new serial connection, different to the one incorporated in the PIC, a *baudrate* of 1200 bauds should be used. Of course, this output has to be different from the RC6 port (TX).

### Additional Serial Transmission (1 token)

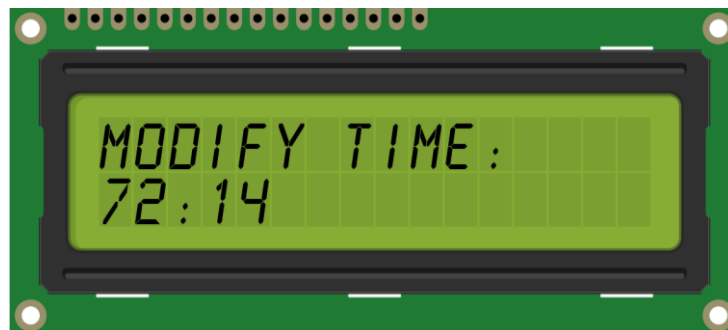
As an optional, a new serial connection (RX) can be implemented that receives characters from the additional connection, again, with a *baudrate* of 1200 bauds. This connection should be able to read the characters sent from the terminal and resend them to the Java interface. Thus, if from the terminal the PIC receives a 'W', this one will be sent to the Java program, changing the position of the sudoku pointer.



## 2. Modify Time

In this mode, the user has to determine the time limit for a Sudoku game. As it is stated in the *Play a Game* part, once the game has started, so will a countdown. If the game does not end before this time is over, it will end immediately once the countdown reaches the end, sending a 'F' through the serial channel, following the game protocol, as it has been explained in *Play a Game part*. If the user does not determine any specific time, the default one will be 30 minutes.

To define the new time limit will be by inputting in the second row of the LCD the desired time, following the mm:ss format. If the '\*' button is pressed, the PIC will redirect the program to the main menu, without storing this time. The time will be saved once the "#" is pressed after introducing the seconds value. As there is no ':' character, this will be written automatically once the minutes are written.

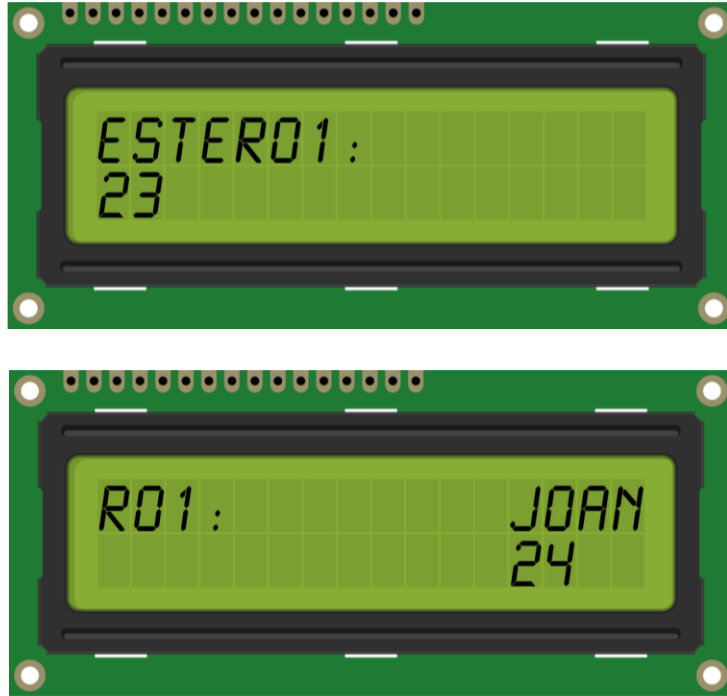


## 3. Show General Top 5 Scores

This mode will have to show the top 5 scores of any user in marquee format. Thus, the best scores achieved in the *Play a Game* option should be display in the LCD, no matter the order. In other words, if a user does not have the best punctuation but one of the best ones, they can be displayed in the first place.

In order to show the users with their respective scores, the username will always appear in the front row, accompanied by their score in the second row. It must be emphasized that if a user has two of the 5 best punctuations, he must appear twice, one per each

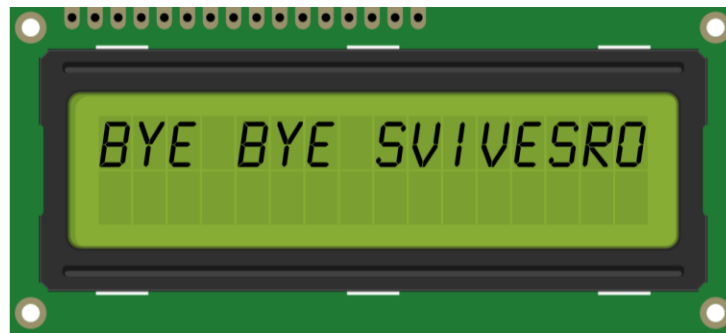
punctuation. If there are less than 5 punctuations, it will only be shown until the last one (no trash can be shown).



The marquee has to be in *ad eternum* format, in a cyclic way. Therefore, once the last punctuation has been displayed, the first one has to be shown on screen. To get back to the menu, the '#' button will be pressed.

#### 4. Logout

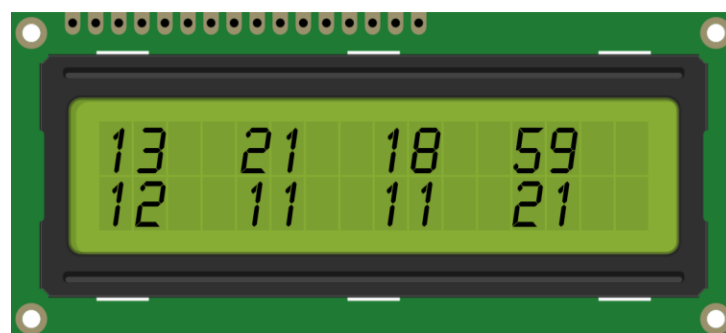
This mode will display a farewell message for 2 seconds and then it will return to the first menu so that a new user can log in or register again. The farewell message must show the username registered.

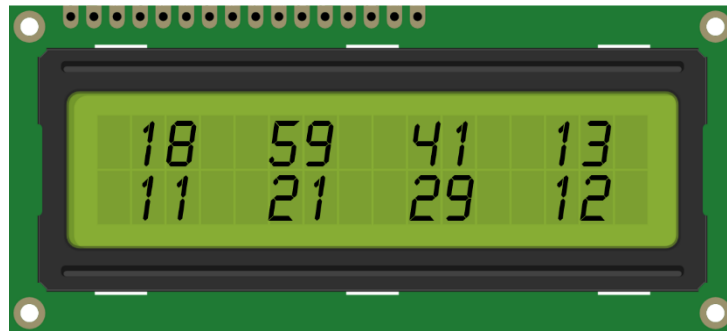


#### 5. Show All Personal Scores (2 tokens)

This mode should show the last 20 scores that the registered user has achieved. These should be displayed in a marquee format from oldest to newest. The marquee will end once the '#' button is pressed, and will return to the main menu. If the user wants to record a new score once it has already been recorded the maximum number of scores, the oldest one must be overwritten, but at the time of showing the scores, it must be in the last position (FIFO format). Scores must be shown on both the first and the second row of the LCD and must be separated by two spaces.

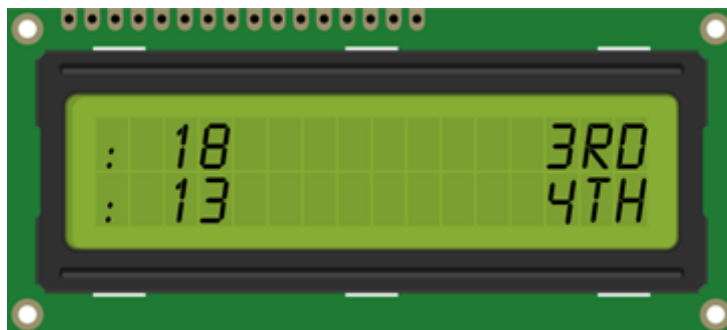
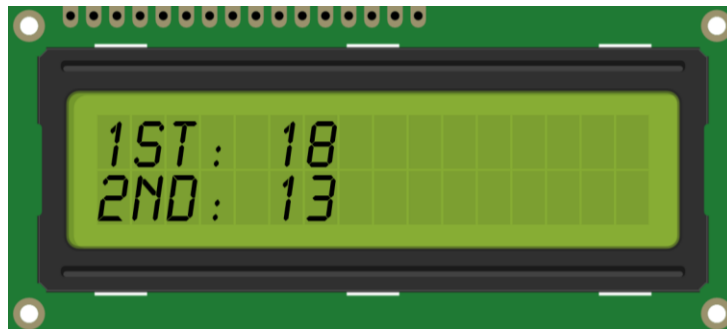
The order of the scores is vertical, meaning the second score to show will be below the first. In other words, following the number of the following example, the '13' is the first score, the '12' the second one, the '21' the third, '11' the fourth ... and so on.





#### 6. Show Personal Top 5 Scores (2 tokens)

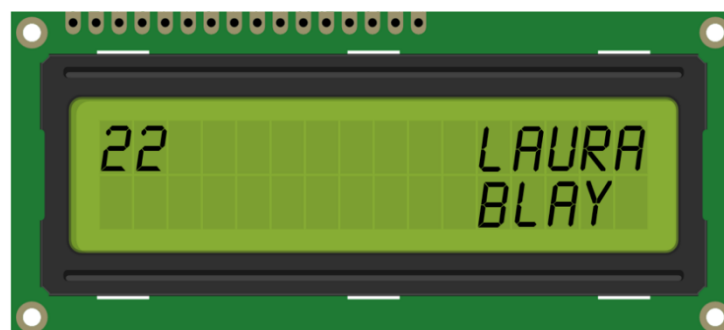
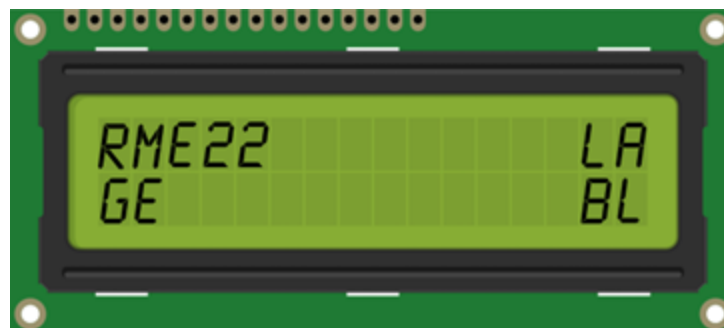
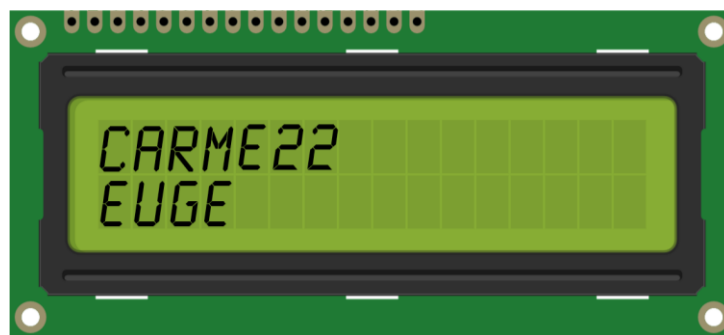
This mode should display in marquee format the top 5 scores of the registered user, in order from best to worst, as shown in the following illustration.



This marquee must be in *ad eternum* format, cyclically, where in order to return to the main menu, the '#' button of the matrix keyboard should be pressed. If there are less than 5 scores, it will only be displayed until the last one. In other words, if there are only 4 scores, "5TH:" should not be displayed.

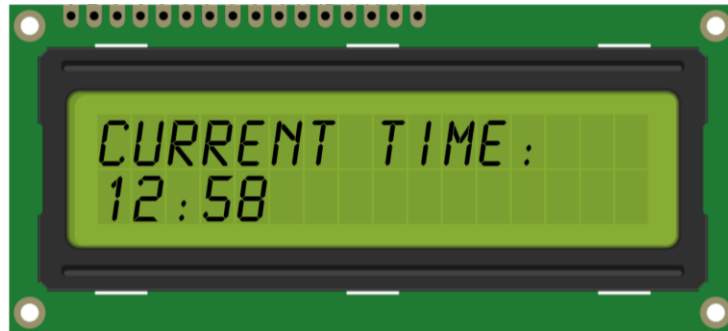
### 7. Show All Users (2 tokens)

In this mode, the LCD will display on screen all registered users, again in a marquee format cyclically *ad eternum*. Therefore, once the last user is displayed, the first user must be shown again. The logged-in user should be the first to displayed, followed by all the users in a LIFO format. Therefore, the last registered user must be displayed the first one, after the logged-in one, and the user registered first should be shown last, except if that user is the one logged-in, who will be the first user to be shown.



#### 8. Show Time (1 token)

From the moment the PIC is turned on, it should count the time that it has been connected. Therefore, in this mode it will be shown on the LCD screen the minutes and seconds that it has been on. This time should change in real time, so every second it passes, the LCD should be updated. The format must be mm:ss, as shown in the following illustration.



## User control

There will be a maximum of 8 users (name and password with an 8-character maximum each), which will need to be stored in the EEPROM of the microcontroller to avoid them being deleted when the PIC is disconnected from the voltage source. If there are 8 users registered and we want to register a new one, the oldest user will be overwritten by the new one, in a *First In, First Out* form. It won't be possible to register usernames with the same name. Additionally, the best 5 scores will also be registered in the EEPROM of the PIC, relating them to their corresponding user. On the other hand, the information needed of the optional functionalities regarding the punctuations will be stored in the RAM.

It is important to consider that the EEPROM and RAM microcontroller space is limited, and therefore it is important to not exceed this space.

### User control registry



In order to login or register in the PIC, a matrix keyboard in SMS format will be needed. This format, already known by the Digital Systems and Microprocessor students, consists of, if for example a user wants to write a 'C', it will need to press key '2' three times. If this key is not pressed in less than a second or another key is pressed, character 'C' will be considered as the definitive. If on the other hand, the same key is pressed several times with less than a second between two consecutive pressings, the associated letters to the key will be increased cyclically ("A", "B", "C", "2", "A", "B", ...). Remember also that the 0 key rotates from '0' to ' ' (blank space) constantly. When a key is pressed, the system will need to show the new value of the key in the LCD. Is important to remember the SMS functioning with the following example:

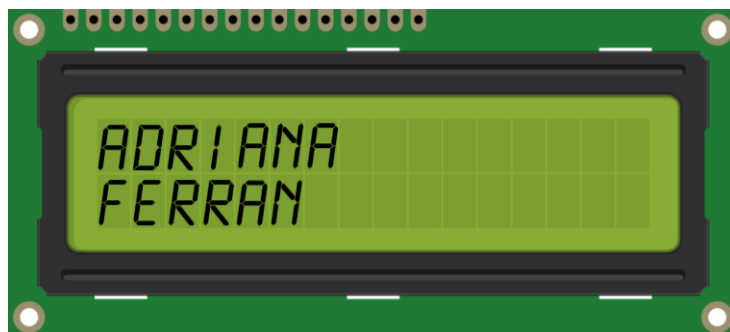
*I press 2, 'A' is shown. I press 2, 'B' is shown. I press 3, BD is shown. I press 3, BE is shown.*

The registry control or login control will be determined by the SMS, and to notify the end of the username or password, the '#' key will be pressed. Moreover, if the username or password is equal to the maximum length (8 characters), once the SMS waiting second happens, it will be detected as '#' has been pressed, and the next action will happen (either to write the password or to redirect to the main menu, accordingly to the logic explained).

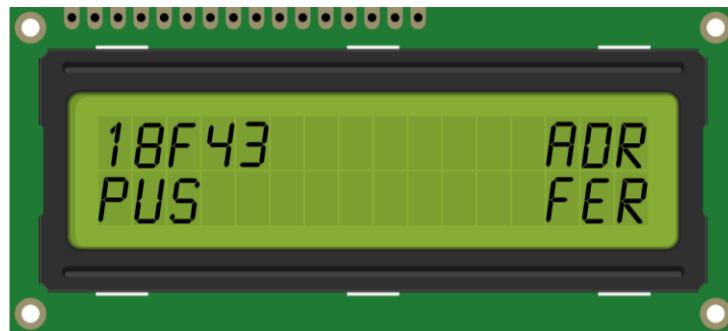
## Marquee control

The marquee will need to be controlled each second. Therefore, for each second, the columns will need to rotate from right to left. The marquee can be omitted in case that is not necessary due to not being more information to display apart from the already being shown on the screen. For example, in the option *Show All Users*, if there are only two registered users, there can be no marquee.

The marquee, as it has been mentioned, will not end until the '#' button is pressed, which will need to return to the main menu automatically. In the next picture, there's a representation of the *Show All Users* marquee, where it is interpreted that each time the last users have been shown, the first ones appear once again from the right.







At the same time, if there's any option in the menu that occupies more than 16 characters, there should be a marquee also, as shown in the following picture:



## General Considerations

1. The maximum number of Users stored will be 8.
2. The name of users and passwords won't have more than 8 characters (not counting the '\0').
3. The menu's text must be the same as that is shown in the statement.
4. A minimum of 2 tokens must be implemented.
5. The additional serial channel can't have the same exit or entry port as the principal serial channel. They must be completely separate systems one from the other.
6. All the logic specific to the statement has to be solved exclusively in microcontroller space.
7. This practice is designed to make you think about the use of memory of the microcontroller. You need to understand that it is a scarce and expensive resource; put special emphasis on your software and data structure design.
8. The microcontroller must be exclusively programmed in C by using the cooperative systems methodology.
9. Obviously, apart from having the usual memory deliverables (cover, index, statement summary, system design, electrical schematic, conclusions and observed problems), the memory will also have to contain the ADTs diagram detailed (including peripherals and the most important variables), the diagram of all the practice motors, along with an explanation from both the ADTs diagram and of each motor. The software dictionary will be delivered as an extra file.
10. To deliver this phase, a .zip file will be delivered, containing a memory that complies with the practice normative, a video or a link to a video of the working practice and a folder with the whole MPLAB project and the results of its compilation (.c and .h files). It is also important to add the file *dictionary.c* as shown in the theoretical classes of the subject. The name of the .zip file must have the following format:

YYYY.MM.DD-SDM-2122-P2-FB-login1-login2.zip

For example: 2022.05.22-SDM-2122-P2-FB-last.compupractice.happyface.zip

## Appendix I: Instructions to play the Sudoku

A Sudoku is played in a 9 x 9 space grid. Inside the files and columns ther are 9 squares, formed by a 3 x 3 squares. Each row, column and square (of 9 spaces each), need to complete with the numbers from 1 to 9, without repeating any number in the row, column or square.

As is possible to see in the picture below, each Sudoku square comes with some spaces already filled in. The more filled spaces, the easier the game becomes.

				3		4	6	
		5		4		2		8
				6	2	3	9	5
2								4
					7			
3	4			5				
	5	8		2	4	6		
4	2		3		6	7		
					5	8	4	