LEBANESE AMERICAN UNIVERSITY
School of Engineering
Department of Electrical and Computer Engineering

Microprocessors Lab
Fall 2018
Nicolas Jamal

Final Project: “Multifactor Authentication”

Background Info

Multifactor Authentication, aka MFA, is a security system that requires multiple authentication methods that belong to different categories in order to verify the user’s identity and allow him/her to login.

Project Description

In your project, you are required to write a Multifactor Authentication program that will use multiple pieces that you already coded in the labs. Basically the user should use a combination of push buttons, rotation angles captured by a potentiometer, and ranges captured by an ultrasonic sensor in order for him/her to set a password, and then login.

Material Needed

- MCO Board
- 9 male-male connectors
- Ultrasonic Sensor

Deliverables

You are required to deliver a CD containing the following:

- The full COMMENTED code in .TXT
- A detailed report explaining your work for every task (Use diagrams and flowcharts as much as possible while limiting the use of long texts)
- A PPT presentation that summarizes both your work and results

Notes:

- You are required to hand in a printed copy of your report, and present your project live.
- The presentation date of the project will be the last day of the semester. Both the date and time for each group will be assigned on Blackboard.
- The grade on the project will be distributed based on the difficulty of every task.

Project Overview

The password combination is made out of 8 characters. Below, you can find an example:

TABLE 1: PASSWORD EXAMPLE

P1	P4	P1	P2	C	8	D	C

- The first 4 characters of our password will be inputted using the push buttons
- The following 2 characters will be inputted using the potentiometer
- The last 2 characters will be inputted using the ultrasonic sensor

In order to get a full grade on this project, the user should be able to set the password in this order (Buttons → Rotation → Range), and then login using the same order. If the combination is correct, you should show a success message, otherwise notify the user and ask him to try again.

Bonus

To make the password stronger, you should allow the user to choose the order of authentication methods. Therefore, when the MCU is booted up, you will ask the user to choose one of the authentication methods (Button, Rotation, and Range) to setup the password. Once the user finalizes setting up the first authentication method (let's say the user started with Range), the user will be able to select one of the 2 remaining methods (Button or Rotation). You can find below an example of a hard password.

TABLE 2: STRONG PASSWORD

D	4	A	E	P4	P1	P2	P3

Once the user finishes setting up the password, the login process should start by asking the user to choose an authentication method. Let's say the user inputs the following password:

TABLE 3: WRONG PASSWORD

P4	P1	P2	P3	D	4	A	E

This represents a wrong password (Compare Table 3 to Table 2). Even though the individual combination of each authentication method is correct, the order of authentication methods was not respected.   The same order of authentication methods must be followed to login. By doing this trick, you will be adding an additional layer of security to your password.

Task 1 (25%)

In this task, you will be setting up the Push Buttons.

Since the push buttons available on the microcontroller board are not enough (SW1 & SW2), you will have to use the push buttons located on the development board. For this project, you will be using PB1 through PB6:

- PB1 → PB4: will be used to input a password combination
- PB5: will be used as ENTER
- PB6: will be used as SELECT/DELETE

Since the push buttons need to be interfaced with the microcontroller board, you will need to connect male to male connectors from J10 (connector located right above the push buttons) to J1 (connector next to microcontroller; check document provided with the project to check pins of J1).

After connecting the push buttons to a specific port of the MCU:

- Initialize that port (data direction register, interrupt, etc.)
- Write the following program:
    - Show on the first line "PB Pass"
    - Initially, when a user presses PB5 or PB6 nothing will happen
    - When a user presses PB1, PB2, PB3 or PB4 the LCD should show P1, P2, P3 or P4 respectively
    - PB5 (Select/Delete) allows the user to delete the last value entered
    - PB6 will have no effect, until 4 values have been entered (e.g. P2P2P3P1). Once 4 values are entered and PB6 is pressed, you should save the combination in memory.
    - Choose any representation to store these values in memory (encode using binary, letters, etc.). But keep in mind that at the end you will need to compare password (Example: you can represent P2P2P3P1 as 2231 in memory)

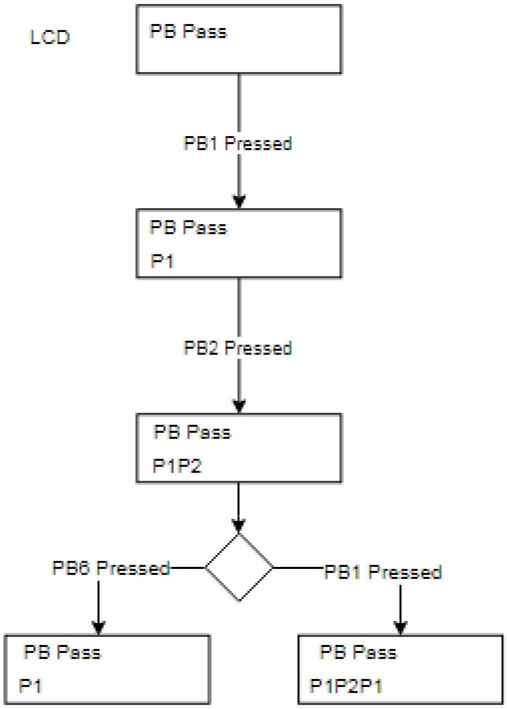

FIGURE 1: PUSH BUTTON EXAMPLE

Task 2 (25%)

In lab 7, you were able to use the potentiometer in order to store 2 values in memory. The range of the potentiometer is 0 → 255. In this task, you will have to translate these values to a single hex value:

- 0 → 15 will represent 0
- 16 →32 will represent 1
- ….
- 239 → 255 will represent F

In this task, you will need to write the following program:

- First line of LCD should display: "POT Pass"
- 2nd line of LCD should display the value of the potentiometer in hex
- Once Enter is pressed (PB5), the current value of the potentiometer should be saved in memory. A space should be introduced. Then the value of the potentiometer should be displayed again.
- If the potentiometer is turned, the second value should be changed.
- If PB6 is pressed, the current value will be deleted and the user will be able to change the first value
- Once the user has pressed Enter twice (saved 2 values), Enter will have no effect (for now). Pressing delete (PB6) will allow the user to change the 2nd value.

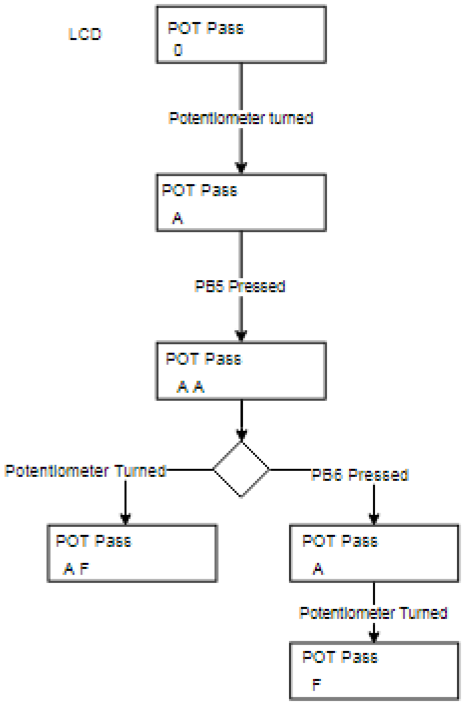

FIGURE 2: POTENTIOMETER EXAMPLE

Task 3 (25%)

In this task, you are required to use the Ultrasonic sensor in order to record the last piece of the password.

- Start by looking up the datasheet of the ultrasonic sensor and getting more information about its pins
- You need to connect the following using male to male connectors for your ultrasonic sensor to work:
  - VCC
  - Ground
  - Analog pin to one of J1's analog inputs
- Once you've done that, all you have to do is use a similar subroutine from the previous task (Potentiometer) in order to read the value of the Ultrasonic sensor. Make sure you read from the correct analog input.

- Once you've done all of that and tested out that your potentiometer works (when your hand is away the value should be large or max, when your hand is really close to the sensor the value should be small or min). Now that you've known the max and min, you will need to translate that range into A, B, C, .. E (A being the closest distance and E the largest). Make sure the ranges between one letter and another is similar.
- Finally, the process should look like that:
  - 1st LCD line should display "US Pass"
  - 2nd LCD line should display the translated value of the potentiometer.
  - When you move your hand in front of the ultrasonic sensor, the value should be changing respectively on the LCD.
  - If PB6 is pressed, the last value should be deleted. If PB5 is pressed, the first value should remain on the LCD, and a second value should be introduced.
  - When PB5 is pressed again both of these values will be stored in memory to continue the password.
  - If PB6 is pressed here, the user will be able to change the second value.

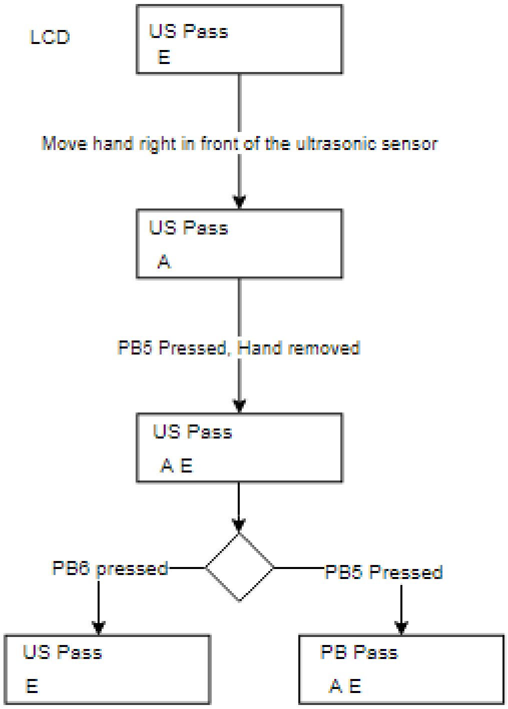
LCD
US Pass
E
Move hand right in front of the ultrasonic sensor
US Pass
A
PB5 Pressed, Hand removed
US Pass
A E
PB6 pressed
PB5 Pressed
US Pass
E
PB Pass
A E

FIGURE 3: ULTRASONIC SENSOR EXAMPLE

Task 4 (25%)

In this final task, you will need to put together your previous work. The user should be able to see first on the screen “Set Pass”. After pressing PB5 or Enter, the user will be able to start setting the password using the push buttons, potentiometer, and then ultrasonic sensor.

Once the password is set, you should show on the screen “Login”. Now, when the user presses PB5 or Enter, the user will be able to go through the last steps in order to login. When the password is inputted, you will need to compare the password that the user just inputted, and the one that the user inputted first. If they are the same, the LCD should show a “Success” message. Now the user can reset the password by pressing PB5 or enter (start again the program). If the password is wrong, the LCD should show an “Error” message. The user will be able to login again in this case by pressing PB5.

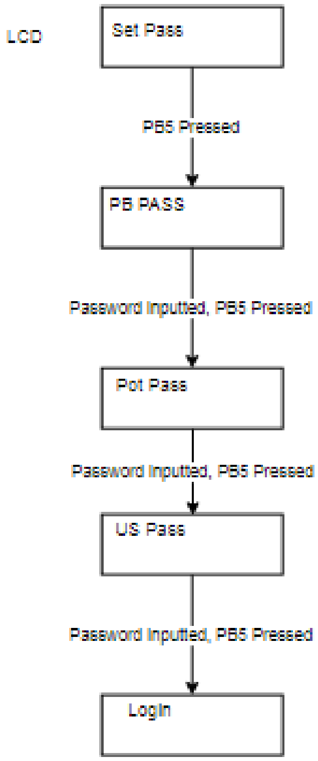
LCD
Set Pass
PB5 Pressed
PB PASS
Password Inputted, PB5 Pressed
Pot Pass
Password Inputted, PB5 Pressed
US Pass
Password Inputted, PB5 Pressed
Login

Figure 5: Program Flowchart (Part 1)

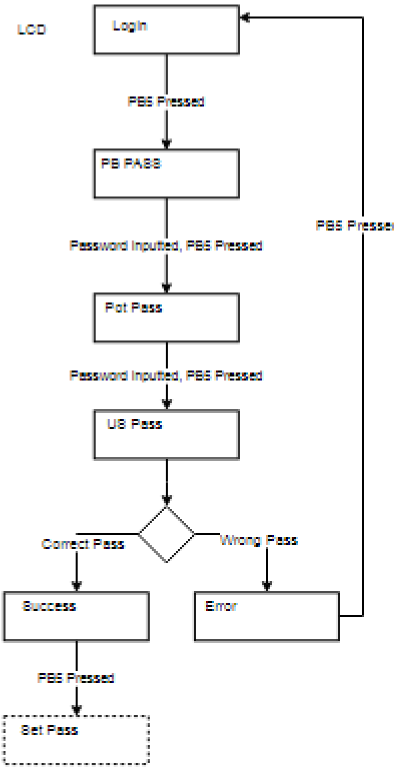

Figure 4: Program Flowchart (Part 2)

Final Notes

- Use diagrams that contain subroutine names, memory locations and registers to explain your logic.
- Take assumptions whenever it is possible and include them in your report.
- Always take into consideration the user experience (UX) while developing your application: show messages on your LCD to guide the user throughout the password setting and login processes.
- You don’t have to re-explain what you have already explained in the labs (LCD initialization, ATD initialization). However, all the new connections and modules should be explained.
- The board and other lab equipment should be returned at the end of your presentations.