
System locked
Enter PWD:456809_


System locked
Enter PWD:_

1. Initialize count with value 10.
2. Assign start address of EEPROM to specific pointer.
3. Make FSR point to start address of array, which will contain the value of EEPROM.
4. Increment FSR.
5. Read value of one location from EEPROM and write it in the array.
6. Increment the pointer that point to EEPROM.
7. Decrement count. If equal to zero, it mean the EEPROM password has been read completely then return. Otherwise, go to step 4.

The other important function is *WriteEEPROM* function, which writes the new password to EEPROM when the user wants to change the password. Writing to the EEPROM is critical operation which should not be performed accidentally [3], because the results would be permanent, unless and until we re-write the same location of EEPROM. For this reason, two values *AAh* and *55h* must be written to EECON2 register one after another to be sure the program want to really write to EEPROM.

Following are the steps performed by the function *WriteEEPROM*:

1. Initialize count with value 10.
2. Assign start address of EEPROM to specific pointer.
3. Make FSR point to start address of array which contain the value that must be store in EEPROM
4. Increment FSR.
5. Read value of one location from array to temporary variable.
6. Write the value 0xAA to EECON2.
7. Write the value 0x55 to EECON2.
8. Write the temporary variable to EEPROM.
9. Increment the pointer that point to EEPROM.
10. Decrement count if equal zero that meaning all password has finished write in all location of EEPROM and return, otherwise will go to step *4*

The system has the buzzer to represent an alarm when the user enters a password three times the buzzer will give a high sound. This pin could be configured with rely to switch a high power alarm when a security breach occurs.

The proposed system is very simple and has very few components involved, because all the functions are implemented in the firmware that is installed in the PIC microcontroller.

Figure 4, shows us the complete hardware components used in the system on the breadboard.

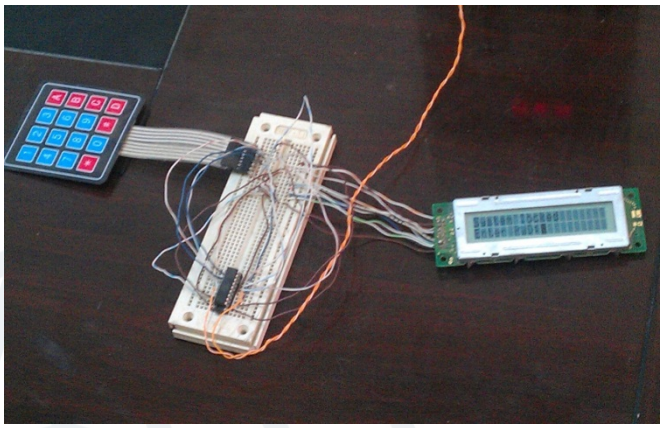

Figure 4: Complete hardware components of the system

It is evident from the figure, that there are only a few electronic components involved in the working of this system. This is really a great benefit of using an embedded system, which makes the hardware architecture very simple, but the complexity ultimately moves to the firmware itself according to the concepts of software and hardware boundary.

V. CONCLUSION

Programming the LCD with 4 bits interface mode is very efficient and the delay of sending a single byte in two steps was not noticeable, irrespective of reserving the 4 bins port in a limited ports microcontroller.

The other issue is using PULL UP on PORTB, which proved beneficial for the system, because it puts the system in the sleep mode within the infinite loop while waiting for a key press. So when there is no interaction with the system, it saves batteries power and improves the functional life of the system. Hence there is no wastage of CPU cycles for unimportant functions.

After implementation of the system using real hardware components, there was one technical difficulty I encountered, and want to emphasize in this paper, was the implementation of keypad. As many researches state that, Columns has to be set as input to be read, to check their status to know which key has been pressed, and the rows as output, which can be set by firmware to 1 or 0 [5,6]. This approach did not work here because, if 4 pins are set to input simultaneously their state would be influenced when one of them is changed causing inaccurate key scan for reading the pressed key. The solution found was to set every port pin of column to input then output every time we check the column status. In this case, only one pin would be set as input at a time and the others are set to output, which would not be influenced by changing state of that pin. This gives us an accurate and smooth keypad operation.

The system is intact and sound, and with some improvements and operational testing it can be considered as a successful product and can be shipped to the market.

VI. REFERENCES

[1] Milan, Verle, "PIC Microcontrollers", mikroElektronika; 1st edition, 2008.

[2] Nebojsa, Matic, "PIC Microcontrollers on line", MikroElektronika, 2000.

[3] Microchip, "PIC16F628x Data Sheet" Microchip.

[4] Hitachi, "LCD HD44780 datasheet", Hitachi.

[5] Peter, JAKAB, "Electronic Combination Lock based on PIC", [Online] Available: http://jap.hu/electronic/combination_lock.html

[6] DAT, "Simple Combination lock Project with keypad and LCD", [Online] Available: http://www.8051projects.net/download-d195-simple-combination-lock-project-keypad-lcd.htm

How to cite

Muhanad Hayder Mohammed, "Secure Electronic Lock using PIC 16F628A Microcontroller". *International Journal of Research in Computer Science*, 2 (5): pp. 43-47, September 2012. doi:10.7815/ijorcs.25.2012.047