Don Bosco Institute of Technology
Department of Electronics and Telecommunication

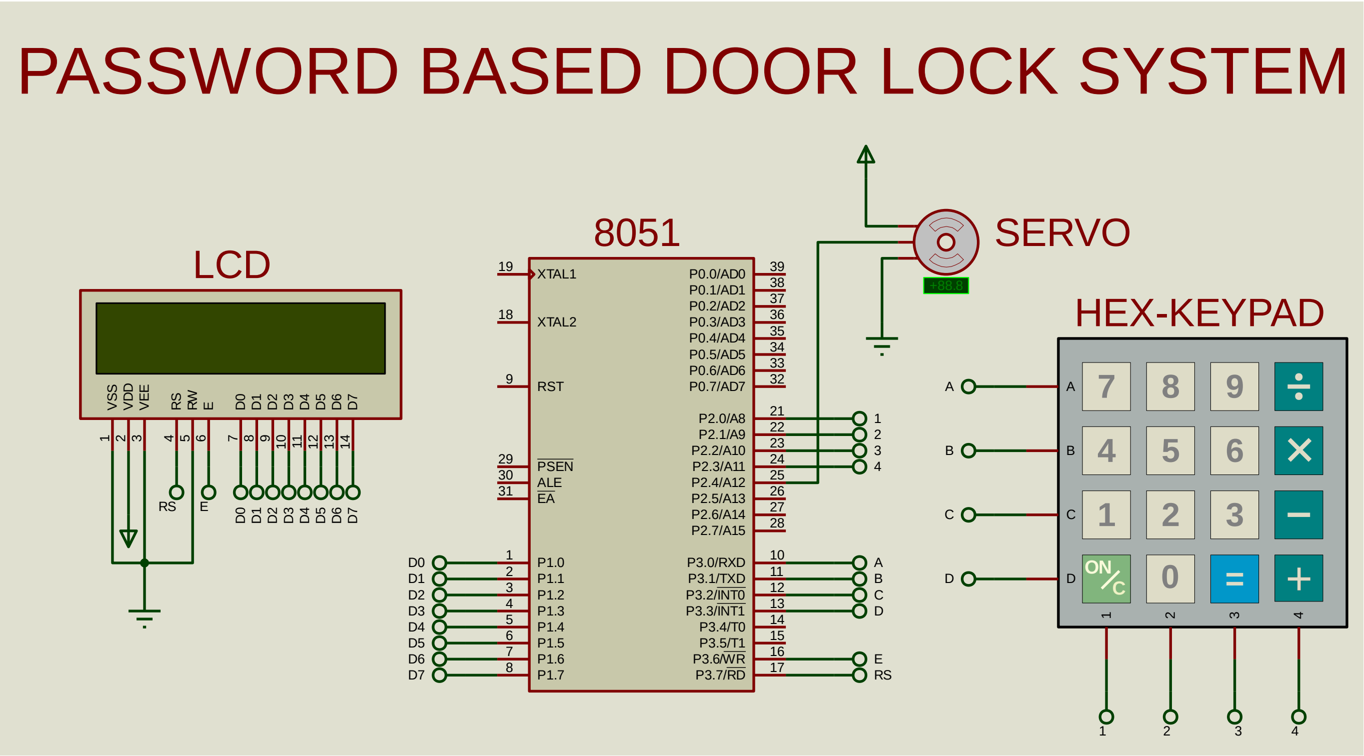
Mini Project Report on

Password Based Door Lock System using Assembly Language (Only simulation)

Subject: Microcontrollers and Applications

Group Members:

1. Hrishikesh Borkar (08)
2. Nikhilesh Chakravarty (12)
3. Saurabh Chaubey (13)
4. Hemantkumar Suman (68)



Abstract: This Project presents the simulation of Password based Door Lock system using 8051 Microcontroller. The user will only be able to enter 4 digits. The password is set in the program. Hence the user should know the password to unlock the door. For simulation purposes, the door has been replaced with a servo motor. When the system will be switched on, the LCD will display "**ENTER PASSWORD**". Password entered by the user will be displayed on LCD will be visible in asterisks "*****". The user will enter the password in the Hex-Keypad. If the user enters the correct password, the servo motor will rotate and "**CORRECT PASSWORD**" will be displayed on LCD. If user enters incorrect password, the servo motor will not rotate and "**WRONG PASSWORD**" will be displayed on LCD. The whole operation is done by a Intel 8051 microcontroller.

Working: The heart of the system is the 8051 microcontroller. When the user enters the password or when the user presses the button, the digit mapped to that switch is stored in to some location, which will be displayed on LCD with "*" instead of the original digit for security reasons. This iteration will go on till 4 digits are entered by the user. The program has a 4 digit stored password. The digit stored in the 1st memory location will be compared with 1st character of the already defined password. If both the digits are equal, it will continue the checking for the next three digits. If any one of the digits is found unequal, it will immediately display "WRONG PASSWORD". If all the four digits are found to be equal, then the servo motor will rotate by 90 degrees. i.e the door will be opened. Also "CORRECT PASSWORD" will be displayed on LCD.

Software Description:

The whole code is divided in to three parts:

1. Interfacing of Hex-Keypad and LCD:

   Here, the hex-keypad is the input component and LCD is the output component. The LCD is connected to the port 1 of the 8051 i.e the port 1 is the output. To make it as output port, we mov 00h value to port 1 by MOV P1,#00h. "ENTER PASSWORD" string is displayed on the LCD with the help of dptr and correct call and jump instructions. The hex keypad is the input from which the user will enter the password. The mapping of each instruction is done by JB and JNB instruction. For eg: JB P2.0,s1 //switch is +ve edge trigered

```
JNB P2.0,$  // wait for the switch to be released.
 MOV A,#'7'
```

In the above example when the switch is pressed ,the Value **7** will be stored in and displayed on the LCD with *. The code for logic is written in data_write subroutine which is used to display characters on LCD. The pressed digit is selected by making that particular row low and rest of the rows high. This is done by looping to make the code efficient and also reduces the number of lines in the program.

2. Storing the password entered by the user in to some memory location:
   The entered password is stored in a location form 30h to 33h. Since the password is of 4 digits, the memory is of 4 addresses. Each digit or byte in stored in single location. This is done my pointers. The pointer is defined by '@'. 8051 has only 2 memory pointers that is R0 and R1. The R0 is first initialized to 30h. After that it acts like a pointer by making it @R0. Now the A value is stored in the location pointed by R0. After storing the 1st byte, it is then incremented to the next memory location. This same operation is performed four times after every button is pressed. The four-time operation is performed with the help of DJNZ operation. i.e DJNZ R4,loop1. The R4 is initialized with immediate data #4. DJNZ operation jumps to label 'loop1' till R4 becomes zero. This ensures that only 4 digits are being entered.

3. Checking the password whether the entered password is right or wrong:
   After the password is stored in the memory location 30h to 33h, it will compare the predefined password by a logical operator XRL(Logical EX-OR). The password is defined with the help of array data_2 db: "8428",00h. The string is to be written in " " when defining in array. The 00h is the null character which tells that the array has ended. With the help of MOVC A,@A+dptr, each character of the data_2 array is stored temporarily in accumulator A to compare the entered byte/digit. The comparing of byte is done XRL operator XRL A,@R1 does the bit wise exor operation of @R1 i.e data in the location pointed by R1 and A, the result will be stored in A. EXOR tells us that the result is zero if both the value are equal. For correct password, the result should be zero for four consecutive times, then only the servo will rotate. After that 'JNZ loop2' instruction is used. JNZ will check whether the A contains a non-zero value or not. If A does not contain a non zero value then, it will jump to loop2 which will display "WRONG PASSWORD" on LCD.

Hardware Description:

There are 4 components used to make this system work.

1. 8051 Microcontroller (Heart of the system.)
   8051 is a CISC type 8 bit microcontroller by Intel. It is a very basic microcontroller with a lot of features. It has 4kb internal ROM and additional 60KB ROM can be connected externally bit . It has various features like timers, interrupts, serial communication, etc. It has eight 8 bit registers and two 16 bit registers which acts like a counter. They are R0 to R7 and PC(Program counter), dptr respectively. R0 to R7 and banked register. It can be programmed in C and Assembly Language. It is available in 40 pin DIP package.

2. 16x2 Alphanumeric LCD (Liquid Crystal Display):
   Alphanumeric LCD display means it can display both alphabets and numbers. 16x2 LCD has 16 columns and 2 rows. It has total 14 pins. There are 8 data pins (D0 to D7)

1 ground(Vss), 1 power pin(Vdd), 1 Vee pin (for contrast), 1 RS pin(Register select), 1 RW pin(Read/write) and 1 enable pin. To get the maximum contrast, Vee pin is connected to ground. RW pin connected to ground because we want to write on LCD. D0 to D7 are connected to Port 1 of 8051.

3. Hex-Keypad:

Hex-Keypad or Hexadecimal Keypad is 16 switch keypad where each key corresponds to a particular row and column. The hex-keypad used in this system is 4x4 i.e it has 4 rows and 4 columns. Hence 16 switches are present. The hex-keypad used in simulation have active high switches. Hence JNB instructions to activate the switch.

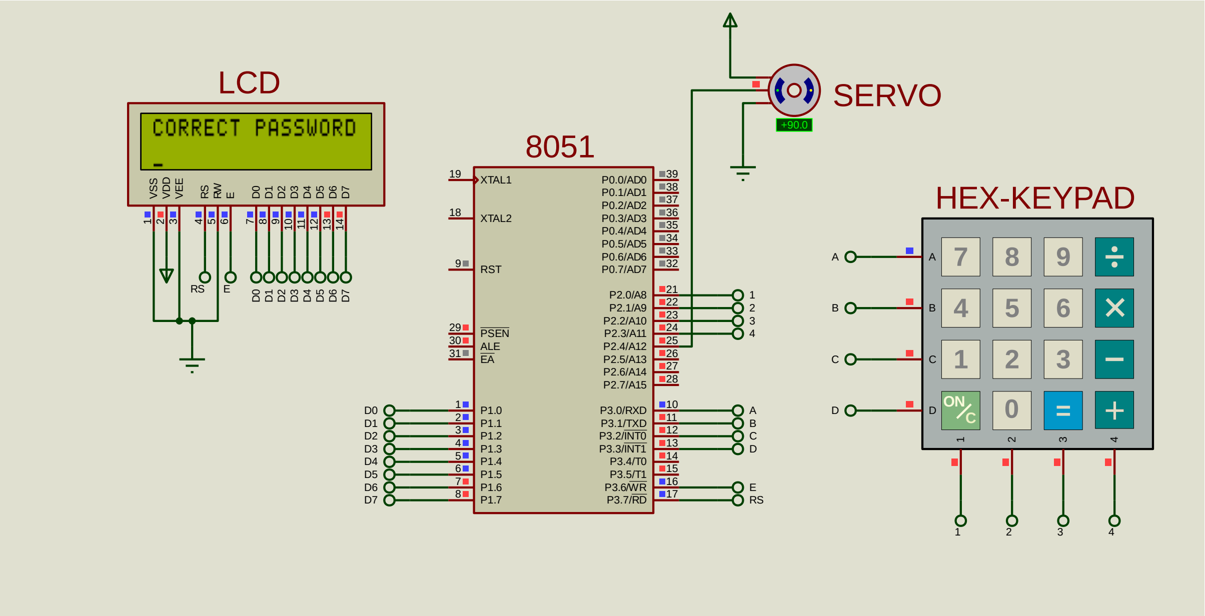
4. Servo Motor:

Servo Motor is a motor which will rotate till a particular angle. In this simulation, the servo motor rotates with angle 90 degree. It is done by making it hight for 1 sec and then again making it low.

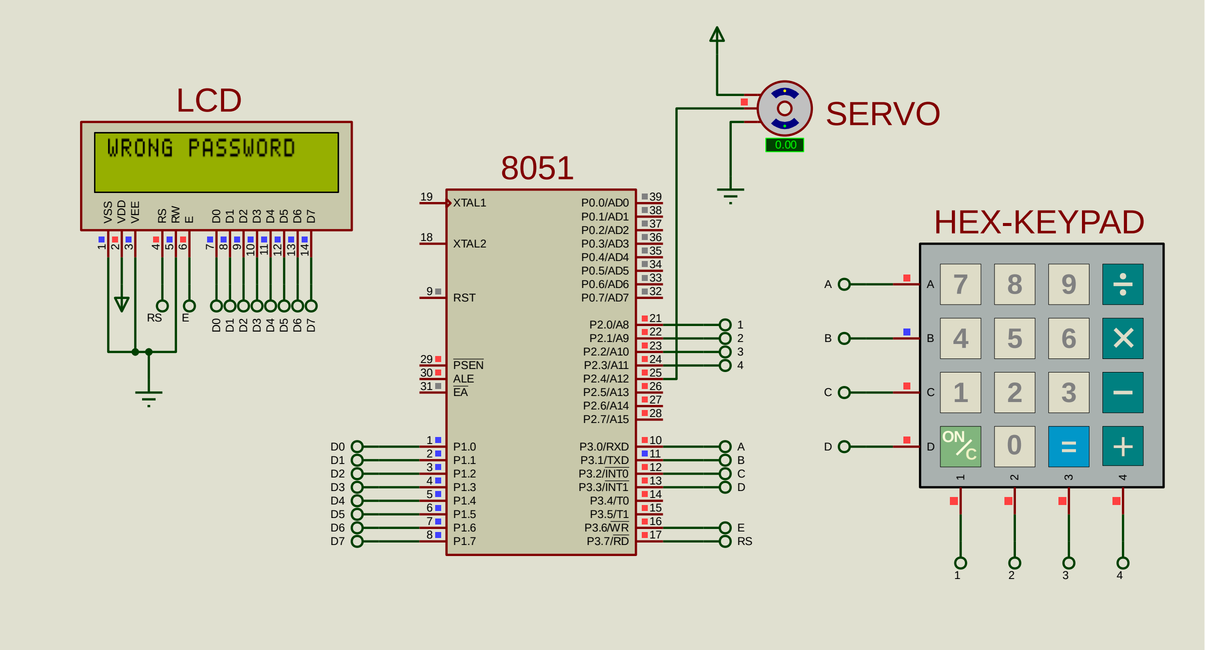
Software used:

1. Proteus 7.7 circuit simulator
2. Keil uvison5 IDE

Screenshots of the output



LCD
CORRECT PASSWORD
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
8051
XTAL1
XTAL2
RST
PSEN
ALE
EA
P0.0/AD0 P0.1/AD1 P0.2/AD2 P0.3/AD3 P0.4/AD4 P0.5/AD5 P0.6/AD6 P0.7/AD7
P2.0/A8 P2.1/A9 P2.2/A10 P2.3/A11 P2.4/A12 P2.5/A13 P2.6/A14 P2.7/A15
P1.0 P1.1 P1.2 P1.3 P1.4 P1.5 P1.6 P1.7
P3.0/RXD P3.1/TXD P3.2/INT0 P3.3/INT1 P3.4/T0 P3.5/T1 P3.6/WR P3.7/RD
SERVO
+90.0
HEX-KEYPAD
A B C D
7 8 9 ÷
4 5 6 ×
1 2 3 −
ON/C 0 = +
1 2 3 4
E RS


LCD
WRONG PASSWORD
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
8051
XTAL1
XTAL2
RST
PSEN
ALE
EA
P0.0/AD0 P0.1/AD1 P0.2/AD2 P0.3/AD3 P0.4/AD4 P0.5/AD5 P0.6/AD6 P0.7/AD7
P2.0/A8 P2.1/A9 P2.2/A10 P2.3/A11 P2.4/A12 P2.5/A13 P2.6/A14 P2.7/A15
P1.0 P1.1 P1.2 P1.3 P1.4 P1.5 P1.6 P1.7
P3.0/RXD P3.1/TXD P3.2/INT0 P3.3/INT1 P3.4/T0 P3.5/T1 P3.6/WR P3.7/RD
SERVO
0.00
HEX-KEYPAD
7 8 9 ÷
4 5 6 ×
1 2 3 −
ON/C 0 = +
A B C D
1 2 3 4
E RS

References

1. https://github.com/ibrahimmkhalid/8051-door-locking-system
2. https://github.com/OsamaMazhar/Electronic-Keypad-Lock
3. http://www.circuitstoday.com/digital-door-lock-password-based-security-8051

Other References:

1. MCA Lab Notebook
2. MCA Classroom Notebook
3. M.A Mazidi’s “The 8051 Microcontroller and Embedded Systems Using Assembly and C”