Smart door lock security system

Author : rnshalinda [github]

Created : 11/07/2022

The main goal of this sample project is to provide an affordable smart and secured security system for hotel rooms and apartments. This system must be cost effective. Therefore suggested using PIN protected keypad unit to meet the requirement.

Considered criteria of the system is when the guest reaches his/her room and touches the door handle the system must automatically activate and prompt the security key (PIN), Otherwise it will turn off unnecessary functions and stay in standby mode. When the guest entered the correct PIN the lock will be unlocked and the guest can get into the room. If the PIN is incorrect door will stay locked.

Block Diagram

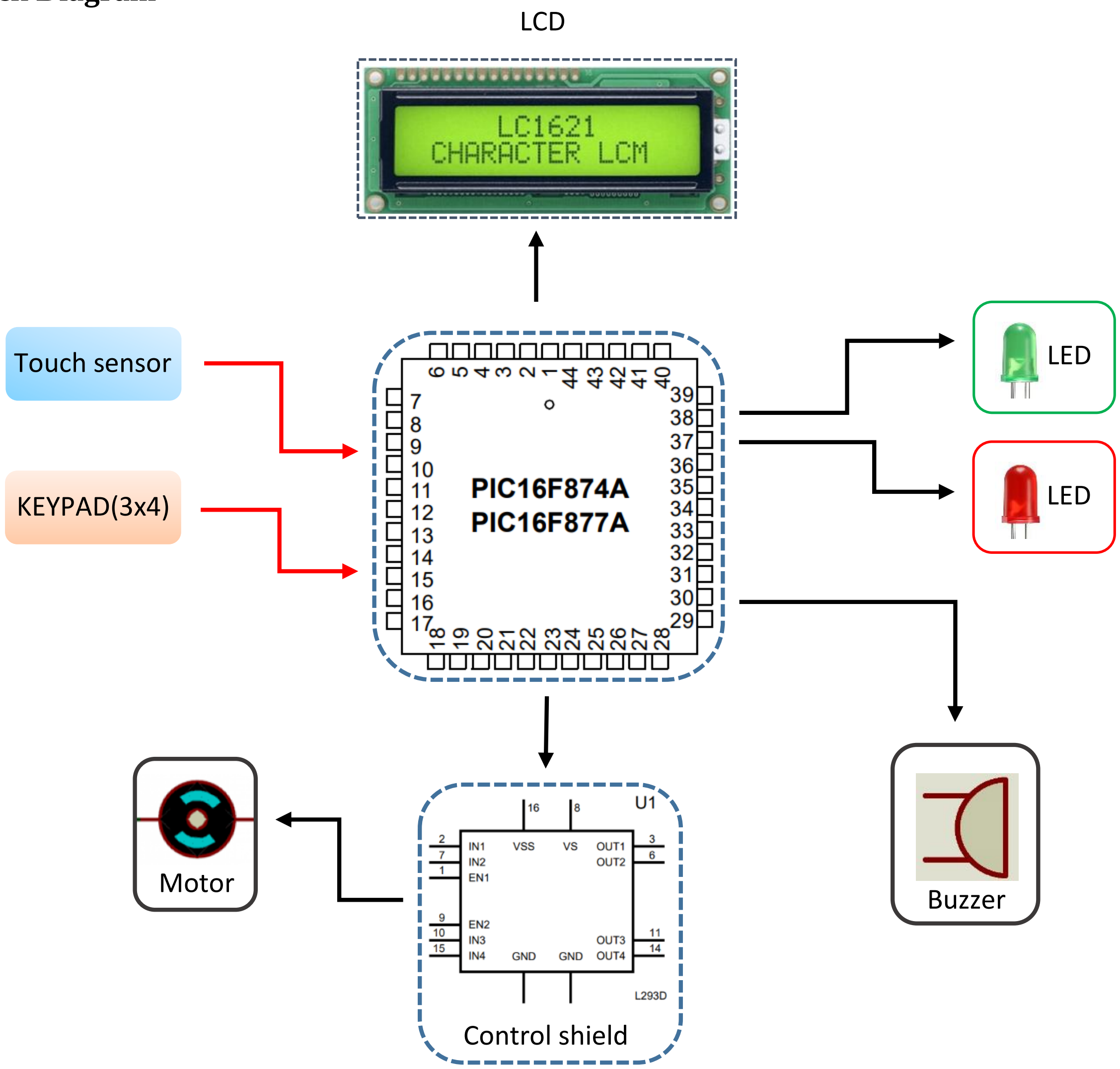
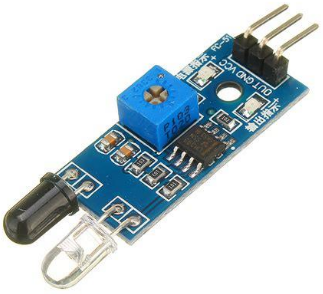


Figure 1 : Door lock Project Block Diagram

Suitable sensors

For the purpose of detecting a guest near the door we can use low cost, IR obstacle detection sensor, close by proximity sensors or touch sensitive sensor that will activate when the guest touches the keypad or door handle.



IR sensor

Proximity sensor

Capacitive touch sensor

3x4 numeric keypad. touch sensitive keypad.

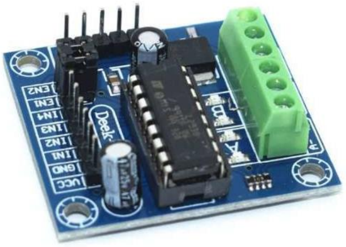
3x4 Keypad unit

Suitable Actuators

Motor

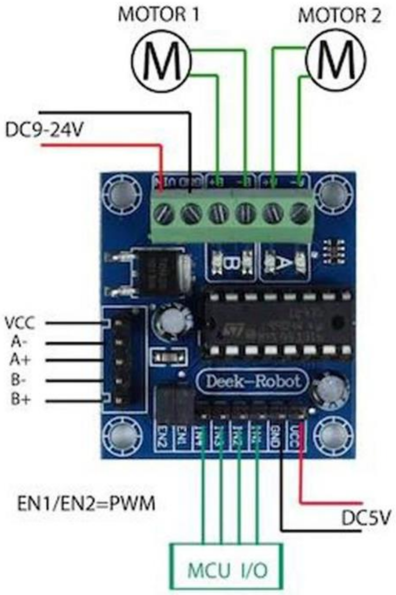
Electrically controlled actuator/motor, used for the locking and unlocking mechanism of the door. Can be either AC or DC powered.

Motor control drive



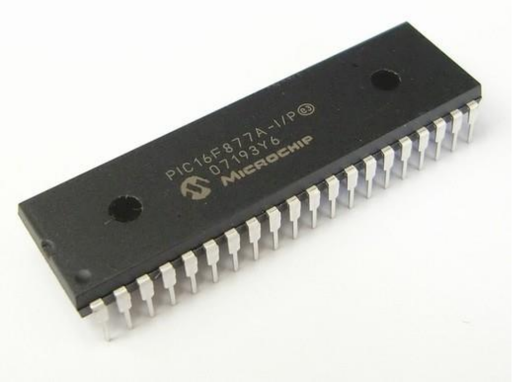
L293D motor control driver can be integrated into most MCU and allows of control two DC motors simultaneously.

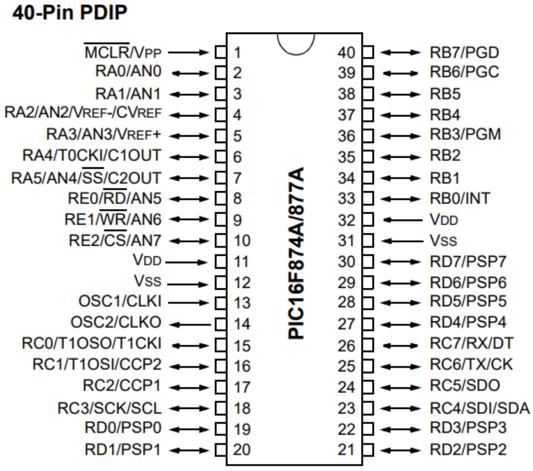
Operating Voltage : 4.5VDC - 36VDC
Max current single motor: 600mA



Motor Shield

Suggested Microcontroller unit





PIC16F877A is one of the common low-cost microcontrollers in the industry it is suggested to use because of its diverse range of compatibility, EEPROM support, 35 instruction set capability and low cost. It has multitude of applications including remote sensors, security and safety devices, home automation and many other industrial instruments.

Proteus simulation schematic.

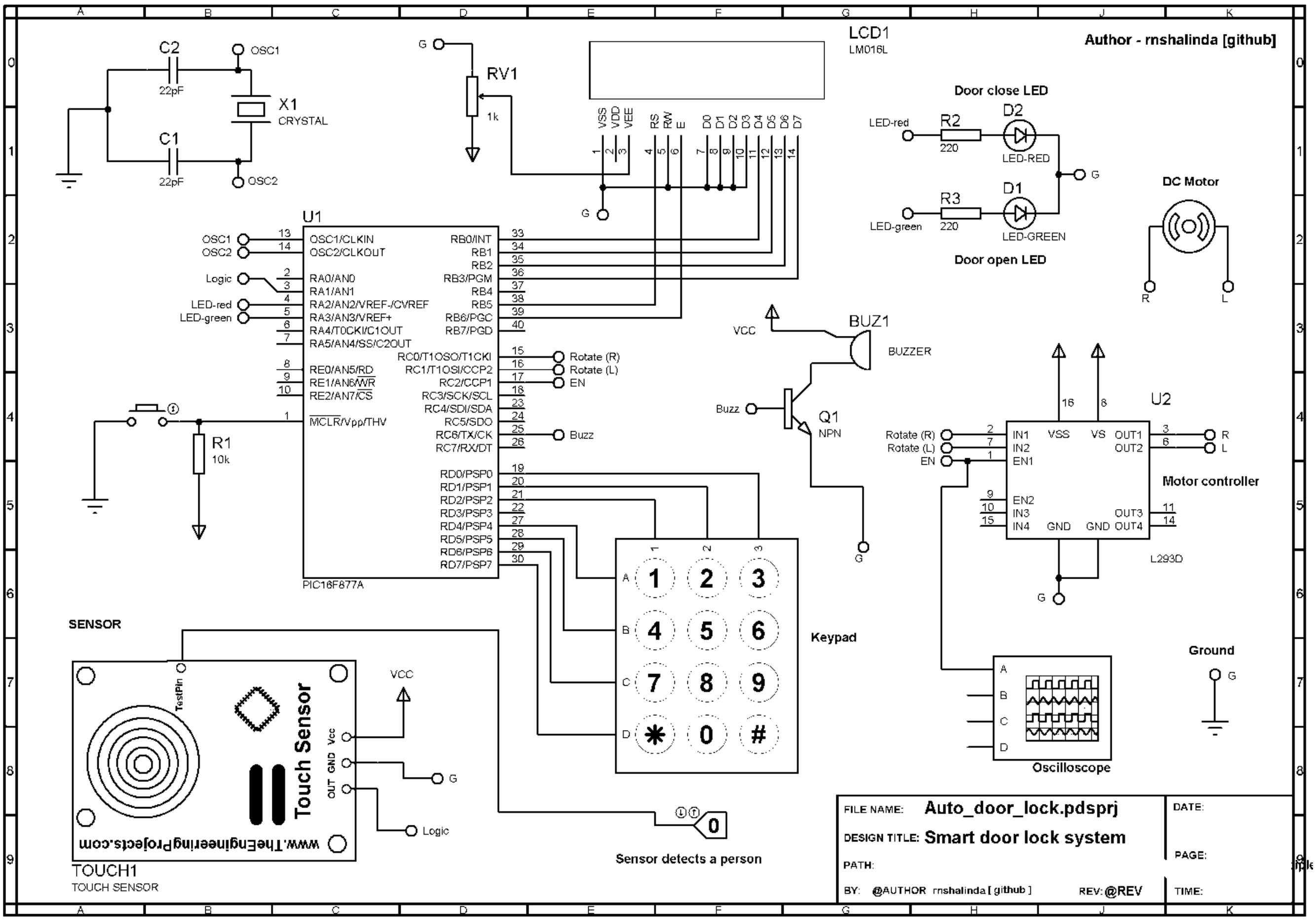


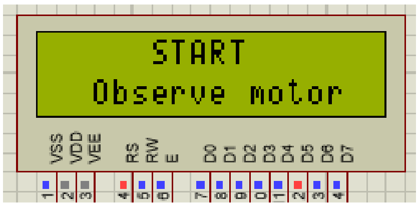
Figure 2: Complete Schematic diagram of the Door lock system

Used main components

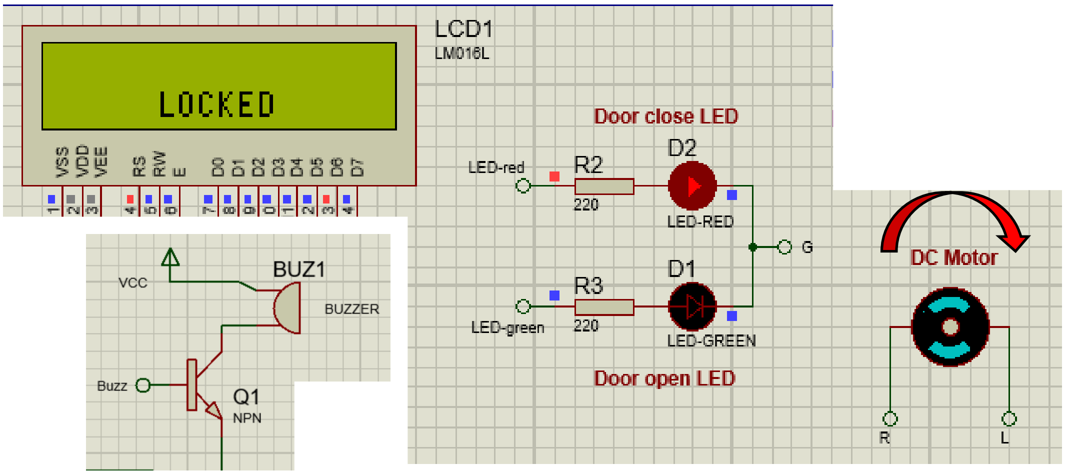
- Touch sensor - Detects a person touching the door handle
- Keypad (3x4) – PIN input
- LCD (LM016L) panel – Display message
- DC motor – Lock/unlock
- L293D motor drive – Control lock mechanism.
- Red and Green LED – Indicate door lock and unlock.
- Buzzer – Produce beep sound. Output sound will be played via computer sound card.
- PIC16F877A Microcontroller unit – Main control unit.

Working procedure

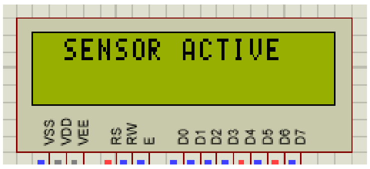
After pressing simulation play button, LCD will display

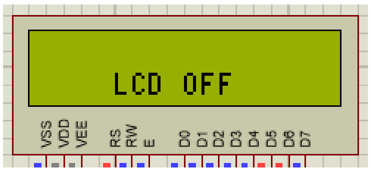


This initial part is only for you to notice the working process and should not be included in real world application. First the door has to be locked.

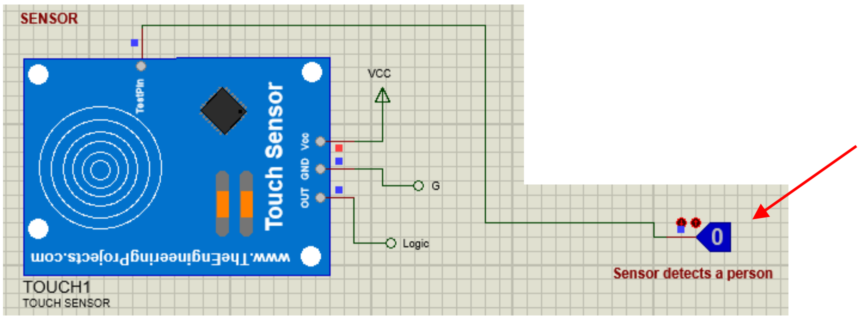


- Motor rotate 90 degrees right to lock the door and lights up RED LED and sound the BUZZER for 1s.



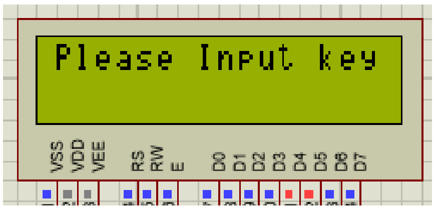


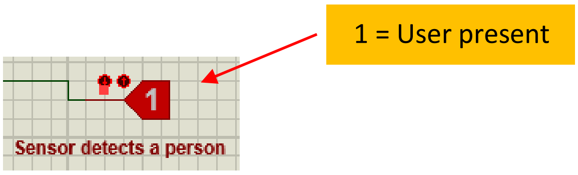
- Initially if there is nobody at the door, the LCD and other functions will remain turned off.



0 = No user

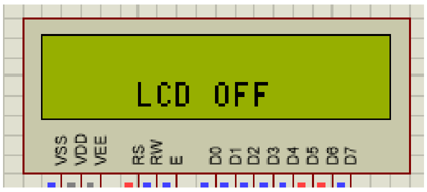
- This touch sensor can be place on door handle to read user presence, When user touches the door handle it gives logic 1 input signal to the microcontroller to start operation.
  (Also can use any other suitable sensor like IR obstacle detection for this)

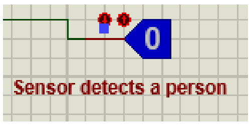




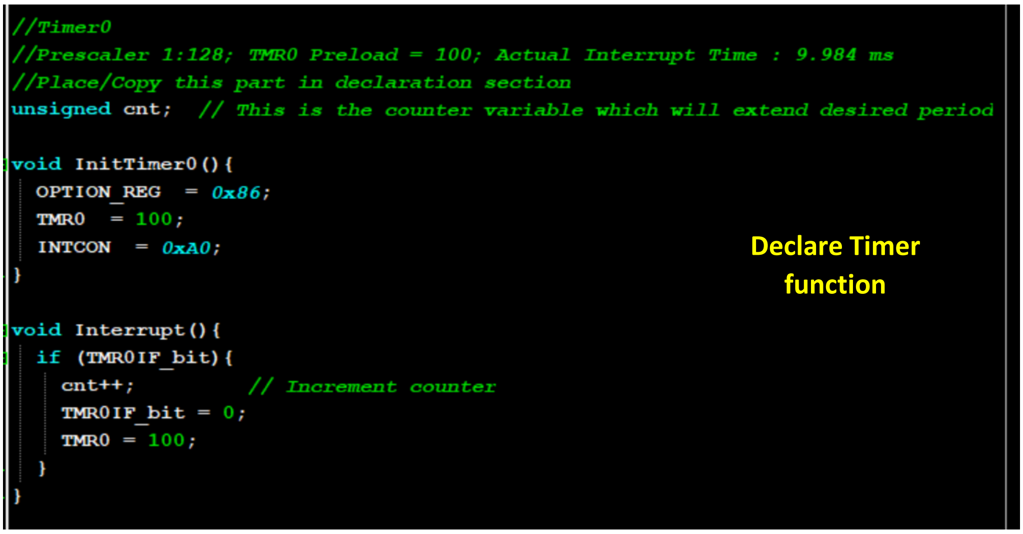
1 = User present

- After user detected it will wait 15s for PIN input.





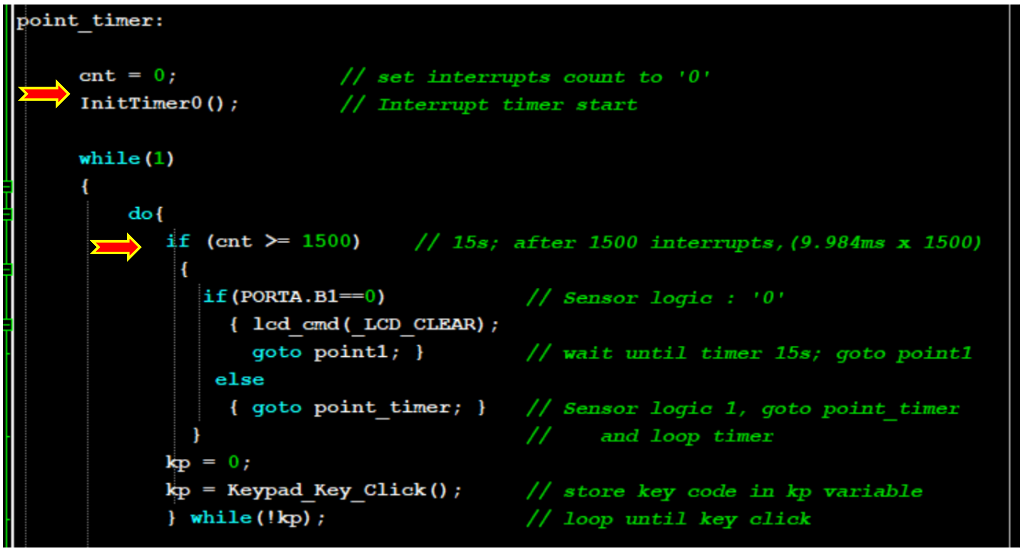
- After 15 seconds, Assuming user had taken his hand off from the door handle, LCD display will turn off.
- If the user had not taken his hand off from the door handle this 15s will keep looping another 15s requesting PIN input.
- For this function I have used **Timer0** register to count time.



```
//Timer0
//Prescaler 1:128; TMR0 Preload = 100; Actual Interrupt Time : 9.984 ms
//Place/Copy this part in declaration section
unsigned cnt;  // This is the counter variable which will extend desired period

void InitTimer0(){
  OPTION_REG   = 0x86;
  TMR0   = 100;
  INTCON   = 0xA0;
}

void Interrupt(){
  if (TMR0IF_bit){
    cnt++;            // Increment counter
    TMR0IF_bit = 0;
    TMR0 = 100;
  }
}
```

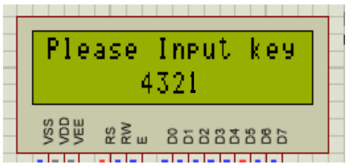
Declare Timer function

- Timer will allow 9.984ms interrupts per single **cnt** count.



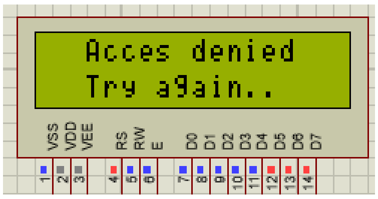
```
point_timer:

     cnt = 0;                  // set interrupts count to '0'
     InitTimer0();             // Interrupt timer start

     while(1)
     {
         do{
            if (cnt >= 1500)     // 15s; after 1500 interrupts,(9.984ms x 1500)
            {
               if(PORTA.B1==0)          // Sensor logic : '0'
                 { lcd_cmd(_LCD_CLEAR);
                   goto point1; }       // wait until timer 15s; goto point1
               else
                 { goto point_timer; }  // Sensor logic 1, goto point_timer
            }                           //    and loop timer
            kp = 0;
            kp = Keypad_Key_Click();    // store key code in kp variable
            } while(!kp);               // loop until key click
```

- Within the code **initTimer0();** will be called and **cnt++** start count. When the count reaches **1500** it will execute the given part.
  (cnt = 9.984 ms x 1500 = 14,976 ms , By converting milliseconds to seconds we get almost **15s**)

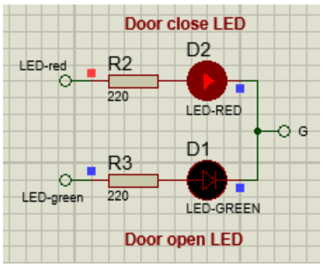


```
Please Input key
      4321
```

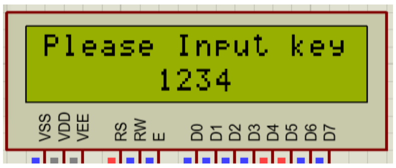
Incorrect PIN



```
 Acces denied
 Try a9ain..
```

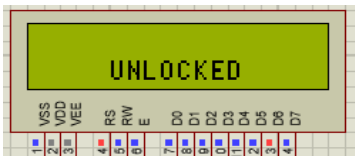


- If user entered wrong PIN, it will display above output and blinks Red-LED.

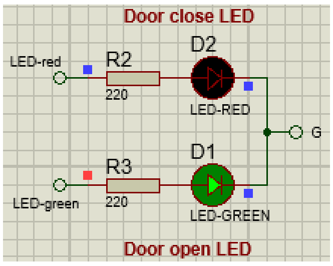


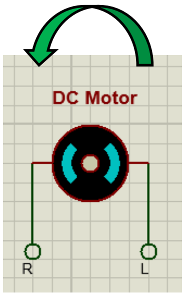
```
Please Input key
      1234
```

Correct PIN

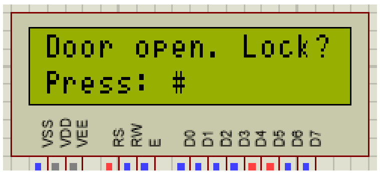


```
    UNLOCKED
```

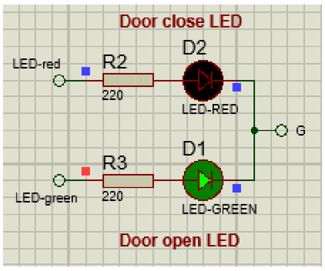




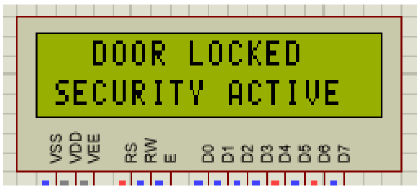
- When the user input correct PIN door motor rotates 90˚ left and Red-LED. Turn OFF and Green-LED stay on.

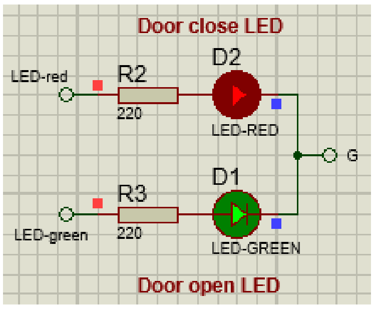


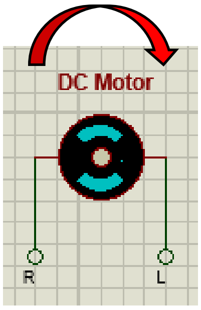
```
Door open. Lock?
Press: #
```

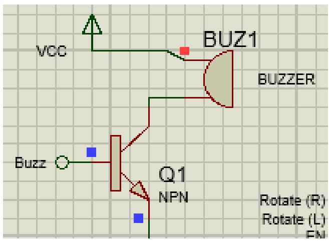


- When the door is open, after 3 second delay system will prompt guest to close the door.
- **Press #** to execute locking mechanism. Green-LED still stay ON until door is locked.

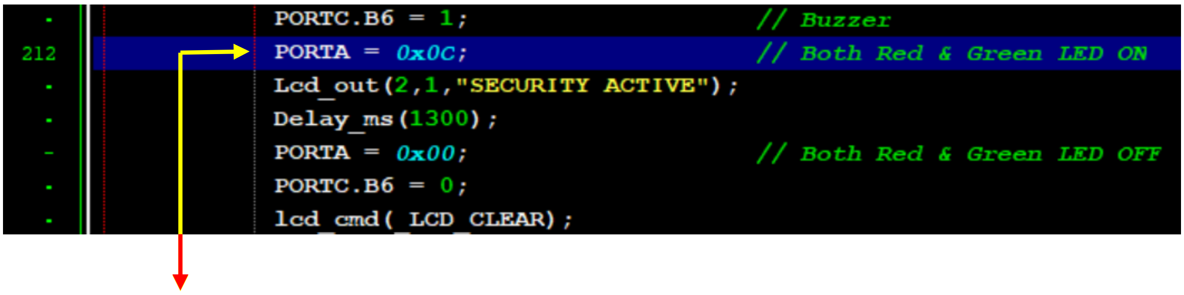




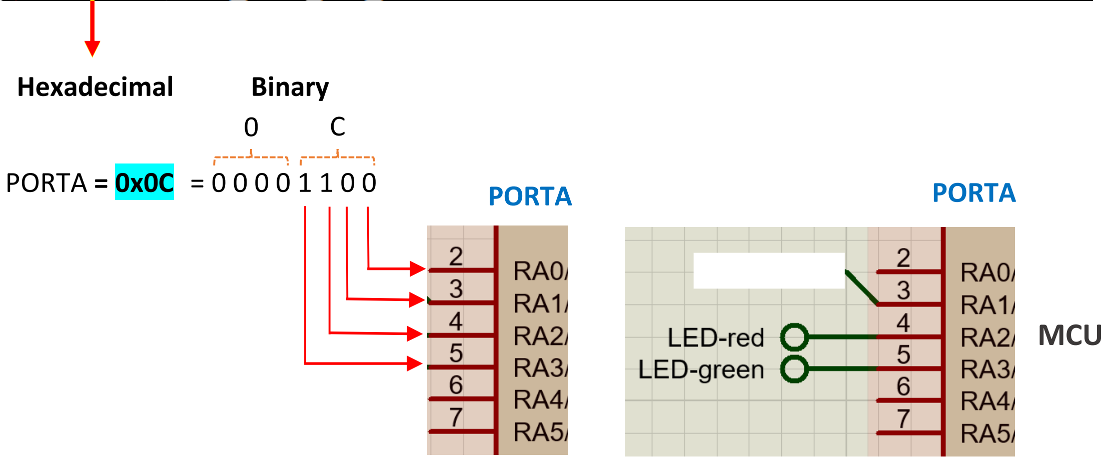




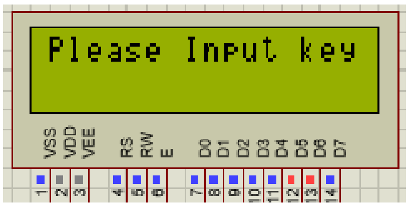
- When the user **press #** the motor will again rotate 90˚ right to lock.
- After locking the door, Both Red/Green LED and Buzzer will simultaneously stay ON for 1.5s and turn OFF after.
- Note ; to turn ON two LEDs (two mcu pins) at the same time, send Hexadecimal values to PORTA.

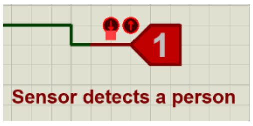


```
PORTC.B6 = 1;                          // Buzzer
PORTA = 0x0C;                          // Both Red & Green LED ON
Lcd_out(2,1,"SECURITY ACTIVE");
Delay_ms(1300);
PORTA = 0x00;                          // Both Red & Green LED OFF
PORTC.B6 = 0;
lcd_cmd(_LCD_CLEAR);
```

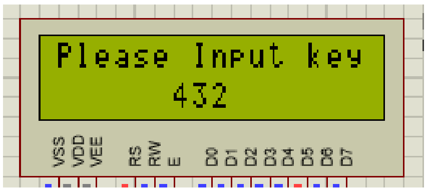


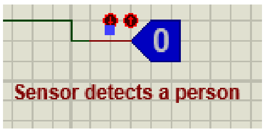
- As shown above it is crucial to provide Hexadecimal value if you need to power more than one pin simultaneously.





- Finally the system loop back to the starting position, But if user is still holding door handle, the staring position will be the PIN input point.





- Likewise if his not holding the handle, system will revert back to initial stand by state to save power.
- This process will repeat whenever someone wants to open the door.