Password-based-door-lock-system-using-8051-microcontroller-ALP

- ARPIT PATAWAT

CONTENTS

Introduction →

Due to the advancement of science and technology throughout the world, there may be a consequent growth inside the fee and sophistication of crime. As an end result, it is important to make certain safety of oneself and one's treasured belongings.in spite of the usage of mechanical locks, the crime price still has multiplied due to the truth that those locks are effortlessly damaged. Consequently, there may be a want for other varieties of locks mainly electronic ones. Traditional locking systems using a mechanical lock and key are now a days being replaced with new advanced techniques of locking systems. With the advancement of digital age these systems are being replaced by digital systems which rely on various, different types of inputs like pins, fingerprints, image processing or a combination of these.
This project is an integration of mechanical and electronic devices and is highly programable and compatible with existing infrastructure with easy integrability. One of the prominent features of these innovative locks is their simplicity and high efficiency.

This work is of an electronic combination lock with a keyboard to be mounted on the door for keying in the secret code. This automatic lock system consists of an electronic control assembly, which controls the output load through a password. This output load here is a Stepper motor which controls the door. LCD is attached with the lock system which display the current status of lock. The code unit, which operates with a 12-switch (matrix) keypad, was designed to control an electromechanical door lock with five– digit code. When a correct password is fed to the keypad, the Door will open for 10 seconds and then close it. A separate counter is added which tells about how many people have entered the room so far.
Unlike other keyboard combination locks this lock is constructed in such a way that once any of the wrong keys is pressed, it resets automatically making it harder for an intruder to break into. And tells about the number of Attempts the person have before making an alarm. It also blows an alarm after 3 error count entry and at the same time it informs user via E-mail where the user has option in his smartphone either to catch the intruder or reset the door lock system, so necessary measures can be taken before happening of a big crime and loss of precious things.

this "Password Based Door Lock System" using 8051 Microcontroller is used in the places where we need more security. It can also use to Secure lockers and other protective doors. Adding Arduino make it IoT based alert system which is different than the conventional door lock system.

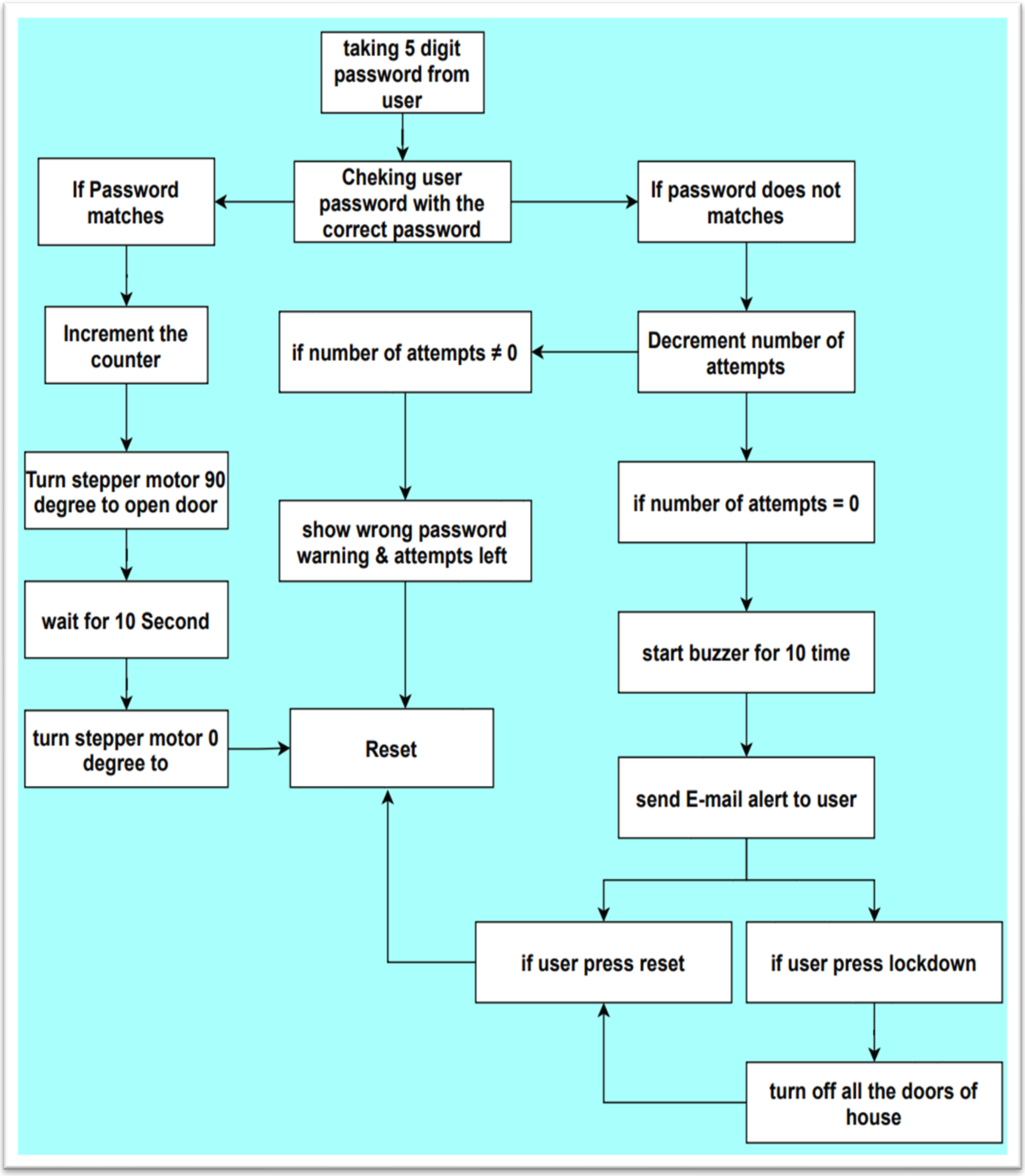
Objective →

The Main objective of our porjects are –

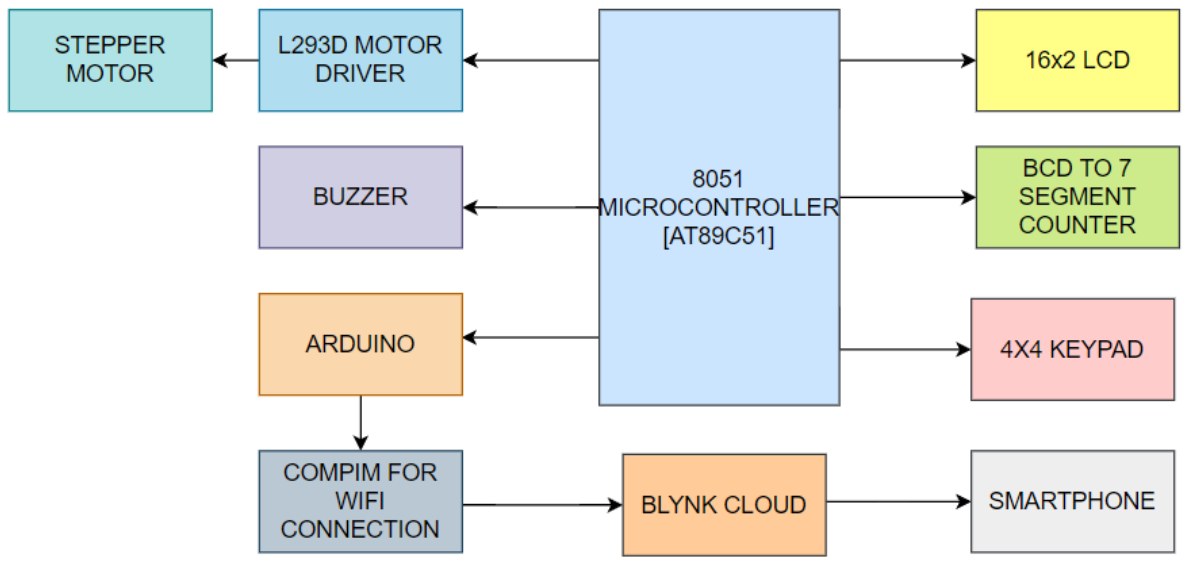
- Build a Secure Door Lock System
- The Door should be only open when correct password is entered
- It should be open for just a short span of intervel and automaticaly close after that
- Owner can see how many people have entered the room so far
- When wrong password is typed, it should not open the door and LCD should tell about number ao attempts left
- When wrong password is typed continueously for 3 time then LCD should tell about intruder alert
- An email to be sent to user’s e-mail address about the intruder warning
- After the E-mail alert, owner need to decide whether to turn on lockdown and close all the doors of the house to catch the intruder or just reset the program.
- The program must not proceed without the owner’s confirmaton.
- LCD must display the status of on-going processes.
- User cannot tamper with the original password

Project description and goals

i. Block diagram of project


taking 5 digit password from user
Cheking user password with the correct password
If Password matches
If password does not matches
Increment the counter
if number of attempts ≠ 0
Decrement number of attempts
Turn stepper motor 90 degree to open door
show wrong password warning & attempts left
if number of attempts = 0
wait for 10 Second
start buzzer for 10 time
turn stepper motor 0 degree to
Reset
send E-mail alert to user
if user press reset
if user press lockdown
turn off all the doors of house

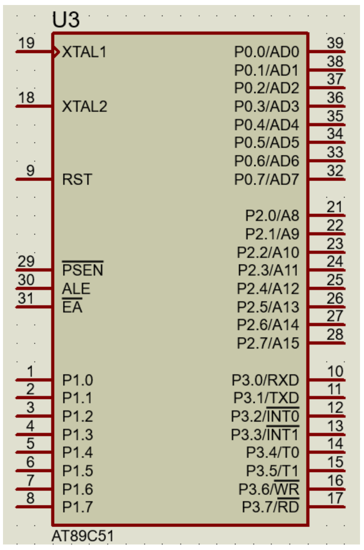
ii. Block diagram of components



Technical Specifications –

i. Components discription –

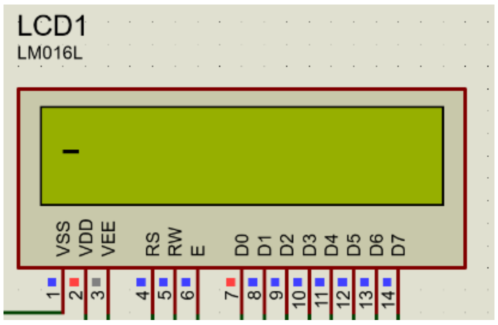
- **8051 Micro-controller** –



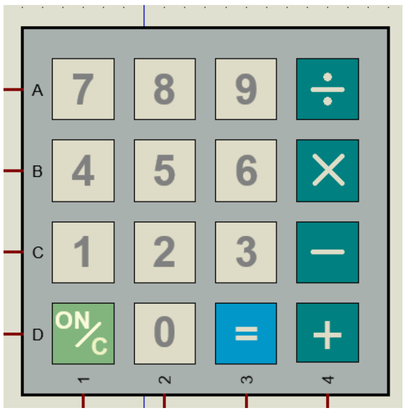
AT89C51 is an 8bit microcontroller and belongs to Atmel's 8051 families. AT89C51 has 4KB of Flash programmable and erasable read-only memory (PEROM) and 128 bytes of RAM. It can be erased and program to a maximum of 1000 times.it has 4 8-bit input/output port. On start-up all ports are set to 1 (input mode). Port 0 needs external pull ups (adding 10K resistor) and the rest of ports have internal pull ups. These ports are also a bit addressable and so their bits can also be accessed individually. The main features of the 8051 microcontrollers are: RAM – 128 Bytes (Data memory) ROM – 4Kbytes (ROM signify the on-chip program space) .

- **16X2 LCD** –

An LCD (Liquid Crystal Display) screen is an electronic display module and has a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each



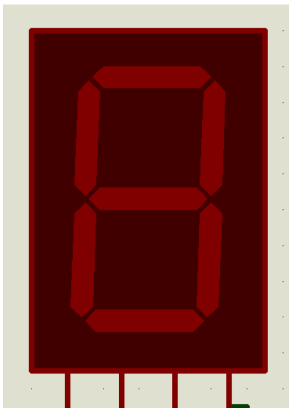
character is displayed in 5x7 pixel matrix. The 16 x 2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. This LCD has two registers, namely, Command and Data. Command register stores various commands given to the display. Data register stores data to be displayed. The process of controlling the display involves putting the data that form the image of what you want to display into the data registers, then putting instructions in the instruction register.



- **4x4 Key-Pad –**

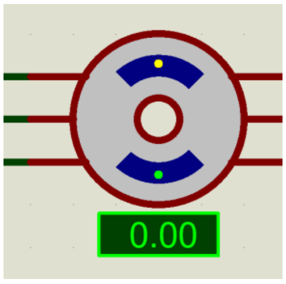
A 4x4 keypad will have eight terminals. In them four are rows of matrix and four are columns of matrix. These 8 pins are driven out from 16 buttons present in the module. Those 16 alphanumeric digits on the module surface are the 16 buttons arranged in matrix formation.

- **BCD to 7 segment counter –**



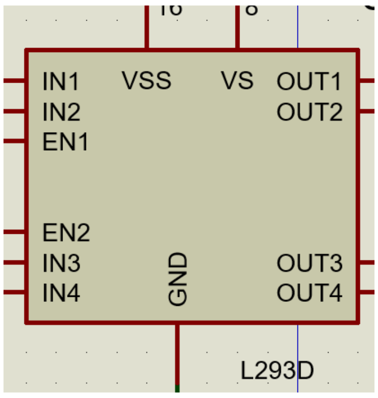
Seven segment displays are the output display device that provide a way to display information in the form of image or text or decimal numbers which is an alternative to the more complex dot matrix displays. A binary coded decimal (BCD) to 7-segment display has 4 BCD inputs and 7 output lines, one for each LED segment. This allows a smaller 4-bit binary number (half a byte) to be used to display all the denary numbers from 0 to 9 and by adding two displays together, a full range of numbers from 0 to 9 can be displayed with just a single byte of eight data bits.

- **Stepper motor –**



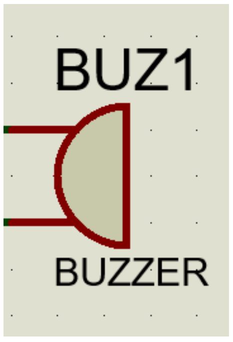
A stepper motor is an electric motor whose main feature is that its shaft rotates by performing steps, that is, by moving by a fixed number of degrees. This feature is obtained thanks to the internal structure of the motor, and allows to know the exact angular position of the shaft by simply counting how may steps have been performed, with no need for a sensor. This feature also makes it fit for a wide range of applications

- **L293D MOTOR DRIVER –**



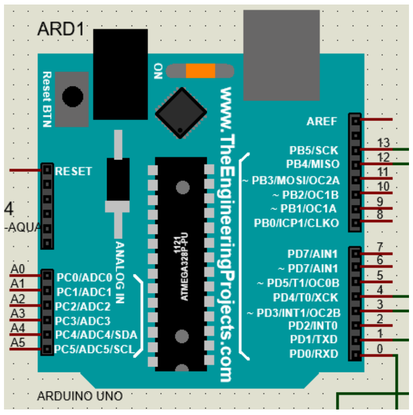
The L293D is a popular 16-Pin Motor Driver IC. it is mainly used to drive motors. A single L293D IC is capable of running two motors at the same time; also, the direction of these two motors can be controlled independently. So, for motors which has operating voltage less than 36V and operating current less than 600mA, which are to be controlled by digital circuits like Op-Amp, 555 timers, digital gates or even Microcontrollers, this is mainly used.

- **BUZZER –**



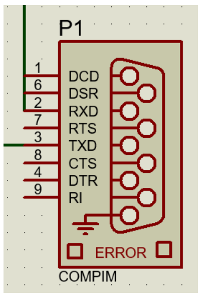
An audio signalling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren.

- **Arduino –**



Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. The Wiring platform consisted of a printed circuit board (PCB) with an ATmega168 microcontroller, an IDE based on Processing and library functions to easily program the microcontroller.

- **COMPIM –**



COMPIM is used to model physical COM interfaces in Proteus. It works by capturing and buffering serial signals which it then presents to the electrical circuit. The computer's serial ports will be used to conduct all serial data originating from the CPU or the UART model.

ii. Software discription –

- **Proteus Design suite**–

Using Proteus, we have designed the circuit and simulated it.

- **BLYNK** –

Using BLYNK app, we were able to control our circuit as well as send the alert mail. And it also helped us to make connection with internet for Arduino in the proteus tool.

- **GMAIL** –

Using G-mail we were able to get the security alert mail

- **Keil micro vision**

Using Keil micro vision, we wrote the assembly language programme that is afterward uploaded to 8051 microcontroller.

- **Arduino IDE**

Using Arduino IDE, we programmed Arduino such that is can get input from 8051 micro controller and it can even send output to it along with the internet communication.

- **Virtual Serial Port emulator**

Using VSPE, we made a connection between hardware port and virtual port (pairing) through which communication with internet was possible in proteus.

- **Command prompt**

Using windows cmd, we were able to create a virtual port.

Design Approach→

1. Component interfacing

i. LCD interfacing –

Category	Pin No.	Pin Name	Circuit Connection	Function
Power Pins	1	VSS	Ground	Ground Pin, connected to ground
	2	VDD or VCC	5V	Voltage Pin +5V
Contrast Pin	3	V0 or VEE	10K variable resistor	Contrast Setting, connected to VCC through a variable resistor
Control Pins	4	RS	P2.0	Register Select Pin, RS = 0 Command Mode, RS =1 Data Mode
	5	RW	P2.1	Read/Write pin, RW = 0 Write mode, RW = 1 Read Mode
	6	E	P 2.2	Enable, a high to low pulse need by the LCD to latch information presented on data bus
Data Pins	7 - 14	D0 - D7	P 3.0 – P 3.7	Data Pins, Stores the data to be displayed on LCD or the command instructions

Algorithm to send command-

- Copy the command value to port 3
- Make register select = 0
- Select write mode by making R/W = 0
- Give a high to low Pulse to enable (port value high, delay then low)

Algorithm to send/display data -

- Copy the ASCII value of data to port 3
- Make register select = 1
- Select write mode by making R/W = 0
- Give a high to low Pulse to enable (port value high, delay then low)

Algorithm to display string text –

- Define string using DB command at some address
- Add 0 at the end
- Move data pointer to that address
- Take first byte of String and move to accumulator A
- Display this data of A to LCD
- Increment DPTR up to we get 0

LCD Command Codes

Code (Hex)	Command to LCD Instruction Register	Code (Hex)	Command to LCD Instruction Register
1	Clear display screen	E	Display on, cursor blinking
2	Return home	F	Display on, cursor blinking
4	Decrement Cursor (Shift cursor to left)	10	Shift cursor position to left
6	Increment Cursor (Shift cursor to right)	14	Shift cursor position to right
5	Shift display right	18	Shift the entire display on left
7	Shift display left	1C	Shift the entire display on right
8	Display off, cursor off	80	Force cursor to beginning to 1st line
A	Display off, cursor on	C0	Force cursor to beginning to 2nd line
C	Display on, cursor off	38	2 lines and 5x7 matrix

By using this above algorithm, we are displaying desired number/string to LCD.

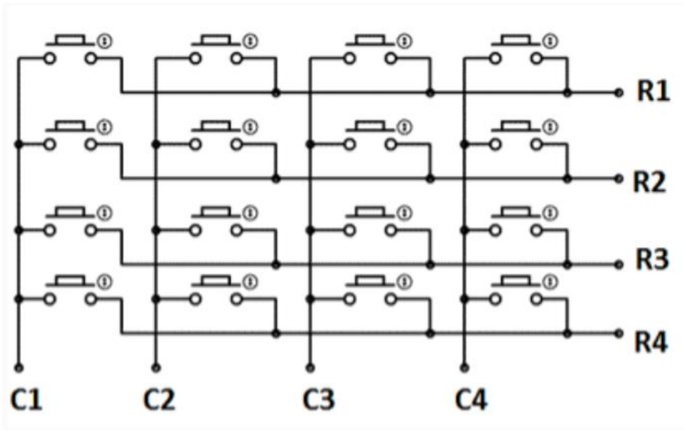
ii. Keypad interfacing

Category	Pin No.	Pin Name	Circuit Connection	Function
Row Pins	1 - 4	A-D	P1.0 – P1.3	To detect Key press in a Row
Column Pin	5-8	1-4	P1.4 - P1.7	To detect Key press in a Column

Algorithm to Read key press-

- make all the rows and columns high/VCC (1)
- make the first starting row = ground/low (0)

- If the user will press any key from starting row, then that column will become grounded indicates that key is pressed
- Check for all the columns if it is grounded then load the ASCII value of the key to accumulator and stop the process
- If none of the column is grounded, then make this row 1/High voltage and repeat the same process for the other rows.



- For 5 input we have to check continuously until we get 5 inputs.

Using this aboe algorithm, we are taking input from user and work accordingly.

iii. Stepper Motor & L293D Interfacing –

PIN layout -

IN1– Port 2.3 IN2 –Port 2.4 IN3 –Port 2.5 IN4 – Port 2.6

EN1, EN2, and VSS– 5V VS – 12-volt battery OUT 1,2,3,4 – 4 ports of Stepper motor

Step angle = 90, number of steps = 4

Rotation (degree)	Port 2.3	Port 2.4	Port 2.5	Port 2.6
270	1	1	0	0
180	0	1	1	0
90	0	0	1	1
0	1	0	0	1

By assigning this value to the port of 8051, we can move stepper motor to these 4 angles.here we are moving only to 90 degrees just to open door

iv. Arduino interfacing

Arduino Port	Connection Port	function
TXD	TXD of COMPIM	To transfer data to hardware port
RXD	RXD	To receive data from Hardware port
Pin 3	P0.2 of 8051	To reset the program
Pin 4	P0.1 of 8051	To initiate lockdown

Pin 12	P0.0 of 8051	To trigger E-mail notification
Pin 13	+ve of LED	To indicate Arduino has received signal

In the Arduino programming, it is defined such that first we will take input from P0.0 of 8051, if it is high which mean that intruder alert, then using blynk library we are sending an e-mail to user predefined address after that the control goes to blynk app.

v. Blynk App interfacing –

Using Virtual port, we have made a connection between Arduino and Blynk app and this will run on Laptop Wi-Fi. we are assigning 2 buttons- 1 for Reset and another for lockdown connected to D.3 and D4 respectively. When e-mail is triggered, we are the control goes to blynk app, here if we press any of these 2 buttons, the function will work. And the 8051 will take input from the high bit of 8051 and work accordingly.

vi. 7 segment counter interfacing –

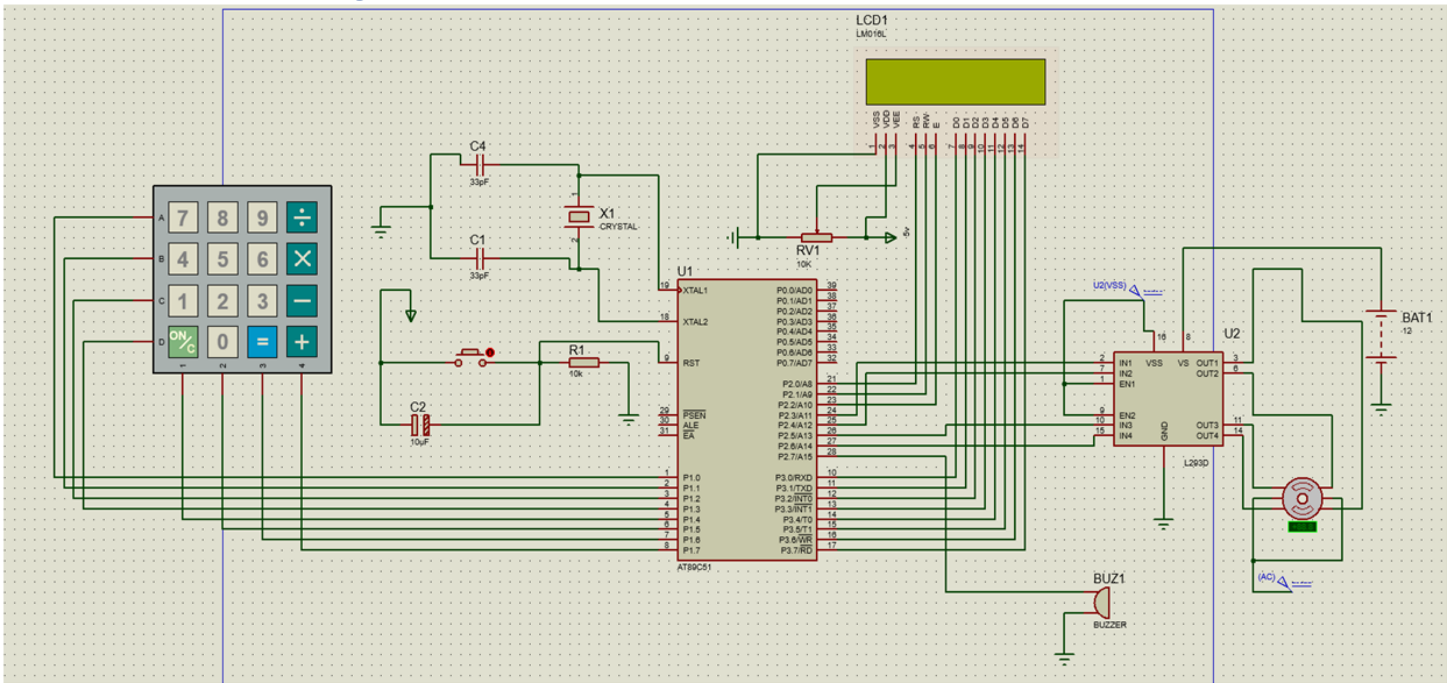
Here the input ports are 4 and the output ports are 7 so using the BCD value of a number, we can represent that number in this counter.

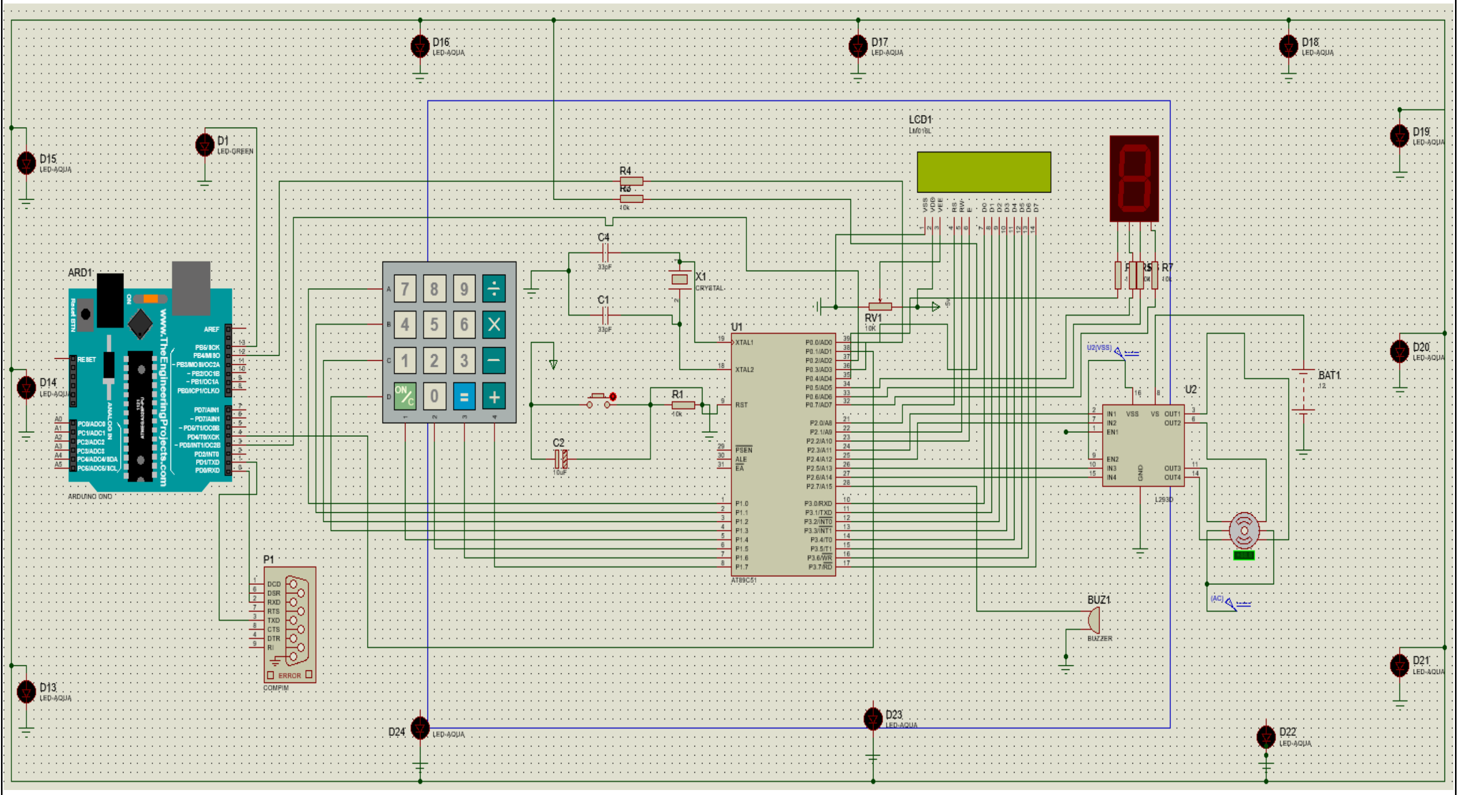
Code	**Value**	**Code**	**Value**
0000	0	0100	4
0001	1	0101	5
0010	2	0110	6
0011	3	0111	7
1000	8	1001	9

2. Working of program –

First we will display "Enter 5 dig. code" string to LCD. Now user will enter 5 digits via keypad which we are taking and storing at memory location 160-164 and along with this we are displaying in the LCD. Now for each digit the user has entered, we are checking with the pre-defined password. If all the 5 digits matches then we will increment the counter by sending the corresponding bits which are defined at location 500H.after that we will display the user that gate is opening and then we will move stepper motor by 90 degrees for 10 second and then we will close it. For the correct password, the counter will go from 1-9. Now if the user has entered wrong password, then we will decrement the attempt value once and if it is not equal to 0 then we just display the "wrong password" and number of attempts left. If the user has given 3 wrong attempts, then will turn on buzzer for 10 times and we will set the e-mail alert pin to be high which is connected directly with the Arduino. Now as Arduino is connected with the interned through the virtual port, we can send an email once and then using blynk app if we press reset button then the whole program will reset and we press the lockdown then 8051 will send high signal to the peripherical Led's which represent all the other doors of the house and their turning on indicates that the house is locked from outside.

3. Circuit Diagram – basic and advanced





4. Code for 8051

```
ORG 0000H
MOV R6,#0D ;counter value
MOV R5,#3D ;number of attempts
MOV A,#0 ; value to display 0 to counter
MOV C,ACC.4
MOV P0.4,C
MOV C,ACC.5
MOV P0.5,C
MOV C,ACC.6
MOV P0.6,C
MOV C,ACC.7
MOV P0.7,C

;--------------------------------
MAIN:
CLR P0.0 ;make it 1 only when we send alert to arduino
CLR P2.7 ; port 2.7 will be used for buzzer so make it 1 when to alarm
CLR P0.3 ; led lockdown make it 1 when lockdown needs to start
ACALL LCD_INIT ; initialize LCD subroutine
MOV DPTR,#INITIAL_MSG ;DPTR point to initial text
ACALL SEND_DAT ;DISPLAY DPTR content on LCD
ACALL DELAY ;GIVE DELAY
ACALL LINE2 ;MOVE TO LINE 2
ACALL READ_KEYPRESS ;take input from keypad
ACALL DELAY ;give some delay
ACALL CLRSCR ; clear our screen
MOV DPTR, #CHECK_CODE_MSG ;send checking code.. msg to lcd
ACALL SEND_DAT
ACALL DELAY2
ACALL CHECK_PASSWORD  ;CHECK for correct password
SJMP MAIN ;short jump to main
;---------------------------------
LCD_INIT:MOV DPTR,#MYDATA ; code to load initial command to LCD
C1:CLR A
MOVC A,@A+DPTR
JZ DAT ; jump if A  = 0
ACALL COMNWRT
ACALL  DELAY
INC DPTR
SJMP C1
DAT:RET
;---------------------------------
SEND_DAT:  ;code to display string to LCD
CLR A
MOVC A,@A+DPTR
JZ AGAIN ; jump if A = 0
ACALL DATAWRT
ACALL DELAY
INC DPTR
SJMP SEND_DAT
AGAIN: RET
;---------------------------------
READ_KEYPRESS: ; to store the password and save it to memory location
MOV R0,#5D  ; R0 = 5
MOV R1,#160D ; R1= 160
ROTATE:ACALL KEY_SCAN   ;take the input key
MOV @R1,A ;take the key pressed value and store at address of R1 i.e. 160
ACALL DATAWRT ; display the key on LCD
ACALL DELAY2 ; delay
ACALL DELAY2 ;
INC R1
DJNZ R0,ROTATE ;repeat this process for 5 time
RET
;----------------------------------
CHECK_PASSWORD:MOV R0,#5D  ;R0 = 5
MOV R1,#160D ; R1= 160
MOV DPTR,#PASSWORD ;DPTR Point to actual PASSWORD
RPT:CLR A ; A = 0
MOVC A,@A+DPTR ; A = FIRST NUMBER OF THE ACTUAL PASSWORD
XRL A,@R1 ; XOR with the actual password
;if both the numbers are equal then A = 0;
JNZ FAIL ; jump if a not = 0
INC R1
INC DPTR
DJNZ R0,RPT ;repeat this process for 5 times
ACALL SUCCESS
RET
;-----------------------------------
SUCCESS:ACALL CLRSCR
ACALL DELAY2
MOV DPTR,#TEXT_S1
ACALL SEND_DAT ;display correct password
ACALL DELAY2
ACALL LINE2
MOV DPTR,#TEXT_S2
ACALL SEND_DAT ;display opening door
ACALL DELAY2
MOV DPTR,#500H
MOV A,R6
MOVC A,@A+DPTR ;to display counter value
MOV C,ACC.4
MOV P0.4,C
MOV C,ACC.5
MOV P0.5,C
MOV C,ACC.6
MOV P0.6,C
MOV C,ACC.7
MOV P0.7,C
INC R6 ;increment counter
ACALL DELAY
CLR P2.3 ;open door
CLR P2.4
;ROTATE MOTOR CLOCK WISE 90 degree TO OPEN DOOR
ACALL DELAY3 ; GIVE 5 SECOND DELAY
ACALL CLRSCR
MOV DPTR, #TEXT_S3
ACALL SEND_DAT
ACALL DELAY2
ACALL DELAY3; GIVE 5 SECOND DELAY
SETB P2.3 ;rotate motor antin clock wise
CLR P2.5
ACALL DELAY2
SETB P2.3 ;initial position
SETB P2.4
SETB P2.5
SETB P2.6
MOV R5,#3D ;reset attempts value
RET
;----------------------------
FAIL:ACALL CLRSCR
MOV DPTR,#TEXT_F1
ACALL SEND_DAT ;display incorrect text
ACALL DELAY2
ACALL LINE2
MOV DPTR, #TEXT_F2
ACALL SEND_DAT ;display access denied text
ACALL DELAY2
DJNZ R5,LOOP ;check for number of attempts
ACALL ALERT ;alert function if attempts valu = 0
LOOP: ACALL ATTEMPT
LJMP MAIN ;go to main funtion
;------------------------------
ATTEMPT: ACALL CLRSCR
MOV DPTR,#ATTEMPT_TEXT ;number of attempts left
ACALL SEND_DAT
ACALL DELAY2
MOV A,#48D ; 48 = 0
ADD A,R5
DA A
ACALL DATAWRT ;here displaying attempts value
ACALL DELAY
ACALL DELAY2
ACALL DELAY2 ;
RET
;-------------------------------
ALERT:SETB P0.0 ;sending alert to arduino
MOV R2,#10D ;to start buzzer for 10 time
```

```
ACALL CLRSCR
MOV DPTR, #ALERT_TEXT ;display alert text
ACALL SEND_DAT
ACALL DELAY2
BUZZ:SETB P2.7 ;buzzer will turn on and off 10 times
ACALL DELAY2
CLR P2.7
ACALL DELAY2
DJNZ R2, BUZZ
MOV R5,#3D
SJMP NOTIFICATION
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~
NOTIFICATION:JB P0.1,LOCKDOWN ;if user press lockdown
JB P0.2,RESET ;if user press reset
SJMP NOTIFICATION
LOCKDOWN:SETB P0.3 ;turn on all led's
ACALL CLRSCR
MOV DPTR,#LOCKDOWN_TEXT ;
ACALL SEND_DAT
ACALL DELAY2
ACALL LINE2
MOV DPTR,#POLICE_TEXT ;
ACALL SEND_DAT
ACALL DELAY2
SJMP NOTIFICATION ;keep cheking for user input

RESET: LJMP MAIN ;reset

;-------------------------------------------------
;algorithm to check for key scan
KEY_SCAN:MOV P1,#11111111B  ;TAKE INPUT FROM PORT 1
;CHECKING FOR ROW 1 COLUMN 1
CLR P1.0  ;first row checking #11111110
JB P1.4, NEXT1 ;when 1 column is 1 then no button is pressed , check for next column
MOV A,#55D ; if above fails then 7 is pressed , A =7
RET

NEXT1:JB P1.5,NEXT2 ; ROW 1 COULMN 2
MOV A,#56D ; A = 8
RET

NEXT2: JB P1.6,NEXT3 ; ROW 1 COLUMN 3
MOV A,#57D ; A=9
RET

NEXT3: JB P1.7,NEXT4 ; ROW 1 COLUMN 4
MOV A,#47D  ; A=/ DIVIDE
RET

NEXT4:SETB P1.0 ; ROW 1 IS RESET
```

```
CLR P1.1 ;CHECK FOR ROW 2
JB P1.4, NEXT5 ; ROW 2 COLUMN 1
MOV A,#52D ; A = 4
RET

NEXT5:JB P1.5,NEXT6 ; ROW 2 COLUMN 2
MOV A,#53D ;A = 5
RET

NEXT6: JB P1.6,NEXT7 ; ROW 2 COLUMN 3
MOV A,#54D ;A = 6
RET

NEXT7: JB P1.7,NEXT8 ; ROW 2 COLUMN 4
MOV A,#42D ; A = *
RET

NEXT8:SETB P1.1 ;ROW IS RESET
CLR P1.2 ; CHECK FOR ROW 3
JB P1.4, NEXT9 ; ROW 3 COLUMN 1
MOV A,#49D  ;A = 1
RET

NEXT9:JB P1.5,NEXT10 ; ROW 3 COLUMN 2
MOV A,#50D ;A =2
RET

NEXT10: JB P1.6,NEXT11 ; ROW 3 COLUMN 3
MOV A,#51D ;A = 3
RET

NEXT11: JB P1.7,NEXT12 ; ROW 3 COLUMN 4
MOV A,#45D ;A = -
RET

NEXT12:SETB P1.2 ; ROW 3 IS RESET
CLR P1.3 ; CHECK FOR ROW 4
JB P1.4, NEXT13 ; ROW 4 COLUMN 1
MOV A,#67D ; A = C
RET

NEXT13:JB P1.5,NEXT14; ROW 4 COLUMN 2
MOV A,#48D ; A = 0
RET

NEXT14: JB P1.6,NEXT15 ; ROW 4 COLUMN 3
MOV A,#61D  ; A = '='
RET

NEXT15: JB P1.7,NEXT16; ROW 4 COLUMN 4
```

```
MOV A,#43D ; A = +
RET
NEXT16:LJMP KEY_SCAN ; again check for keys
;---------------------------------------------

COMNWRT:MOV P3,A  ;to send command
CLR P2.0 ; R/s = 0
CLR P2.1 ;R/w =0
SETB P2.2 ;high
ACALL DELAY ; delay
CLR P2.2 ;low
RET

DATAWRT: MOV P3,A  ;to send data
SETB P2.0;R/s= 1
CLR P2.1;R/w =0
SETB P2.2;high
ACALL DELAY; delay
CLR P2.2;low
RET
;-----------------------------------------------
LINE2: MOV A,#0C0H ;move to line 2 of LCD
ACALL COMNWRT
RET

;--------------------------------
DELAY: MOV R3,#65 ; r3 = 65 , m = 1
HERE2: MOV R4,#255 ;r4 = 255 , m =1
HERE: DJNZ R4,HERE ; m = 2
DJNZ R3,HERE2 ;m =2
RET ;m =2
;for here loop , 2 * 255 * 1.085 uS = 553.35 us
;HERE 2 loop repeats HERE loop 65 times then  553.35 us * 65 = 35967.75uS
;mov r4 is also repating 65 times  and djnz r3 too so 3 * 65 * 1.085 us = 211uS
;for return 2 * 1.085 = 2.17uS
;total machine cycle = 35967.75 + 211 + 2.17 = 36180.92 uS
;time delay = 0.036 S

;-----------------------------------------
DELAY2:      MOV R3,#250D ; R3  = 250
        MOV TMOD,#01 ; timer 0 mode 1
BACK2:  MOV TH0,#0FCH
        MOV TL0,#018H  ;initial count value = FC18 is loaded into timer
        SETB TR0 ;starting timer
HERE5:  JNB TF0,HERE5 ;monitor Timer flag if it is 1
        CLR TR0 ; stop the timer
        CLR TF0 ; reset the timer flag
        DJNZ R3,BACK2 ; repeat this process 250 times
        RET
;COUNT = 65535 - 64536 + 1 = 1000
```

```
; 1000 * 1.085 uS = 1085 uS
; 1085uS * 250 = 0.271 S
;---------------------------------------------

DELAY3:MOV TMOD,#10H ;Timer 1, mod 1
MOV R3,#70 ; for multiple delay
AGAIN1: MOV TL1,#00H ;TL1=08,low byte of timer
MOV TH1,#00H ;TH1=01,high byte , TIMER = 0000
SETB TR1 ;Start timer 1
BACK: JNB TF1,BACK ;until timer rolls over
CLR TR1 ;Stop the timer 1
CLR TF1 ;clear Timer 1 flag
DJNZ R3,AGAIN1 ;if R3 not zero then
RET
;COUNT = 65535 - 0000 + 1 = 65536
;65536 * 1.085 uS = 71.1065mS
;71.1065 mS * 70 = 4977.45mS = 5S

;------------------------------------------
CLRSCR: MOV A,#01H
ACALL COMNWRT
RET
;----------------------------------------
ORG 500H
DB
10000000B,01000000B,11000000B,00100000B,10100000B,01100000B,11100000B,00010000B,001100
00B
MYDATA: DB 38H,0EH,01,06,80H,0;
;initializer 5 X 7 MATRIX lcd
;display on cursor blinking
;clear the display screen
;cursor shift --> towards right
;start from the first line
INITIAL_MSG:   DB "ENTER 5-DIG.CODE",0
CHECK_CODE_MSG:  DB "CHECKING CODE...",0
PASSWORD:DB 49D,50D,51D,52D,53D,0  ;PASSWORD = 1 2 3 4 5
TEXT_F1: DB "WRONG CODE",0
TEXT_F2: DB "ACCESS DENIED",0
TEXT_S1: DB "ACCESS GRANTED",0
TEXT_S2: DB "OPENING DOOR",0
TEXT_S3: DB "CLOSING DOOR", 0
ALERT_TEXT: DB "INTRUDER ALERT !",0
ATTEMPT_TEXT: DB "ATTEMPTS LEFT:0",0
LOCKDOWN_TEXT: DB "LOCKDOWN STARTED",0
POLICE_TEXT: DB "CALLING POLICE!",0
END
```

5. Code of Arduino –

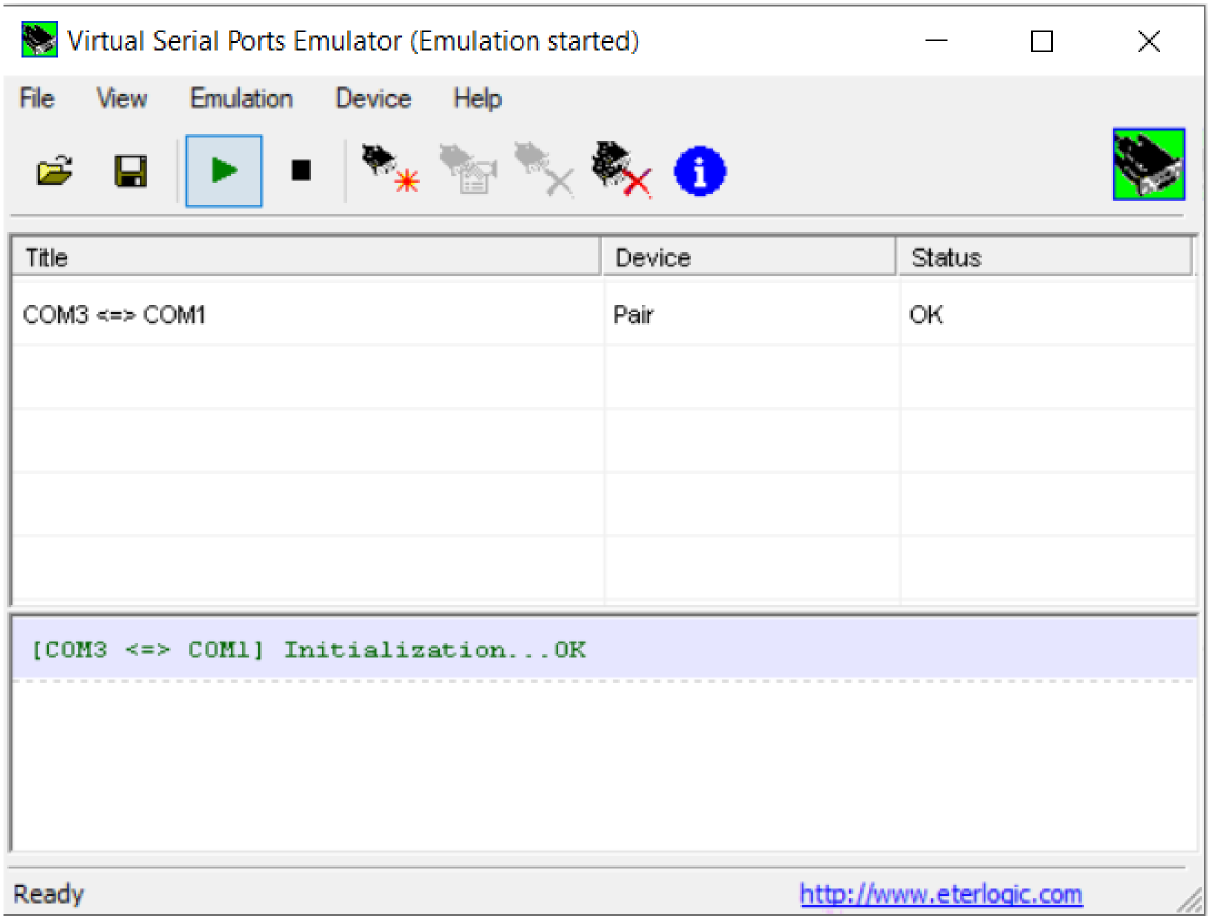
```
#define BLYNK_PRINT DebugSerial
#include <SoftwareSerial.h>
SoftwareSerial DebugSerial(2, 3); // RX, TX
#include <BlynkSimpleStream.h>
char auth[] = "Ri25KxPoXT36A9rPBjxpm0BGT4mn5u_E";
const int buttonPin = 12;     // the number of the pushbutton pin
const int ledPin =  13;
int i = 0;
int buttonState = 0;

void setup()
{
  DebugSerial.begin(9600);
  Serial.begin(9600);
  Blynk.begin(Serial, auth);
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  Blynk.run();
  buttonState = digitalRead(buttonPin);
    if (buttonState == HIGH) {
      if (i == 0)
      {
        Blynk.email("arpitpatawat2000@gmail.com", "Intruder Alert", "someone has tried to
open door!. this is a Security Email.");
        digitalWrite(ledPin, HIGH);
        i = 1;
      }
        }
    else if(buttonState == LOW){
      digitalWrite(ledPin, LOW);
      i = 0;
      }
}
```

Project demonstration →

Creating a connection between original port COM3 and virtual port COM1

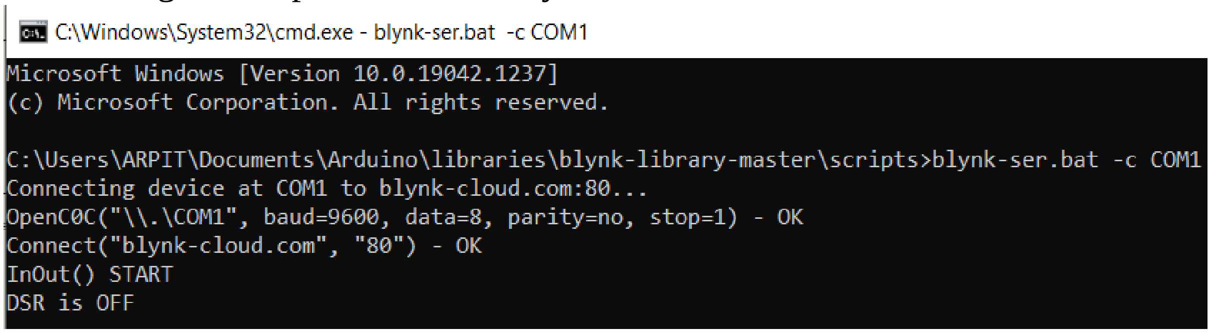

Virtual Serial Ports Emulator (Emulation started)
File View Emulation Device Help

Title	Device	Status
COM3 <=> COM1	Pair	OK

```
[COM3 <=> COM1] Initialization...OK
```

Ready
http://www.eterlogic.com

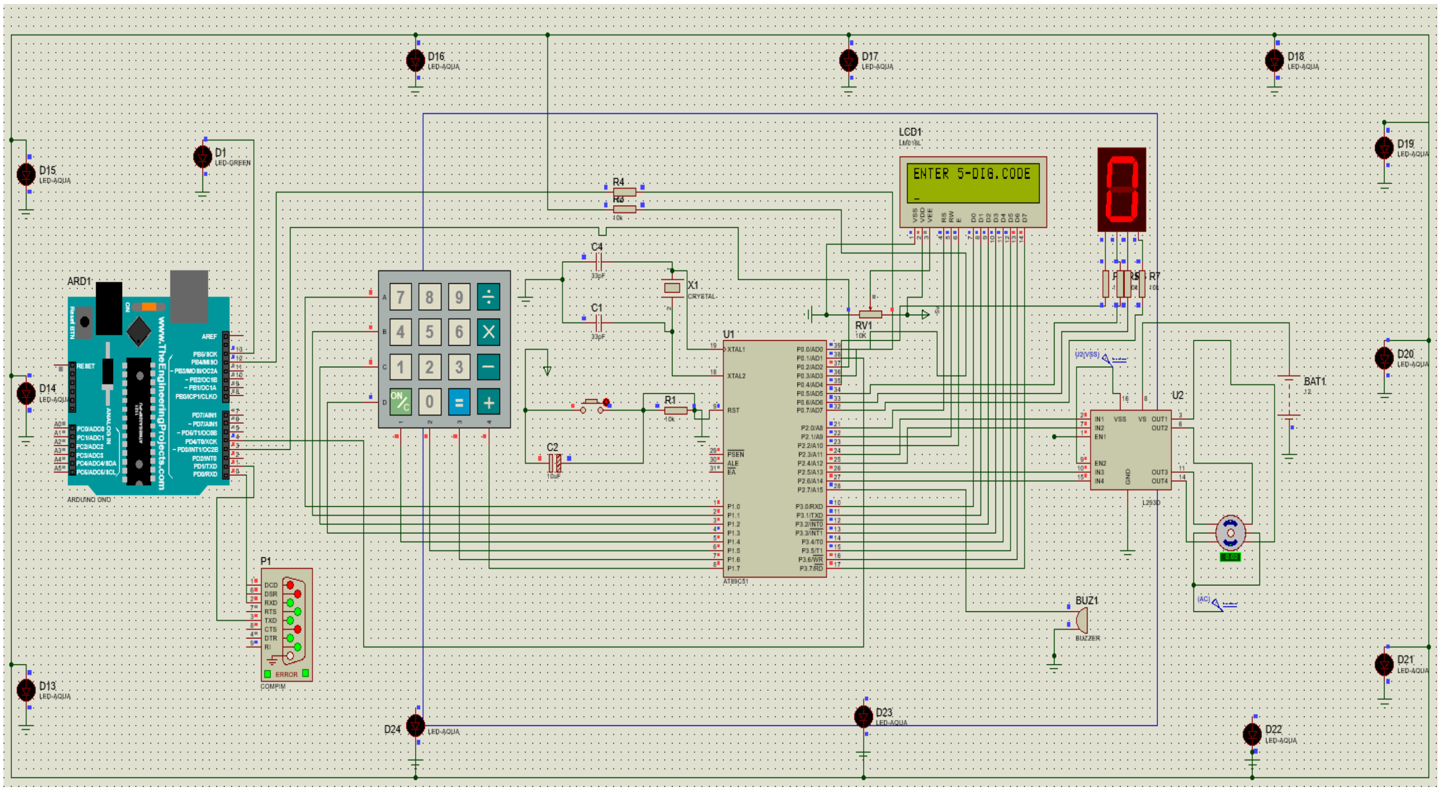
Connecting virtual port COM1 to blynk server



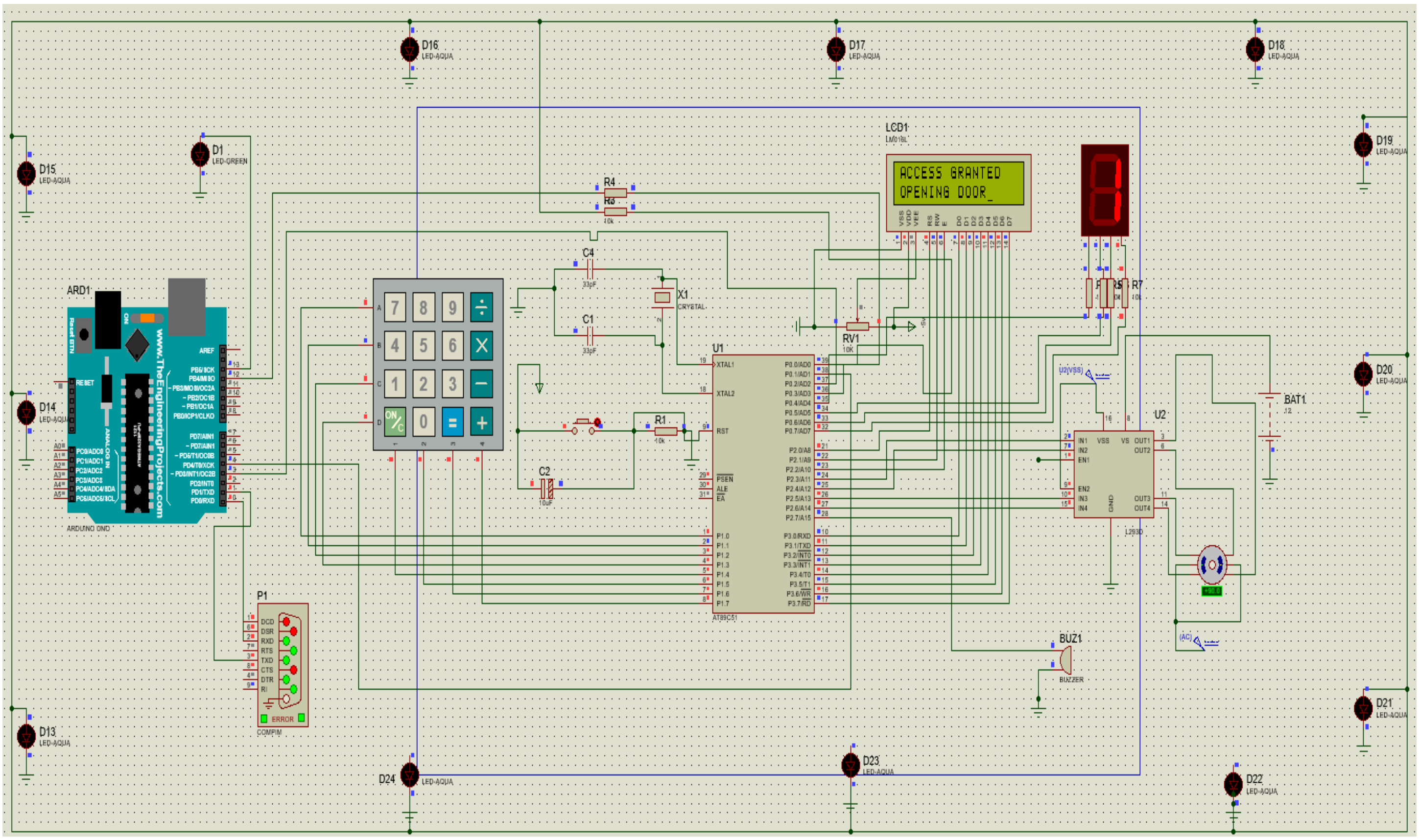
```
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ARPIT\Documents\Arduino\libraries\blynk-library-master\scripts>blynk-ser.bat -c COM1
Connecting device at COM1 to blynk-cloud.com:80...
OpenC0C("\\.\COM1", baud=9600, data=8, parity=no, stop=1) - OK
Connect("blynk-cloud.com", "80") - OK
InOut() START
DSR is OFF
```

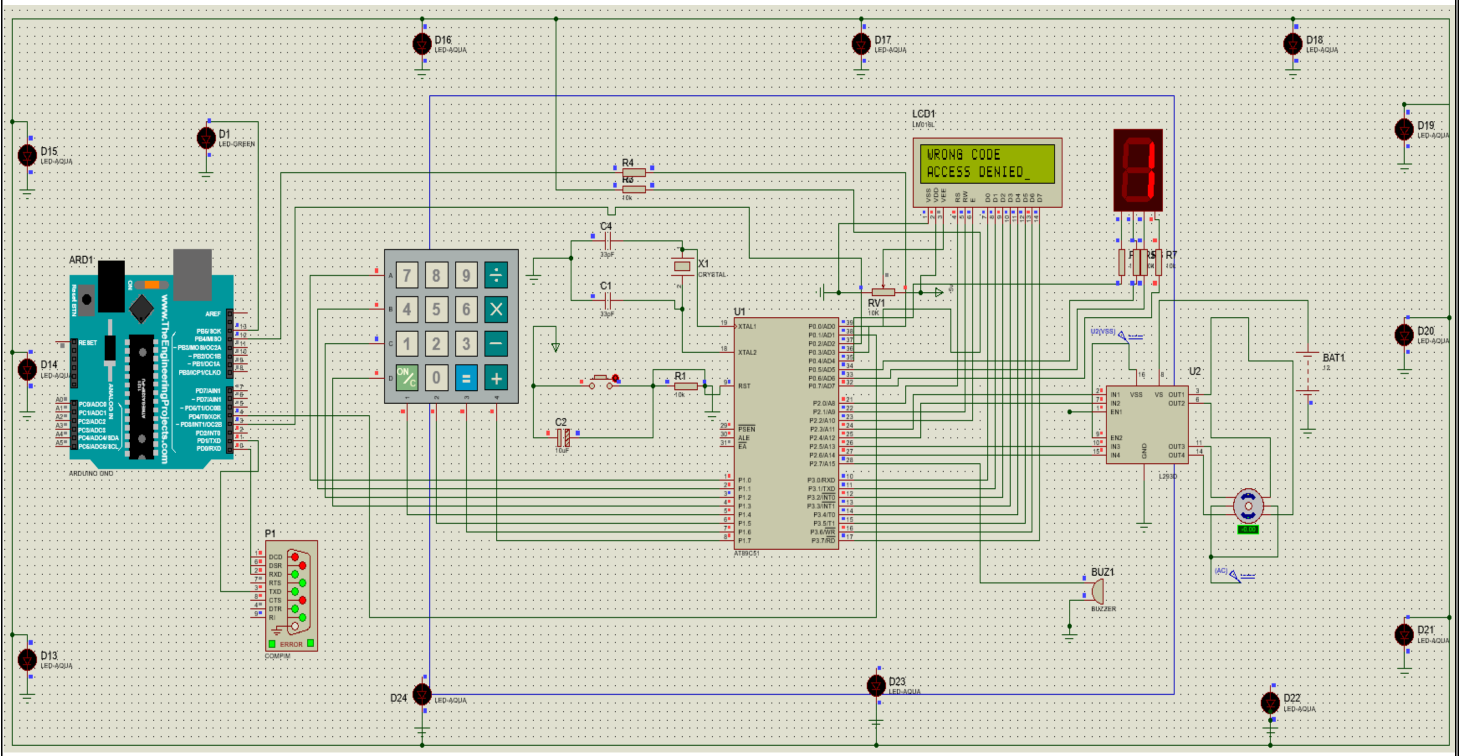
when program started →



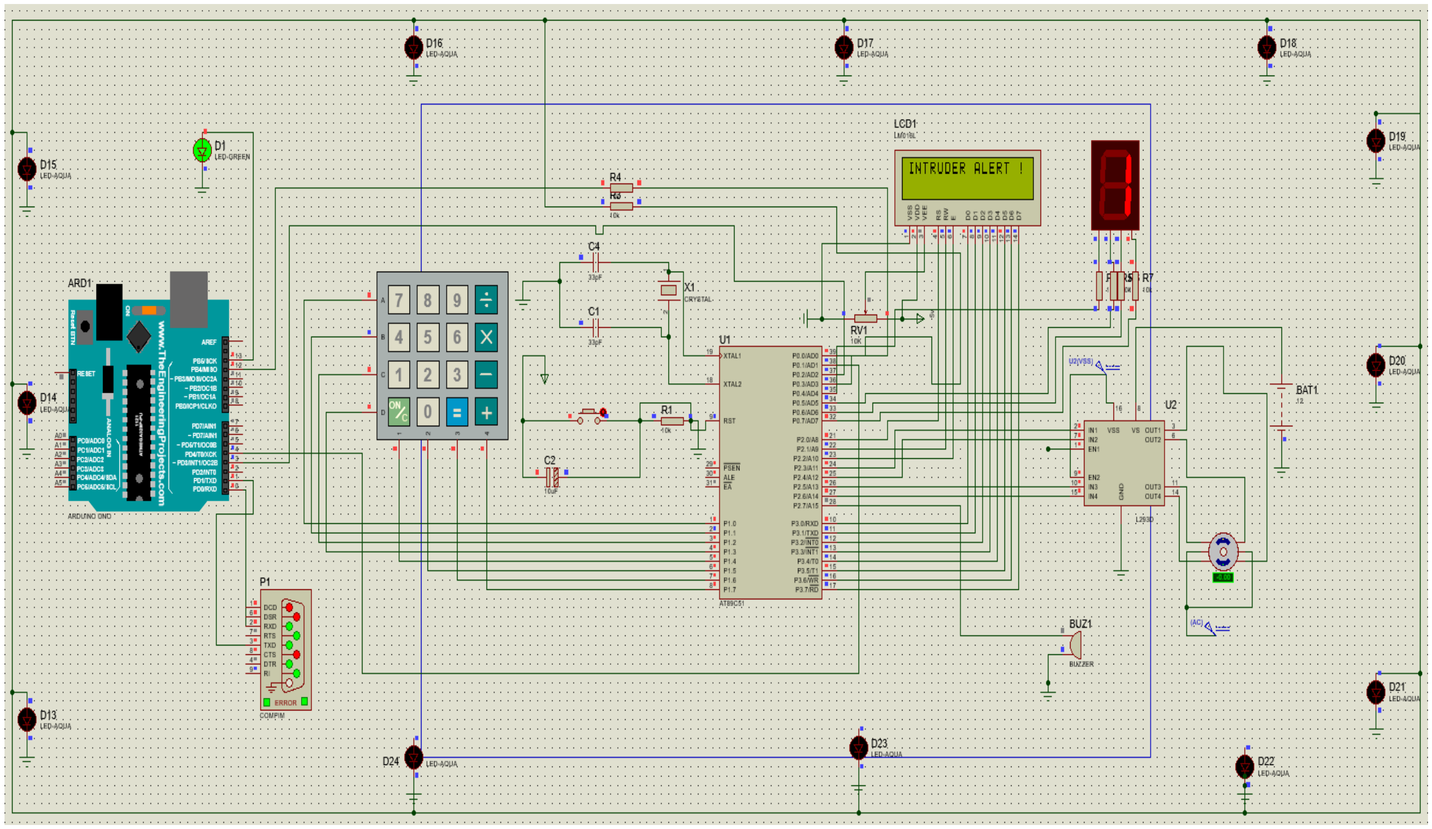
when correct password is given →



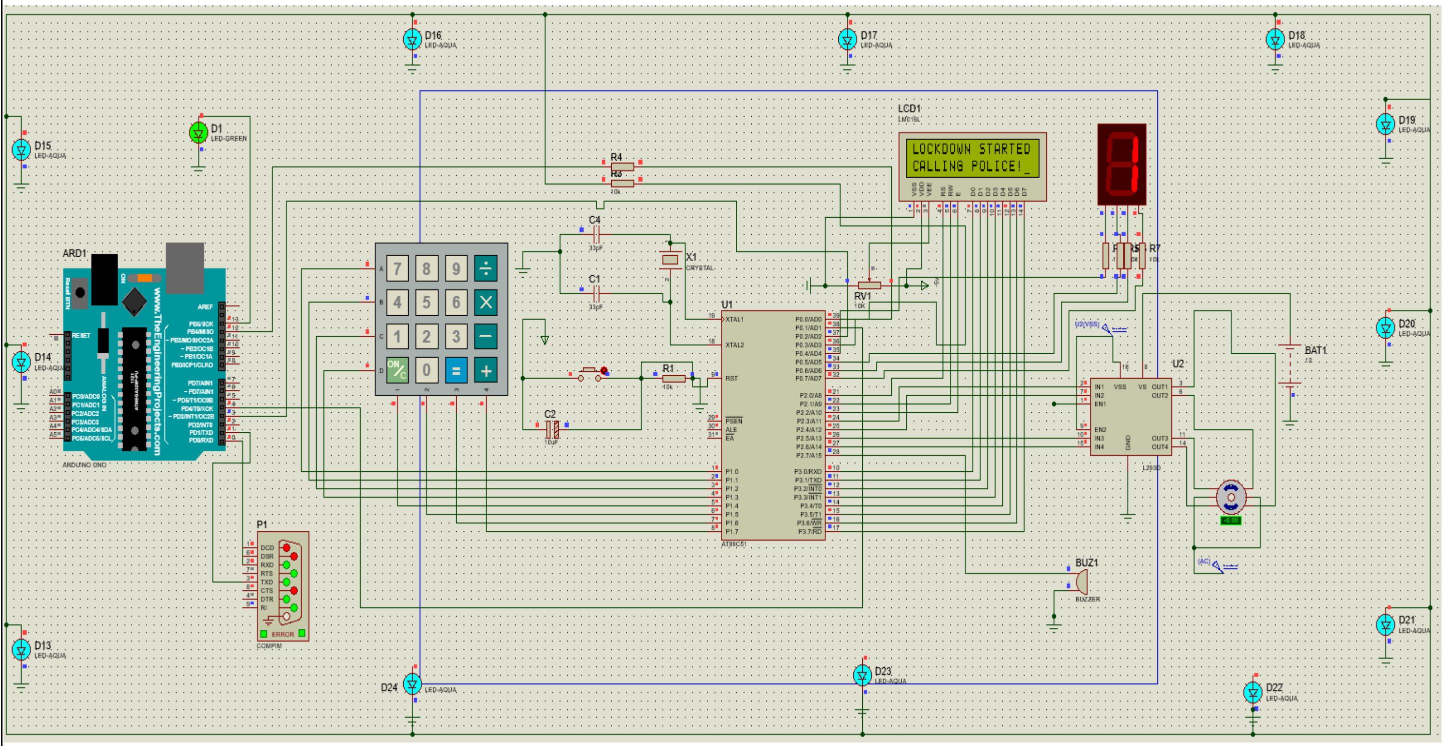
when 1-time wrong password is given →



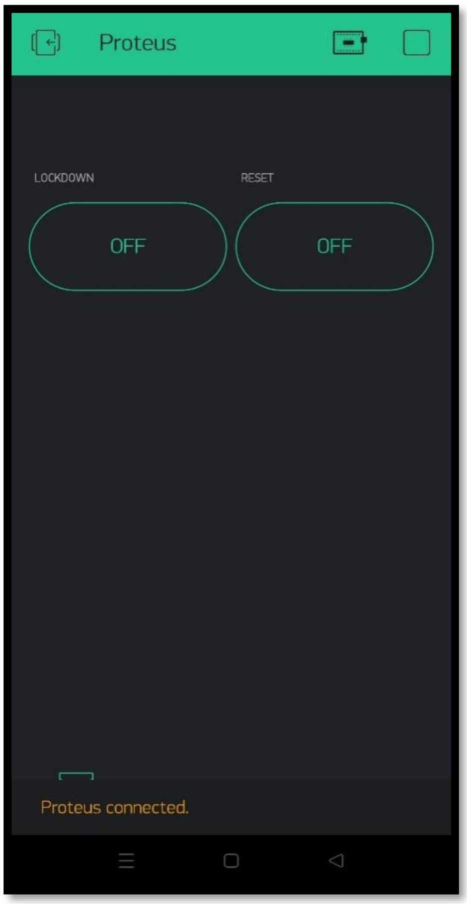
when someone tried to give wrong password 3 time →



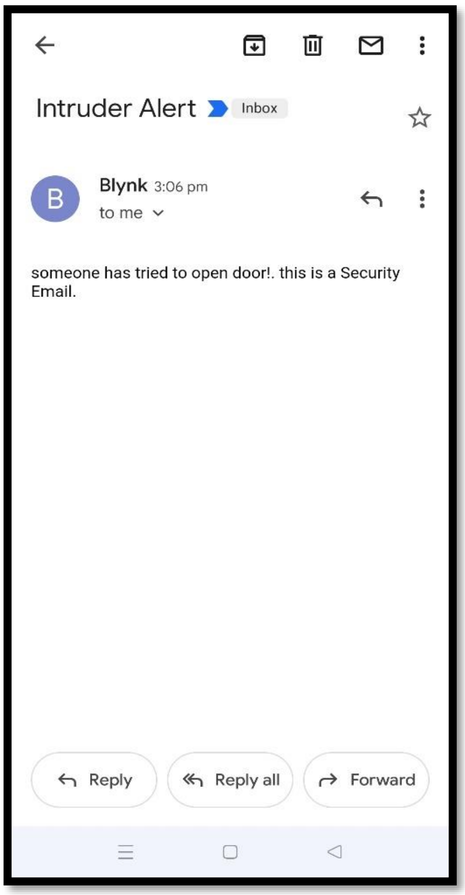
When lockdown is pressed →



App screenshot and mail screenshot →



Proteus connected.



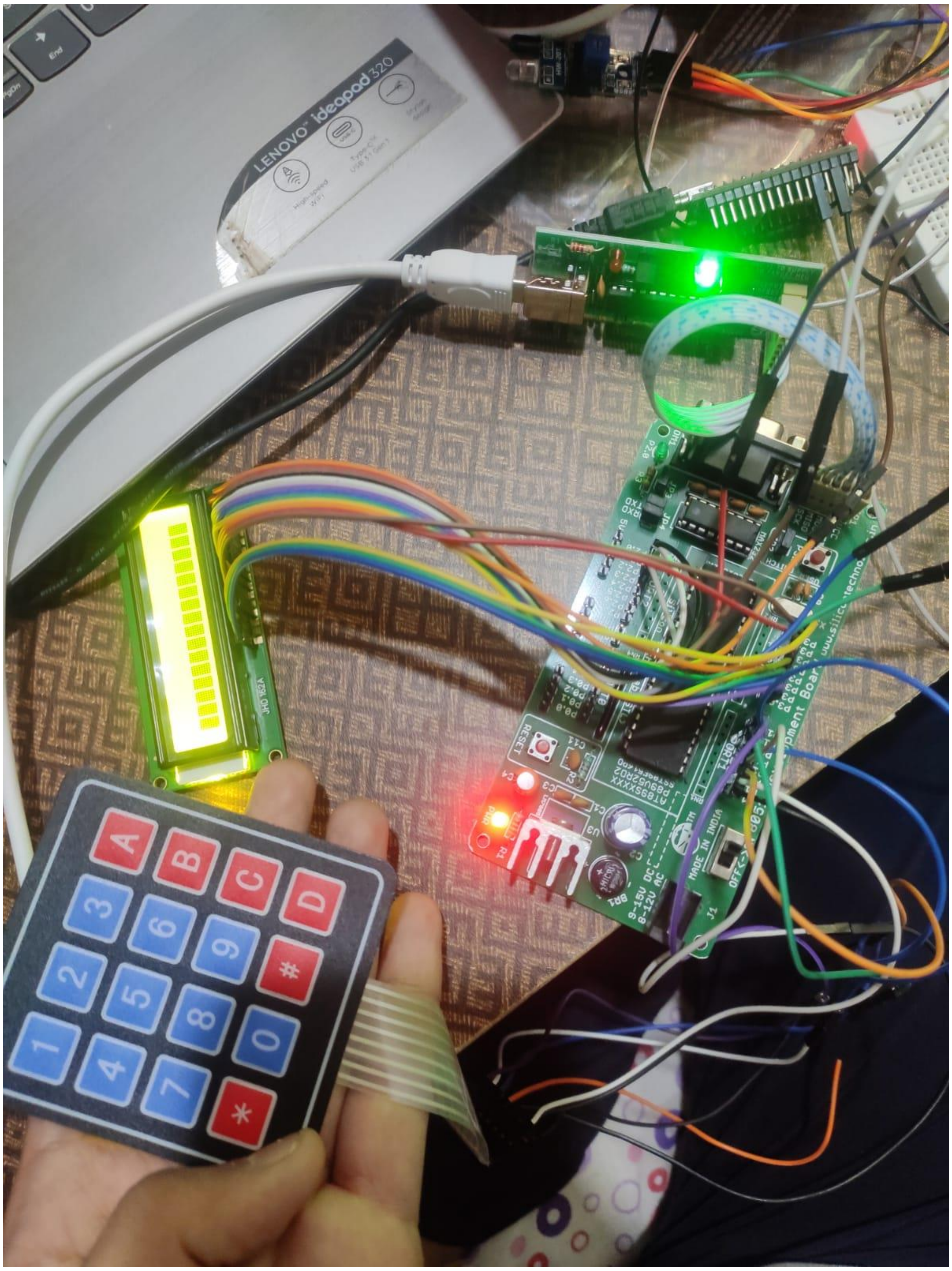
Intruder Alert

Blynk 3:06 pm

to me

someone has tried to open door!. this is a Security Email.

Hardware implementation →



Conclusion →

In future, we can increase the security level by adding a biometric fingerprint scanner. We can interface sensors like Fire, LPG, PIR motion detector to microcontroller in case of any accident so that door will open automatically. Also, we can interface camera to the microcontroller so that it could capture the picture of the thief who is trying to breach the security.

Our project is productive in providing enough security as long as the password is not shared. In future this “Password based Door Lock System” can be provided maximum security by the above enhancements in order to completely satisfy user’s needs. Hence, a common man can afford to buy such locking system in minimal cost to keep his valuables safely without any worries.

Advantages:

1. Components are easily available.
2. Our project is simple and easy.
3. Power consumption is less.
4. It provides security.
5. The user need not to carry the keys along with him, he can just remember the password and use it later to open the door.
6. Also, the e-mail alert is a great security notification which can save from robbery.

Applications:

1. It can be used at organisations to ensure authorised access to highly secured places.
2. It can be used in residential places to ensure better safety.
3. It can also be used in ATM, offices, classrooms to ensure safety.

Limitations:

1. It should connect with wi-fi in order to send and read information.
2. If we forget the password, it is not possible to open the door.
3. If there is no charge left in the battery, then our system doesn’t work.

Please visit my work at - https://github.com/arpitpatawat/password-based-door-lock-system-using-8051-microcontroller-ALP and star it.

---XXX-------------------------------------