Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE ENGINEERING

KNOCK DETECTION LOCK SYSTEM WITH SANITIZING DISPENSER

By
Anuska Rakshit - 19BCB0106
Pallavi Mishra - 19BCB0119
Mansi Raturi - 19BCE0488

Project Report
of
CSE2006 – MICROPROCESSOR & INTERFACING

Fall Semester 2021-22

DECLARATION

I hereby declare that the report entitled "KNOCK DETECTION LOCK SYSTEM WITH SANITIZING DISPENSER" submitted by us, for the CSE2006 Microprocessor and Interfacing (EPJ) to Vellore Institute of Technology is a record of bonafide work carried out by me under the supervision of Dr. Manish Kumar.

I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for any other courses in this institute or any other institute or university.

- ANUSKA RAKSHIT 19BCB0106
- PALLAVI MISHRA 19BCB0119
- MANSI RATURI 19BCE0488

ACKNOWLEDGEMENT

With due regards, we would like to thank our faculty Prof. Dr Manish Kumar for constantly supporting us in making our project work. Without her much needed support, it would have been impossible for us to carry out our work. Her guidance laid the foundation of our project.

At last, we would also like to thank VIT University for providing us the proper ambience to carry out our project work successfully. We even learned to carry out a project in a group by dividing the workload.

CONTENTS

ABSTRACT

In our day-to-day lives, security has always been a major concern. Whether it's the basic entryway lock, interconnecting door lock, restroom door lock, furniture lock, baggage lock, shutter, and grill door lock, and so on, locks are a part of it. Modern locks and locking systems are significantly more complicated, and they frequently include a dotted mechanism on the key to increase security. However, the key can always be copied with some effort, which is a drawback. One option is to do away with the 'lock and key' method altogether. This project seeks to accomplish the same goal by providing robust security.We have proposed to create a Secret Knock Detecting Door Lock in this project.The main purpose is to develop a security mechanism that enables the user to unlock the door just by giving a certain number of knocks on the surface. As there is no key to be replicated, this technique totally eliminates the concern of duplication. In this way, only those with known knock patterns can enter the house.

Using Arduino to detect the pattern of your knocks at the door and only open the lock if the pattern is recognised. The pattern of knocking corresponds to the correct pattern. Our project is based on a custom-built Arduino whose primary purpose is to reduce the cost of the final product and the time it takes to produce it.The Arduino programming condition is used to create an algorithm for detecting knock patterns. We've got includes a reset button as well as two status LEDs that are useful for task testing and new knock patterns joining or registering When the user knocks, a custom Arduino board is activated.Organizes the the knock design with the knocking algorithm and opens the door if the knock pattern matches with the example registered.

1. INTRODUCTION

Security is a major concern in our day to day life, and digital locks have become an important part of these security systems. There are many types of security systems available to secure our place. Some examples are PIR based Security System, RFID based Security System, Digital Lock System, bio-matrix systems, Electronics Code lock etc. We propose to build a smart door security system to increase public safety from intruders. This is implemented using an Arduino board. It is designed to prevent unauthorized access, trespassing and intrusion. The motivation behind this project was

- To make a locking system which is cost effective
- To provide a secure locking system without the need of a physical key.
- Easy authentication app portal which everyone with a smartphone can utilize.
- Accessible by only authorized people and notifies all the users when the lock is opened(either when an authorized person enters or god forbid if there is a break into the house).

2. LITERATURE REVIEW

Serial No	Title	Journal Name & Year	Author Name	Summary
1	IoT and Wi-Fi Based Door Access Control System using Mobile Application	ResearchGate, 2019	Rehnuma Reza Deepty; Albina Alam; Md. Ezharul Islam	An IoT and Wi-Fi based door access control system, where a user can control the access to the door using his phone remotely within the coverage area provided by the wireless Access-Point that is situated inside the home.Raspberry Pi is used as the door access controller and other state-of-the art devices to create testbed of the system and verified its use in practical environment.
2	Security System Based on KnockPattern Using Arduino and GSM Communication	International Journal of Engineering and Techniques - Volume 4 Issue 1, Jan – Feb 2018	- R.SaiCharan Reddy	A smart security system with the use of 'Arduino' microcontroller, PiezoElectric and GSM Module is developed. Here, the security system is based on a "secret knocking pattern" which can be installed to a ‘safe’ or any other similar object which needs protection. The lock unlocks only when a certain secret knocking pattern is implemented and an SMS alert is sent if anyone tries to sneak into contents by knocking differently.

				This concept eliminates the fear of replication as there is no physical Unlocking Object to start with. Thus, the smart 'Knock Based Security System' can be an added protection in our everyday lives. Arduino Uno board which act as a microcontroller unit. The Piezo sensor takes the knocking input and then passes it to the Arduino board where the input pattern is compared with the original Secret pattern. In case of wring pattern, the GSM sends an alert to the given Phone Number.
3	Smart Home Management Using Wireless Sensor Network,	International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)	Ms.Jayshri V.Ekshinge Dr. Santosh S. Sonavane	In this paper, a low-power consumption remote home security alarm system developed by applying WSN and GSM technology is presented. It can detect theft, leaking of raw gas and fire, and send alarm messages remotely. The hardware of this system includes the single chip C5081F310, wireless receiving and sending chip CC1100 as well as the SIMENS TC35 GSM module. The system software developed in C51 language has the ability

				of collecting, wireless receiving and sending data, and can send a piece of alarm short message to the user's mobile phone when some dangerous condition has been detected
4	Intelligent Lock Applied for Smart Door	International Journal of Computer Science and Information Security(IJCSIS), Vol. 17, No. 6, June 2019	Elshaimaa Nada, Sarah Aljudaibi, Abrar Aljabri, Hafsa Raissouli	Use knocking patterns to identify and allow only authorised users. In case the pattern is forgotten, using a login-password to unlock the door.
5	SMART KNOCK DETECTING DOOR LOCK USING ARDUINO	International Research Journal of Modernization in Engineering Technology and Science (2021)	Shaik Ateeq Ur Rahman, Jinesh Thakker, M.Sai Keerthana	Technological approach for implementation of door unlocking system through partial technical and manual handling. This design has many benefits like reasonable cost, enhanced security, and small size. This system uses the microcontroller in conjunction with sensor, buzzer and servo motor.
6	RFID-Based Digital Door Locking System	Indian Journal of Microprocessors and Microcontroller (IJMM) ISSN: 2582-8835 (Online), Volume-1 Issue-	Shubham Soni, Rajni Soni, Akhilesh A. Waoo	A servo motor is used that is controlled by an Arduino. The Arduino board operates with all of the code saved on it. LCD, RFID Module LED Lights in Red, Green, and Yellow, &Buzzer, have been

		2, September 2021		used. When the door is locked, the yellow LED will be on and when the door lock is open, the green led light will turn on. The Red LED work will alert you that your card is wrong.

3. DRAWBACK IN THE EXISTING WORK

Our world is rapidly developing in technology. All the things around us are being digitized and automated. Smart Home is the term commonly used to design a residence that uses a home controller to integrate the residence's different automation system. Hence, aim at creating a low-cost, smart lock system using Arduino uno.

Need for the system:

- To simplify home security
- To increase accessibility without compromising security.
- To increase connectivity
- Remote security

4. PROPOSED WORK

Considering the above problems, we propose a smart knock detecting door using Arduino which suits optimum with regards to cost, efficiency and security as per our knowledge and observation. This device will be installed on the door and will detect any discrepancy or unknown pattern if recorded. Door opens only when the knocked pattern matches with the one updated in the system by the rightful person. This pattern along with the buffer period can be updated by the owner as and when required. The device isn’t connected to a loud alarm when the wrong pattern is knocked, keeping in mind the noise and disturbance it could cause to neighbours and also because it might happen the user himself knocks incorrectly by mistake.

4.2 NOVELTY

Focusing on the COVID-19 crisis we have decided to add a new feature to our knock safe system by attaching Automatic Sanitizer Dispenser. Before using the lock the person has to first get his hands sanitized. Only if the Dispenser sensor has worked the system will lead the person to input the pattern in the lock. In case he fails to sanitize his hands a message would be displayed on the screen and won’t be allowed to use the safe system.

The photodiode i.e the Light sensor detects light difference and thus activates the system.

5. ARCHITECTURE

5.1 COMPONENTS

1. Arduino UNO - The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output pins that may be interfaced to various expansion boards and other circuits.
2. Piezo sensor - A piezoelectric sensor is a device that uses the piezoelectric effect to measure changes in pressure, acceleration, temperature, strain or force.
3. Photodiode - Light Sensor Module Detects light and outputs digital and analog signals. The digital signal can have a trigger level. LEDs indicate power and output signal.
4. Breadboard
5. LED’s
6. Micro Servo
7. Capacitor
8. Pushbutton
9. Resistors

Why Use Piezoelectric Sensors?

A piezoelectric sensor is a device that uses the piezoelectric effect to measure changes in pressure, acceleration, temperature, strain, or force by converting them to an electrical charge. The prefix piezo- is Greek for 'press' or 'squeeze'

The Piezo sensor takes the knocking input and then passes it to the Arduino board where the input pattern is compared with the original Secret pattern.

The advantages of piezoelectric sensors are

- wide frequency bandwidth
- high sensitivity
- high signal-to-noise ratio
- simple structure
- reliable operation
- light weight.

Why use Servo Motor?

They provide high levels of torque at high speed – something stepper motors can't do.

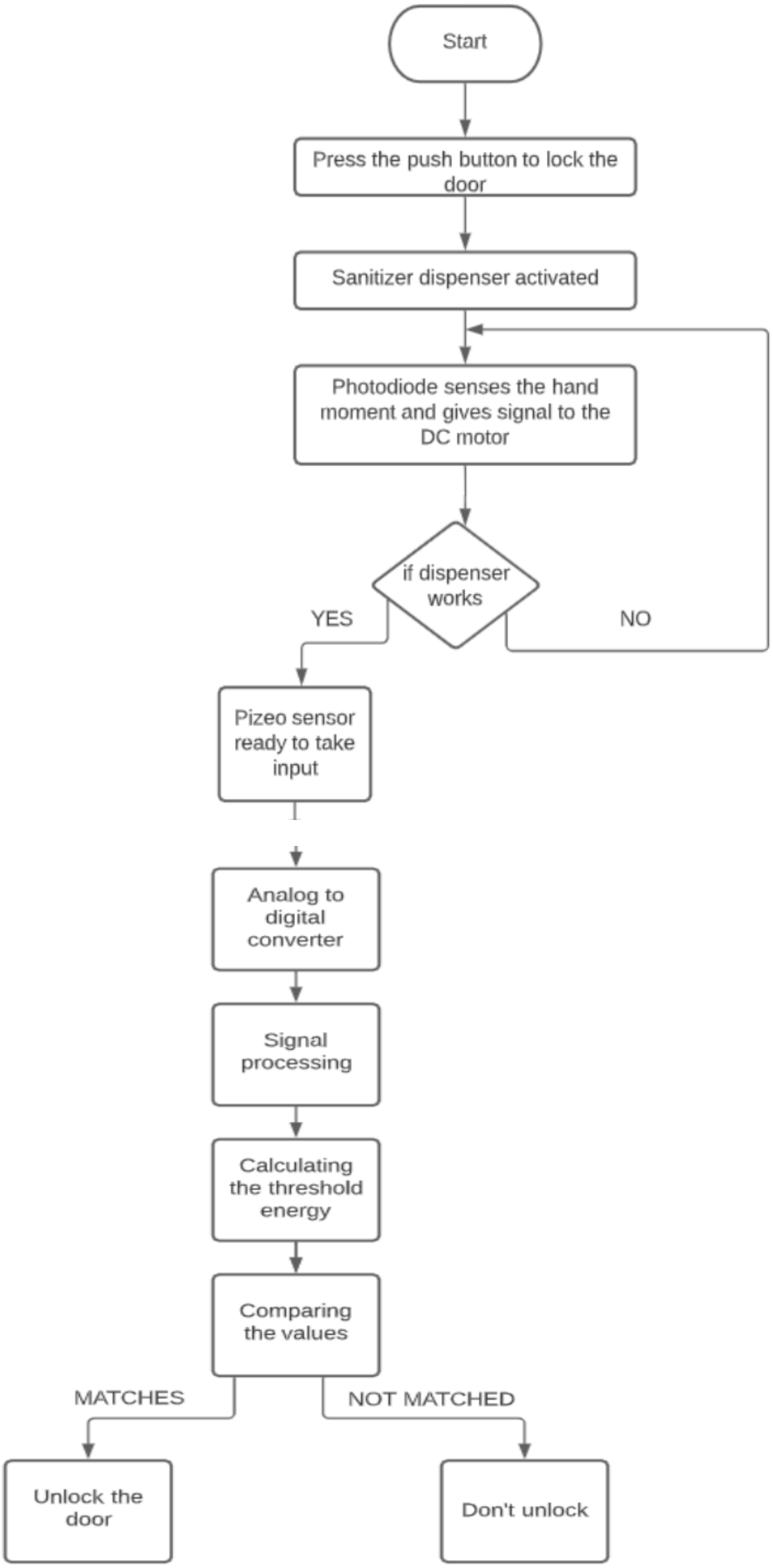
- They also operate at 80 – 90% efficiency
- Servo motors can work in AC or DC drive, and do not suffer from vibration or resonance issues

Why use Arduino Uno?

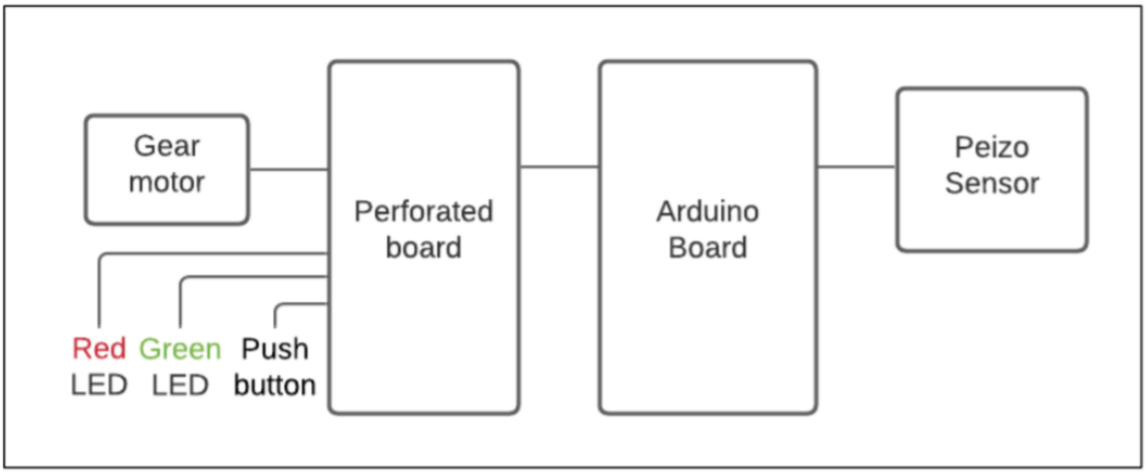
Arduino is an open-source electronics platform that uses simple hardware and software to make it easy to use. It has a microcontroller board based on ATmega 3287P. It has 14 digital I/O pins, 6 analog inputs, a 16 MHz ceramic resonator, USB connection, power jack, 1CSP header, and a reset button. Arduino boards can take inputs - such as light from a sensor, a finger on a button, or a Twitter tweet - and convert them to outputs - such as turning on an LED, triggering a motor, or posting anything online.

Arduino UNO is preferred over Raspberry PI as Raspberry PI does not support analog inputs which is an important aspect in this project.

5.2 FLOW DIAGRAM:


Start
Press the push button to lock the door
Sanitizer dispenser activated
Photodiode senses the hand moment and gives signal to the DC motor
if dispenser works
YES
NO
Pizeo sensor ready to take input
Analog to digital converter
Signal processing
Calculating the threshold energy
Comparing the values
MATCHES
NOT MATCHED
Unlock the door
Don't unlock

5.3 BLOCK DIAGRAM:



6. IMPLEMENTATION:

6.a Sensor

We have used PIEZO SENSOR to detect the duration of knocks between each knock. The piezo sensor has electric plates which measure changes in pressure, acceleration, strain or force by converting them to electric charge. One can measure the smallest electric charge deviation caused by rubbing, static charge or any other charge source. We have used this concept to detect Knocks. Also, we have increased the threshold value to 100 so that it doesn't record unnecessary noise signals.

6.b Execution

The time of knocks between each knock was detected using a PIEZO SENSOR. Electric plates in the piezo sensor translate changes in pressure, acceleration, strain, or force into an electric charge. The tiniest electric charge variance generated by friction, static charge, or any other charge source may be measured. This idea has been used to detect knocks. We've also upped the threshold value to 100 to prevent it from recording unwanted noise signals. Basic Procedures:

1- Define the unlockable zones.

2- Gathering knocking data – A piezo sensor gathers vibration (analogue) signals.

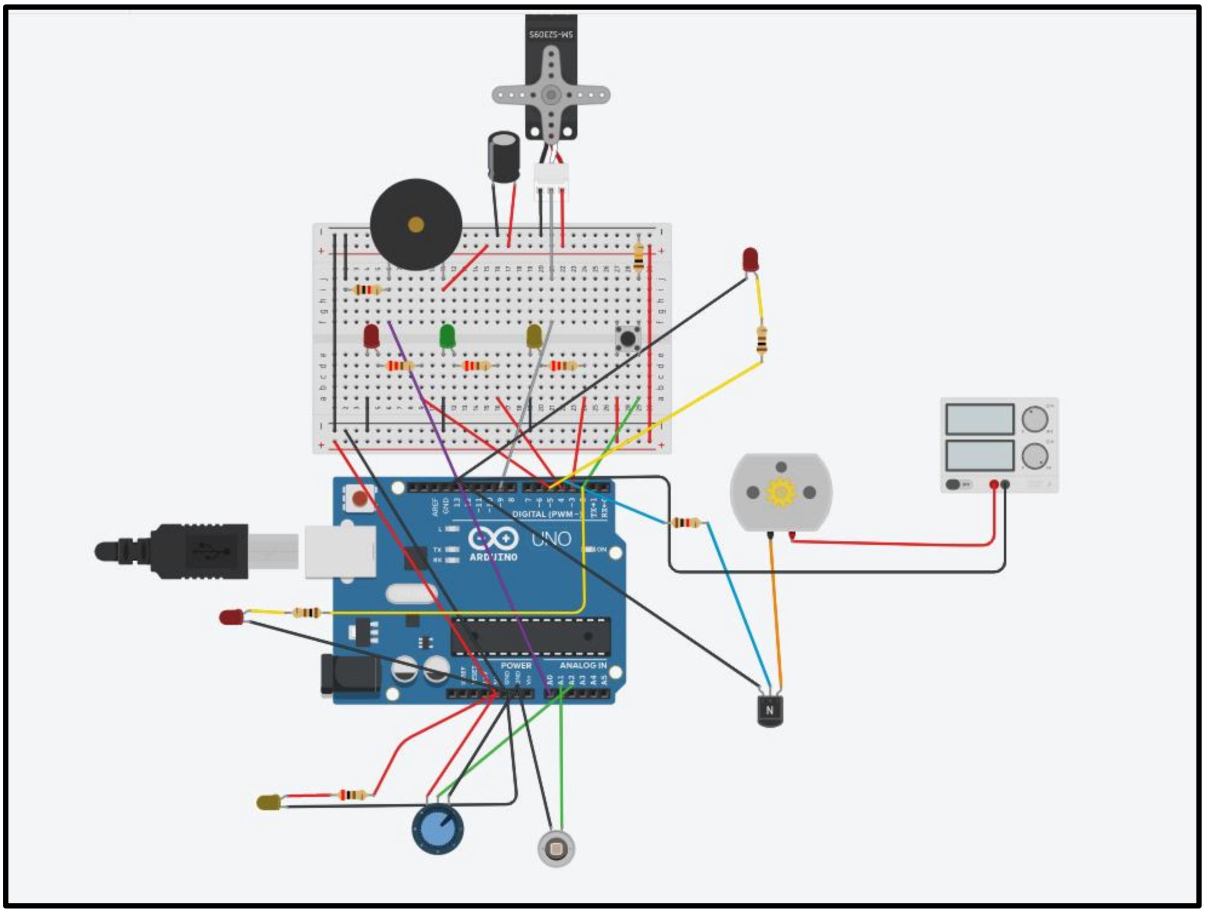
3- Getting the signal to the ARDUINO - The analogue signals from the sensors are received by the ARDUINO, which uses an analogue to digital converter (ADC) to power the actuator, which opens the door by transforming the signal's energy into mechanical motion. The shine of LEDs linked to the PCB gives away the effects.

6.c Sanitizer dispenser

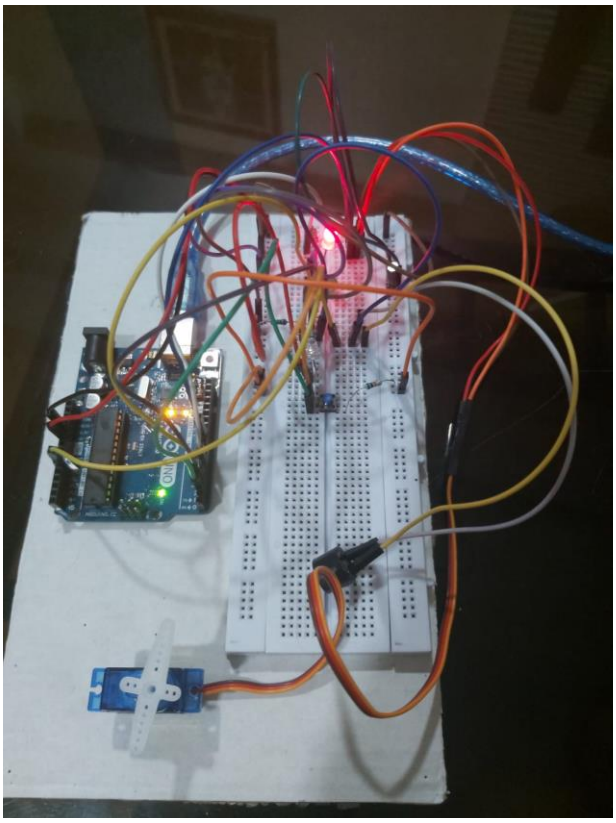
A sanitiser dispenser is added to the knock detection system in the project. So before knocking the door, the person is required to sanitize their hands by placing it in front of a photodiode sensor that detects its presence and sends the input to the arduino board. upon detection, the servo motor automatically pulls the faucet on the sanitizer to clean the hands.

7. SCREENSHOTS OF THE PROTOTYPE

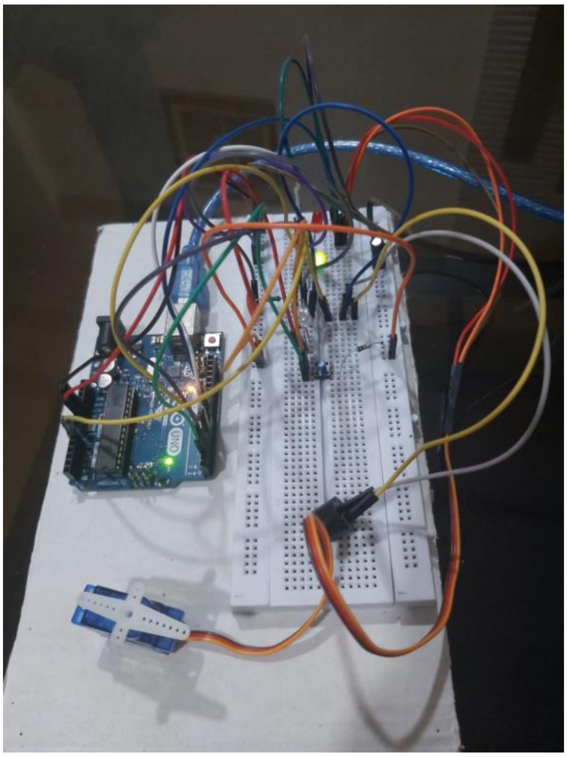
7.1 Software Simulation (Tinkercad)



7.2 Hardware implementation



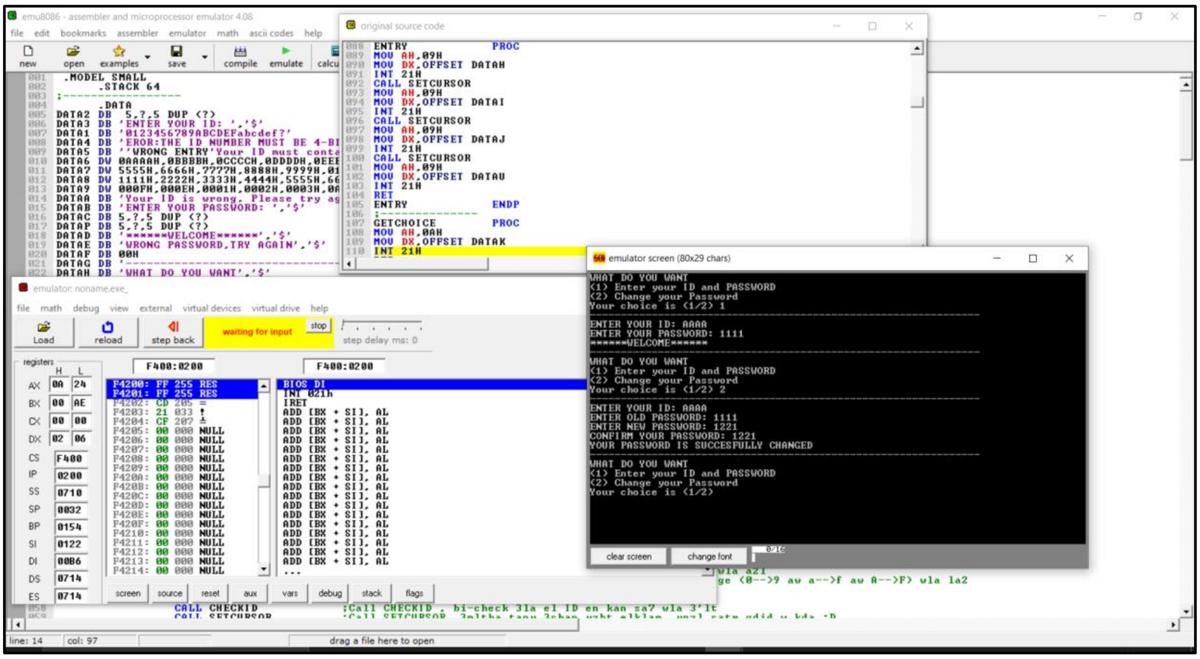
Door in the locked state(Red LED glows)



Door in unlocked state (Green LED glows)

The main objective of the program is to turn 90º the servomotor and turn the red light on when the button is pressed. Then, it will start the knock countdown. Each time the buzzer detects a knock the yellow light will blink. When the system has detected 3 knocks will return the servo to its natural position and turn the green light on.

7.3 ALP Program Execution



emu8086 - assembler and microprocessor emulator 4.08

```
 .MODEL SMALL
     .STACK 64
;-----------------
     .DATA
DATA2 DB  5,?,5 DUP (?)
DATA3 DB 'ENTER YOUR ID: ','$'
DATA1 DB '0123456789ABCDEFabcdef?'
DATA4 DB 'EROR:THE ID NUMBER MUST BE 4-BI
DATA5 DB ''WRONG ENTRY'Your ID must conta
DATA6 DW 0AAAAH,0BBBBH,0CCCCH,0DDDDH,0EEE
DATA7 DW 5555H,6666H,7777H,8888H,9999H,01
DATA8 DW 1111H,2222H,3333H,4444H,5555H,66
DATA9 DW 000FH,000EH,0001H,0002H,0003H,0A
DATAA DB 'Your ID is wrong, Please try ag
DATAB DB 'ENTER YOUR PASSWORD: ','$'
DATAC DB 5,?,5 DUP (?)
DATAP DB 5,?,5 DUP (?)
DATAD DB '******WELCOME******','$'
DATAE DB 'WRONG PASSWORD,TRY AGAIN','$'
DATAF DB 00H
DATAG DB '---------------------------------
DATAH DB 'WHAT DO YOU WANT','$'
```

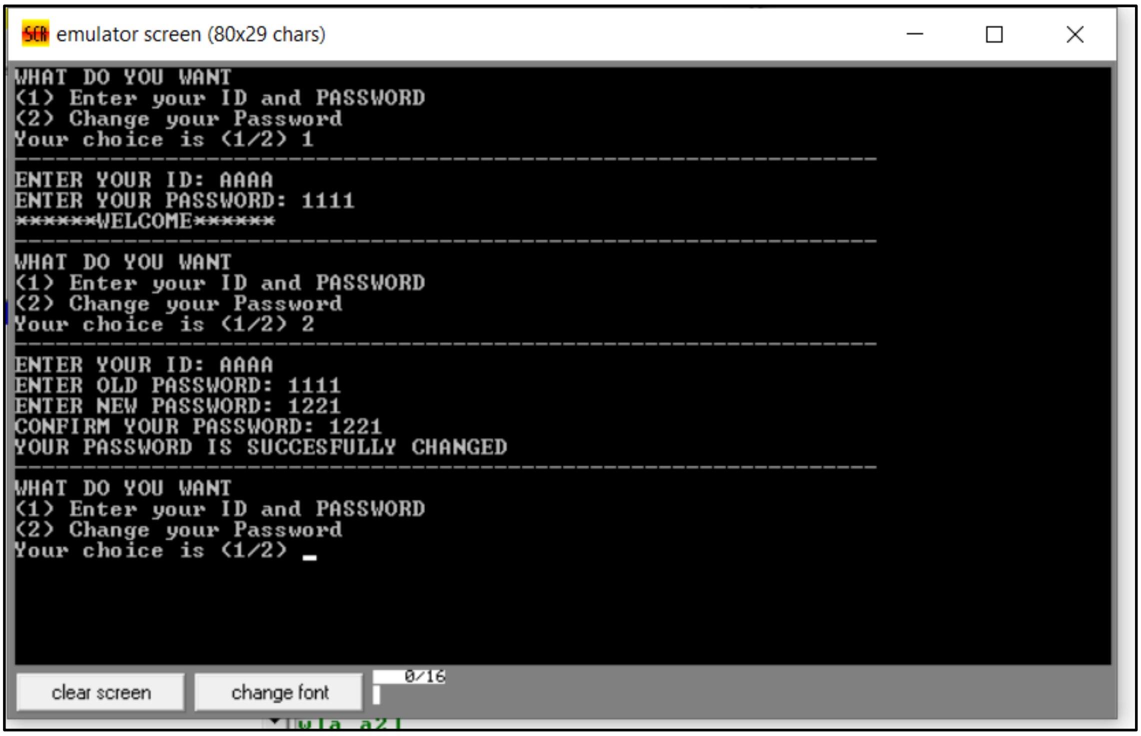
original source code

```
ENTRY              PROC
MOV AH,09H
MOV DX,OFFSET DATAH
INT 21H
CALL SETCURSOR
MOV AH,09H
MOV DX,OFFSET DATAI
INT 21H
CALL SETCURSOR
MOV AH,09H
MOV DX,OFFSET DATAJ
INT 21H
CALL SETCURSOR
MOV AH,09H
MOV DX,OFFSET DATAU
INT 21H
RET
ENTRY              ENDP
;---------------
GETCHOICE          PROC
MOV AH,0AH
MOV DX,OFFSET DATAK
INT 21H
```

emulator: noname.exe_ — waiting for input

```
registers
    H  L
AX  0A 24
BX  00 AE
CX  00 00
DX  02 06
CS  F400
IP  0200
SS  0710
SP  0032
BP  0154
SI  0122
DI  00B6
DS  0714
ES  0714

F400:0200                  F400:0200
F4200: FF 255 RES          BIOS DI
F4201: FF 255 RES          INT 021h
F4202: CD 205 =            IRET
F4203: 21 033 !            ADD [BX + SI], AL
F4204: CF 207 ±            ADD [BX + SI], AL
F4205: 00 000 NULL         ADD [BX + SI], AL
F4206: 00 000 NULL         ADD [BX + SI], AL
F4207: 00 000 NULL         ADD [BX + SI], AL
F4208: 00 000 NULL         ADD [BX + SI], AL
F4209: 00 000 NULL         ADD [BX + SI], AL
F420A: 00 000 NULL         ADD [BX + SI], AL
F420B: 00 000 NULL         ADD [BX + SI], AL
F420C: 00 000 NULL         ADD [BX + SI], AL
F420D: 00 000 NULL         ADD [BX + SI], AL
F420E: 00 000 NULL         ADD [BX + SI], AL
F420F: 00 000 NULL         ADD [BX + SI], AL
F4210: 00 000 NULL         ADD [BX + SI], AL
F4211: 00 000 NULL         ADD [BX + SI], AL
F4212: 00 000 NULL         ADD [BX + SI], AL
F4213: 00 000 NULL         ADD [BX + SI], AL
F4214: 00 000 NULL         ...
```

```
          CALL CHECKID          ;Call CHECKID , bi-check 3la el ID en kan sa7 wla 3'lt
```



emulator screen (80x29 chars)

```
WHAT DO YOU WANT
(1) Enter your ID and PASSWORD
(2) Change your Password
Your choice is (1/2) 1
--------------------------------------------------------------
ENTER YOUR ID: AAAA
ENTER YOUR PASSWORD: 1111
******WELCOME******
--------------------------------------------------------------
WHAT DO YOU WANT
(1) Enter your ID and PASSWORD
(2) Change your Password
Your choice is (1/2) 2
--------------------------------------------------------------
ENTER YOUR ID: AAAA
ENTER OLD PASSWORD: 1111
ENTER NEW PASSWORD: 1221
CONFIRM YOUR PASSWORD: 1221
YOUR PASSWORD IS SUCCESFULLY CHANGED
--------------------------------------------------------------
WHAT DO YOU WANT
(1) Enter your ID and PASSWORD
(2) Change your Password
Your choice is (1/2) _
```

8. RESULT AND DISCUSSION:

In this generation of advanced technology, theft and crime has taken the aid of technology itself in achieving its results many times. To avoid such circumstances, sometimes even small things can lead to a great change. Thus, implementation of smart devices in Security Control like the Knock Based Security System can prove to be very valuable to the person using it and also to the valuables themselves. In this paper, the issue of safety is being addressed through easily and affordable technology like piezo sensor, GSM module, Push button, and Arduino microcontroller.This can also be improved by interfacing with various technologies like fingerprint reader, voice detection etc., thus making it much more secure while not being too costly and out of reach in terms of availability.

8.2 FUTURE WORK:

- Set up an alarm in case the password is incorrect for three consecutive times thus warning the user. This can be done using the GSM module.
- Users can change passwords giving the correct credentials.
- Hardware implementation of the entire system.
- Can also be improved by interfacing with various technologies like finger-print reader, voice detection etc., thus making it much more secure while not being too costly and out of reach in terms of availability.

9. CONCLUSION

- A technological way to implement a door unlocking system using a combination of technical and manual methods was developed in the project.
- This design provides a number of advantages, including a low cost, increased security, and a small size.
- The microcontroller is used in combination with a sensor, a buzzer, and a servo motor in this system.
- As a result, the detection efficiency of the knock has improved.
- This system can be introduced in places that are more prone to thefts like banks, jewelry stores, even private property.
- This system can help reduce thefts and problems caused when keys are lost and ensures safety of the public and private property.
- It reduces time for opening door locks.

.

10. REFERENCES

[1] Thillai, Thilagam J. Salai, T. Sarath Babu, and B. Siva Reddy. "Arduino based safeguarding system by using sound." *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 2019.

[2] Reddy, R. Sai Charan, et al. "Security System Based on Knock-Pattern Using Arduino and GSM Communication." *International Journal of Engineering and Techniques* 4.1 (2018).

[3] Wei-Hsun Lee 1, and Chi-Yi Chiu "Design and Implementation of a Smart Traffic Signal Control System for Smart City Applications".

[4] Nada, Elshaimaa, et al. "Intelligent Lock Applied for Smart Door." *International Journal of Computer Science and Information Security (IJCSIS)* 17.6 (2019).

[5] Rahman, Shaik Ateeq Ur, Jinesh Thakker, and M. Sai Keerthana. "SMART KNOCK DETECTING DOOR LOCK USING ARDUINO."

[6]Soni, Shubham, Rajni Soni, and Akhilesh A. Waoo. "RFID-Based Digital Door Locking System."

[7] Sitompul, Budi Paul, et al. "Design and build of automatic hand sanitizer using arduino." *Eduvest-Journal Of Universal Studies* 1.3 (2021): 119-127.

[8] https://www.electronicshub.org/different-types-sensors/

[9]https://www.elprocus.com/atmega328-arduino-uno-board-working-and-its-applications/

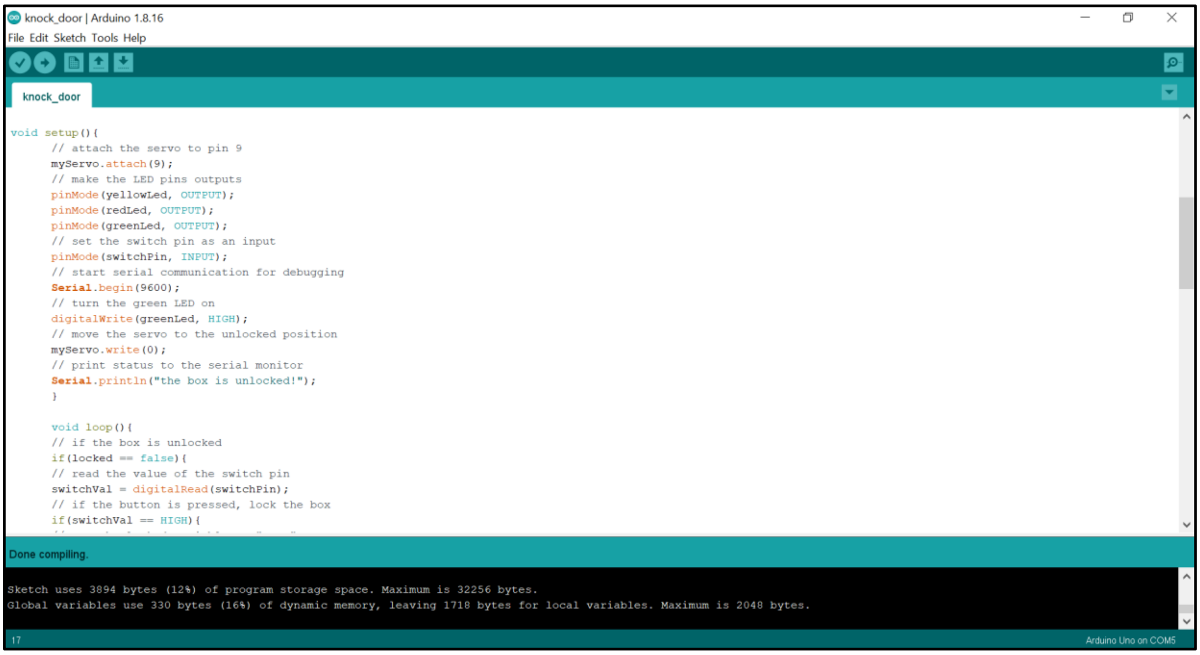
[10] Das, Arnab, et al. "Development of a Novel Design and Subsequent Fabrication of an Automated Touchless Hand Sanitizer Dispenser to Reduce the Spread of Contagious Diseases." *Healthcare*. Vol. 9. No. 4. Multidisciplinary Digital Publishing Institute, 2021.

11. APPENDIX

11.a Arduino code:

```
// import the library
 #include <Servo.h>
 // create an instance of the servo library

Servo myServo;
int onOffTime;
int IRSense;
int autoOffTrigger=0;
void setupled()
{
  pinMode(5, OUTPUT); //Motor pump control pin
  pinMode(2, OUTPUT); //Sensor sensing Pin
  Serial.begin(9600);
}

void loopled()
{
  int IRSense= analogRead(A0); // Read Sensor Value
  int onOffTime= analogRead(A1); // Read How much volume to dispense

  int time=map(onOffTime,0,1023,0,10);//convert to simple scale
  Serial.println("IR: "+String(IRSense));
  Serial.println("pot: "+String(onOffTime)+ ": "+String(time));

  if(IRSense >78) //If sense higher than 78 LED INDICATE
  {
    digitalWrite(2,1);
  }
  else
  {
    digitalWrite(2,0);
  }
  //IF IR sense higher than 78
  //Motor pump will ON for Sometime mentioned in "time"
  //Then Turn Off
  if(IRSense >78 && autoOffTrigger==0)
  {
          digitalWrite(5,1);
          delay(time*1000); // 1000 is 1000 millisecond(s)
          digitalWrite(5,0);
    autoOffTrigger=1;
    Serial.println("Dispensing... ");
  }

else if(IRSense <78)
  {
    //AutoOFFTrigger is used to cut off motor pump
    //Make for each sense, only onetime dispenser will come out
    autoOffTrigger=0;
  }
}

  //Smart lock

 const int piezo = A0;
 const int switchPin = 2;
 const int yellowLed = 3;
 const int greenLed = 4;
 const int redLed = 5;


//defines LED's and piezo's pins.
// variable for the piezo value
 int knockVal;
 // variable for the switch value
 int switchVal;
// variables for the high and low limits of the knock value
 const int quietKnock = 10;
 const int loudKnock = 100;
// variable to indicate if locked or not
 boolean locked = false;
 // how many valid knocks you've received
 int numberOfKnocks = 0;
void setup(){
 // attach the servo to pin 9
 myServo.attach(9);
// make the LED pins outputs
 pinMode(yellowLed, OUTPUT);
 pinMode(redLed, OUTPUT);
 pinMode(greenLed, OUTPUT);
// set the switch pin as an input
 pinMode(switchPin, INPUT);
// start serial communication for debugging
 Serial.begin(9600);
// turn the green LED on
 digitalWrite(greenLed, HIGH);
// move the servo to the unlocked position
 myServo.write(0);
// print status to the serial monitor
 Serial.println("the box is unlocked!");
 }
void loop(){
// if the box is unlocked
 if(locked == false){
// read the value of the switch pin
 switchVal = digitalRead(switchPin);
// if the button is pressed, lock the box
 if(switchVal == HIGH){
 // set the locked variable to "true"
 locked = true;
// change the status LEDs
 digitalWrite(greenLed,LOW);
 digitalWrite(redLed,HIGH);
// move the servo to the locked position
 myServo.write(90);
// print out status
 Serial.println("the box is locked!");
// wait for the servo to move into position
 delay (1000);
 }
 }
// if the box is locked
 if(locked == true){
// check the value of the piezo
 knockVal = analogRead(piezo);
// if there are not enough valid knocks
 if(numberOfKnocks < 3 && knockVal > 0){
// check to see if the knock is in range
 if(checkForKnock(knockVal) == true){
// increment the number of valid knocks
 numberOfKnocks++;
 }
// print status of knocks
 Serial.print(3 - numberOfKnocks);
 Serial.println(" more knocks to go");
 }
// if there are three knocks
 if(numberOfKnocks >= 3){
 // unlock the box
 locked = false;
// move the servo to the unlocked position
 myServo.write(0);
// wait for it to move
 delay(20);
// change status LEDs
 digitalWrite(greenLed,HIGH);
 digitalWrite(redLed,LOW);
 Serial.println("the box is unlocked!");
 }
 }
 }
// this function checks to see if a
 // detected knock is within max and min range
 boolean checkForKnock(int value){
 // if the value of the knock is greater than
  // the minimum, and larger than the maximum
 if(value > quietKnock && value < loudKnock){
 // turn the status LED on
 digitalWrite(yellowLed, HIGH);
 delay(50);
 digitalWrite(yellowLed, LOW);
 // print out the status
 Serial.print("Valid knock of value ");
 Serial.println(value);
return true;
 }
 // if the knock is not within range
 else {
 // print status
 Serial.print("Bad knock value ");
 Serial.println(value);
return false;
 }
 }
```

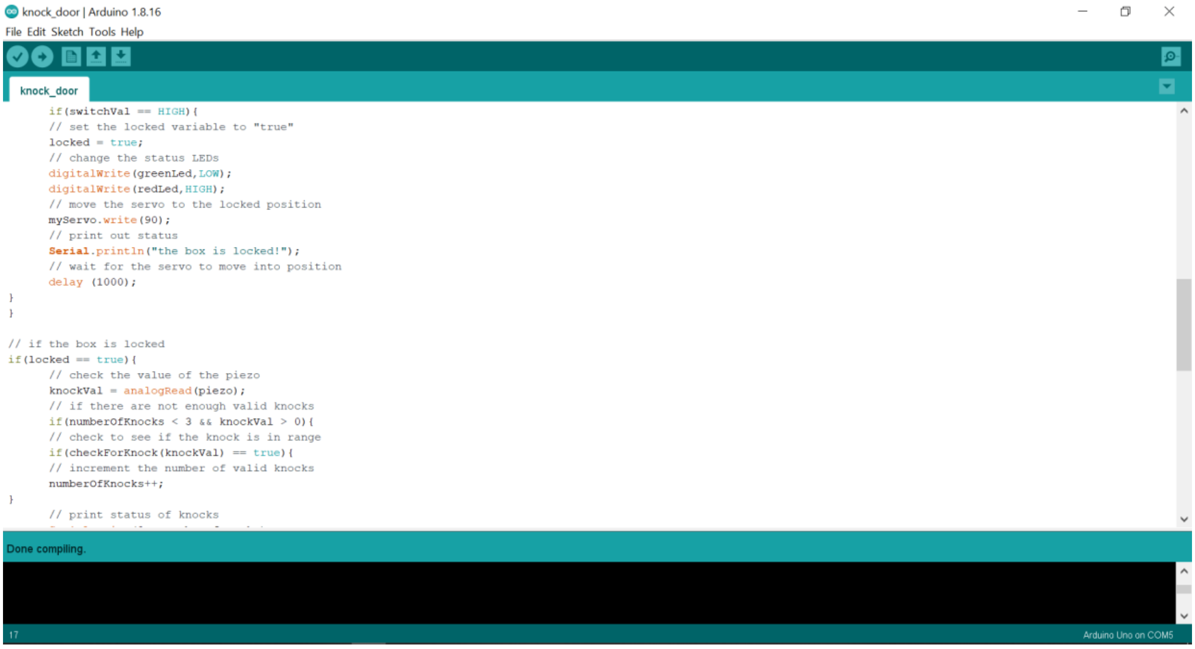
11.b Screenshot of CODE:



```
void setup(){
      // attach the servo to pin 9
      myServo.attach(9);
      // make the LED pins outputs
      pinMode(yellowLed, OUTPUT);
      pinMode(redLed, OUTPUT);
      pinMode(greenLed, OUTPUT);
      // set the switch pin as an input
      pinMode(switchPin, INPUT);
      // start serial communication for debugging
      Serial.begin(9600);
      // turn the green LED on
      digitalWrite(greenLed, HIGH);
      // move the servo to the unlocked position
      myServo.write(0);
      // print status to the serial monitor
      Serial.println("the box is unlocked!");
      }

      void loop(){
      // if the box is unlocked
      if(locked == false){
      // read the value of the switch pin
      switchVal = digitalRead(switchPin);
      // if the button is pressed, lock the box
      if(switchVal == HIGH){
```

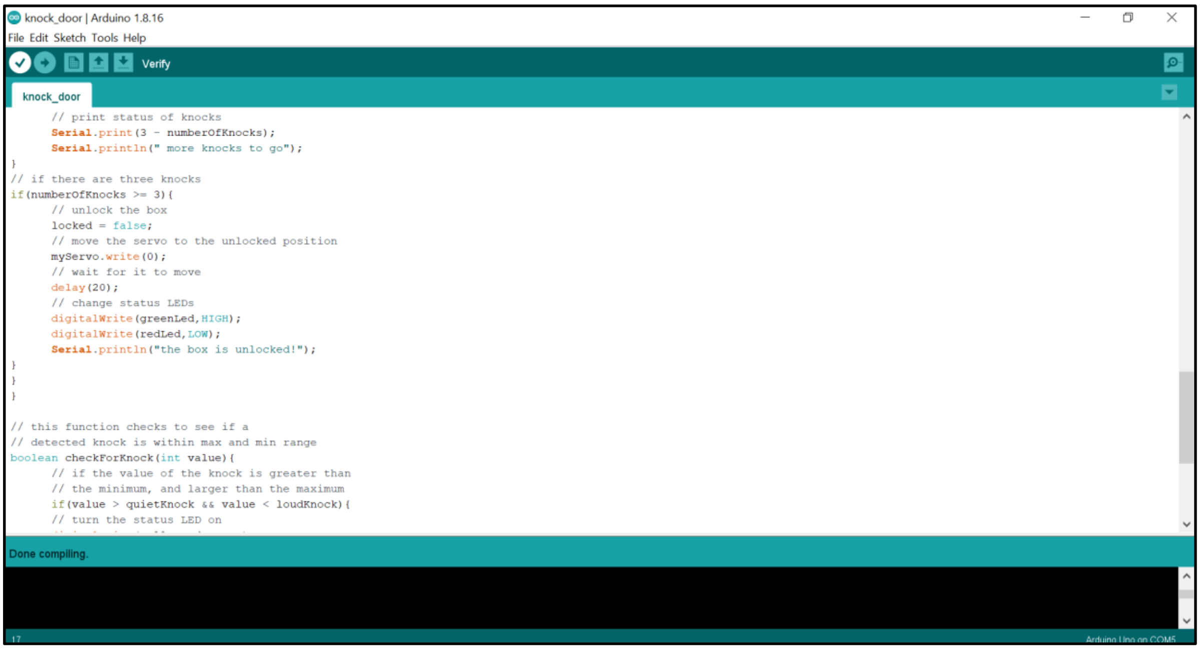
Done compiling.

```
Sketch uses 3894 bytes (12%) of program storage space. Maximum is 32256 bytes.
Global variables use 330 bytes (16%) of dynamic memory, leaving 1718 bytes for local variables. Maximum is 2048 bytes.
```



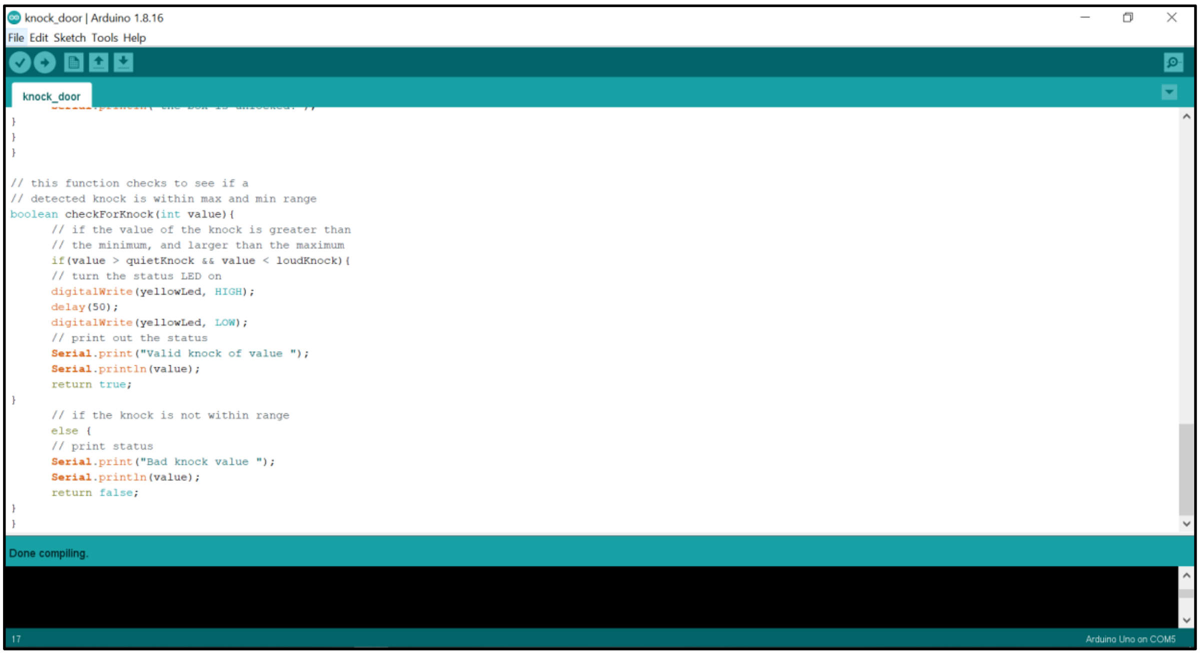
```
      if(switchVal == HIGH){
      // set the locked variable to "true"
      locked = true;
      // change the status LEDs
      digitalWrite(greenLed,LOW);
      digitalWrite(redLed,HIGH);
      // move the servo to the locked position
      myServo.write(90);
      // print out status
      Serial.println("the box is locked!");
      // wait for the servo to move into position
      delay (1000);
}
}

// if the box is locked
if(locked == true){
      // check the value of the piezo
      knockVal = analogRead(piezo);
      // if there are not enough valid knocks
      if(numberOfKnocks < 3 && knockVal > 0){
      // check to see if the knock is in range
      if(checkForKnock(knockVal) == true){
      // increment the number of valid knocks
      numberOfKnocks++;
}
      // print status of knocks
```

Done compiling.



```
      // print status of knocks
      Serial.print(3 - numberOfKnocks);
      Serial.println(" more knocks to go");
}
// if there are three knocks
if(numberOfKnocks >= 3){
      // unlock the box
      locked = false;
      // move the servo to the unlocked position
      myServo.write(0);
      // wait for it to move
      delay(20);
      // change status LEDs
      digitalWrite(greenLed,HIGH);
      digitalWrite(redLed,LOW);
      Serial.println("the box is unlocked!");
}
}
}

// this function checks to see if a
// detected knock is within max and min range
boolean checkForKnock(int value){
      // if the value of the knock is greater than
      // the minimum, and larger than the maximum
      if(value > quietKnock && value < loudKnock){
      // turn the status LED on
```

Done compiling.



```
}
}
}

// this function checks to see if a
// detected knock is within max and min range
boolean checkForKnock(int value){
      // if the value of the knock is greater than
      // the minimum, and larger than the maximum
      if(value > quietKnock && value < loudKnock){
      // turn the status LED on
      digitalWrite(yellowLed, HIGH);
      delay(50);
      digitalWrite(yellowLed, LOW);
      // print out the status
      Serial.print("Valid knock of value ");
      Serial.println(value);
      return true;
}
      // if the knock is not within range
      else {
      // print status
      Serial.print("Bad knock value ");
      Serial.println(value);
      return false;
}
}
```

Done compiling.

11.c ALP CODE:

```
 .MODEL SMALL
     .STACK 64
;-----------------
     .DATA
DATA2 DB  5,?,5 DUP (?)
DATA3 DB 'ENTER YOUR ID: ','$'
```

```
DATA1 DB '0123456789ABCDEFabcdef?'

DATA4 DB 'EROR:THE ID NUMBER MUST BE 4-BIT HEX','$'

DATA5 DB "WRONG ENTRY 'Your ID must contain data from 0-->9 or A-->F','$'

DATA6 DW 0AAAAH,0BBBBH,0CCCCH,0DDDDH,0EEEEH,0FFFFH,1111H,2222H,3333H,4444H

DATA7 DW 5555H,6666H,7777H,8888H,9999H,0100H,0200H,0300H,0400H,5667H

DATA8 DW 1111H,2222H,3333H,4444H,5555H,6666H,000AH,000BH,000CH,000DH

DATA9 DW 000FH,000EH,0001H,0002H,0003H,0A00H,0B00H,0C00H,0D00H,0A0AH

DATAA DB 'Your ID is wrong, Please try again!!','$'

DATAB DB 'ENTER YOUR PASSWORD: ','$'

DATAC DB 5,?,5 DUP (?)

DATAP DB 5,?,5 DUP (?)

DATAD DB '******WELCOME******','$'

DATAE DB 'WRONG PASSWORD,TRY AGAIN','$'

DATAF DB 00H

DATAG DB '----------------------------------------------------------------','$'

DATAH DB 'WHAT DO YOU WANT','$'

DATAI DB '(1) Enter your ID and PASSWORD','$'

DATAJ DB '(2) Change your Password','$'

DATAU DB 'Your choice is (1/2) ','$'

DATAT DB 'EROR:WRONG CHOICE','$'

DATAK DB 2,?,2 DUP (?)

DATAR DB 'ENTER YOUR ID: ','$'

DATAQ DB 'ENTER OLD PASSWORD: ','$'

DATAY DB 'ENTER NEW PASSWORD: ','$'

DATAO DB 'CONFIRM YOUR PASSWORD: ','$'

DATAV DB 'YOUR PASSWORD IS SUCCESFULLY CHANGED','$'

DATAW DB 'WRONG ENTRY!! PLEASE,RE-ENTER NEW PASSWORD: ','$'

DATAZ DW ?

;------------------

        .CODE

MAIN          PROC FAR

         MOV AX,@DATA           ;move offset of data segment to AX
```

```
            MOV DS,AX             ;Mov AX to DS
            MOV ES,AX             ;Make DS and ES OVERLAPPED
            MOV  DH,00H           ;Initialize DH With zeros
            CALL CLEAR            ;Call CLEAR screen procedure
            MOV  BP,OFFSET DATAF    ;Mov Offset dataf to BP to use it in setting cursor
START:        CALL SETCURSOR        ;Call SETCURSOR procedure
            CALL ENTRY            ;Call ENTRY, elmsg ely 7i2olk fiha enta 3aiz eh!!
            CALL GETCHOICE        ;Call GETCHOICE, 7ia5od mnk e5tiark eh 1 or 2
            CALL CHECKNO          ;Call CHECKNO ,hishof elrkam ely md5lo 1 aw 2 wla 7aga
tania !!
            CALL SETCURSOR        ;Call SETCURSOR, 3mltha tany 3shan yzbt elklam ,ynzl satr
gdid w kda :D
            CALL ENTERORCHANGE     ;Call ENTERORCHANGE, hishofk en kont 3aiz td5l
elPASSWORD wla t3'iro
            CALL HANDLE           ;Call HANDLE, elproc di bt5lik t-handle elmwdo3, y3ny enter
old pass,re-enter ....
            CALL CONVERT          ;Call CONVERT , elproc di bt3'ir elpassword
ID:           CALL WELCOME          ;Call WELCOME,, lw enta e5trt enk 3aiz td5al elpassword,
di awl 7aga htzhrlk
            CALL GET_IN           ;Call GET_IN , bta5od mnk elpassword
            CALL NO.LET           ;Call NO.LET , bishof en kan elrakm ely md5lo 4 arkam wla a2l
            CALL CHECK            ;Call CHECK , bi-check iza kan elrakm ely d5lto in range (0-->9
aw a-->f aw A-->F) wla la2
            MOV  SI,OFFSET DATA2+2  ;Initialize SI to point to the ID data in memory
            CALL PUTIDINAX        ;Call PUTIDINAX, bt7ot el ID ely gy mn elmemory f AX
            CALL CHECKID          ;Call CHECKID , bi-check 3la el ID en kan sa7 wla 3'lt
            CALL SETCURSOR        ;Call SETCURSOR, 3mltha tany 3shan yzbt elklam ,ynzl satr
gdid w kda :D
            CALL GETPASS          ;Call GETPASS ,elproc di hta5od mn eluser elpassword
            MOV  SI,OFFSET DATAC+2  ;Initialize, SI to point to datac in memory
            CALL PUTIDINAX        ;Call PUTIDINAX, bt7ot el PASSWORD ely gy mn elmemory
f AX
            CALL CHECKPASS        ;Call CHECKPASS, bi-check en kan elpassword sa7 wla 3'lt
            CALL SETCURSOR        ;Call SETCURSOR, bizbt elklam ely bizhr 3la elDOS
            CALL ENTER            ;Call ENTER, lw elpassword sa7 bitl3lo gomla 3la elshasha
```

```
NO_EROR:        CALL SETCURSOR         ;As shown before
                CALL NOEROR            ;check if the entered nmber is less than 4!!
WR_ENT:         CALL SETCURSOR         ;As shown before
                CALL WRONGENTRY        ;check if the number is between 0 --> 9 or a --> f or A --> F
WRONGID:        CALL SETCURSOR
                CALL WRONG_ID
WRONGPASS:      CALL SETCURSOR
                CALL WRONG_PW
OPERA:          MOV  AH,4CH
                INT 21H
MAIN            ENDP
;----------------
CLEAR           PROC
                MOV AX,0600H
                MOV BH,07
                MOV CX,0000
                MOV DH,24
                MOV DL,79
                INT 10H
                RET
CLEAR           ENDP
;----------------
ENTRY           PROC
                MOV AH,09H
                MOV DX,OFFSET DATAH
                INT 21H
                CALL SETCURSOR
                MOV AH,09H
                MOV DX,OFFSET DATAI
                INT 21H
                CALL SETCURSOR
                MOV AH,09H
```

```
            MOV DX,OFFSET DATAJ
            INT 21H
            CALL SETCURSOR
            MOV AH,09H
            MOV DX,OFFSET DATAU
            INT 21H
            RET
ENTRY       ENDP
;--------------
GETCHOICE     PROC
            MOV AH,0AH
            MOV DX,OFFSET DATAK
            INT 21H
            RET
GETCHOICE     ENDP
;-----------------
CHECKNO      PROC
            LEA BX,DATAK+2
            CMP [BX],31H
            JZ  RETURN2
            CMP [BX],32H
            JZ  RETURN2
            CALL EROR
RETURN2:     CALL 5AT
            RET
CHECKNO      ENDP
;---------------
EROR        PROC
            CALL SETCURSOR
            MOV AH,09H
            MOV DX,OFFSET DATAT
            INT 21H
            CALL 5AT
            JMP START
            RET
ERROR       ENDP
;--------------
SETCURSOR   PROC
            MOV AH,02H
            MOV BH,00
            MOV DL,00
            MOV DH,DS:[BP]   ;coloumn   ;row
            INT 10H
            ADD DS:[BP],1
            RET
SETCURSOR   ENDP
;----------------
WELCOME     PROC
            MOV AH,09H
            LEA DX,DATA3
            INT 21H
            RET
WELCOME     ENDP
;----------------
GET_IN      PROC
            MOV AH,0AH
            MOV DX,OFFSET DATA2
            INT 21H
            RET
GET_IN      ENDP
;----------------
NO.LET      PROC
            LEA SI,DATA2+1
            CMP [SI],04H
```

```
                JNZ NO_EROR
                RET
NO.LET          ENDP
;----------------
CHECK           PROC
                MOV AH,4
                LEA SI,DATA2+2
AGAIN:          LEA DI,DATA1
                MOV CX,23
                MOV AL,[SI]
                REPNZ SCASB
                CMP CX,00
                JZ  END
                INC SI
                DEC AH
                JNZ AGAIN
                RET
END:            JMP WR_ENT
CHECK           ENDP
;----------------
NOEROR          PROC
                MOV AH,09H
                MOV DX,OFFSET DATA4
                INT 21H
                CALL 5AT
                JMP START
                RET
NOEROR          ENDP
;---------------
WRONGENTRY      PROC
                MOV AH,09H
                MOV DX,OFFSET DATA5
```

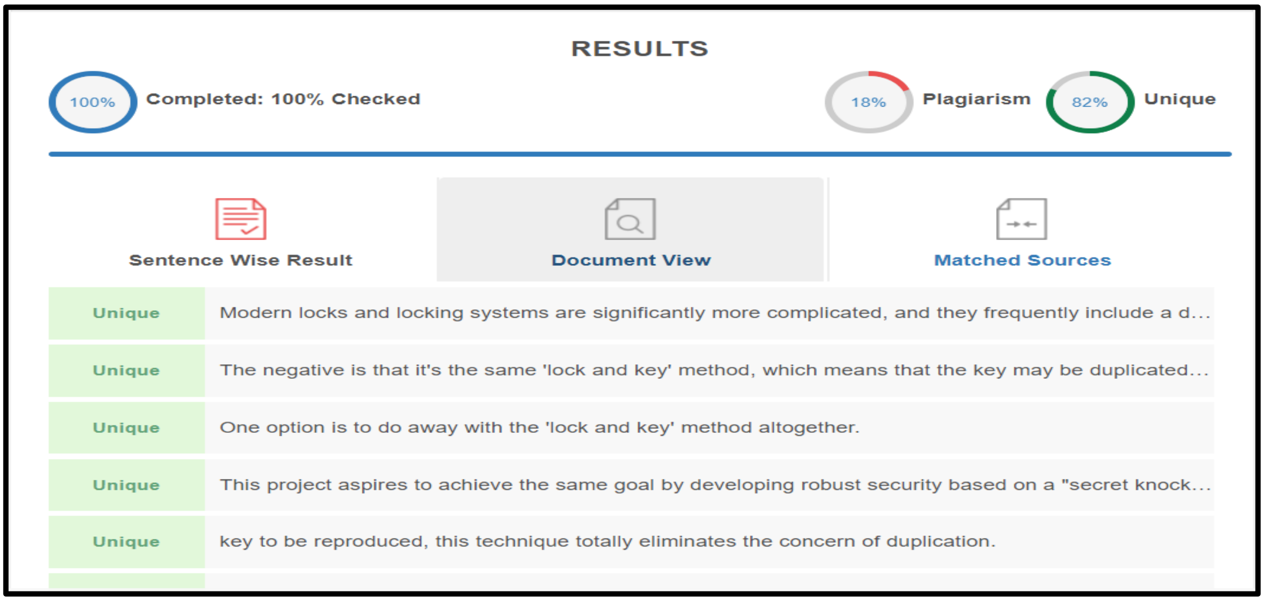
```
            INT 21H
            CALL 5AT
            JMP START
            RET
WRONGENTRY      ENDP
;--------------
PUTIDINAX      PROC
            MOV CX,04H
AGAIN2:        CMP [SI],39H
            JZ  ZERO
            JB  ZERO
            JA  OVER
ZERO:          SUB [SI],30H
            JMP STAR
OVER:          CMP [SI],70
            JZ  CAPITAL
            JB  CAPITAL
            JA  SMALL
CAPITAL:       SUB [SI],55
            JMP STAR
SMALL:         SUB [SI],87
            JMP STAR
STAR:          INC SI
            DEC CX
            JNZ AGAIN2
            SUB SI,4
            MOV AH,[SI]
            MOV AL,[SI+2]
            MOV BH,[SI+1]
            MOV BL,[SI+3]
            SHL AX,4
            OR  AX,BX
            RET
PUTIDINAX       ENDP
;--------------
CHECKID         PROC
            MOV CX,21           ; Set the counter to 21 decimal
            LEA DI,DATA6        ; DI = OFFSET DATA6
            CLD                 ; DF = 0 (AUTO INCREAMENT)
            REPNE SCASW         ; Check if the ID exists or not
            CMP CX,0000H        ; Check if the ID exists or not
            JZ WRONGID          ; If not exists jump to WRONGID
            RET
CHECKID         ENDP
;--------------
WRONG_ID        PROC
            MOV AH,09H
            MOV DX,OFFSET DATAA
            INT 21H
            CALL 5AT
            JMP START
            RET
WRONG_ID        ENDP
;-------------
GETPASS         PROC
            MOV AH,09H
            MOV DX,OFFSET DATAB
            INT 21H
            MOV AH,0AH
            MOV DX,OFFSET DATAC
            INT 21H
            RET
GETPASS         ENDP
;-------------
```

```
CHECKPASS       PROC
                MOV BX,AX
                ADD DI,38           ; If exist, jump to the password which equivalent to thad ID
                CMP BX,[DI]         ; Check if the password correct or not
                JNZ WRONGPASS
                RET
CHECKPASS       ENDP
;-------------
ENTER           PROC
                MOV AH,09H
                MOV DX,OFFSET DATAD
                INT 21H
                CALL 5AT
                JMP START
                RET
ENTER           ENDP
;------------
WRONG_PW        PROC
                MOV AH,09H
                MOV DX,OFFSET DATAE
                INT 21H
                CALL 5AT
                JMP START
WRONG_PW        ENDP
;------------
5AT             PROC
                CALL SETCURSOR
                MOV AH,09H
                MOV DX,OFFSET DATAG
                INT 21H
                RET
5AT             ENDP
```

```
;-----------
ENTERORCHANGE    PROC
                LEA BX,DATAK+2
                CMP [BX],31H
                JZ  ID
                RET
ENTERORCHANGE    ENDP
;--------------
HANDLE          PROC
                MOV AH,09H
                MOV DX,OFFSET DATAR
                INT 21H
                CALL GET_IN
                CALL NO.LET
                CALL CHECK              ;check if exists
                MOV  SI,OFFSET DATA2+2
                CALL PUTIDINAX          ;PUT ID IN AX
                CALL CHECKID
                ;MOV BX,OFFSET DATAZ
                MOV BX,OFFSET DATAZ
                LEA DX,[DI]
                MOV [BX],DX
                CALL SETCURSOR
                MOV AH,09H
                MOV DX,OFFSET DATAQ
                INT 21H
                MOV AH,0AH
                MOV DX,OFFSET DATAC
                INT 21H
                MOV  SI,OFFSET DATAC+2
                CALL PUTIDINAX          ;PUT ID IN AX
                CALL CHECKPASS
```

```
            CALL SETCURSOR
            MOV AH,09H
            MOV DX,OFFSET DATAY
            INT 21H
AGAIN3:         MOV AH,0AH
            MOV DX,OFFSET DATAC
            INT 21H
            CALL SETCURSOR
            MOV AH,09H
            MOV DX,OFFSET DATAO
            INT 21H
            MOV AH,0AH
            MOV DX,OFFSET DATAP
            INT 21H
            CALL CHECKCONFIRM
            RET
HANDLE          ENDP
;---------------
CHECKCONFIRM    PROC
            CLD
            MOV SI,OFFSET DATAC+2
            MOV DI,OFFSET DATAP+2
            MOV CX,05H
            REPE CMPSB
            CMP CX,0000H
            JNZ  PUTITAGAIN
            RET
PUTITAGAIN:     ;CALL 5AT
            CALL SETCURSOR
            MOV AH,09H
            MOV DX,OFFSET DATAW
            INT 21H
            JMP AGAIN3
CHECKCONFIRM    ENDP
;--------------
CONVERT        PROC
            MOV SI,OFFSET DATAP+2
            CALL PUTIDINAX
            MOV BX,OFFSET DATAZ
            ADD [BX],38
            MOV DI,[BX]
            MOV [DI],AX
            CALL SETCURSOR
            MOV AH,09H
            MOV DX,OFFSET DATAV
            INT 21H
            CALL 5AT
            JMP START
CONVERT        ENDP
;-------------
            END MAIN
```

12. PLAGIARISM REPORT



Unique	Modern locks and locking systems are significantly more complicated, and they frequently include a d...
Unique	The negative is that it's the same 'lock and key' method, which means that the key may be duplicated...
Unique	One option is to do away with the 'lock and key' method altogether.
Unique	This project aspires to achieve the same goal by developing robust security based on a "secret knock...
Unique	key to be reproduced, this technique totally eliminates the concern of duplication.