Finger print and keypad lock design information

MATERIALS AND METHODOLOGY

-INTRODUCTION
-DESIGN PARAMETERS
-SYSTEM DESIGN

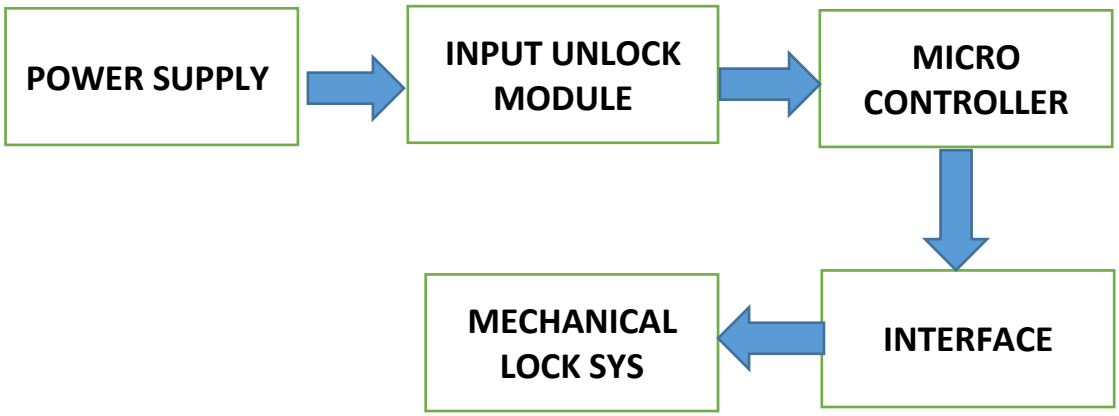
INTRDUCTION:
Based on the already developed information on the previous chapter,our device is found to be a consisting body of an active and passive electronic components,coupled with a micro controller system and few modules,which are are further systematize to provide a smart locking system, which comprises of biometric and keypad lock system.

DESIGN PARAMETERS:
-Input system for lock:
This is a very important parameter to look into in our design,since the efficiency of the device is dependent on its variable input system for unlocking process, where in this project is the biometric sensor and the keypad system.

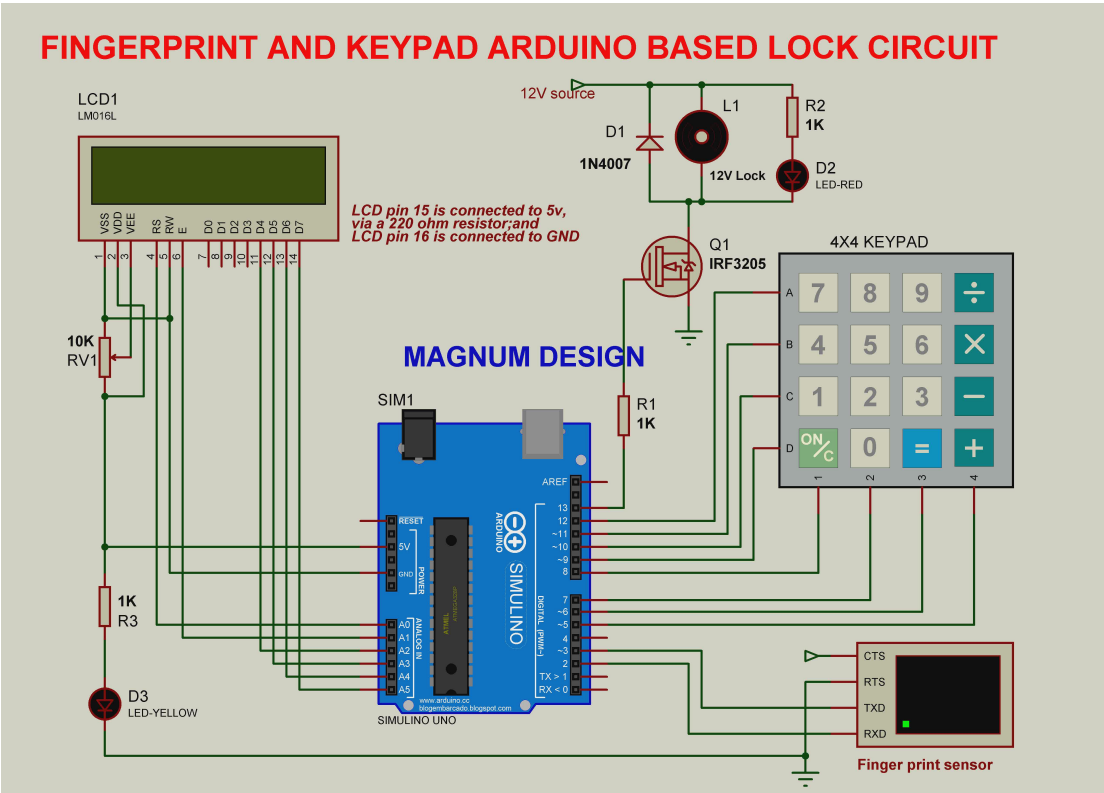
-Data processing for values generated:
The data, generated from the sensor,are further processed and converted to a digital value that the computer can understand. This process is achieved using a micro controller,and the essential calculations were made in the conversion process,for the micro controller to use for the whole process .

-High level data interface:
This system at this point, requires a method of which all the processed information can be interfaced to the user, and the suitable way to achieve that is to use an LCD (liquid crystal display), which will relate both the lock state of the system, and the relative input for the unlock.

SYSTEM DESIGN:
In this design, the building block or working principle of the system is evidently presented using a simple block diagram below.



The above diagram is further elaborated using the simple circuit diagram presented below.



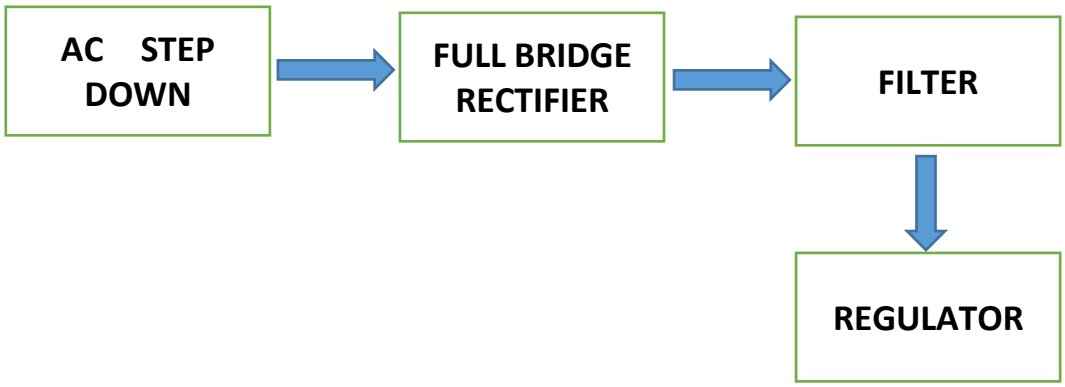
Below is the list of components I used for this project

Parts:

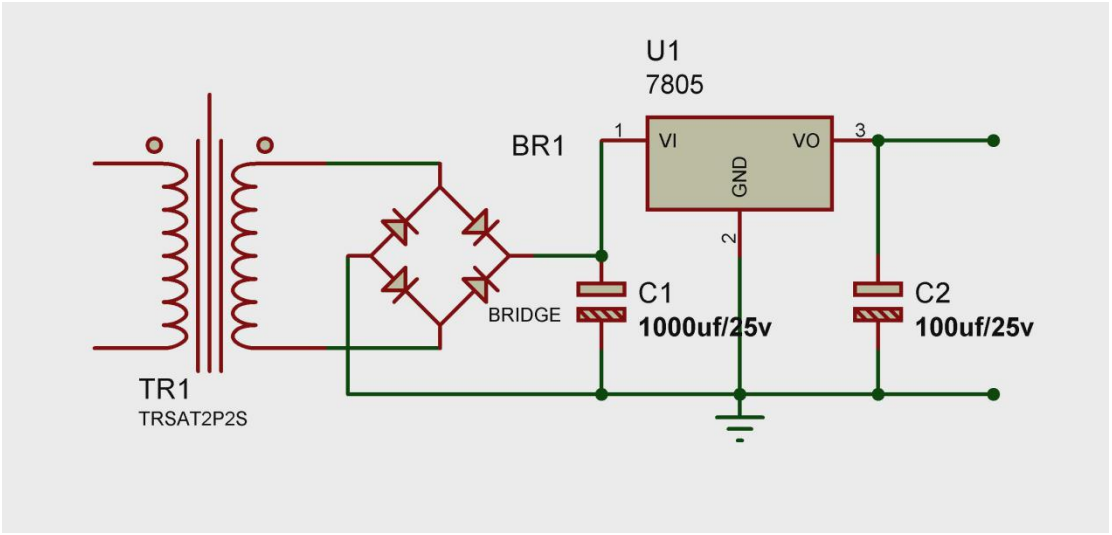
Atmega328p with arduino bootloader
Breadboard
16x2 character LCD
ZFM-206SA Fingerprint sensor
4x4 Matrix Keypad
ABS Plastic Box enclosure 180x120x55mm
LM7805 Voltage Regulator
SPDT switch
22pF ceramic capacitor x4
220 Ohm Resistor
1K Resistor
5V Voltage Regulator
Tactile push button
16Mhz crystal
10K trim pot
Red LED
Green LED
Diode

POWER SUPPLY

the power supply of the circuit, is a different circuit all together, in order to make work lots more easier, we used a 5volt power regulator,or any compartible 5v adapter,though,we can choose to build our stand by power supply,which simulate further the model of the 12v power adapter that we used and it constitute of the simple block diagram presented below.



The circuit below, further elaborate the block diagram above;



AC STEP DOWN:

The transformer devise is used to bring or step down the incoming AC grid line voltage which is 220v AC down to 12v AC, for the proper working functionality of the rest of other passive components in the system

FULL BRIDGE RECTIFIER:

The full bridge rectifier circuit, comprises four set of semiconductor device known as diode, which is basically configured in such a manner that the stepped down AC voltage

from the transformer devise, will be easily converted into a DC voltage, in order to suit the operational working of the system.

FILTER:

The filter circuit in this system, constitute of a simple electrolytic capacitor, which is connected across the converted DC voltage from the transformer.the filter network serve the basic purpose to smooth out ripple that made their way out of the DC output,and to ensure a constant dc voltage output, all through the whole process of power supply.

REGULATOR:

The surge effect in most electronics is quite bad and destructive in nature, and in this system, we would not like the same effects of surge prevailing in the system, and in order to subdue these effects, we integrate the use of voltage regulator 7805, to protect our system from any sudden surge that might likely surface.

The regulator also,provides the actual voltage requirement of the sensor we are using in this project,and hence 5volt rating.

INPUT UNLOCK MODULE:

The sensor for the biometrics used is ZFM-206SA Fingerprint sensor, while the keypad used is 4x4 keypad. These two module ar the main input system for the varaible for the unlocking process.

The biometrics utilises the finger as the parameter for unlocking the system, and the keypad provides an auxillary method of pening the system, if the finger print is not prefered.

The use of these two main module, cut accrss the current topology for smart phone development,where both the finger print and the password lock, prvides the essential framework of securty protocol in a phone, and as such, we projected to do the same with the project.

MICRO CONTROLLER CIRCUIT:

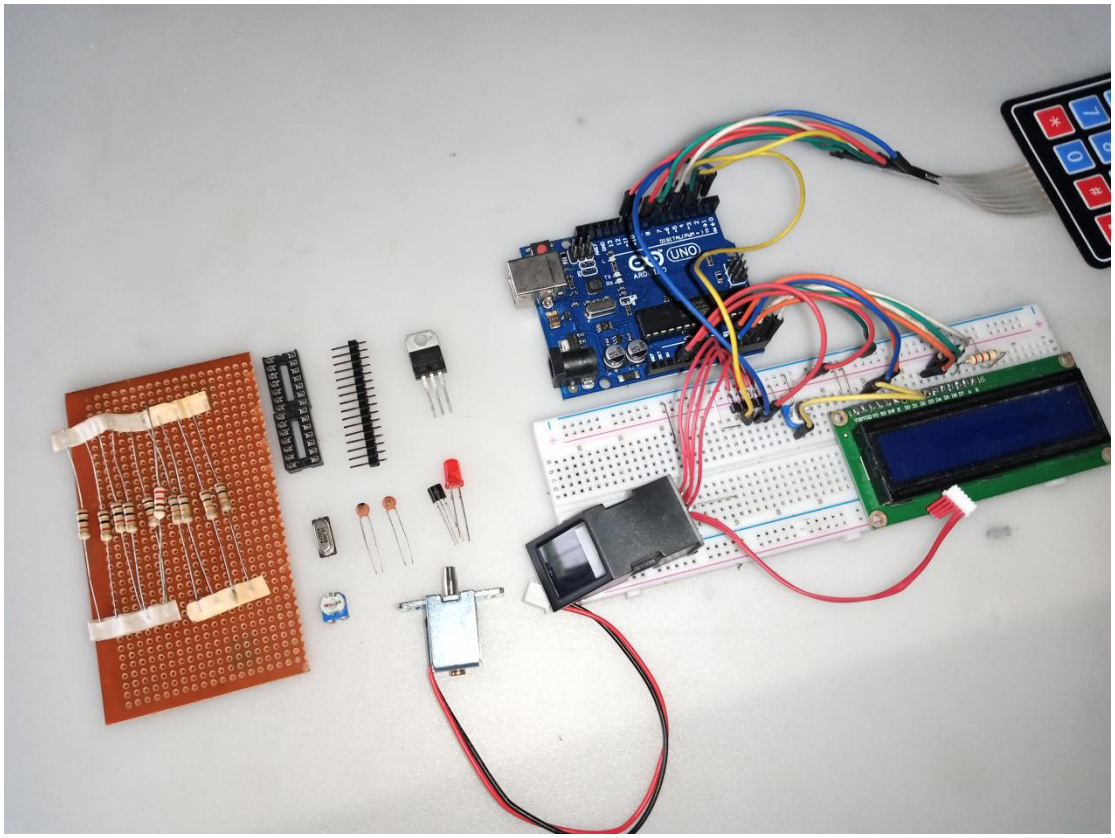
The micro controller used in this project was a simple ATMEGA328 chip(compartible with arduino).
The micro controller in the project is task to recieve the incoming data from both the biometric scanner and keypad,and with the code written to the system, the micro controller process the infromation and do the necessary calculations in order to generate the essential taskthat will be interfaced on the LCD, and further lock the system.

INTERFACE:
The interface part of the project includes the use of an LCD (liquid crystal display),in order to show the current state of the lock system. The lcd is further programmed and connected to the system, for display of all the processes carried out by the micro controller, and that creats a flexible utility feature for the system.

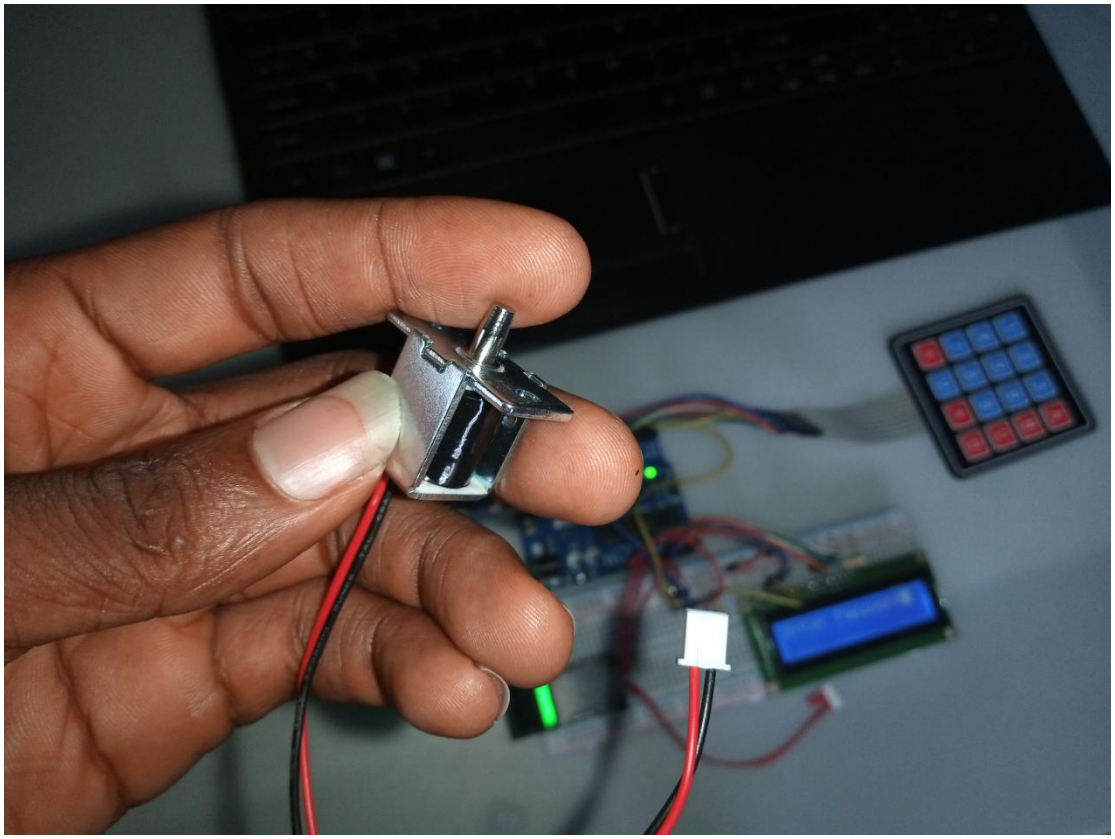
MECHANICAL LOCK SYSTEM:
This part of the system uses the solenoid mechanical lock system,to either lock or unluck the dor or any mode of assess to the building.

The relay, from the schematic, once triggered will be able to either set the solenoid key to active mode, which is lock, deactivate it, which is unlock, and as such, we have a smart system that perfomes a high degree for lock system.

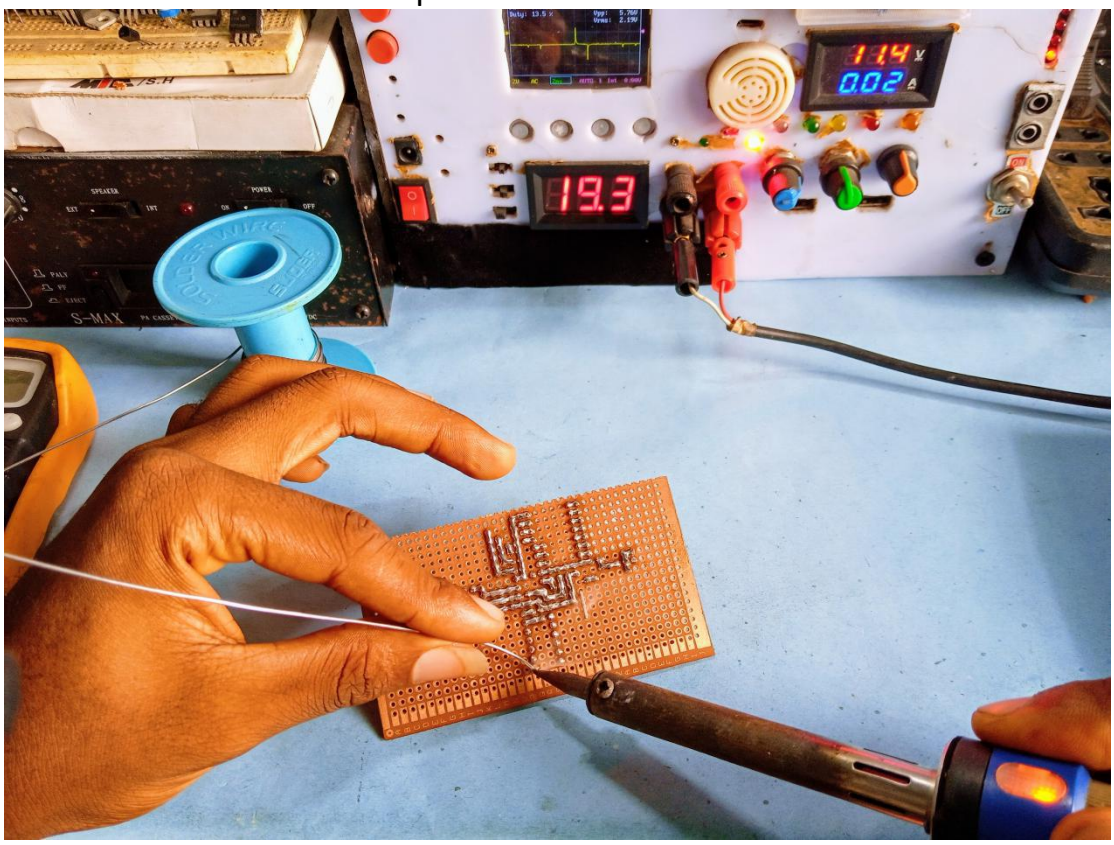
CONSTRUCTION OF THE SYSTEM:
This part of the development of this project includes the accumulatio of materilas thaat were proposed for the design applications. Below the the image of some these comnponets as they wwere assembled.



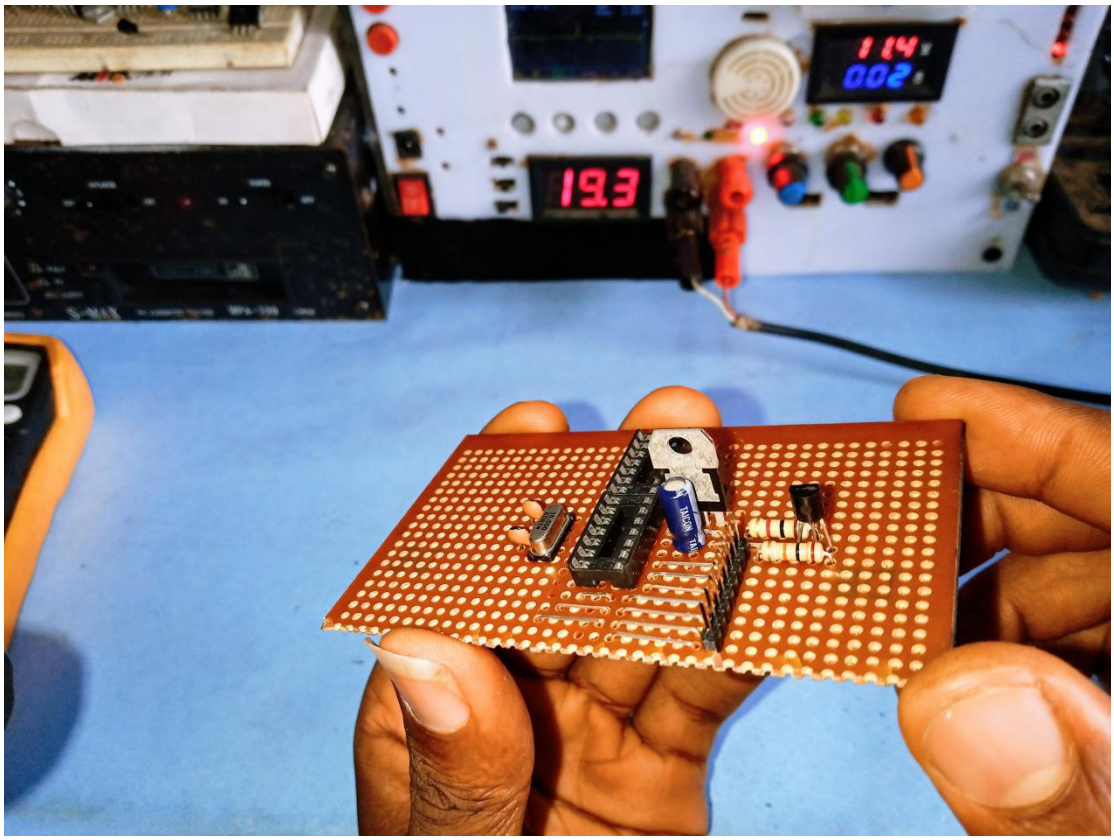
The solenoid



the soldring of the system is made on a perf board in order to ensure firm connections from components



More



The source code for the project is further presented below.

```
#include <LiquidCrystal.h>
#include <Adafruit_Fingerprint.h> //Libraries needed
#include <SoftwareSerial.h>
#include <Keypad.h>

const int rs = 14, en =15, d4 = 16, d5 = 17, d6 = 18, d7 = 19;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);


SoftwareSerial mySerial(2, 3);      //Fingerprint sensor wiring RX 3, TX 2

const byte numRows= 4;                  //number of rows on the keypad
const byte numCols= 4;                  //number of columns on the keypad

char keypressed;

char code[]={'1','2','3','4'}; //Passcode needed you can change it just keep the format "4 digits as char array"
char c[4];                                  //Array to get the code from the user
int ij;


//keymap defines the key pressed according to the row and columns just as appears on the keypad
char keymap[numRows][numCols]=
{
{'1', '2', '3', 'A'},
{'4', '5', '6', 'B'},
{'7', '8', '9', 'C'},
{'*', '0', '#', 'D'}
};
```

```
byte rowPins[numRows] = {12,11,10,9}; //Rows 0 to 3 //if you modify
your pins you should modify this too
byte colPins[numCols]={8,7,6,5}; //Columns 0 to 3

//initializes an instance of the Keypad class
Keypad myKeypad= Keypad(makeKeymap(keymap), rowPins, colPins,
numRows, numCols);


Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

uint8_t id;
#define lockOutput 13
void setup()
{
   Serial.begin(9600);
   pinMode(lockOutput, OUTPUT);
   finger.begin(57600);
      lcd.begin(16, 2);
   // Print a message to the LCD.
   lcd.print("SMART LOCK SYS!");
   lcd.blink();
   delay(2000);
   lcd.setCursor(0,1);
   lcd.print("Initializing sys");
   delay(2000);


   digitalWrite (lockOutput, LOW );
}



void loop()
{
   getFingerprintIDez();                    //Waiting for a fingerprint to
scan if it's valid or not
   keypressed = myKeypad.getKey();          //Reading the buttons typed
by the keypad
```

```
   if(keypressed == 'A'){                    //If it's 'A' it triggers "Adding
new template" sequence

                    ij=0;                          //ij is set to 0
                      lcd.clear();
       lcd.setCursor(0, 0);
       lcd.write("ENROLL NEW USER");
       lcd.setCursor(0, 1);
       lcd.write("Read mode...");
       delay(4000);
                    lcd.clear();
                    lcd.setCursor(0,0);
                    lcd.print("Enter Password");
                    getPassword();              //Getting the passcode
function
                    if(ij==4){                  //If the passcode is correct
we can start enrolling new finger template
                    Enrolling();                //Enrolling function
                    delay(2000);
                    lcd.clear();
                    }
                    else{                       //If the code is wrong we
can't add new users (templates)
                    lcd.setCursor(0,0);
                    lcd.print("Wrong code");
                    delay(2000);
                    lcd.clear();
                    }


   }
  if(keypressed == 'B')          //As the keypad is meant to stay inside
the 'B' button opens the door lock
  {                                   //Because the system I used needs
an electrical signal to open
       lcd.clear();
       lcd.setCursor(0, 0);
       lcd.write("SYSTEM BYPASS!");
       lcd.blink();
       lcd.setCursor(0, 1);
```

```
      lcd.write("Instant Unlock!");
      lcd.blink();
      delay(4000);

  ij=0;                                 //ij is set to 0
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Enter Password");
  getPassword();                    //Getting the passcode function
  if(ij==4)
  {
  digitalWrite (lockOutput, HIGH );
        lcd.clear();
      lcd.setCursor(0, 0);
      lcd.write("SYSTEM BYPASS!");
      lcd.blink();
      lcd.setCursor(0, 1);
      lcd.write("Door is now open!");
      lcd.blink();
  delay(5000);                          //Opening delay
 digitalWrite (lockOutput, LOW );
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.write("SYSTEM BYPASS!");
      lcd.blink();
      lcd.setCursor(0, 1);
      lcd.write("Door is closed!");
       delay(2000);

 }


  else{                                 //If the code is wrong we can't add new
users (templates)
                    lcd.setCursor(0,0);
                    lcd.print("Wrong code");
                    delay(2000);
                    lcd.clear();
  lcd.clear();
  }
```

```
  }

   lcd.clear();
      lcd.setCursor(0, 0);
      lcd.write("PLACE FINGER NOW!");
      lcd.setCursor(0, 1);
      lcd.write("Read mode...");
      delay(50);
}

//Getting password (code) function, it gets the characters typed
//and store them in c[4] array
void getPassword(){
     for (int i=0 ; i<4 ; i++){
       c[i]= myKeypad.waitForKey();
       lcd.setCursor(i,1);
       lcd.print("*");
      }
     lcd.clear();
     for (int j=0 ; j<4 ; j++){ //comparing the code entred with the code
stored
       if(c[j]==code[j])
         ij++;                                  //Everytime the char is correct we
increment the ij until it reaches 4 which means the code is correct
     }                                           //Otherwise it won't reach 4 and it
will show "wrong code" as written above
}

//The Enrolling and getFingerprintEnroll functions are mainly based on
the "Enroll" example code from the library
//Only the modifications will be commented, return to the example to
see each step in details, as here it's shortened

void Enrolling()    {
   keypressed = NULL;
   lcd.clear();
   lcd.print("Enroll New");
   delay(2000);
   lcd.clear();
   lcd.setCursor(0,0);
```

```
   lcd.print("Enter new ID");
   id = readnumber();                                        //This function
gets the Id it was meant to get it from the Serial monitor but we
modified it
   if (id == 0) {// ID #0 not allowed, try again!
        return;
   }

   while (!    getFingerprintEnroll() );
   }

//Enrolling function only the modifications are commented
uint8_t getFingerprintEnroll() {

   int p = -1;
   lcd.clear();
   lcd.print("Enroll ID:"); //Message to print for every step
   lcd.setCursor(10,0);
   lcd.print(id);
   lcd.setCursor(0,1);
   lcd.print("Place finger"); //First step
   while (p != FINGERPRINT_OK) {
      p = finger.getImage();
   }

   // OK success!

   p = finger.image2Tz(1);
   switch (p) {
      case FINGERPRINT_OK:
         break;
      case FINGERPRINT_IMAGEMESS:
         return p;
      case FINGERPRINT_PACKETRECIEVEERR:
         return p;
      case FINGERPRINT_FEATUREFAIL:
         return p;
      case FINGERPRINT_INVALIDIMAGE:
         return p;
      default:
```

```
            return p;
    }

    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Remove finger"); //After getting the first template
successfully
    lcd.setCursor(0,1);
    lcd.print("please !");
    delay(2000);
    p = 0;
    while (p != FINGERPRINT_NOFINGER) {
        p = finger.getImage();
    }

    p = -1;

    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Place same"); //We launch the same thing another time to
get a second template of the same finger
    lcd.setCursor(0,1);
    lcd.print("finger please");
    while (p != FINGERPRINT_OK) {
        p = finger.getImage();
    }

    // OK success!

    p = finger.image2Tz(2);
    switch (p) {
        case FINGERPRINT_OK:
            break;
        case FINGERPRINT_IMAGEMESS:
            return p;
        case FINGERPRINT_PACKETRECIEVEERR:
            return p;
        case FINGERPRINT_FEATUREFAIL:
            return p;
        case FINGERPRINT_INVALIDIMAGE:
```

```
      return p;
    default:
      return p;
  }


  p = finger.createModel();
  if (p == FINGERPRINT_OK) {
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    return p;
  } else if (p == FINGERPRINT_ENROLLMISMATCH) {
    return p;
  } else {
    return p;
  }

  p = finger.storeModel(id);
  if (p == FINGERPRINT_OK) {
  lcd.clear();                    //if both images are gotten without
problem we store the template as the Id we entred
  lcd.setCursor(0,0);
  lcd.print("Stored in");      //Print a message after storing and showing
the ID where it's stored
  lcd.setCursor(0,1);
  lcd.print("ID: ");
  lcd.setCursor(5,1);
  lcd.print(id);
  delay(3000);
  lcd.clear();
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    return p;
  } else if (p == FINGERPRINT_BADLOCATION) {
    return p;
  } else if (p == FINGERPRINT_FLASHERR) {
    return p;
  } else {
    return p;
  }
}
```

```
//This function gets the ID number as 3 digits format like 001 for ID 1
//And return the number to the enrolling function
uint8_t readnumber(void) {
   uint8_t num = 0;
     while (num == 0) {
         char keey = myKeypad.waitForKey();
      int    num1 = keey-48;
              lcd.setCursor(0,1);
              lcd.print(num1);
              keey = myKeypad.waitForKey();
      int    num2 = keey-48;
              lcd.setCursor(1,1);
              lcd.print(num2);
              keey = myKeypad.waitForKey();
      int    num3 = keey-48;
              lcd.setCursor(2,1);
              lcd.print(num3);
              delay(1000);
              num=(num1*100)+(num2*10)+num3;
              keey=NO_KEY;
   }
   return num;
}

//Main function taken from the "fingerprint" example and modified
//Only the modifications are commented
//This function waits for a fingerprint, scan it and give it access if
recognised

int getFingerprintIDez() {
   uint8_t p = finger.getImage();              //Image scanning
   if (p != FINGERPRINT_OK)    return -1;

   p = finger.image2Tz();                        //Converting
   if (p != FINGERPRINT_OK)    return -1;

   p = finger.fingerFastSearch();         //Looking for matches in the
internal memory
   if (p != FINGERPRINT_OK){                //if the searching fails it means
that the template isn't registered
```

```
      lcd.clear();                          //And here we write a message
or take an action for the denied template
      lcd.print("Access denied");
      delay(2000);
      lcd.clear();
      lcd.print("Place finger");
      return -1;
   }
   //If we found a match we proceed in the function
    lcd.clear();
      lcd.setCursor(0, 0);
      lcd.write("COMPLETED! ");
      lcd.setCursor(0, 1);
      lcd.write("Found match. ID");
      lcd.setCursor(15, 1);
      lcd.print(finger.fingerID, DEC);
      delay(5000);

   lcd.clear();
      lcd.setCursor(0, 0);
      lcd.write("INITIALISING...");
      lcd.setCursor(0, 1);
      lcd.write("Password mode!");
  delay(3000);


   ij=0;                                 //ij is set to 0
   lcd.clear();
   lcd.setCursor(0,0);
   lcd.print("Enter Password");
   getPassword();                    //Getting the passcode function
   if(ij==4)
   { //If the passcode is correct we can start enrolling new finger
template
      digitalWrite (lockOutput, HIGH );
        lcd.clear();
      lcd.setCursor(0, 0);
      lcd.write("       WELCOME      ");
      lcd.setCursor(0, 1);
      lcd.write("Identity no.      #");
```

```
        lcd.setCursor(12, 1);
        lcd.print(finger.fingerID, DEC);
        delay(5000);
          digitalWrite (lockOutput, LOW );



    }
    else{                               //If the code is wrong we can't add new
users (templates)
                    lcd.setCursor(0,0);
                    lcd.print("Wrong code");
                    delay(2000);
                    lcd.clear();
    lcd.clear();
    }
    lcd.print("Place finger");
    return finger.fingerID;

}
```