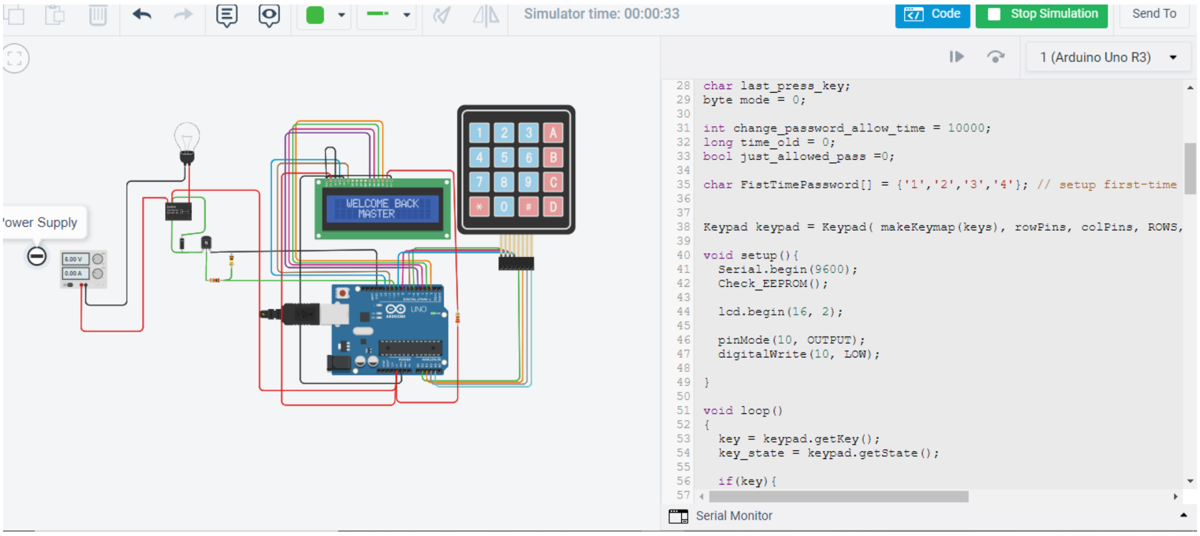

```
char last_press_key;
byte mode = 0;

int change_password_allow_time = 10000;
long time_old = 0;
bool just_allowed_pass =0;

char FistTimePassword[] = {'1','2','3','4'}; // setup first-time


Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS,

void setup(){
  Serial.begin(9600);
  Check_EEPROM();

  lcd.begin(16, 2);

  pinMode(10, OUTPUT);
  digitalWrite(10, LOW);

}

void loop()
{
  key = keypad.getKey();
  key_state = keypad.getState();

  if(key){
```

```
#include <LiquidCrystal.h>
#include <Keypad.h>
#include <EEPROM.h>
#define Password_Lenght 5

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {6, 7, 8, 9};
byte colPins[COLS] = {A1, A2, A3, A4};

char Data[Password_Lenght];
char Data2[Password_Lenght];
char Data3[Password_Lenght];
char Master[Password_Lenght];
byte data_count = 0, master_count = 0;
bool Pass_is_good;
    char key;
byte key_state = 0;
char last_press_key;
byte mode = 0;

int change_password_allow_time = 10000;
long time_old = 0;
bool just_allowed_pass =0;

char FistTimePassword[] = {'1','2','3','4'}; // setup first-time password here
```

```
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

void setup(){
  Serial.begin(9600);
  Check_EEPROM();

  lcd.begin(16, 2);

  pinMode(10, OUTPUT);
  digitalWrite(10, LOW);

}

void loop()
{
  key = keypad.getKey();
  key_state = keypad.getState();

  if(key){
    last_press_key = key;
    Serial.println(key);
  }


  if(mode == 0){
    lcd.setCursor(1,0);
    lcd.print("Enter Password");
    if(last_press_key == '#' && key_state == 2){
      mode = 1;
    }

  }else if(mode == 1){
      lcd.clear();
      lcd.setCursor(2,0);
      lcd.print("Old Password");
      mode = 3;
    }
    else if(mode == 4){
    lcd.setCursor(0, 0);
    lcd.print("Set New Password");
  }

    else if(mode == 2){
    lcd.setCursor(1,0);
    lcd.print("Password Again");

  }
```

```
if (key && key != '#')
{
  collectKey();
}

if(data_count == Password_Lenght-1)
{
   if(mode == 3){
     lcd.clear();
    if(!strcmp(Data, Master)){
      clearData();
      mode = 4;

    }
  else{
      lcd.setCursor(4, 0);
      lcd.print("PASSWORD");
      lcd.setCursor(2, 1);
      lcd.print("NOT MATCHED!");
      mode = 0;
      clearData();
      delay(2000);
      lcd.clear();

  }
}
 else if(mode == 0){
    lcd.clear();
    if(!strcmp(Data, Master)) {
      lcd.setCursor(2, 0);
      lcd.print("WELCOME BACK");
      lcd.setCursor(4, 1);
      lcd.print("MASTER");

      just_allowed_pass = 1;
      time_old = millis();

      if (digitalRead(10) == 0){
      digitalWrite(10, HIGH);
      delay(1000);
      }
      else{
          digitalWrite(10, LOW);
          delay(1000);

      }


    }else{
```

```
    lcd.setCursor(2, 0);
    lcd.print("INCORRECT !");
    lcd.setCursor(4, 1);
    lcd.print("PASSWORD");
    delay(1000);
  }
  delay(1000);//
  lcd.clear();
  clearData();

}else if( mode == 4){
  lcd.clear();
  mode = 2;
  for(int i = 0; i < Password_Lenght; i = i+1){
  Data2[i] = Data[i];
 }
  clearData();
}else if(mode == 2){
  if(!strcmp(Data, Data2)){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("New Password is ");
    lcd.setCursor(4, 1);
    lcd.print(Data);
    delay(1000);
    lcd.clear();
    lcd.setCursor(4, 0);
    lcd.print("Saving...");
    for(int i =0; i <= 100; i =  i+10){
      lcd.setCursor(4, 1);
      lcd.print(i);
      lcd.setCursor(7, 1);
      lcd.print("%");
      delay(200);
    }
    EEPROM.put(0, Data);
    EEPROM.get(0, Master);
    delay(500);
  }else{
    lcd.clear();
    lcd.setCursor(4, 0);
    lcd.print("PASSWORD");
    lcd.setCursor(2, 1);
    lcd.print("NOT MATCHED!");
    delay(2000);
  }
  mode = 0;
  clearData();
  lcd.clear();
}
```

```
  }
  }

void collectKey(){
  Data[data_count] = key;
  lcd.setCursor(5+data_count,1);
  lcd.print("*");
  data_count++;
}

void clearData()
{
  while(data_count !=0)
  {
    Data[data_count--] = 0;
  }
}


void Check_EEPROM(){
  EEPROM.get(0, Master);
  if(Master[0] == 0 && Master[1] == 0 && Master[2] == 0 && Master[3] == 0){ // check if EEPRM have store password ?
    Serial.println("No EEPROM PASSWORD FOUND"); // if not found will burn EEPROM a first time password
    EEPROM.put(0, FistTimePassword);
    EEPROM.get(0, Master);
  }
}
```