Password Protected Security System

Introduction

This project is a simple password-protected security system using an Arduino, keypad, LEDs, and a piezo buzzer. It checks for the correct passcode entered via the keypad and provides visual and audio feedback.

Key Components

- Arduino UNO
- 4x4 Matrix Keypad
- Red LED
- Green LED
- Piezo Buzzer
- Jumper Wires
- Breadboard

Working Principle

1. **Startup:**
   - The system initializes by turning on the red LED and displaying "Enter Passcode:" on the Serial Monitor.
2. **Entering Password:**
   - User inputs the 4-digit password using the keypad.
   - Each keypress is masked with '*' on the Serial Monitor.
3. **Password Validation:**
   - Once 4 digits are entered, the system checks the entered code.
   - If the password matches the preset code ("6556"):
     - Green LED turns on.

        - A short beep sounds from the piezo buzzer.
        - "Enter" is displayed on the Serial Monitor.
    - If the password is incorrect:
        - Red LED remains on.
        - Two long beeps are sounded.
        - "Wrong Passcode" is displayed on the Serial Monitor.

4. **Reset:**
    - After the attempt (correct or wrong), the system resets and prompts for passcode input again.

Circuit Overview

- **Keypad:**
    - Rows connected to Arduino pins 9, 8, 7, 6.
    - Columns connected to Arduino pins 5, 4, 3, 2.
- **Red LED:** Connected to digital pin 10.
- **Green LED:** Connected to digital pin 11.
- **Piezo Buzzer:** Connected to digital pin 12.
- **Common ground and Vcc:** All components share a common ground and appropriate power supply.

Arduino Code

```
/*Code written by-

Fuad Hasan

BME, KUET */


/*Password protected Security System*/
```

```
#include<Keypad.h>

const byte row = 4;
const byte col = 4;
const int redLed = 10;
const int greenLed = 11;
const int piezo = 12;

char numPad[row][col] = {
 {'1', '2', '3'},
 {'4', '5', '6'},
 {'7', '8', '9'},
 {'*', '0', '#'}
};

byte rowPin[row] = {9, 8, 7, 6};
byte colPin[col] = {5, 4, 3, 2};

String password = "6556";
String vstup = "";

Keypad cKeypad = Keypad(makeKeymap(numPad), rowPin, colPin, row, col);

void setup()
{
  pinMode(redLed, OUTPUT);
 pinMode(greenLed, OUTPUT);
 pinMode(piezo, OUTPUT);
 digitalWrite(redLed, HIGH);
 Serial.begin(9600);
 Serial.print("Enter Passcode: ");
}

void loop()
{
 char cKey = cKeypad.getKey();

 if (cKey){
   if(vstup.length() < 4){
     Serial.print("*");
     vstup += cKey;
   }
 }

 if(vstup.length() == 4) {
   delay(1500);
   if(password == vstup) {
     Serial.println("\nEnter");
     digitalWrite(redLed, LOW);
     digitalWrite(greenLed, HIGH);
     tone(piezo, 500);
     delay(100);
```

```
   noTone(piezo);
  }
  else{
   Serial.println("\nWrong Passcode");
   digitalWrite(redLed, HIGH);
   digitalWrite(greenLed, LOW);
   tone(piezo, 1000);
   delay(800);
   tone(piezo, 1000);
   delay(800);
   noTone(piezo);
  }
  delay(1500);
  vstup = "";
  Serial.println("Enter Passcode: ");
  digitalWrite(redLed, HIGH);
  digitalWrite(greenLed, LOW);
 }
}
```

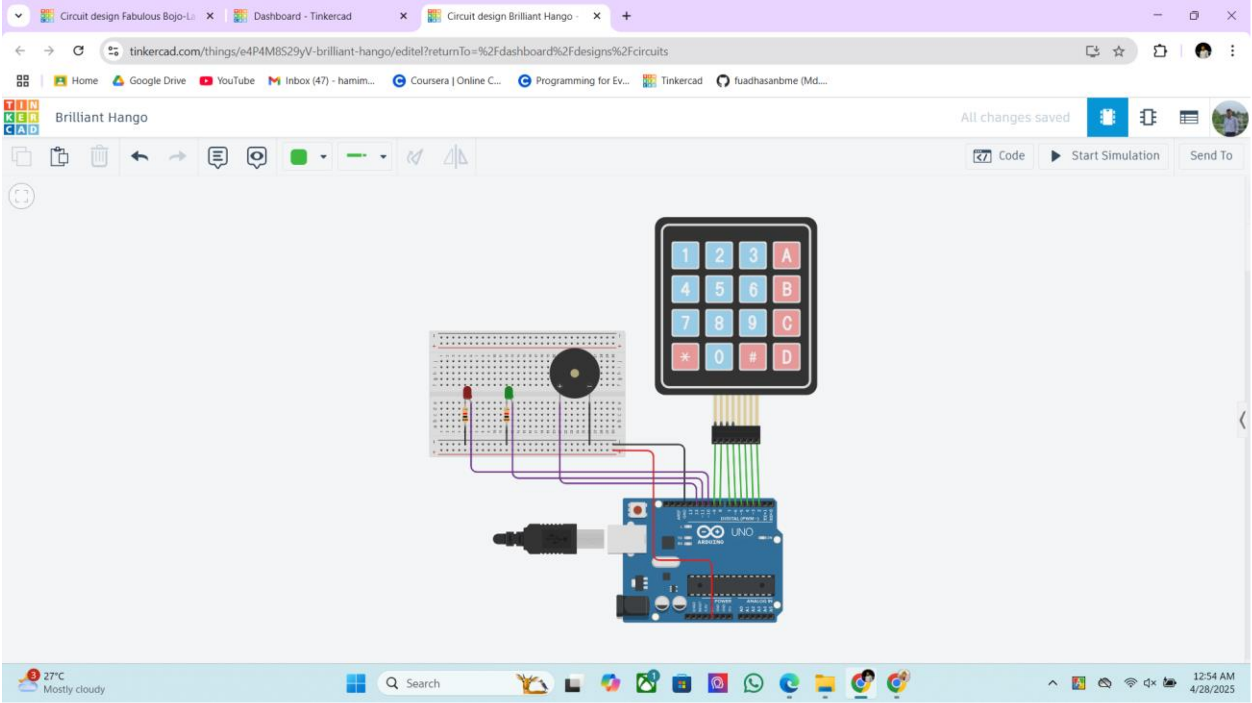
Code Explanation

- **Library Inclusion:**
  - #include<Keypad.h>: Includes the keypad library to simplify keypad handling.
- **Pin Definitions:**
  - Defines rows, columns, and LED/buzzer pins.

- **Keypad Setup:**
  - numPad array represents the keypad layout.
  - rowPin and colPin arrays define which Arduino pins connect to the keypad rows and columns.
- **Password Setup:**
  - password variable holds the correct 4-digit code ("6556").
  - vstup stores user-entered digits.
- **Keypad Initialization:**
  - cKeypad is created using Keypad constructor, mapping keys and pins.
- **Setup Function:**
  - Sets pin modes.
  - Turns on the red LED (indicating locked state).
  - Starts Serial Monitor communication.
- **Loop Function:**
  - Waits for keypad input.
  - Each keypress adds a masked * to Serial Monitor and appends character to vstup.
  - After 4 digits entered:
    - Waits briefly.
    - Compares input with password.
    - If correct: turns green LED on and makes a short beep.
    - If wrong: keeps red LED on and makes two long beeps.
  - Resets the input buffer and returns to waiting for next entry.

Conclusion

This Arduino-based password protected security system demonstrates a basic authentication method using a keypad and provides both visual and audio feedback. It is a perfect beginner-friendly project to learn about keypads, LEDs, buzzers, and conditional programming with Arduino.


Circuit design Fabulous Bojo-La
Dashboard - Tinkercad
Circuit design Brilliant Hango
tinkercad.com/things/e4P4M8S29yV-brilliant-hango/editel?returnTo=%2Fdashboard%2Fdesigns%2Fcircuits
Home
Google Drive
YouTube
Inbox (47) - hamim...
Coursera | Online C...
Programming for Ev...
Tinkercad
fuadhasanbme (Md....
TIN KER CAD
Brilliant Hango
All changes saved
Code
Start Simulation
Send To
1 2 3 A
4 5 6 B
7 8 9 C
* 0 # D
UNO
ARDUINO
27°C
Mostly cloudy
Search
12:54 AM
4/28/2025