

SISTEM DE SIGURANȚĂ REALIZAT CU ARDUINO

Denis DUNĂ, Alin GOANȚĂ

Facultatea de Inginerie Hunedoara, Universitatea Politehnică Timișoara, Romania

Rezumat: Aplicația prezentată în această lucrare descrie funcționarea un sistem de siguranță care permite utilizatorului să fie avertizat la introducerea unei parole greșite utilizând o tastatură cu 4x4 butoane și o placă de dezvoltare Arduino.

Cuvinte cheie: cod parolă, keypad 4x4, LCD display, Arduino

Introducere

Sistemele embedded au devenit omniprezente în activitățile cotidiene. Plăci de dezvoltare atât de populare ca Arduino sunt disponibile pe piață în diverse tehnologii. Cu ajutorul lor și a unor compilatoare adecvate, este la îndemână pentru pasionați să își urmeze un hobby în electronică sau programare.

1. Descrierea părții hardware a aplicației

Schema bloc a aplicației hardware este prezentată în Figura 1. În continuare sunt prezentate caracteristicile tehnice și modul de funcționare a componentelor aplicației.

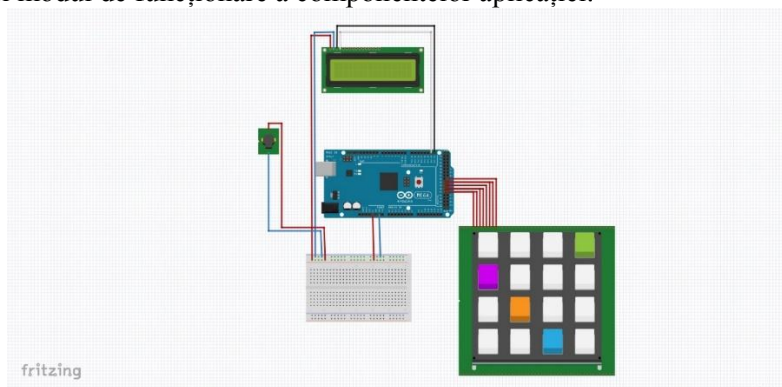


Figura 1. Schema bloc a aplicației hardware.

1.1. Placa de dezvoltare Arduino

Arduino este un microcontroler open-source format dintr-o singură placă, proiectată să faciliteze crearea aplicațiilor în diverse proiecte din diferite domenii. Partea hardware constă dintr-un design simplu cu un microcontroler Atmel AVR și porturi I/O încorporate (Figura 2). Software-ul de programare este bazat pe un compilator standard al limbajului de programare C++.

Arduino este bazat pe procesoarele Atmega de la Atmel și poate fi programat foarte ușor folosind IDE-ul oferit.



Figura 2. Placă de dezvoltare Arduino.

1.2. Keypad 4x4

Tastatura keypad 4x4 are 16 taste conectate matricial. Are o mufă cu 8 pini care se conectează la sistemul de dezvoltare. După cum se vede în Figura 3, cei 8 pini reprezintă semnalele digitale aplicate liniilor și coloanelor matricii. Tastatura este utilizată pentru introducerea codului de acces.

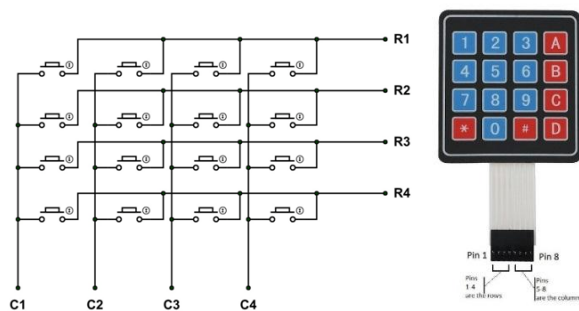


Figura 3. Keypad 4x4.

1.3. Dispozitiv de afișare LCD alfanumeric

În vederea afișării parolei și pentru avertizarea cu privire la corectitudinea acesteia, se utilizează un dispozitiv de afișare alfanumeric cu 2x16 caractere (Figura 4). Acest modul este conectat la sistem la magistrala I2C utilizând pinii SCL și SDA ai plăcii de dezvoltare Arduino. Principalele caracteristici ale magistralei I2C sunt:

- este o magistrală serială, bidirecțională, alcătuită doar din 2 linii: SDA (Serial Data) și SCL (Serial Clock);
- un modul compatibil I²C are înglobată o interfață care îi permite conectarea directă la cele 2 linii ale interfeței; astfel, la nivelul magistralei nu vor mai fi necesare nici un fel de circuite pentru funcționarea corectă a sistemului;
- fiecare modul conectat la magistrală este adresabil prin program printr-o singură adresă;
- modulele pot fi transmițătoare sau receptoare;



Figura 4. LCD alfanumeric

1.4. Alarmă auditivă piezobuzzer

Alarma auditivă la tastarea incorectă a parolei este realizată cu modul piezobuzzer (Figura 5). Acest dispozitiv se alimentează cu un semnal dreptunghiular cu lățime de puls variabilă. Prin variația lățimii pulsului se poate varia volumul sunetului, iar prin variația frecvenței se variază tonalitatea.

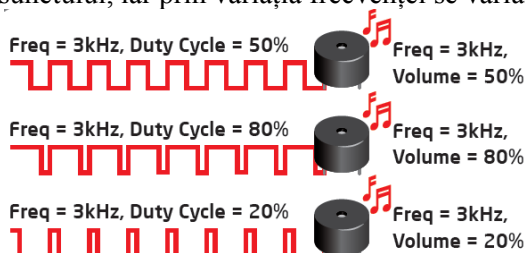


Figura 5. Piezobuzzer – diagrama de timp a generării sunetelor.

2. Descrierea părții software a aplicației

Codul sursă include librăriile *Wire.h* pentru utilizarea magistralei I2C, *LiquidCrystal_I2C.h* pentru funcționarea dispozitivului de afișare și *Keypad.h* pentru funcționarea tastaturii.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>
```

```

#define Password_Length 8
int signalPin = 12;
char Data[Password_Length];
char Master[Password_Length] = "123A456";
byte data_count = 0, master_count = 0;
bool Pass_is_good;
char customKey;
const byte ROWS = 4;
const byte COLS = 4;
char hexaKeys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

byte rowPins[ROWS] = {2,3,4,5}; //rand 0-3
byte colPins[COLS] = {6,7,8,9}; //col 0-3

Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);

LiquidCrystal_I2C lcd(0x3F, 16, 2);

void setup(){
  lcd.begin();
  lcd.backlight();
  pinMode(signalPin, OUTPUT);
}

void loop(){

  lcd.setCursor(0,0);
  lcd.print("Introduceti parola:");

  customKey = customKeypad.getKey();
  if (customKey){
    Data[data_count] = customKey;
    lcd.setCursor(data_count,1);
    lcd.print(Data[data_count]);
    data_count++;
  }

  if(data_count == Password_Length-1){
    lcd.clear();

    if(!strcmp(Data, Master)){
      lcd.print("Corect");
      digitalWrite(signalPin, HIGH);
      delay(5000);
      digitalWrite(signalPin, LOW);
    }
    else{
      lcd.print("Incorect");
      delay(1000);
    }
  }
}

```

```
    lcd.clear();
    clearData();
}
}

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

Concluzii

Aplicația prezentată este un sistem de siguranță foarte des întâlnit. În acest caz sistemul presupune programarea unei plăci de dezvoltare Arduino. Avantajul față de un sistem clasic îl constituie posibilitatea de îmbunătățire din punct de vedere al componentelor și al securității. Utilizatorul poate oricând să reprogrameze sistemul, adăugând diverse facilități.

Bibliografie

1. <https://www.instructables.com/id/Arduino-Keypad-4x4-Tutorial/>
2. <https://www.openhacks.com/uploadsproductos/eone-1602a1.pdf>