

Sistem de Securitate pentru o mini căsuţă

Proiect

Arhitectura şi organizarea calculatoarelor

Îndrumător: Şef lucr.dr.ing Carp Marius-Cătălin

Student: Grigoraş Victor-Emil

Grupa: 4LF291

Program de studii: Electronică Aplicată

2021 – 2022

Cuprins

1.REZUMAT	3
1.1.DESCRIEREA PROIECTULUI.....	3
1.2.DESCRIEREA IMPLEMENTĂRII	3
2.INTRODUCERE.....	4
2.1.PREZENTAREA TEMEI	4
2.1.SISTEME ACTUALE.....	4
3.PREZENTAREA HARDWAREULUI	5
1.PREZENTAREA HARDWAREULUI	5
3.1.CARACTERISTICI TEHNICE ALE PLĂCII DE DEZVOLTARE	6
3.2.CARACTERISTICI TEHNICE ALE COMPONENTELOR FOLOSITE	7
3.2.1.Modul senzor de distanță ultrasonic.....	7
3.2.2.Buzzer activ și pasiv	7
3.2.3Micro Servomotor SG90.....	8
2.ARHITECTURA SISTEMULUI	9
3.PREZENTAREA SOFTWAREULUI.....	10
4.PREZENTAREA MONTAJULUI REALIZAT	11
6.1.PREZENTAREA FUNCȚIONALITĂȚII	12
7.CONCLUZII	14
8.BIBLIOGRAFIE.....	14
9.ANEXA – COD SURSĂ.....	15

1. REZUMAT

1.1. DESCRIEREA PROIECTULUI

Acest proiect reprezintă un sistem de securitate pentru o mini căsuță, pe care bineînțeles îl putem aplica și unei case adevărate foarte ușor. La acest proiect am folosit o placă de dezvoltare Arduino Uno R3, un senzor ultrasonic HC-SR04 care declanșează alarma atunci când un obiect sau chiar un individ intră în raza lui de acțiune, un buzzer activ și unul pasiv, un servomotor SG90 pentru a arma sau dezarma căsuța și o tastatură matricială 3x3 care mai are încă 2 butoane, unul pentru a deschide ușa căsuței și un buton pentru modificarea codului pin.

1.2. DESCRIEREA IMPLEMENTĂRII

Baza proiectului este placa de dezvoltare Arduino Uno R3, cu un microcontroller ATmega328p. Comunicarea dintre placa de dezvoltare și celelalte componente se realizează prin porturile digitale 2,3,4,5,6,7,8,9,10,11,12,13.

- Porturile 2, 3, 4, 5, 6, 7 sunt legate la tastatura matricială 3x3.
- Portul 8 este legat la ieșirea buzzer-ului.
- Porturile 9 și 10 sunt legate la butonul destinat modificării codului pin al sistemului, respectiv la butonul de deschidere al ușii.
- Portul 11 este legat la servomotor.
- Porturile 12 și 13 sunt legate la pinul echo respectiv pinul trigger al senzorului ultrasonic

2. INTRODUCERE

2.1. PREZENTAREA TEMEI

Tema principală pe care m-am concentrat în desfășurarea proiectului este dezvoltarea și implementarea unui sistem de securitate pentru o casă. Sistemul este unul destul de simplu, care se bazează pe armarea și dezarmarea casei introducând pinul de acces, dar și detectarea unor obiecte în apropierea casei.

2.1. SISTEME ACTUALE



Fig. 1. Kit sistem alarma wireless Ajax Start

Kit-ul de securitate are o unitate de control care comunica cu toate dispozitivele sistemului de securitate și colectează informații de la acestea, le analizează iar în caz de alarmă notifică utilizatorii și compania de securitate. Acest kit mai are în componență și un senzor de mișcare fără fir care este un element esențial al sistemului de alarmă, acesta detectează intrușii încă de la primii pași. Există și aplicația Ajax Security Systems care permite să armezi sau să dezarmezi total sau parțial sistemul și să fii la curent cu funcționalitatea dispozitivelor (inclusiv informații despre starea semnalului, nivelul bateriei și temperatura).

3. PREZENTAREA HARDWAREULUI

1. PREZENTAREA HARDWAREULUI

Microcontrollerul care stă la baza sistemului implementat este un Arduino Uno R3. Acesta conține un microchip denumit ATmega328P. Arduino este o companie open-source care produce atât plăcuțe de dezvoltare bazate pe microcontrolere, cât și partea de software destinată funcționării și programării acestora. Pe lângă acestea include și o comunitate uriașă care se ocupă cu creația și distribuirea de proiecte care au ca scop crearea de dispozitive care pot sesiza și controla diverse activități sau procese în lumea reală.

Aceste plăci pun la dispoziția utilizatorului pini I/O, digitali și analogici, care pot fi interfațați cu o gamă largă de plăcuțe numite scuturi (shield-uri) și/sau cu alte circuite. Plăcile au interfețe de comunicații seriale, inclusiv USB pe unele modele, pentru a încărca programe din calculatoarele personale. Pentru programarea microcontrolerelor, Arduino vine cu un mediu de dezvoltare integrat (IDE) care include suport pentru limbaje de programare ca C și C++.

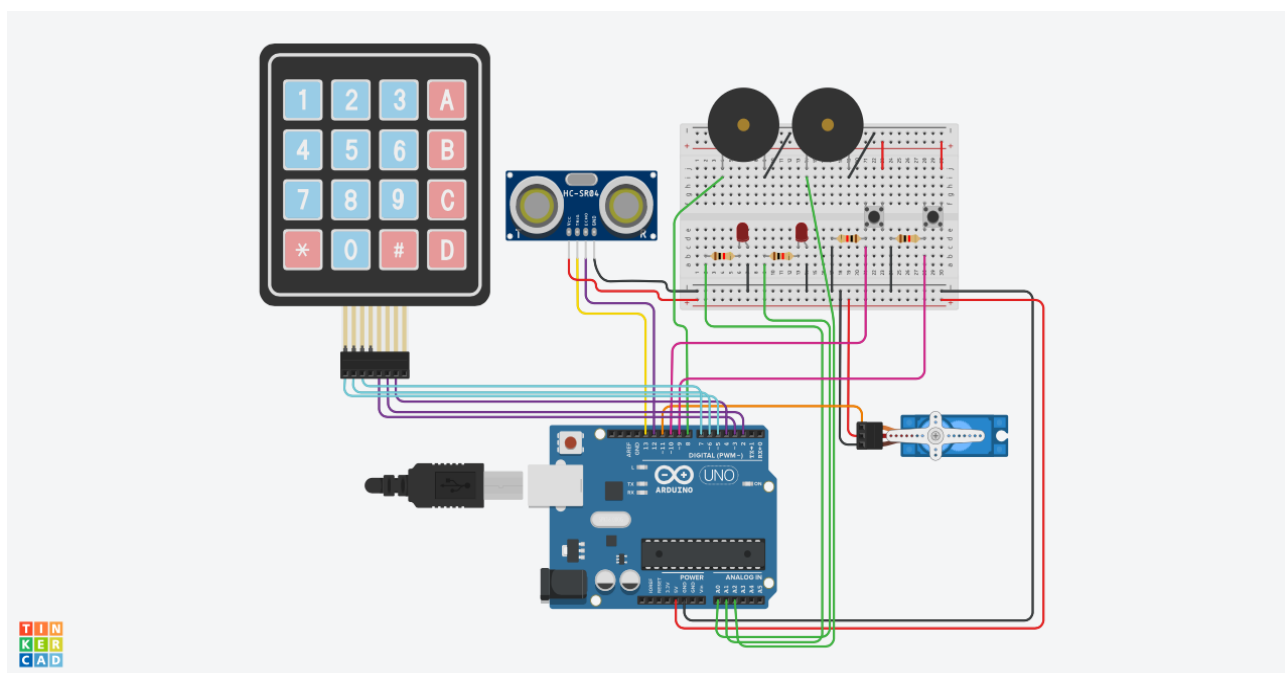


Fig. 2. Schema pentru realizare

3.1. CARACTERISTICI TEHNICE ALE PLĂCII DE DEZVOLTARE

Arduino Uno este o placă de dezvoltare bazată pe microchipul ATmega328P. Are 14 porți digitale pentru intrare/ieșire, din care 6 pot furniza ieșire PWM, are 6 pini analogi de intrare. Placa are interfața de comunicație serială, inclusiv USB, pentru a încărca codul de sursă în microcontroller. Are un mediu de dezvoltare integrat (IDE), care include suport pentru programare C și C++.

Caracteristicile:

- Microcontroller: ATmega328P
- Tensiune de funcționare: 5 Volți
- Tensiune de intrare (recomandat): 7-12 Volți
- Pini I/O digitali: 14 (din care 6 pot furniza ieșire PWM)
- Pini de intrare analogici: 6
- Curent continuu pentru un pin I/O: 40 mA
- Curent continuu pentru pinul de 3,3 V: 50 mA
- Memorie Flash: 32 KB din care 0,5 KB utilizată de bootloader
- SRAM: 2 KB
- EEPROM: 1 KB
- Viteza ceasului: 16 MHz

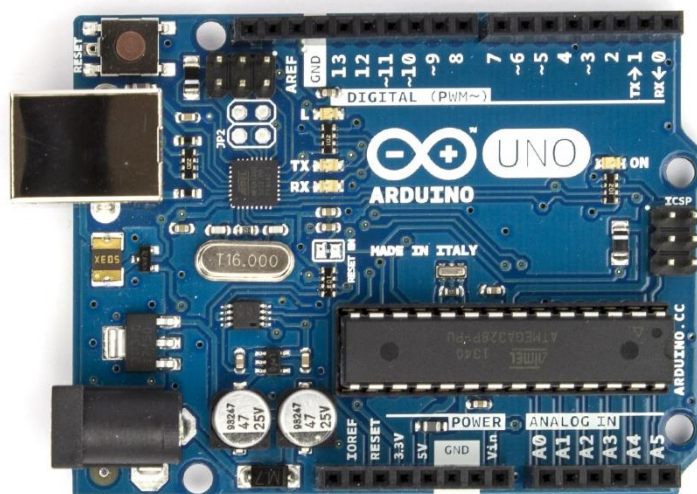


Fig. 3. Arduino Uno R3

3.2. CARACTERISTICI TEHNICE ALE COMPONENTELOR FOLOSITE

3.2.1. Modul senzor de distanță ultrasonic

Modulul de senzor de distanță ultrasonic HC-SR04 măsoară distanța pe o scală de 2cm-400cm, fără contact, cu o precizie mare.

Caracteristicile:

- Tensiune de funcționare: 5 V
- Curent de funcționare: 15 mA
- Frecvența de funcționare: 40Hz
- Distanța maximă: 400 cm
- Distanța minimă: 2 cm
- Unghiul de măsurare: 15°



Fig. 4. Modul de senzor ultrasonic

3.2.2. Buzzer activ și pasiv

Caracteristicile:

- Tensiune recomandată: 3V
- Tensiune de funcționare: 2-5V
- Curent de funcționare: 30mA(MAX)
- Minim 80dB, la distanța de 10 cm
- Frecvență de rezonanță: 2300-+300Hz



Fig. 5. Buzzer activ și pasiv

3.2.3 Micro Servomotor SG90

Caracteristicile:

- Tensiune de alimentare: 4.8-6V;
- Consum redus de current
- Frecvență PWM: 50 Hz
- Temperatura de funcționare: -30 °C --+ 60 °C



Fig. 6. *Micro Servomotor SG90*

2. ARHITECTURA SISTEMULUI

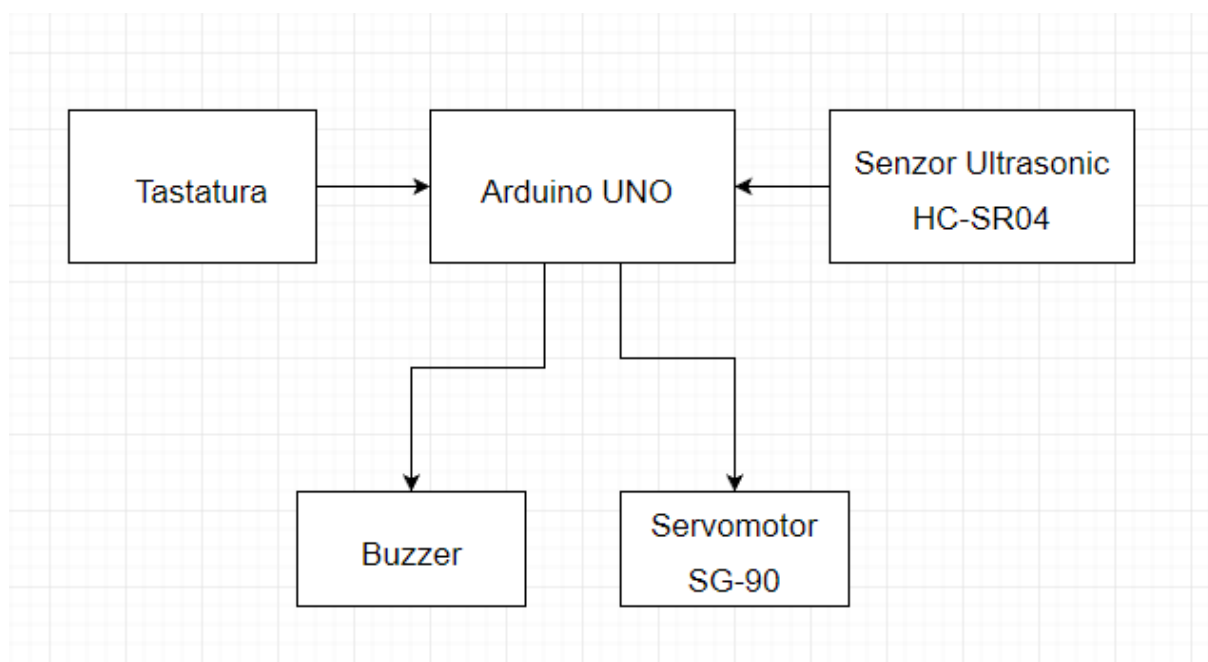


Fig. 7. Schema bloc a circuitului

Senzorul comunică prin protocolul UART(Universal Asynchronous Recive Transmit). Necesită două fire de comunicație pentru fiecare direcție (intrare în Arduino – echo, ieșire din arduino – trigger).

3. PREZENTAREA SOFTWAREULUI

Arduino Integrated Development Environment(IDE) este o aplicație cross-platform, open-source care conține un editor de text pentru scrierea codului în limbaj C/C++, o zonă de afișare a mesajelor, o consolă text și o bară de instrumente pentru funcții comune și o serie de meniuri. Programele scrise folosind acest mediu se numesc schițe și se salvează cu extensia .ino.

Acesta se conectează la plăcuțe Arduino sau Genuino pentru a încărca programe și a comunica cu ele. Când o schiță este încărcată este folosit de fapt Arduino Bootloader, un program mic care a fost încărcat pe microcontrolerul de pe plăcută. Acesta permite încărcarea de cod fără a mai folosi hardware adițional.

Pentru a se construi programul sunt necesari mai mulți pași. Acțiuni de pre-procesare sunt utilizate pentru a transforma schița într-un program C/C++. Apoi sunt localizate dependențele schiței, iar apoi codul este pasat unui compilator avr-gcc/avr-g++. Următorul pas este că acest cod să fie combinat cu librării standard Arduino care să asigure funcționalități de bază. Rezultatul este un fișier hex, care conține byții specifici care trebuie scriși pe chip ul de memorie al plăcuței Arduino.

AVR-GCC este un compilator care preia cod scris în limbaj de nivel înalt C și creează o sursă binară ce poate fi încărcată într-un microcontroler AVR

Mediul de dezvoltare Arduino se folosește de librari pentru a asigura funcționalități extra în dezvoltarea schițelor. Acestea sunt încărcate pe plăcută împreună cu schița, astfel ocupând un spațiu mai mare pe această.

În codul meu de sursă am folosit trei librării, Wire.h, Keypad.h si Servo.h.

4. PREZENTAREA MONTAJULUI REALIZAT

Acest montaj reprezintă un sistem de securitate pentru o mini casuta. Atunci cand casuta este armata si senzorul ultrasonic detecteaza un obiect, alarma va suna de 3 ori.

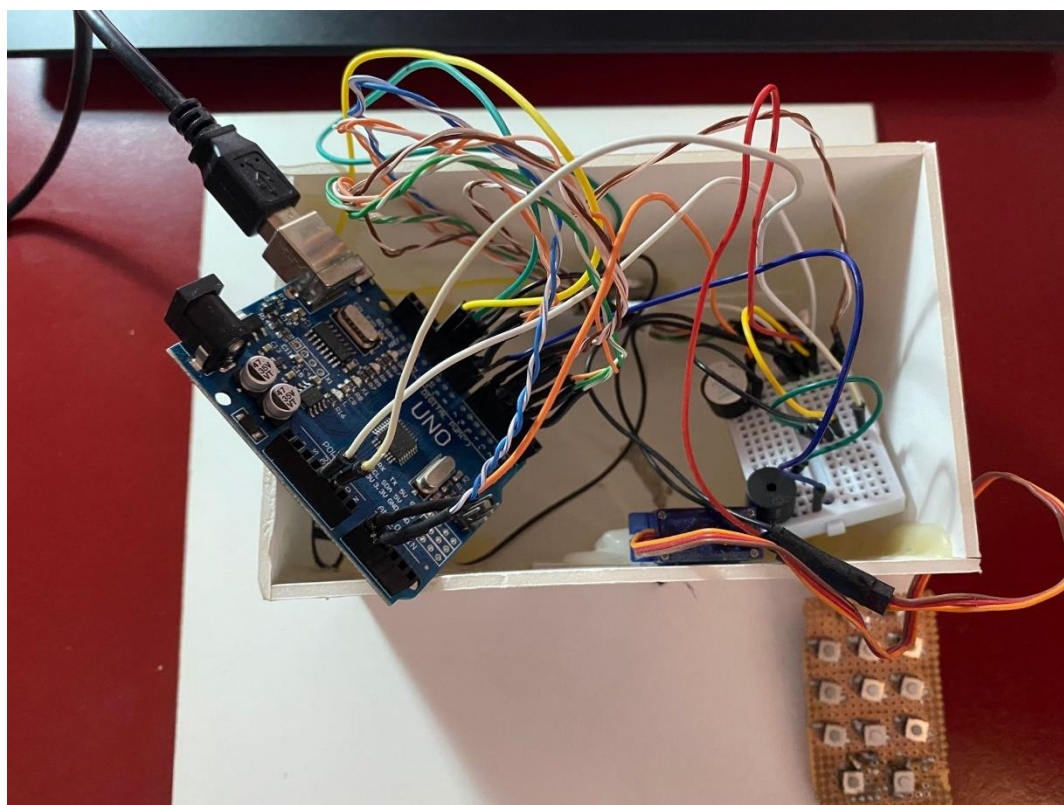


Fig. 8. Montajul realizat.

Pentru realizarea motajului am folosit un senzor ultrasonis HC-SR04, placa de dezvoltare Arduino, Buzzer activ si pasiv, Servomotor SG-90 si o tastatura matriciala 3x3 care mai are 2 butoane pentru deschiderea usitei, respective pentru resetarea pinului de acces.

PERFORMANȚE

Sistemul realizat este unul destul de simplu. Sistemul funcționează după următorul principiu:

Cand porneste sistemul, acesta porneste dezarmat, pentru a arma sistemul trebuie sa introducem pinul (1234). Dupa ce introducem pinul, usa se inchide si sistemul de securitate devine activ (ledul din partea dreapta al tastaturii se aprinde). Daca cineva intra in raza de actiune a senzorului de distanta se declanseaza alarma (suna de 3 ori). Pentru a dezarma casuta repetam aceeasi pasi (pinul), usa se deschide, iar sistemul devine inactiv (se aprinde ledul din partea stanga).

6.1. PREZENTAREA FUNCȚIONALITĂȚII

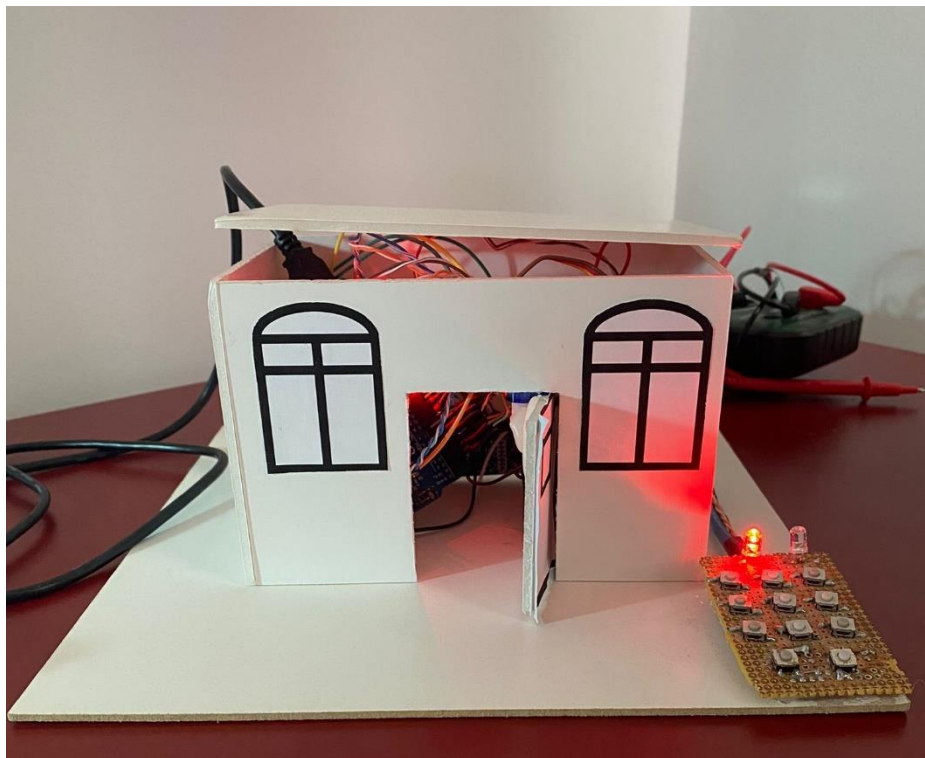


Fig. 9. Casa dezarmata(led-ul din stanga al tastaturii se aprinde)

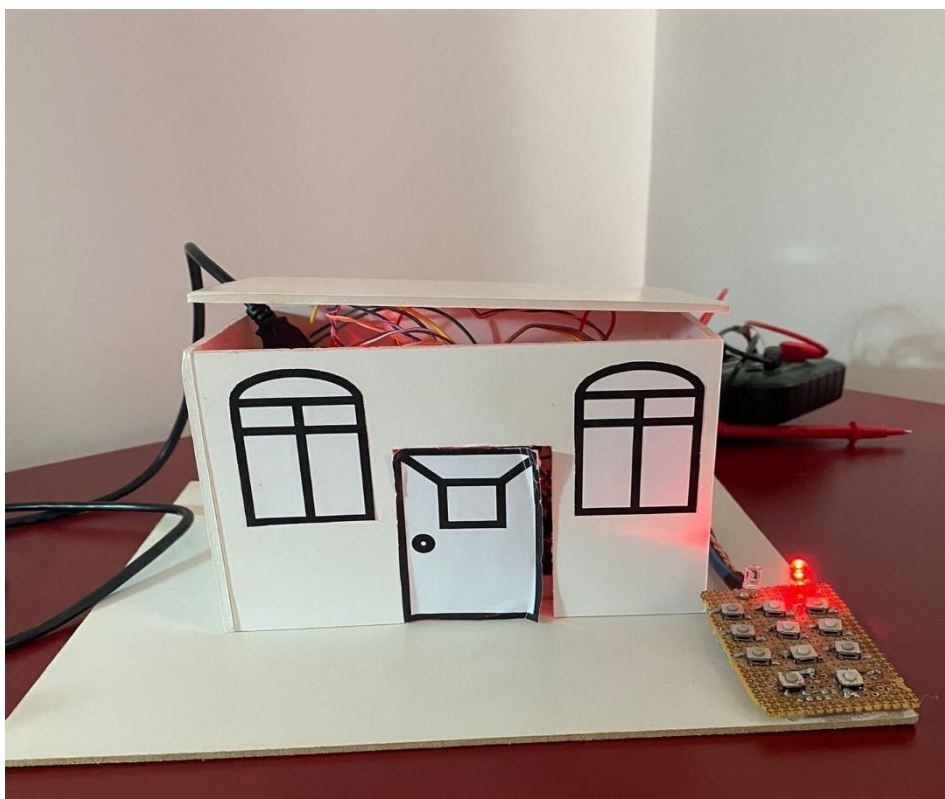


Fig. 10. Casă armată(led-ul din dreapta al tastaturii se aprinde)

Cand dorim să schimbăm pinul sistemului de securitate apăsăm pe butonul destinat pentru acest lucru. Intorducem pinul vechi, iar dacă am introdus pinul corect acest lucru ne este semnalat printr-un semnal sonor (dacă nu este corect procesul se închide). Introducem noul pin, îl mai introducem încă odată pentru confirmare. Dacă introducem pinul greșit de 3 ori se declanșează alarma

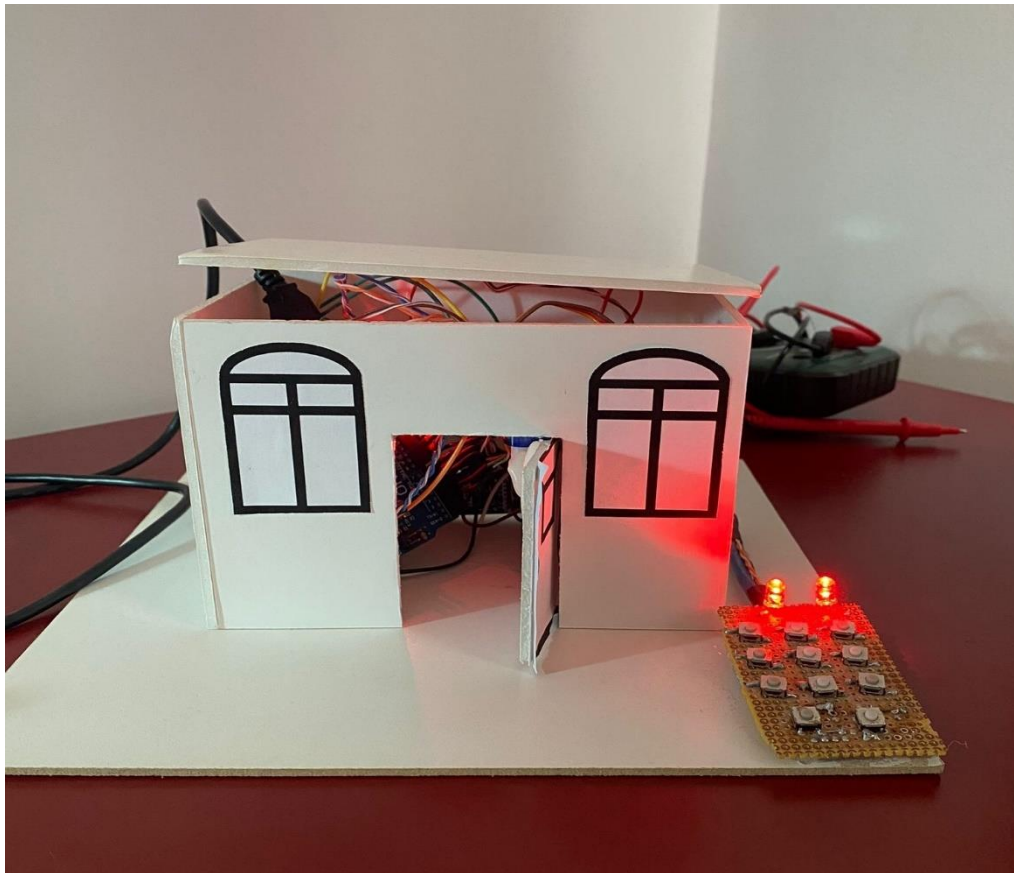


Fig. 11. Schimbarea pinului

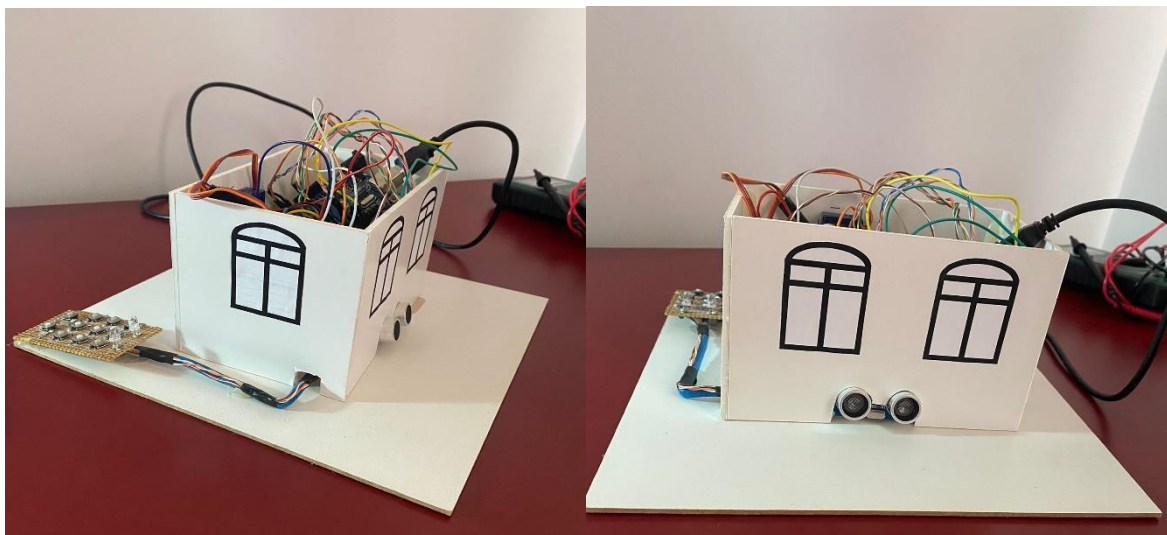


Fig. 12. Casa lateral/spate

7. CONCLUZII

Sistemul prezentat este unul dintre cele mai simple sisteme de securitate. Acest sistem mai poate fi dezvoltat și cu o aplicație mobilă care să permită armarea sau dezarmarea sistemului de la distanță prin conexiune wireless.

8. BIBLIOGRAFIE

- [1] Arduino Uno R3
<https://www.farnell.com/datasheets/1682209.pdf>
- [2] An introduction to arduino uno pinout
<https://www.circuito.io/blog/arduino-uno-pinout/>
- [3] Senzorul ultrasonic
<https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>
- [4] Buzzer activ
https://www.mouser.com/datasheet/2/400/ef532_ps-13444.pdf
- [5] Servomotor SG-90
<https://datasheetpdf.com/pdf/791970/TowerPro/SG90/1>
- [6] Arduino Software
<https://en.wikipedia.org/wiki/Arduino#Software>

9. ANEXA – COD SURSĂ

Codul de sursă al sistemului:

```
const unsigned int TRIG_PIN=13;
const unsigned int ECHO_PIN=12;
const unsigned int BAUD_RATE=9600;

#include <Wire.h>
#include <Keypad.h>
#include <Servo.h>
Servo myservo;

#define lungime_parola 5

//int signalPin = 12;
int arm=A0,deza=A1;
int buzzerA=A2;
int buzz=8;
int dezarmat=1;
int buttonChange=10;
int change=0;
int pos = 0;
int cancelbutton=9;
int gresit=0;
int servom=11;

char Data[lungime_parola];
char Parola[lungime_parola] = "1234";
char newpass[lungime_parola];
char newpassverif[lungime_parola];
char verifpass[lungime_parola];
byte data_count = 0, master_count = 0,newcount=0,newcountverif=0,verifcount=0;
bool Pass_is_good;
char customKey;
int buttonState=0,cancelbuttonState=0;

const byte ROWS = 3;
const byte COLS = 3;

char hexaKeys[ROWS][COLS] = {
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'}
};

byte rowPins[ROWS] = {5, 6, 7};
```

```

byte colPins[COLS] = {2, 3, 4};

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

void setup(){
  Serial.begin(9600);
  pinMode(arm,OUTPUT);
  pinMode(deza,OUTPUT);
  digitalWrite(deza,HIGH);
  pinMode(buzzerA, OUTPUT); //buzzer
  //Serial.println("Enter Password:");
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(buttonChange, INPUT);
  myservo.attach(servom);
  opendoor();
}

void loop(){

  buttonState = digitalRead(buttonChange);
  cancelbuttonState=digitalRead(cancelbutton);
  if(buttonState == HIGH && dezarmat==1)
  {
    change=1;
    digitalWrite(deza,HIGH);
    digitalWrite(arm,HIGH);
    Serial.print("Change");
    digitalWrite(buzzerA,HIGH);
    delay(100);
    digitalWrite(buzzerA,LOW);

  }
  if(change==1 && dezarmat==1)
  {
    //changepass();
    passinit();
  }
  else if(change==2)
  {
    //passverif();
    changepass();
  }
  else if(change==3)
  {
    passverif();
  }
}

```



```

else
{
    parola();
}
if(cancelbuttonState == HIGH)
{

    //Serial.print("Cancel");
    change=0;
    clearData();
    clearData1();
    clearData2();
    clearintverif();
    digitalWrite(buzzerA,HIGH);
    delay(100);
    digitalWrite(buzzerA,LOW);
    digitalWrite(deza,LOW);
    digitalWrite(arm,LOW);
    if(dezarmat==1)
        digitalWrite(deza,HIGH);
    else
        digitalWrite(arm,HIGH);
}

digitalWrite(TRIG_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);

const unsigned long duration= pulseIn(ECHO_PIN, HIGH);
int distance= duration/29/2;
if(duration==0){
    Serial.println("Warning: no pulse from sensor");
}
else{
    //Serial.print("distance to nearest object:");
    Serial.println(distance);
    //Serial.println(" cm");
    if(distance>3 && distance<14 && dezarmat==0){
        //digitalWrite(buzzerA,HIGH);
        siren();
    }
    else{
        //digitalWrite(buzzerA,LOW);
    }
}
}
}

```

```

void clearData1(){
    while(newcount !=0){
        newpass[newcount--] = 0;
    }
    return;
}
void clearData(){
    while(data_count !=0){
        Data[data_count--] = 0;
    }
    return;
}
void clearintverif(){
    while(verifcount !=0){
        verifpass[verifcount--] = 0;
    }
    return;
}
void opendoor()
{
    for (pos = 80; pos >= 0; pos -= 1) {
        myservo.write(pos);
        delay(15);
    }
}
void closedoor()
{
    for (pos = 0; pos <= 80; pos += 1) {
        myservo.write(pos);
        delay(15);
    }
}
void siren()
{
    digitalWrite(buzzerA,HIGH);
    for(int i=0;i<5;i++){
        delay(300);
noTone(buzz);
tone(buzz,494,500);
delay(300);
noTone(buzz);
tone(buzz,523,300);
delay(200);
noTone(buzz);
        digitalWrite(buzzerA,LOW);
    }
}

```

```

}

void parola()
{
    customKey = customKeypad.getKey();
    if (customKey){
        Data[data_count] = customKey;
        Serial.print(Data[data_count]);
        data_count++;
        digitalWrite(buzzerA,HIGH);
        delay(100);
        digitalWrite(buzzerA,LOW);
    }

    if(data_count == lungime_parola-1){

        if(!strcmp(Data, Parola)){
            if(dezarmat==1)
            {
                digitalWrite(arm,HIGH);
                digitalWrite(deza,LOW);
                digitalWrite(buzzerA,HIGH);
                delay(50);
                digitalWrite(buzzerA,LOW);
                dezarmat=0;
                closedoor();
            }
            else
            {
                digitalWrite(arm,LOW);
                digitalWrite(deza,HIGH);
                digitalWrite(buzzerA,HIGH);
                delay(50);
                digitalWrite(buzzerA,LOW);
                dezarmat=1;
                opendoor();
            }
            Serial.print("Correct");
        }
        else
        {
            Serial.print("Incorrect");
            digitalWrite(arm,HIGH);
            digitalWrite(deza,HIGH);
            digitalWrite(buzzerA,HIGH);
            delay(500);
            digitalWrite(arm,LOW);
            digitalWrite(deza,LOW);
            digitalWrite(buzzerA,LOW);
        }
    }
}

```

```

        delay(500);
        digitalWrite(arm,HIGH);
        digitalWrite(deza,HIGH);
        digitalWrite(buzzerA,HIGH);
        delay(500);
        gresit++;
        if(gresit==3)
        {
            gresit=0;
            siren();
        }
        digitalWrite(arm,LOW);
        digitalWrite(deza,LOW);
        digitalWrite(buzzerA,LOW);
        delay(500);

        if(dezarmat==1)
        digitalWrite(deza,HIGH);
        else
        digitalWrite(arm,HIGH);
    }
    clearData();
}
}

void passinit()
{
    customKey = customKeypad.getKey();
    if (customKey){
        verifpass[verifcount] = customKey;
        verifcount++;
        digitalWrite(buzzerA,HIGH);
        delay(100);
        digitalWrite(buzzerA,LOW);
    }
    if(verifcount == lungime_parola-1){
        //change=2;

        if(!strcmp(Parola, verifpass))
        {
            delay(500);
            digitalWrite(deza,LOW);
            digitalWrite(arm,LOW);
            delay(500);
            digitalWrite(deza,HIGH);
            digitalWrite(buzzerA,HIGH);
            delay(500);
            digitalWrite(deza,LOW);

```

```

        digitalWrite(buzzerA, LOW);
        delay(500);
        digitalWrite(deza, HIGH);
        delay(500);
        digitalWrite(deza, LOW);
        digitalWrite(deza, HIGH);
        digitalWrite(arm, HIGH);
        changepass();
        change=2;

    }
    else
    {
        change=0;
        clearintverif();
        delay(500);
        digitalWrite(arm, HIGH);
        digitalWrite(buzzerA, HIGH);
        delay(500);
        digitalWrite(arm, LOW);
        digitalWrite(buzzerA, LOW);
        delay(500);
        digitalWrite(arm, HIGH);
        digitalWrite(buzzerA, HIGH);
        delay(500);
        digitalWrite(arm, LOW);
        digitalWrite(buzzerA, LOW);
    }
}
}

void passverif()
{
    customKey = customKeypad.getKey();
    if (customKey){
        newpassverif[newcountverif] = customKey;
        Serial.print(newpassverif[newcountverif]);
        newcountverif++;
        digitalWrite(buzzerA, HIGH);
        delay(100);
        digitalWrite(buzzerA, LOW);
    }
    if(newcountverif == lungime_parola-1){
        change=0;
        digitalWrite(deza, LOW);
        digitalWrite(arm, LOW);
        if(!strcmp(newpass, newpassverif))
        {
            delay(500);

```

```

    digitalWrite(deza,HIGH);
    digitalWrite(buzzerA,HIGH);
    delay(500);
    digitalWrite(deza,LOW);
    digitalWrite(buzzerA,LOW);
    delay(500);
    digitalWrite(deza,HIGH);
    delay(500);
    digitalWrite(deza,LOW);
    /*digitalWrite(buzzerA,HIGH);
    delay(100);
    digitalWrite(buzzerA,LOW);*/
    for(int i=0;i<lungime_parola;i++)
        Parola[i]=newpass[i];
    }
    else
    {
        delay(500);
        digitalWrite(arm,HIGH);digitalWrite(buzzerA,HIGH);
        delay(500);
        digitalWrite(arm,LOW);
        digitalWrite(buzzerA,LOW);
        delay(500);
        digitalWrite(arm,HIGH);digitalWrite(buzzerA,HIGH);
        delay(500);
        digitalWrite(arm,LOW);
        digitalWrite(buzzerA,LOW);
    }

    if(dezarmat==1){
        digitalWrite(deza,HIGH);
    }
    else
    {
        digitalWrite(arm,HIGH);
    }
    clearData1();
    clearData2();
    clearintverif();

}
}
void clearData2(){
    while(newcountverif !=0){
        newpassverif[newcountverif--] = 0;
    }
    return;
}
void changepass()

```

```

{
  customKey = customKeypad.getKey();
  if (customKey){
    newpass[newcount] = customKey;
    Serial.print(newpass[data_count]);
    newcount++;
    digitalWrite(buzzerA,HIGH);
    delay(100);
    digitalWrite(buzzerA,LOW);
  }
  if(newcount == lungime_parola-1){
    change=3;
    digitalWrite(deza,LOW);
    digitalWrite(arm,LOW);
    delay(500);
    digitalWrite(deza,HIGH);
    digitalWrite(buzzerA,HIGH);
    delay(500);
    digitalWrite(deza,LOW);
    digitalWrite(buzzerA,LOW);
    delay(500);
    digitalWrite(deza,HIGH);
    delay(500);
    digitalWrite(deza,LOW);
    digitalWrite(deza,HIGH);
    digitalWrite(arm,HIGH);
    //for(int i=0;i<lungime_parola;i++)
      // Parola[i]=newpass[i];
    passverif();
    //clearData1();
  }
}
}

```