Arduino Home Security System

Bálint Roland
Dobner Szabolcs
Sasu-Simon Kinga
Varga Zoltán

Students at Technical University of Cluj-Napoca, Computer Science, Group 30414

Abstract. This paper presents an Arduino based multilevel security system. It offers two levels of security using a PIR sensor and a password based digital lock.

1. Introduction

The main concept of this project is to protect our home from unwanted thieves. When the sensor gets a signal or the wrong password is introduced, the Arduino controller sets up the alarm. This sensor network is made up into a single automated architecture for the purpose of practical implementation in smart home system. The advantage of this system is that it is easy to implement and you can modify it to satisfy your needs at any time.

2. Components, working mechanism and conditions

2.1 The full list of components that are used in the project is listed below.

- Arduino Uno
- 2-Channel 5V Relay Module (Black)
- Servomotor, SG92R 9g 2.5 kg.cm, 4.8 V
- 4x4 matrix keyboard
- Pir Motion Sensor Detector Module
- Full size breadboard.
- Resistors.
- Connecting cables.
- Lamp

2.2 Functioning mechanism

This home security system is easy to use and it can provide safety for your home and family. It can be installed on the door, so when somebody is trying to enter must know the correct password or the door won't open and the alarm will remain armed. When the correct password is entered the door is unlocked and the alarm system is disarmed. If the burglar tries to enter on the window or in any other place, he will be caught, because of the motion sensor. The motion sensor will signal the alarm to go off so you will know that somebody entered your home (the alarm will ring every 5 seconds and the light will go on and off).

2.3 Conditions

For the project to work properly the PIR sensor must be placed at most 7 meters from any window or door.

3. The code alongside the problem-solving process

```
// For the project to work properly we will use two libraries which implement basic functions for the
// keypad and the servo motor such as reading pressed keys or rotating the servo motor at a certain
// degree
#include "Keypad.h"
#include "Servo.h"

// as global variables we give names to pins from the Arduino for the legibility of the code
const int buzzer = 12; // pin 12 is linked to the buzzer and the leds
const int light = 11; // pin 11 is linked to the input number 2 of the relay which controls the light
const int pir = 2; // pin 2 is linked to the output of the pir sensor (motion sensor)
const int servo = 13; // pin 13 is linked to the input of the servo motor

// we define a map of the keys from the keypad as a 4x4 matrix
char keys[4][4] =
{
{'1','2','3','a' },
{'4','5','6','b' },
{'7','8','9','c' },
{'*','0','#','d' }
};
byte colPins[4] = {7, 8, 9, 10}; // link the pins from the Arduino to the columns of the keypad
byte rowPins[4] = {3, 4, 5, 6}; // link the pins from the Arduino to the rows of the keypad
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, 4, 4 ); // create our keypad type
                                                                      // object using the above
                                                                     // initialized parameters

Servo myServo; // define an object of type servo in order to control our servo motor

const int PASSWORD_LENGTH = 4; // define the length of the passcode
char password[PASSWORD_LENGTH] = {'0','0','0','1'}; // define the passcode

char keysPressed[8]; // the array in which the pressed keys will be stored
int nrKeysPressed=0; // we will store the number of keys that were pressed to be able to check if the
                     // keys match to the passcode
int isSystemArmed = 0; // indicates the state in which the system is (if it is 0 means the alarm is disarmed,
                       // if it is 1 means the alarm is armed)

// the function will get no parameters
// it will arm the system by turning of the light, closing the door, and by changing the state of the system
void armSystem()
{
 digitalWrite(light, HIGH); // turns of the light
 myServo.write(0); // closes the door
 isSystemArmed = 1; // changes the state of the system
}
```

```
// it will be called when the Arduino is plugged in a power supply
void setup() {

  myServo.attach(servo); // links to the myServo object the pin corresponding to it
  pinMode(light,OUTPUT); // sets the type of the pin controlling the light to output in order to be able
                         // to send signals to it
  pinMode(pir,INPUT); // sets the type of the pin controlling the pir sensor to input in order to be able
                      // to read signals from it
  armSystem(); // the alarm system will start in an armed state when plugged in
}

// resets the array containing the keys pressed and the variable containing the number of the keys pressed
void resetNumbers()
{
  for(int i=0;i<8;i++)
  {
    keysPressed[i] = '0';
  }
  nrKeysPressed = 0;
}

// opens the door by rotating the servo
void openDoor()
{
  myServo.write(110); // if the servo is rotated to 110 degrees the door lock opens
}

// turns on the light
void lightsOn()
{
  digitalWrite(light,LOW); // if the relay's input gets a LOW state it will connect the power source to the
                           // bulb
}

// produces an acoustical signal which consists of 2 500 Hz sounds which last for 0.3 seconds with a 0.3
// second gap between them
void successBuzz()
{
  for(int i=0;i<2;i++)
  {
    tone(buzzer,500);
    delay(300);
    noTone(buzzer);
    delay(300);
  }
}
```

```
// produces an acoustical signal which consists of 3 300 Hz sounds which last for 0.5 seconds with a 0.3
// second gap between them
void wrongBuzz()
{
 for(int i=0;i<3;i++)
 {
  tone(buzzer,300);
  delay(500);
  noTone(buzzer);
  delay(300);
 }
}

// checks whether the passcode from the keypad corresponds to the one given and takes actions
// respectively
void checkPassword()
{
// checks whether the password given in the keysPressed array corresponds to the password of the system
// if it corresponds the match variable will be 1 else it will get the value 0
 int match = 1;
 for(int i=0;i<PASSWORD_LENGTH;i++)
 {
  if(password[i] != keysPressed[i])
  {
   match = 0;
  }
 }
// it the password is correct the door will open, after the light turns on a sound will follow and the state
// of the alarm system will be disarmed
 if(match == 1)
 {
   openDoor();
   lightsOn();
   successBuzz();
   isSystemArmed = 0;
 }
// it the passcode is wrong a sound will indicate it and the system will remain in its previous state
else{
  wrongBuzz();
 }
// after each attempt the variables which store the input information will be re-initialized
 resetNumbers();
}

// will turn on the alarm of the anti-burglar system
void alarm()
{
 int alarm = 1; // indicates the state of the alarm (if it is 1 the alarm is on if it is 0 the alarm is off)
 char stopAlarmKey;
```

```
// produces the sound and light effects until the key ‘B’ is pressed on the keypad
while(alarm)
  {
   // checks whether the key ‘B’ is was pressed on the keypad, if it is so the alarm will stop
    stopAlarmKey = keypad.getKey();
    if(stopAlarmKey && stopAlarmKey == 'b')
    {
      alarm = 0;
    }

 // will produce a sound of 1000 Hz, and will turn on the light for 0.5 seconds
    tone(buzzer,1000);
    lightsOn();
    delay(500);

  // checks whether the key ‘B’ is was pressed on the keypad, if it is so the alarm will stop
     stopAlarmKey = keypad.getKey();
    if(stopAlarmKey && stopAlarmKey == 'b')
    {
      alarm = 0;
    }

// will stop the sound and the light for 0.5 seconds
    noTone(buzzer);
    digitalWrite(light,HIGH);
    delay(500);
  }

}

// will repeat the instructions given until the Arduino is removed from the power supply
void loop() {
  char keyPressed = keypad.getKey();
  if(keyPressed)
  {
// produces a sound effect and turns on the LEDs when a key is pressed
    tone(buzzer,200);
    delay(100);
    noTone(buzzer);

// checks which key was pressed and acts accordingly
// key ‘*’ resets the variables which store input data if, for example, the passcode is wrong
    if(keyPressed=='*')
    {
     resetNumbers();
    }else
 // key ‘#’ submits the entered passcode and will start to check it
    if(keyPressed == '#')
    {
```

```
    checkPassword();
  }
  else
// key ‘A’ arms the system
  if(keyPressed == 'a')
  {
   armSystem();
  }
  Else
// if any else key is pressed it will be recorded in the keysPressed array
  {
   keysPressed[nrKeysPressed] = keyPressed;
   nrKeysPressed++;
  }
 }

// after processing the inputs from the keypad, the system will start to process the information sent by
// the pir sensor if the state of the alarm is armed
 int pirState = digitalRead(pir);
 if(pirState == HIGH && isSystemArmed == 1)
 {
// if motion was detected the alarm will go off
  alarm();
 }

}
```

4. Circuit diagram and building process

4.1 Circuit diagram

The **Figure 1** represents the full circuit of the project.

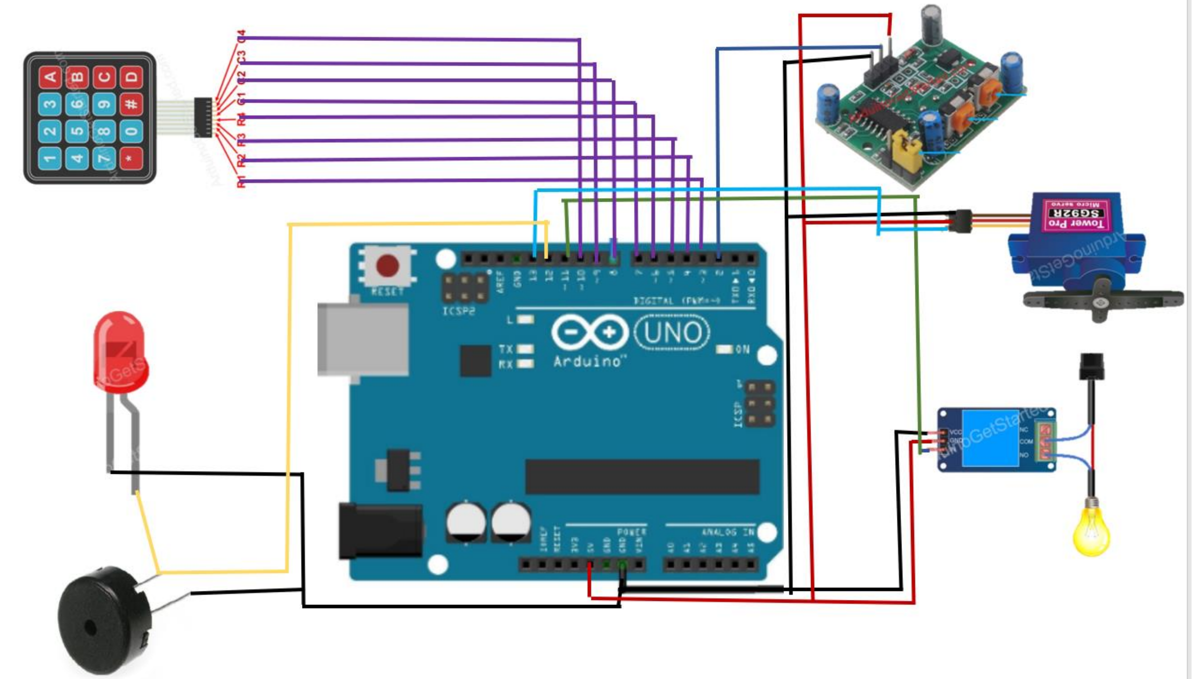

Figure 1. Circuit diagram.

In **Figures 2 and 3** is the whole project finished.

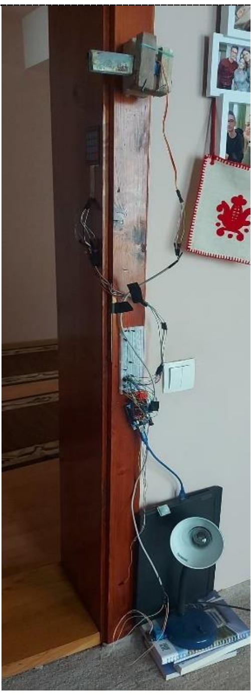

Figure 2.

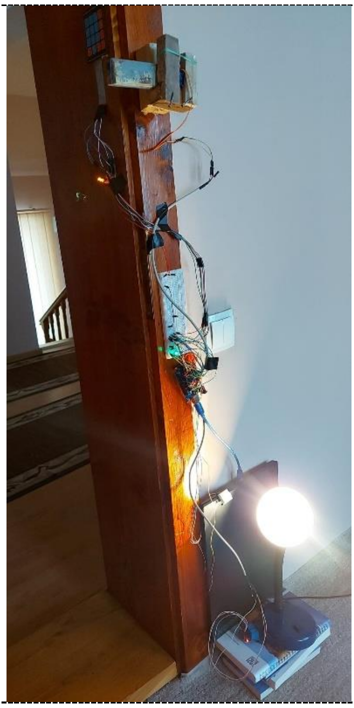

Figure 3.

4.2 Building process

Figures 4 and 5 show us working on the project, building, and testing it.

Figure 4.

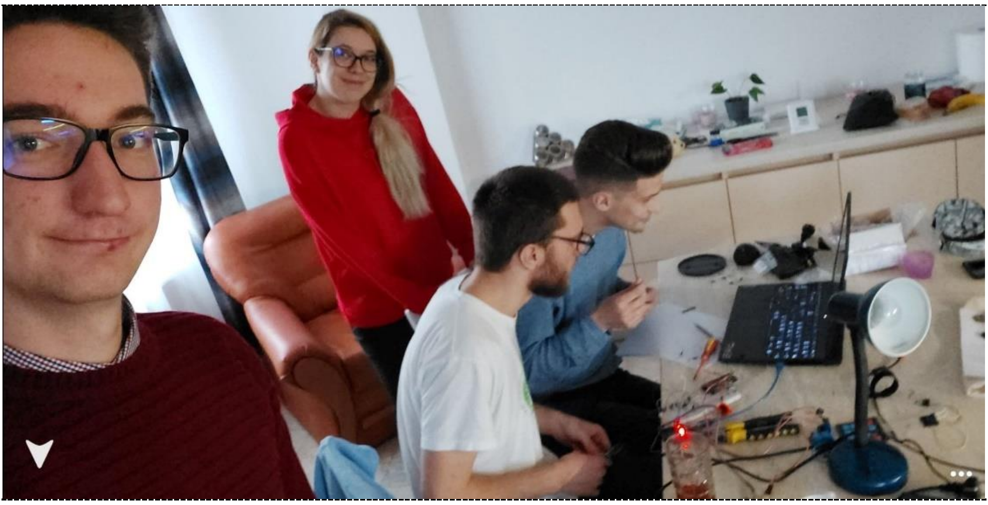

Figure 5.

5. References

1. https://www.youtube.com/watch?v=lTAo_H5eqsk
2. https://www.youtube.com/watch?v=RByUATNB6RQ
3. https://www.youtube.com/watch?v=6Fdrr_1guok
4. https://www.youtube.com/watch?v=8gzh95w4Hmk&feature=youtu.be&fbclid=IwAR36jE_9jM1eIEwJuor4BzjcaW0E64OXYX33J16daWa76DmKr_YkgHtZz1Q
5. https://www.youtube.com/watch?v=r3z5GIRbzCY&fbclid=IwAR2nCsyVu56Sa8CTHOyctVjER2lNUATeiqVWd8LXYMP8j13QzBIsKV-fLRM
6. https://www.youtube.com/watch?v=F-yk4Tyc44g&fbclid=IwAR2RgwgRkLmETQ2OQZKRkm42PuVEar1qIQy4yXDPdUjgLgmrFLh3Y-rQ36g
7. https://www.electronicshub.org/arduino-pir-sensor-tutorial/
8. https://arduinogetstarted.com/tutorials/arduino-servo-motor
9. https://arduinogetstarted.com/tutorials/arduino-keypad
10. https://arduinogetstarted.com/tutorials/arduino-relay