

Security alarm

Nume: Olteanu Teodora-Silvia
FILS

Introducere

The purpose of this project is to ensure safety. One can activate the security alarm by introducing a password and deactivate it only by entering the correct password.

General description

First, a 4 digit password must be set. When one wants to activate the alarm, the password that was previously set must be introduced. After that, in a matter of a few seconds, the alarm will be activated. The sensor will detect whether there is any movement in the nearby and the alarm will be triggered. The buzzer will then start to emit a buzz sound, which can only be stopped if the right password is introduced. If a wrong password is entered, the LCD will display a message that will let the person know that the password is wrong. Furthermore, the user has the option to change the password, but this is possible only if he knows the one already set and introduces it firstly. If, when entering the password(0000), one presses a wrong digit(01), there can be used a cancel option, which will allow the user to start over again, without having to introduce all 4 required digits.

Block scheme:



Hardware Design

For the hardware part the following components have been used:

Arduino Uno breadboard

 I used this picture from the ocw labs.

Keypad 4x4

- the keypad has 8 pins, 4 of them are for the rows and the other 4 for the columns;
- each button is a push button switch which makes a short between a row and a column when pressed.



Senzor Ultrasonic HC-SR04:

- it can measure distance from 2cm to 4 meters;
- it emits an ultrasound which travels through the air and if it meets an obstacle on its path, it will bounce back to the module;
- it has 4 pins: Ground(GND), VCC, Trig(input pin) and Echo(output pin);
- GND and VCC need to be connected to the Ground and VCC from the Arduino board.



16x2 LCD:

- it has 16 pins:
- ground **GND**;
- VCC**(supply voltage pin of LCD),
- VO**(control pin, adjusts the contrast of the LCD),
- RS**(register select-control pin, toggles between Command/Data Register),
- R/W**(read/write-control pin, toggles the LCD between Read/Write Operation),
- E**(enable-control pin, must be held high to perform Read/Write Operation),
- D0-D7**(data bits-data/command pin, pins used to send Command or data to the LCD),
- A**(LED positive-normal LED like operation to illuminate the LCD),
- K**(LED negative-normal LED like operation to illuminate the LCD connected with GND).



Buzzer:

- it has 2 pins: a positive and a negative one;



Potentiometer

Breadboard

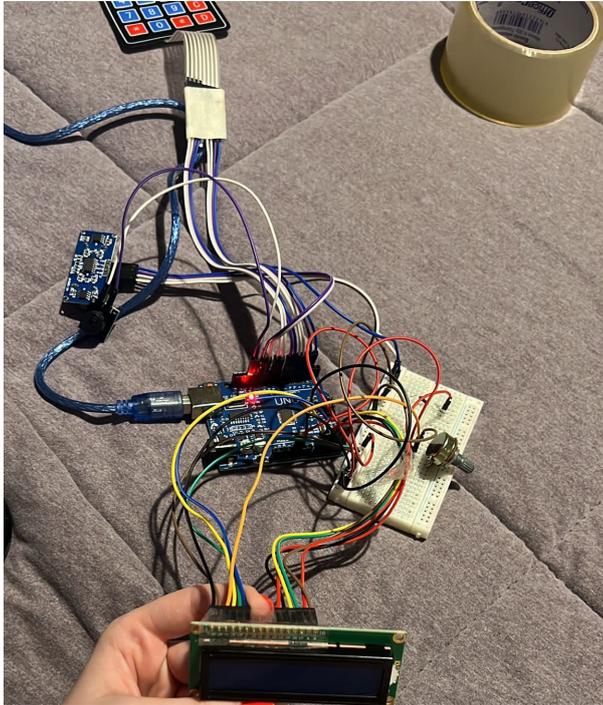
Wires

Resistor

Here is the electric scheme of the circuit that I have built in TinkerCad:



Here is a picture of the circuit:



The 8 pins of the keypad have been connected to the pins 2-5 and 6-9 of the Arduino board (4 for the rows and 4 for the columns). The VCC and GND of the sensor are connected to the VCC and respectively to the GND of the board, while the TRIG and ECHO pins are connected to pins 11, 12. The buzzer is connected to GND (breadboard) and pin 13. The GND of the Arduino is connected to the ground of the board. As for the LCD, again the VCC and the GND are connected on the breadboard. The cathode(K) is connected to - and the anode(A) to + by the use of a resistor. V0 is connected to the potentiometer.

Software Design

Development environment:

I have used **Arduino IDE** to write the code for my project.

Libraries:

The libraries that I have used are **Keypad** and **LiquidCrystal**:

1. The **LiquidCrystal.h** library is used to control LCD (Liquid Crystal Display) modules. It provides functions to initialize the LCD, set the cursor position, display text, create custom characters, and more.
2. The **Keypad.h** library defines the layout and pins for the 4x4 matrix keypad. It handles keypad inputs and provides functions to read and interpret the button presses on the keypad.

Implementation

I have implemented the following functions:

- 1.**setup()**
- 2.**setPassword()**
- 3.**verifyPassword()**
- 4.**loop()**
- 5.**checkDistance()**
- 6.**setOffAlarm()**

The code is uploaded in the archive below.

Final Result

<https://youtu.be/pAhXtH74ZVc>

Download

Here you will find the archive with the code and the final results: [arhiva_teodora_olteanu.rar](#)

Bibliography / References

<https://playground.arduino.cc/Code/Keypad/#Download>

<https://docs.arduino.cc/learn/electronics/lcd-displays>

https://www.youtube.com/watch?v=ITAo_H5eqsk

[Export to PDF](#)

From:
<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2023/apredescu/securityalarm>



Last update: **2023/05/28 21:35**

