Graphical user interface, text

Description automatically generated

**PROJECT REPORT**

Name & Surname: Eren Başaran

Student ID: A00303007

Degree: Computer Engineering Higher Cert

Year 2, Semester 2

Assignment: Alarm Project Report

Lecturer: Kevin McDermott

Date due: 21/04/2023

STUDENT PLAGIARISM DISCLAIMER FORM

**PLAGIARISM DISCLAIMER**

STUDENT NAME: Eren Başaran

STUDENT NUMBER: A00303007

PROGRAMME: Computer Engineering, Higher Cert

YEAR: 2022/2023

MODULE: Project 2

LECTURER: Kevin McDermott

ASSIGNMENT TITLE: Alarm Project Report

DUE DATE: 21/04/2023

DATE SUBMITTED:

I understand that plagiarism is a serious academic offence, and that AIT deals with it according to the AIT Policy on Plagiarism.

I have read and understand the AIT Policy on Plagiarism and I agree to the requirements set out therein in relation to plagiarism and referencing. I confirm that I have referenced and acknowledged properly all sources used in preparation of this assignment. I understand that if I plagiarise, or if I assist others in doing so, that I will be subject to investigation as outlined in the AIT Policy on Plagiarism.

I understand and agree that plagiarism detection software may be used on my assignment. I declare that, except where appropriately referenced, this assignment is entirely my own work based on my personal study/or research. I further declare that I have not engaged the services of another to either assist in, or complete this assignment.

**Signed: Eren Başaran**

**Dated:** 21st April 2023

**Contents Page**

1. Introduction
2. Background Research

1. Hardware Design

1. Software Design

1. Implementation

1. Testing & Evaluation

1. Future Development

1. Conclusions

**Introduction**

This report details the creation of an alarm project using Arduino that has both a hardware and software component. The project's goal was to develop an alert system that is dependable and usable in a variety of environments. The alarm system's software development, testing, and implementation are all covered in this report, along with the choice and assembly of hardware components. It also identifies possible areas for future development and acts as a reference for other programmers interested in making alarm systems similar to this one.

**Aims**

This project's primary goal is to use Arduino to develop and construct a home security system. The system uses a number of sensors and modules to track movement, determine distance, sound an alert, and show the state of the alarm on an LCD display and LED lights. A keypad for arming and disabling the alarm is also a part of the system, as is password security to prevent unauthorised entry.

Using Processing to visualise data is another goal of this project. The user may see the state of the sensors and the alert on a graphical user interface by sending sensor data from Arduino to Processing.

**Objectives**

* Create and implement an Arduino-based alarm system that can recognise and react to invasions.
* Use an ultrasonic sensor to measure distance and a PIR motion sensor to precisely detect movement.
* Use an LCD screen to display the sensor data and the alarm's current condition.
* Utilize LED lighting to clearly indicate the alert state.
* Include a buzzer so that when motion is detected or the alarm is activated, it will emit an audible alarm.
* The alarm should include a keypad that users may use to arm and deactivate it. Password protection should also be used to prevent unauthorised access.
* To show sensor data and give insights on alarm activity, create a user-friendly data visualisation interface in Processing.
* Test the alarm system to confirm that it can detect intruders and act appropriately under a variety of circumstances.
* Analyze the alarm system's operation and efficacy, and pinpoint any possible enhancement opportunities.

**Background Research**

For the Arduino Alarm System project to provide the intended results, a number of technologies are used. The project makes use of PIR motion sensors, ultrasonic sensors, a buzzer to sound an alarm, LED lights to show the state of the alarm, and an LCD display to show the status and sensor data. In order to prevent unauthorised entry, the project additionally incorporates password protection and a keypad to activate and disengage the alarm. Additionally, Processing is used to create the project's data visualisation.

All of the components such as sensors, LED lights, buzzer , LCD monitor and keypad were connected to the Arduino Board using jumper wires. Each connection will be declared on the Hardware Design section.

The project must satisfy the necessary security criteria for alarm systems as defined by industrial standards. This involves making sure the password security mechanism is reliable and difficult to go around.

During the project, health and safety issues were considered to ensure the safety of the developer and anyone who interacts with the project. This includes using low voltage components, insulating and boxing the project to prevent accidental contact with live wires, and checking wiring before powering up the system.

The circuit board and components may be recycled, but batteries need to be disposed of separately and in compliance with the WEEE requirement, which is a crucial factor in the project's design.

**Hardware Design**

The Arduino alarm system circuit consists of various components connected to an Arduino Uno board. The components include 2 PIR motion sensors, an ultrasonic sensor, a buzzer, a LED light, an LCD display, and a keypad.

One PIR motion sensor is connected to the digital input pin 49, and the other one is connected to the pin 41 of the Arduino board, and they detect motion in their field of view. These two sensors work differently from each other, one of them is assigned as ZONE1 and the other one as ENTRY\_EXIT. More details can be found on the Software Design section. The trigPin of the ultrasonic sensor is connected to the digital pin 9 and The trigPin the echoPin is connected to the pin 10 of the board and measures the distance to an object in front of it.

The buzzer is connected to digital pin 48 of the board and is used to sound an alarm when motion is detected or when the distance is breached. The LED light is connected to digital 47. The LED and the buzzer both trigger at the same time when one of the alarm conditions are satisfied.

The LCD display is connected to the board using the LiquidCrystal library, and it displays the status of the system and the readings from the sensors. There are 6 pins on the display to connect to the Arduino board. RS is connected to the digital pin 52, EN to digital pin to 53, d4 to digital pin 4, d5 to digital pin 5, d6 to digital pin 6, d7 to digital pin 7. A potentiometer is also connected to the V0 pin of the LCD display so that the user can control the brightness.

The keypad has 8 pins to connected in total. 4 of them for row pinouts and the other 4 is for column pinouts. Row pins are connected to these digital pins of Arduino board; 23,25,27,29. Column pins are connected to these digital pins of Arduino board; 31,33,35,37.

The keypad is used to arm and disarm the alarm system and to enter the password.

The operation of the alarm system is as follows: when the system is powered on, the LCD display shows an “Enter password” message and prompts the user to enter the password. If the correct password is entered, the system is armed and starts monitoring for motion using the PIR sensors and measuring the distance using the ultrasonic sensor.

If motion is detected or an object is detected within a certain distance, the buzzer sounds an alarm, and the LED light indicate the status of the system. If the alarm is triggered, the user can disarm the system using the keypad by entering the correct password. If an unauthorized access attempt is made, the system prompts the message “Wrong Password!” and asks for user to re-enter the password.

The circuit is designed to ensure the safety of the user by using low voltage components and insulating the project. The project can be recycled at the end of its life by following the WEEE directive, which outlines the safe and environmentally responsible disposal of electronic waste.

**Software Design**

The Arduino Alarm System's software is made up of a number of parts that work together to operate output devices, detect motion, estimate distance, and communicate with the user through a keypad and an LCD display.

There are 2 types of alarm variables: ENTRY\_EXIT and ZONE. ENTRY\_EXIT can be seen as the main door of a house. Imagine the tenant walks out the house in the morning and comes back in the evening. The alarm should be armed after the tenant walks out of the home and then the alarm should trigger when the door remains open more than a certain amount of time, check the countdown() function, in the evening. One of our PIR sensors refer to this type of variable. The ZONE variable triggers as soon as there is a breach.

The libraries must be defined at the first section of the code and then the necessary variables should be defined just before setup() function. Each variable’s responsibility has been commented out. See Figure 1to observe the appendix of the first part of the code:

Text

Description automatically generated

Figure 1

setup() is the default function where triggers when the Arduino board is powered up. In here, all the components which has been used on the system is defined depending if they are output or input. With those, LCD display is being initialized and the first “Enter password.” Message prompts up. See Figure 2.

Text

Description automatically generated

Figure 2

Before talking about the loop() function, it would be better to talk about the user-defined functions which is being used in that function. Those functions are essential to use for future developments and to have a code which can be read easily.

Countdown() function: It is for the ENTRY\_EXIT alarm variable. This function triggers when a motion has been detected. It provides 5 seconds of delay until the buzzer and LED flashes. My\_Delay() function is used for delay instead of Arduino’s built-in delay() function in order for user to disable the alarm by inputting the password when the countdown is still on. See Figure 3 for countdown() and Figure 4 for My\_Delay() function.

Text

Description automatically generated

Figure 3

Text

Description automatically generated

Figure 4

my\_key() function is used to read password from the user. When the # pressed it detects it as Enter and triggers the checkPassword() function which prints out the screen whether the password is correct or not. Depending on the result, it sets the ON\_OFF\_STATE high or low. resetPassword() is to reset the password at the end of the process. processNumberKey() is to generate the password so that it can be checked if the password is correct or not. See Figure 5 for my\_key(), Figure 6 for checkPassword(), Figure 7 for resetPassword() and Figure 8 processNumberKey() functions.

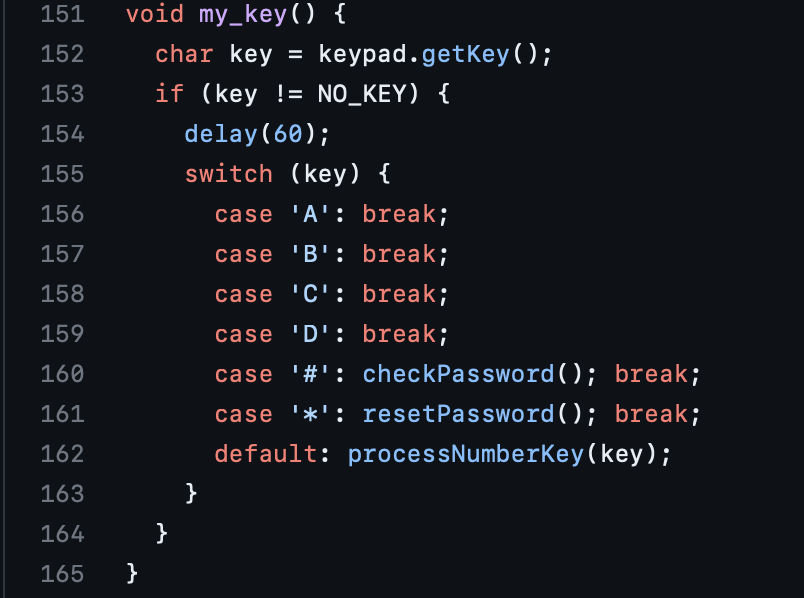


Figure 5

Text

Description automatically generated

Figure 6

Text

Description automatically generated

Figure 7

Text

Description automatically generated

Figure 8

Flash() function: It is used to trigger LED and buzzer. They both flash every 1 second. During Flashing My\_Delay() is also used to check if the user inputs the correct password or not. See Figure 9.

Text

Description automatically generated

Figure 9

distanceReturner() function: It detects the distance between the object and the ultrasonic sensor. The calculation of distance is based on the speed of sound, which travels at approximately 343 meters per second in air at room temperature. To convert the time in microseconds to distance in centimeters, the duration is multiplied by the speed of sound in centimeters per microsecond (which is 0.034 cm/μs), and then divided by 2 because the sound waves travel to the object and back. See Figure 10.

Text

Description automatically generated

Figure 10

loop() function: The function where the every action is handled. It is an infinite loop which every single sensor is being checked all the time. The alarm states are being re-assigned and the necessary text is being printed on the LCD monitor and Serial Monitor so that on the processing part it can be read through that monitor and then visualize the data based on that information. See Figure 11.

Text

Description automatically generated

Figure 11

**Implementation**

The project's execution entailed putting the different hardware pieces together in accordance with the design requirements stated in the hardware design section. Jumper wires were used to link each component to the Arduino board, and each connection was clearly shown on the design schematics.

Software-wise, the code was written, compiled, and uploaded to the Arduino board via a USB connection using the Arduino 1.8.19 IDE. All essential functions for controlling the various hardware components were included in the C++ code, which was created. The code was created to interpret data from the PIR motion sensor and ultrasonic sensor and, in response, to turn on the alarm and LED lights. Additionally, the code had keypad commands for arming and disarming the alarm, and password protection was used to block unauthorized access.

To show the sensor values on a computer, data visualization was also made using the Processing IDE. The data visualization featured 3 rectangles referring to each sensor on the alarm system. Whenever a zone or an entry\_exit is triggered, the rectangles turn red.

Overall, the project's implementation required both developing and compiling the software code to manage the system's numerous operations as well as putting the hardware components into place.

**Testing & Evaluation**

There are some tests have been implemented to the Arduino Alarm Project to evaluate the performance of the project.

* Functional Testing: Both ENTRY\_EXIT and ZONE1 PIR sensors have been tested by moving in front of it to ensure they detect motion and trigger the alarm as they should be. For example, ENTRY\_EXIT alarm sets a countdown when a motion is triggered and ZONE1 just triggers it. The ZONE2 ultrasonic sensor has also been tested by getting closer to the sensor.
* Integration Testing: This test is essential to ensure that all components work together as a whole. LCD display perfectly displays the proper information for the alarm system. This feature was tested multiple times to make sure that the user sees the correct info for the alarm status.
* Usability Testing: Keypad has a user-friendly interface where user can arm and disarm the alarm easily. All sensors have good sensitivity to detect motion and distance from the intruder. ENTRY\_EXIT sensor triggers at the first time it detects motion but in the second time it must see another motion so that user will not have to disarm the alarm as he/she leaves the home in the morning.
* Safety Testing: To ensure the safety of the project, a series of tests were conducted to ensure that it would not cause harm to the user or the environment. These tests included checking the insulation of all components, verifying that all wires were properly connected and grounded, and testing the emergency shutdown features of the system.
* Endurance Testing: To test the endurance of the project, project was continuously run for an extended period to see how it would hold up under normal usage conditions. Based on our tests, we concluded that the project was able to operate reliably for an extended period without any significant degradation in performance.

**Future Development**

The use of facial recognition technology to increase security and usability is one possible upgrade.

The system might take a picture of the individual attempting to deactivate the alarm by integrating a camera module into the current arrangement. To decide whether access should be permitted, the face recognition software might compare the acquired image to a database of authorized people. This would make it unnecessary to employ a keypad and password system, giving users a more smooth and safe experience.

A possible security breach might also be identified, and the system could be set up to notify the user through email or text message. This would add an extra layer of protection and enable quick response from the user in an emergency.

Overall, integrating face recognition technology into the Arduino alarm system might significantly improve its usability and security, and there are several further possible enhancements and expansions that could be investigated with more time and money.

**Conclusion**

In conclusion, the project successfully shown how to build a home security system using PIR motion sensor and Ultrasonic sensor. A buzzer to sound an alert, LED lights to show the state of the alarm, and an LCD display to show the status and sensor data were all included in the system. A keypad was used to arm and disable the alarm, and password security was added to prevent unauthorized access. The project demonstrated the student's expertise in both software and hardware design as well as their capacity to assess the system critically and make improvements. The project was successful overall and may be used as a base for future advancements in home security systems. The link to the GitHub repository can be found down below[1].

[1] : GitHub Repository: https://github.com/Erenco16/arduino-alarm-project