

GEBZE TECHNICAL UNIVERSITY

CSE 101 Term Project Report

Project name: Combination Safe

GROUP 13

1. Contents

I. Description of the project	3
II. Equipment used	3
III. Task distribution	4
IV. Hardware diagram	5
V. Code explanations	6
VI. Summary	9
VII. Testing	10

1.1. Description of the project

We have designed a safe which works with a keypad and RFID NFC scanner. The safe asks for the user to enter PIN code, when an incorrect PIN is entered for 3 times in a row, the safe locks itself and only opens if an authorized card gets scanned.

1.2. Equipments used

- 16x2 I2C LCD Display
- 3x4 Keypad
- Servo motor
- RC522 RFID NFC module
- Red and green LEDs and a buzzer



1.3. Task distribution

Hardware and software development

Melih KAZANCI - **230104004909**

Mehmet Arda GÖNEL - **230104004093**

Fatma ÖZTÜRK - **230104004152**

Gabil RAHİMLİ - **230104004902**

Report

Görkem UYSAL - **230104004174**

Esmanur SÖNMEZ – **230104004908**

Melih KAZANCI - **230104004909**

Video testing

Mervan Deniz YILDIRIM - **230104004173**

Tuana Melisa AKSOY - **230104004903**

Hüseyin SEVİNÇ - **230104004913**

Slide preparation for the presentation

Alp KAZAZ - **230104004915**

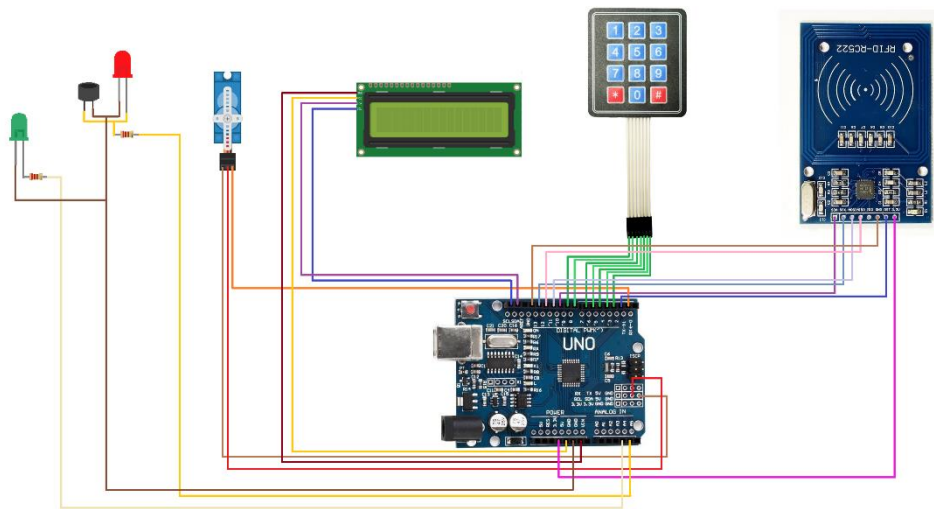
Presentation speakers

Mehmet Arda GÖNEL - **230104004093**

Zülfikar KARASU - **230104004164**

2. Technical information

2.1. Circuit diagram



Project consists of 7 different peripheral devices which are connected to Arduino's digital and analog pins.

Digital pins between 3 and 9 are used by the keypad, 2,10,11,12,13 are used by the RFID NFC module, 1st pin is used by the servo motor and the leftmost SDA and SCL pins are used by the I2C display.

At first the LEDs and the buzzer were also connected to the digital pins but when we decided to add RFID module we needed to free some digital PWM pins, after some research we have found that analog pins can also be used like a digital pin to control LEDs.

In order to leave some GND pins available on the Arduino, we connected 2 LEDs and the buzzer to the same ground line, which let us adding the RFID module to the project.

2.2. Code explanations

2.2.1. Controlling the RFID module

```
while (1)
{
    // . . .

    if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial())
        continue;

    if (/*Comparing the scanned card's UID with the correct UID number*/)
    {
        //Run motor functions to unlock the safe
        //Wait for the user input to relock the safe

        // . . .
        break;
    }

    rfid.PICC_HaltA();
}
```

RFID functions are only used in this section of the program. When the user enters 3 incorrect passwords in a row, the safe asks for the authorized card in order to determine if the user is the owner of the safe. When the program falls into this while loop, the only way to exit is to scan the correct NFC card which is recognized by the RFID scanner.

`rfid.PICC_IsNewCardPresent()` is a function from the RFID library “<MFRC522.h>”.

With this function we can check if a card is getting scanned or not. If so, this function will return 1.

rfid.PICC_ReadCardSerial() is also from "<MFRC522.h>" and it lets the scanner to get the UID number from the card and store it inside its' memory. It returns 1 when the card gets scanned.

By the if condition (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial()) we can check if a card is being scanned at that moment, if not, the if block gets executed and the while loop starts again. We save time with not comparing the unscanned UID info by doing this.

Lastly, rfid.PICC_HaltA() is used to prevent the RFID module to run when no card gets scanned. It makes the communication between the module and the previously scanned card stop until another card gets scanned.

2.2.2. Taking the input from the Keypad

```
int take_input()
{
    char customKey;
    int input=0, i=0, j=1000, temp;

    while (i<4)
    {
        customKey = customKeypad.getKey();
        if (customKey)
        {
            lcd.setCursor(i,1);
            lcd.print(customKey);
            temp = customKey - '0';
            temp = temp*j;
            input += temp;
            j /= 10;
            i++;
        }
    }
}
```

```
    }  
    return input;  
}
```

The function “getKey()” is from the library “<Keypad.h>” which lets us to take keypad inputs. In this program we take inputs as char and convert it into an integer by using the conversion variables “temp” and “j”.

Firstly, temp gets customKey’s ASCII number and we subtract the ASCII value of zero in order to get the pressed number on the keypad. And then “temp” gets multiplied by its’ digit number, since we have used a 4 digit PIN system in the project it starts with the value 1000 and gets divided by 10 with each run until the 4th input gets taken.

For example, if the user enters the number “1415” the program works like $1000+400+10+5$ which lets us to get the integer representation of the PIN 1415 from the char-based keypad.

The if condition (customKey) is to prevent unwanted returning values from the keypad when no button is pressed. With this if block the program executes the functions only when the user presses one of the buttons.

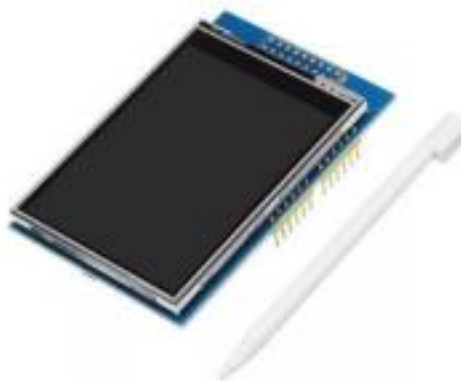
3. Summary

3.1. Other project ideas

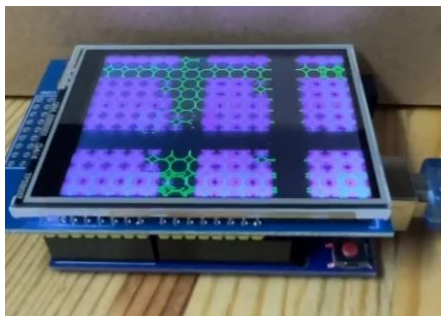
- Arduino project that teaches you how to use Arduino
- LED Memory game
- Smart Safe with Arduino
- LED Arcade game
- Memory Card game

3.2. Updates during the development of the project

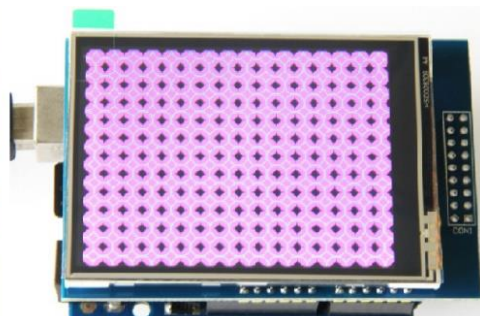
Several ideas were put forward at the beginning of the project and as a result of voting we picked the Memory Card game. After choosing that project we bought a 2,8" Touchscreen module



But after some tests we have seen some incorrectness in how the display module responds to our inputs. Following picture is from one of the test results



Our display



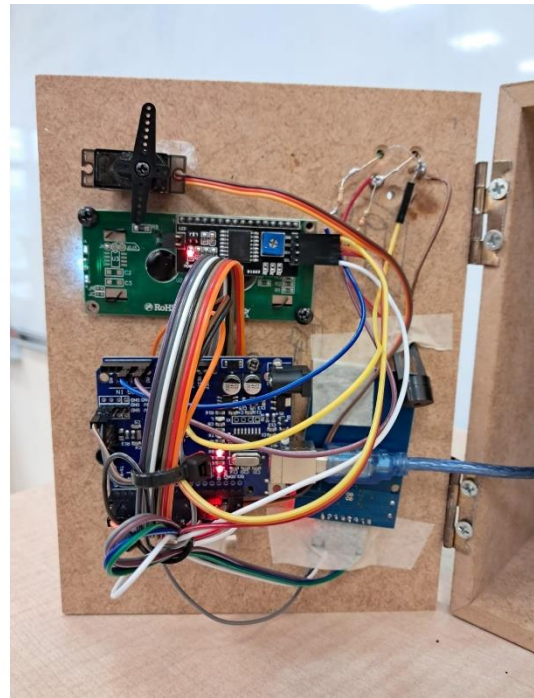
Expected output

Eventually we decided to change the display module and change the project to the Smart Safe.

While advancing this project, we encountered another problem. Arduino's digital pins were insufficient for the RFID NFC module. After some research we have seen that analog pins can also be used as a digital pin to control LEDs.

We solved this problem by connecting the LEDs and the buzzer to the analog pins and leaving the remaining digital pins to RFID module.

After all these developments, our safe has this appearance:



3.3. Testing

Here you can watch the testing video of the project:

<https://youtu.be/pHUeD6ZdF3c>