

Seif

- Student: **Borbei Andreea**
- Grupa: **333CC**

Introducere

Proiectul ales de mine este un seif pentru pastrarea in siguranta a obiectelor de valoare. Pentru securizarea bunurilor si inchiderea seifului, se va introduce un cod PIN. Utilizatorul va putea introduce codul PIN de maxim 3 ori pentru a incerca deblocarea seiful. Daca codul nu este introdus corect dupa 3 incercari, un semnal de alarma va porni (semnal sonor emis de buzzer si se va aprinde un LED rosu), iar detinatorul seifului va primi o notificare pe telefon cum ca s-a incercat deschiderea seifului. Detinatorul seifului va putea opri alarma din cadrul aplicatiei mobile.

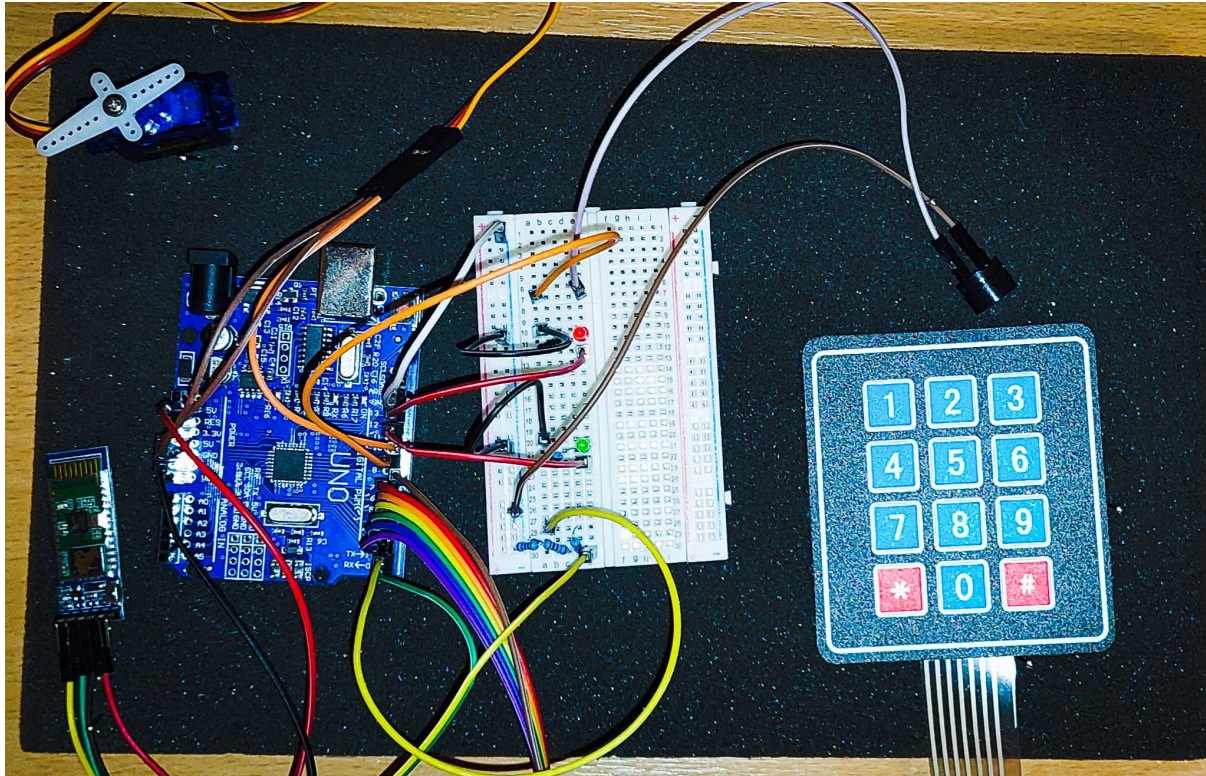
Descriere generală



Schema electrică



Circuit



Hardware Design

Lista de piese:

- Arduino UNO
- Motoras
- modul Bluetooth/Wifi
- Buzzer
- Keyboard
- LED-uri
- baterie de alimentare

Software Design

Descrierea codului aplicației (firmware):

- Pentru dezvoltarea aplicației am folosit **Arduino IDE**
- Pentru a programa motorasul servo si keypad-ul 4×3 am folosit **librariile Servo.h si Keypad.h**
- **Descriere implementare:**

Dispozitivul are 2 mari moduri de functionare: cel in care asteapta input remote de la aplicatie prin intermediul modului bluetooth HC-05 (in aceasta situatie poate primi fie un cod PIN, caz in care

returneaza o valoare in aplicatie pentru a declansa o notificare ce anunta utilizatorul ca PIN-ul a fost setat cu succes, sau poate primi comanda de lock/unlock care blocheaza/deblocheaza seiful de la distanta) sau asteapta input de la keypad si verifica daca codul introdus de utilizator este corect. La fiecare introducere gresita de PIN, LED-ul rosu va lumina pentru a indica acest lucru. La 3 incercari esuate de deblocare, va fi pornita o alarma ce va putea fi oprita doar de utilizator din aplicatia LockIt. Aplicatia LockIt a fost implementata prin intermediul platformei MIT App Inventor.

- Descrierea codului:
- **setup():**
- Initializarea pinilor folositi pentru cele doua LED-uri si buzzer
- Setarea pinului pentru motoras si setarea pozitiei sale
- **setLocked(int locked):** Blocheaza sau deblocheaza seif-ul (schimband pozitia elicei motorasului servo) in functie de valoarea variabilei locked

```
if(locked){
    servo.write(180);
}
else{
    delay(500);
    servo.write(90);
}
```

- **loop():**

Se asteapta input de la interfata seriala UART prin intermediulul modulului Bluetooth din aplicatie:

```
// Checks for input from LockIt app via UART interface
if(Serial.available() > 0){
    String command = Serial.readString(); //read until timeout
    command.trim();                       // remove any \r \n whitespace at
the end of the String
    // "Lock" command -> lock the safe box remotely using the app
    if(command == "lock"){
        setLocked(true);
    }
    // "Unlock" command -> unlock the safe box remotely using the app
    } else if(command == "unlock"){
        setLocked(false);
    }
    else if(command == "stop"){
    }
    // The app sent the PIN code for the safe box
    else{
        strcpy(passwd, command.c_str());
        // Check if PIN is of length 4
        if(strlen(passwd) == 4){
            digitalWrite(ledPinGreen, HIGH);
            delay(500);
            digitalWrite(ledPinGreen, LOW);
        }
        // Write back to the app in order to trigger a notification that
        // lets the user know the PIN has been set correctly
    }
```

```
    Serial.write('0');  
    setLocked(true);  
    delay(500);  
  }  
}
```

Fie de la keypad, caz in care se verifica fiecare caracter primit cu caracterul de la pozitia respectiva din PIN. Pentru a tine cont de pozitia curenta din PIN, am folosit o variabila position. Daca caracterul primit nu corespunde cu cel de la position, codul este gresit si position devine 0 pentru urmatoarea verificare.

```
// If there is input from the keypad -> used variable position to keep  
// track of how much of the input from the keypad matches the PIN;  
// when a wrong key is pressed, position resets to 0.  
else if((key >= '0' && key <= '9') || (key == '*' || key == '#') &&  
chars_read < 12){  
  // Lock the safe box after unlocking it using the keypad  
  if(key == '#' || key == '*'){  
    position=0;  
    chars_read = 0;  
    setLocked(true);  
    digitalWrite(ledPinGreen, LOW);  
  }  
  // Check if the current pressed key is correct  
  else if(key == passwd[position]){  
    position++;  
    chars_read++;  
  }  
  // The current pressed key is wrong -> reset positions  
  else if(key != passwd[position]){  
    position = 0;  
    chars_read++;  
  }  
  // Signal that a key from the keypad has been received  
  digitalWrite(ledPinGreen, HIGH);  
  delay(500);  
  digitalWrite(ledPinGreen, LOW);  
}
```

La introducerea corecta a codului PIN, seiful se va debloca si LED-ul verde va ramane aprins tot timpul cat seiful este deschis.

```
// If the user inputs the PIN correctly -> unlock the safe and reset  
variables  
if(position == 4){  
  digitalWrite(ledPinGreen, HIGH);  
  setLocked(false);  
  chars_read = 0;  
  position = 0;  
}
```

La introducerea gresita a unui cod PIN, LED-ul rosu va lumina intermitent pentru a semnaliza eroarea.

```
// If the user inputs the wrong PIN -> blink the red LED
if(chars_read % 4 == 0 && chars_read / 4 >= 1){
    digitalWrite(ledPinRed, HIGH);
    delay(700);
    digitalWrite(ledPinRed, LOW);
}
}
```

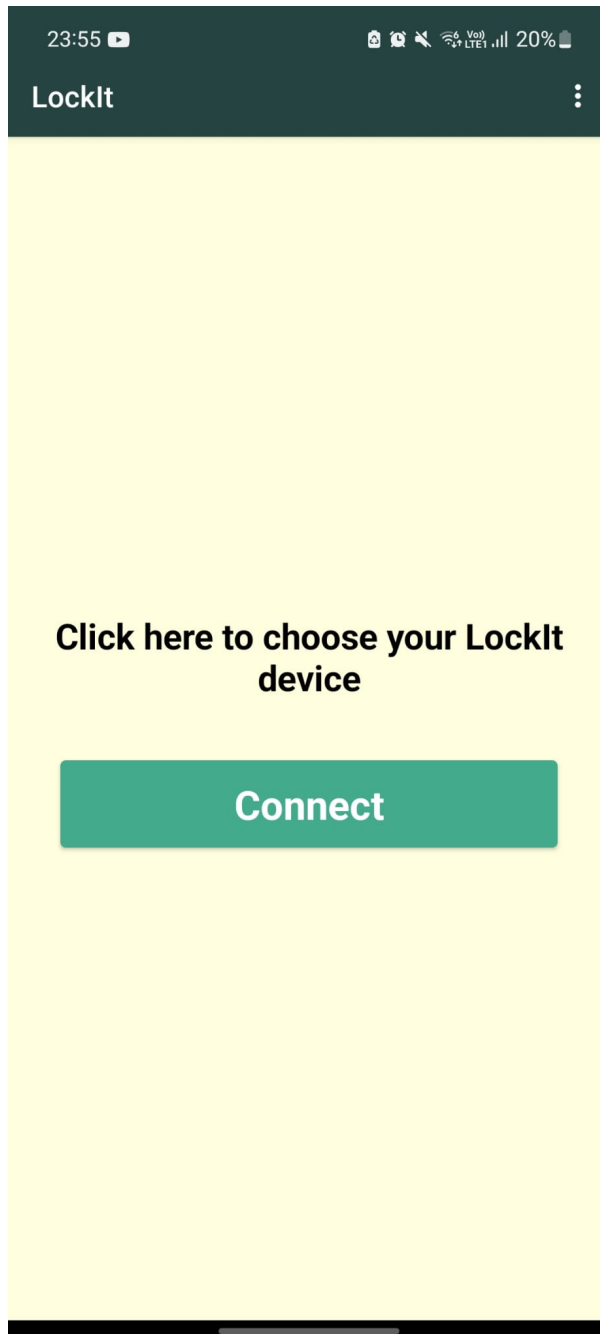
La 3 incercari gresite de deblocare, se activeaza tonul de alarma pentru buzzer. In acelasi timp, se verifica si daca exista input din aplicatie pentru a opri alarma.

```
// If there were 3 failures of opening the safe box,
// send a message to the app to notify the user and start the alarm
// Waits for input from the owner via LockIt app to stop the alarm
if(chars_read == 12)
    Serial.write('3');
while(chars_read == 12){
    if(alarm_stop == 0){
        for(int i=600;i<700;i++){
            unsigned long currentMillis = millis();
            if(currentMillis - previousMillis > interval) {
                previousMillis = currentMillis;
                if (ledState == LOW)
                    ledState = HIGH;
                else
                    ledState = LOW;
            }
            // Switch the LED
            digitalWrite(ledPinRed, ledState);

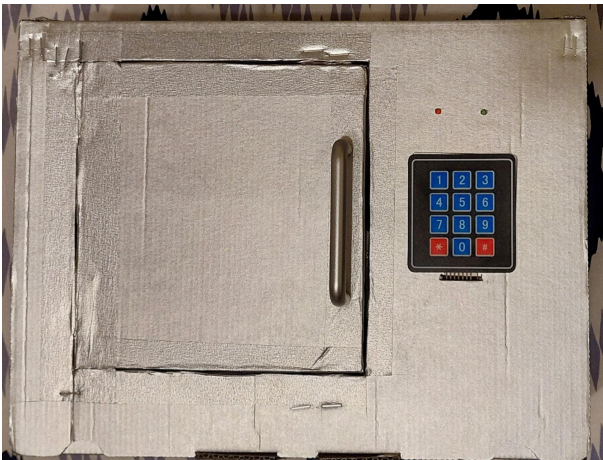
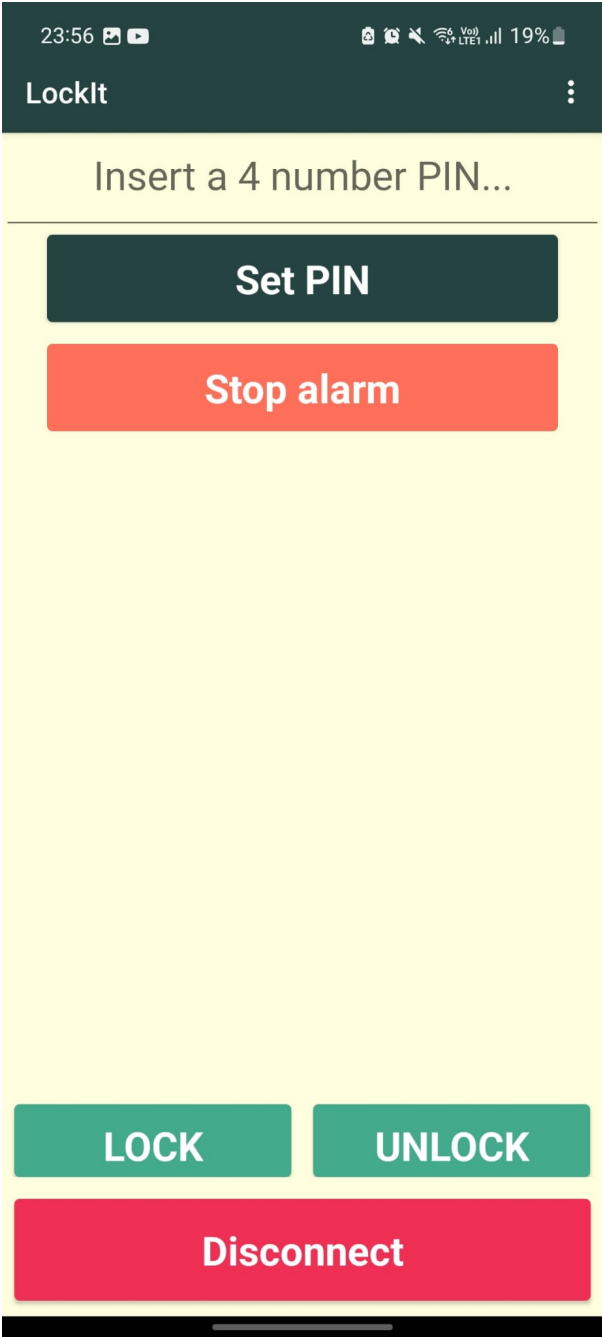
            if(Serial.available() > 0){
                String stop_command = Serial.readString();
                stop_command.trim();
                if(stop_command == "stop"){
                    chars_read = 0;
                    //Serial.write('4');
                    digitalWrite(ledPinRed, LOW);
                    alarm_stop = 1;
                    noTone(buzzerPin);
                    break;
                }
            }
            tone(buzzerPin,i);
            delay(15);
        }
    }
    if(alarm_stop == 0){
        for(int i=700;i>600;i--){
            unsigned long currentMillis = millis();
            if(currentMillis - previousMillis > interval) {
```

```
        previousMillis = currentMillis;
        if (ledState == LOW)
            ledState = HIGH;
        else
            ledState = LOW;
    }
    // Switch the LED
    digitalWrite(ledPinRed, ledState);
    if(Serial.available() > 0){
        String stop_command = Serial.readString();
        stop_command.trim();
        if(stop_command == "stop"){
            chars_read = 0;
            //Serial.write('4');
            digitalWrite(ledPinRed, LOW);
            alarm_stop = 1;
            break;
        }
    }
    tone(buzzerPin,i);
    delay(15);
}
}
noTone(buzzerPin);
}
```

Rezultate Obținute



Aplicatia LockIt:



Seiful:

Concluzii

Am realizat un proiect interesant in care nu am avut contact doar cu lumea hardware unde a trebuit sa implementez lucrurile cat mai user-friendly, dar am putut experimenta zona de dezvoltare de aplicatii mobile, zona pe care nu am experimentat-o inca la facultate, aceste doua lumi fiind conectate prin intermediul modulului Bluetooth. Am invatat nu numai cum sa conectezi niste componente astfel incat sa se intample ceva, ci si importanta de a lucra ordonat si de a ascunde complexitatea hardware de utilizatorul final.

Download

- Surse: [proiect_pm_borbei_andreea_333cc.zip](#)

Bibliografie/Resurse

- Link-uri:
- <https://appinventor.mit.edu/>
- <https://arduinogetstarted.com/faq/how-to-input-a-multiple-digits-number-using-the-keypad>
- <https://www.instructables.com/Arduino-AND-Bluetooth-HC-05-Connecting-easily/>

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2023/dene/andreea.borbei>

Last update: **2023/05/29 22:19**

