

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Arvutisüsteemide instituut

Taavi Tapupere 163890IASB,
Sander Tsõbulski 164826IASB,
Paul Raatpalu 164054IASB

SEIFIUKS

Tarkvaraprojekt

Juhendajad: assistent Marina Brik,
teadur Sergei Kostin

Tallinn 2018

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autorid: Taavi Tapupere 163890IASB, Sander Tsõbulski 164826IASB,
Paul Raatpalu 164054

Sisukord

Lühikirjeldus	4
Ajakulud	5
Vajalike komponentide nimekiri	6
Skeemi pilt (Breadboard view)	7
Põhimõtteskeem (Schematic)	8
Süsteemi funktsionaalsus	9
Tegevusdiagramm	9
Programmi kood	10
Süsteemi kasutusjuhtum	11
Allikad	12
Kokkuvõte	13
Tulevaste ideede loetelu	14

Lühikirjeldus

Käesoleva projekti eesmärgiks on luua seifiuks, mis küsiks kasutajalt kaardi tuvastamist ja 4 numbrilist PIN-koodi sisestamist, et avada seifiuks. Projekt on tehtud kasutades “breadboardi”, millel realiseeritakse elektroonika komponendid, ning sellele vastav kood, mis on kirjutatud Arduinos. Projekti tegemiseks oli kasutusel Arduino UNO.

Projektis kasutasime RFID - kaardi lugejat (RFID - RC522), 1x4 numbri platvormi, “servo” mootorit (SM-S2309S), OLED 128x64 ekraani ja “biezo puzzer” ehk piristi.

Ajakulud

Projekti tegemiseks kuluks kokku ligikaudu 25 kuni 35 tundi. Programmeerimisele kulus kõige rohkem aega, ligikaudu 20 kuni 25 tundi. ning ülejäänud 5 kuni 10 tundi kulus süsteemi kokkupanemisele.

Sander Tsõbulski - tegeles enamasti programmeerimisega ning natuke ka elektrooniliste komponentide paigaldusega.

Taavi Tapupere - tegeles rohkem elektroonika komponentide paigaldamisega ning panustas ka väikse osa programmeerimisele.

Paul Raatpalu - tegeles elektroonika komponentide paigaldusega.

Vajalike komponentide nimekiri

- Breadboard
- Arduino UNO
- RFID - lugeja
- 1x4 numbri platvorm
- Piezo buzzer
- 128x64 OLED ekraan
- Juhtmed

Skeemi pilt (Breadboard view)

Põhimõtteskeem (Schematic)

Süsteemi funktsionaalsus

Tegevusdiagramm

Programmi kood

```
//Lisame algul erinevate elektrooniliste osade header'ite libary'id, et neid saaks kasutada
#include <Servo.h>
#include <MFRC522.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
Servo myServo; //Initsialeerib Servo mootori, et saaks seda kasutada
//int const potPin = A0;
//int potVal;
int angle; //väärtus, mis määrab ukse hetkenurga

//Siin toimub ekraani initsialeerimine ja muude väärtuse paika panemine, et ekraan töötaks
//#include <SPI.h>
#define OLED_RESET 4
Adafruit_SSD1306 display(OLED_RESET);
#define SS_PIN 10
#define RST_PIN 9
//Initsialiseerib rfid, et saaks salvestada RFID kaardi koode
MFRC522 rfid(SS_PIN, RST_PIN); // Instance of the class

//Kui OLED ekraani suurus ei ole õige andakse vea teade
#if (SSD1306_LCDHEIGHT != 64)
#error("Ekraani kõrgus vale, palun parandage Adafruit_SSD1306.h fail!");
#endif

MFRC522::MIFARE_Key key;

// Initsialiseerime RFID jaoks kaustatavad muutujad, kuhu salvestame RFID kaardi koodi
byte nuidPICC[4];
byte kaardikood[] = {0x55,0x1A,0x10,0x85}; //RFID kaardisobilik kood
byte kaardikood2[] = {0xF6,0x35,0x2E,0x07};

const int buttonPin[] = {2,3,4,5};
int buttonState = 0; //vaatab, kas keypadil on nuppu alla vajutatud
int notes[] = {262,294,330,349};
int kood[] = {0,0,0,0}; //deklaleeritakse kood mida sisestad
int valeKood = 0; //kontrollib mitu korda järjest valekoodi on sisestatud
```

```

void setup() {
  // Serial begin algatab Arduino Uno peal olevate elektrooniliste osade tööle hakkamise
  Serial.begin(9600);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);//vajalik, et saaks ekraani kasutada
  myServo.attach(8);//mootor on 8. pordiga yhendatud
  for(int x=0; x<4; x++)//deklaleerib neli nuppu
  {
    pinMode(buttonPin[x], INPUT);
  }

  SPI.begin(); // Init SPI bus
  rfid.PCD_Init(); // Init MFRC522

  // Algseks RFID võtmeks olev kood
  for (byte i = 0; i < 6; i++) {
    key.keyByte[i] = 0xFF;
  }
  //Prindib välja esialgse RFID koodi numbri, kuigi see ei ole otseselt vajalik programmis
  Serial.print("Default key:");
  printHex(key.keyByte, MFRC522::MF_KEY_SIZE);
}

void loop(){

  display.clearDisplay();//teeb ekraani puhtaks
  display.setTextSize(1);//teksti suurus on 1
  display.setTextColor(WHITE);
  display.setCursor(0,0);
  display.println("Sisetsage kood ja n@idake kaarti!");
  display.display();//kirjutab teksti ekraanile
  //delay(2000);

  // Kontrollib, kas uus RFID kood on loetud, kui ei ole siis loeb uue RFID kaardi koodi sisse
  if(!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial()){
    if (rfid.uid.uidByte[0] != nuidPICC[0] ||
        rfid.uid.uidByte[1] != nuidPICC[1] ||
        rfid.uid.uidByte[2] != nuidPICC[2] ||
        rfid.uid.uidByte[3] != nuidPICC[3] ) {
      Serial.println(F("Uus kaart on tuvastatud."));
      tone(6, notes[0]);
      delay(200);
      noTone(6);
    }
  }
}

```

```

// Salvestab uue RFID kaardi koodi nuidPICC muutujasse
for (byte i = 0; i < 4; i++) {
  nuidPICC[i] = rfid.uid.uidByte[i];
}

Serial.println(F("NUID silt on: "));
// Väljastab RFID NUID sildi 10-nd ja 16-nd süsteemi arvudena
Serial.print(F("In hex: "));
printHex(rfid.uid.uidByte, rfid.uid.size);
Serial.println();
Serial.print(F("In dec: "));
printDec(rfid.uid.uidByte, rfid.uid.size);
Serial.println();
}
}

int i = 0;
//Siin for tsükliis toimub nuppude kontroll.
//Kontrollitakse iga tsükliga läbi 4 erinevat nuppu, kui ühte nuppu vajutatakse, see
//salvestatakse
for(int x=0; x<4; x++)//vaatab neli nuppu läbi, kas neid on alla vajutatud
{
  buttonState = digitalRead(buttonPin[x]);
  if (buttonState == HIGH && buttonPin[x] == 2) { //kui nupp, mis meil on teises pordis on
//vajutatud, salvesta nupuvajutus , teeb seda järgnevat nuppudega ka
  while(i<4)
  {
    if(kood[i] == 0){
      kood[i] = 1;
      break;
    }
    i++;
  }
  Serial.println("Nupp 1");
  tone(6, notes[0]);
  delay(200);
  noTone(6);
  i=0;
}
if (buttonState == HIGH && buttonPin[x] == 3) {
  while(i<4)
  {

```

```

    if(kood[i] == 0){
    kood[i] = 2;
    break;
    }
    i++;
}
Serial.println("Nupp 2");
tone(6, notes[0]);
delay(200);
noTone(6);
i=0;
}
if (buttonState == HIGH && buttonPin[x] == 4) {
    while(i<4)
    {
        if(kood[i] == 0){
        kood[i] = 3;
        break;
        }
        i++;
    }
    Serial.println("Nupp 3");
    tone(6, notes[0]);
    delay(200);
    noTone(6);
    i=0;
}
if (buttonState == HIGH && buttonPin[x] == 5) {
    while(i<4)
    {
        if(kood[i] == 0){
        kood[i] = 4;
        break;
        }
        i++;
    }
    Serial.println("Nupp 4");
    tone(6, notes[0]);
    delay(200);
    noTone(6);
    i=0;
}
}

```

```

}
if(kood[0] != 0 && kood[1] != 0 && kood[2] != 0 && kood[3] !=0)//kontrollib kas neli
//nuppu on vajutatud, et kas kood on sisestatud
{
//Kontrollib, kas sisestatud RFID kaart on õige
if(kood[0] == 1 && kood[1] == 3 && kood[2] == 3 && kood[3] ==2 &&
rfid.uid.uidByte[0] == kaardikood[0] &&
rfid.uid.uidByte[1] == kaardikood[1] &&
rfid.uid.uidByte[2] == kaardikood[2] &&
rfid.uid.uidByte[3] == kaardikood[3])
{

display.clearDisplay();
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(0,0);
display.println("Kood on õige, ukse sulgemiseks vajutage suvalist klahvi!");
display.display();
valeKood = 0;//paneb järjestikuse vale koodi lugeja 0
delay(5000);

Serial.println("Kood on õige");
for(int k=0; k<4;k++)
{
Serial.print(kood[k]);
kood[k] = 0;
}
angle = 0;//avab mootoriga seifi ukse, selleks leiab praeguse nurga ja läheb sealt 5 kraadi
//haaval nulli suunas
for(angle = 90; angle>=1; angle-=5)
{
myServo.write(angle);
//Serial.print("angle: ");
//Serial.println(angle);
delay(100);
}
noTone(6);
int n = 0;
while(n==0)
{
for(int x=0; x<4; x++)//kontrollib kas nuppu on vajutatud
{

```

```

buttonState = digitalRead(buttonPin[x]); //sulgeb seifi ukse
if (buttonState == HIGH && (buttonPin[x] == 2 ||
    buttonPin[x] == 3 || buttonPin[x] == 4 ||
    buttonPin[x] == 5))
{
    n++;
    tone(6, notes[0]);
    delay(200);
    noTone(6);
    for(angle = 0; angle < 90; angle += 5)
    {
        myServo.write(angle);
        //Näitab servo mootori hetke nurka. Kuid seda me ei kasuta hetkel.
        //Serial.print("angle: ");
        //Serial.println(angle);
        delay(100);
    }
}
}
}

else{
    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(0,0);
    display.println("Kood on vale, oodake!");
    display.display();
    delay(5000);
    valeKood++; //lisa järjestikuse vale koodi lugejale 1

    Serial.println("Kood vale, oodake !"); // nulli sisestatud kood
    kood[0] = 0;
    kood[1] = 0;
    kood[2] = 0;
    kood[3] = 0;
} //nullib RFID kaardi sisestatud koodi
rfid.uid.uidByte[0] = 0;
rfid.uid.uidByte[1] = 0;
rfid.uid.uidByte[2] = 0;
rfid.uid.uidByte[3] = 0;

```

```

}
if (valeKood < 3)//kui kolm korda järjest vale kood, siis lühikene paus, kui ei, siis pikk paus
{
    delay(200);
}
else
{
    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(0,0);
    display.println("Kood on vale, oodake 30 sekundit!");
    display.display();
    tone(6, notes[0]);
    delay(30000);
    noTone(6);
    valeKood = 0;//nullida vale koodi lugeja
}
}

/**
 * Helper routine to dump a byte array as hex values to Serial.
 */
void printHex(byte *buffer, byte bufferSize) {
    for (byte i = 0; i < bufferSize; i++) {
        Serial.print(buffer[i] < 0x10 ? " 0" : " ");
        Serial.print(buffer[i], HEX);
    }
}

/**
 * Helper routine to dump a byte array as dec values to Serial.
 */
void printDec(byte *buffer, byte bufferSize) {
    for (byte i = 0; i < bufferSize; i++) {
        Serial.print(buffer[i] < 0x10 ? " 0" : " ");
        Serial.print(buffer[i], DEC);
    }
}

```


Süsteemi kasutusjuhtum

Allikad

Arduino Projects Book,

https://bastiaanvanhengel.files.wordpress.com/2016/06/arduino_projects_book.pdf

(2.05.2018)

Arduino Code:

Adafruit GFX

Adafruit SSD1306

Kokkuvõte

Selle tarkvara projektiks oli koostada süsteem, mis avaks seifi ukse, ning selleks vajalik kirjutatud programmikood, mis paneks kogu süsteemi tööle. Projekt on tehtud kasutades Arduino tarkvara, süsteem on samuti realiseeritud Arduino Uno peal. Süsteemi põhimõte - küsida kasutajalt kaarti ning 4-kohalist PIN-koodi, et avada seifi uks. Õige kaardi ja PIN-koodi sisestamisel avatakse uks kuniks kasutaja vajutab suvalist nuppu, mille tagajärjel uks sulgub. Vale koodi või kaardi näitamisel/sisestamisel tuleb kiri, et kaart või PIN-kood on vale ning tuleb oodata 30 sekundit, et uuesti proovida sisestada PIN-kood ja kaart.

Tulevaste ideede loetelu

- Ehitada koodi põhjal töötav seif.
- Lisada automaatselt avanev uks.
- Teha automaatselt sulgumisel lukustuv uks.
- Teha admini RFID kaart, mis saab seifi vale koodi korral koheselt avada.
- Admin saab muuta koodi.
- Suurem(rohkemate sisenditega) numberpad.
- Arduino Mega 2560 kasutuselevõtt(rohkem mälu).