



Digitaalilukko-Raportti

Iiro Hämäläinen, Daniel Schmidt, Julian Levä

Harjoitustyö
7.12.2023

SISÄLLYS

1	JOHDANTO	3
2	Suunnittelu	4
2.1	Toimintasuunnitelma	4
2.1	Työnkulku	4
2.2	Työnjako.....	5
2.3	KytKentä kaavio.....	5
3	Toteutus	6
3.1	HW-kytKentä kuva.....	6
3.2	Koodi selostus	6
4	Työnkulku	8
	Liitteet	9

1 JOHDANTO

Valitsimme harjoitustyöksi digitaalilukon, jonka toiminnon periaate on, että annetaan koodi näppäimistöllä ja kun koodi on oikein servo moottori avaa lukon joksikin ajaksi. Tavoittelemme arvosanaa 4, joten teimme työn 4 arvosanan kriteerien perustalla.

Työssä käytetyt lohkot EEPROM, a/d-muunnin, watchdog ja TWI(I2C). EEPROM on pin koodin tallennukseen. A/D-muunnin on potentiometristä tulevan signaalin muokkaamiseen, että nano ymmärtää sitä. Watchdog on resetoimiseen eri kohdissa. TWI on lcd näytön käyttöön. Keskeytyksenä on käyetty ulkoista nappi keskeytystä, kun nappia painaa tapahtuu watchdog reset.

2 Suunnittelu

2.1 Toimintasuunnitelma

Komponentit:

- Näppäimistö: tunnusluvun syöttäminen ja lähettäminen
- LCD-näyttö: tilaviestien näyttäminen
- Servomoottori: simuloi lukon salvan asentoa (lukossa – avoinna)
- LED:it: lukon tila: punainen = lukossa, vihreä = avoinna
- Reset-painike: Watchdog reset
- Potentiometri: vastusarvo -> A/D-muunnos -> säädetään lukon aukioloaikaa T välillä 10-20 s. Muutettu vastusarvo rekisteröidään resetoinnin yhteydessä

Ennakkovalmistelut:

- PIN-koodi kirjoitettava EEPROMille erillisellä ohjelmalla

Prosessit:

1. Punanen ledi sytytetään
2. Servomoottori säädetään lukko-asentoon (0 astetta)
3. PIN-koodi kirjoitettuna EEPROMille -> haetaan ohjelmaan
4. Näyttöön kirjoitetaan jotai "Syötä PIN: "
5. Odotetaan syötettä näppikseltä
6. Kun painettu tähteä niin verrataan syötettä oikeaan PIN:iin
7. Jos väärin niin print "Väärin", odota 2 s ja palaa kohtaan 4
8. Jos oikein niin sytytä vihreä LED ja käännä servomoottoria (180 astetta)
9. Odota aika T ja palaa kohtaan 1

2.1 Työnkulku

Toimintasuunnitelmasta otettiin yksittäisiä ominaisuuksia kehitykseen, ja lopuksi näistä luotiin valmis kokonaisuus. Suunnitelmaa muokattiin työn edetessä.

2.2 Työnjako

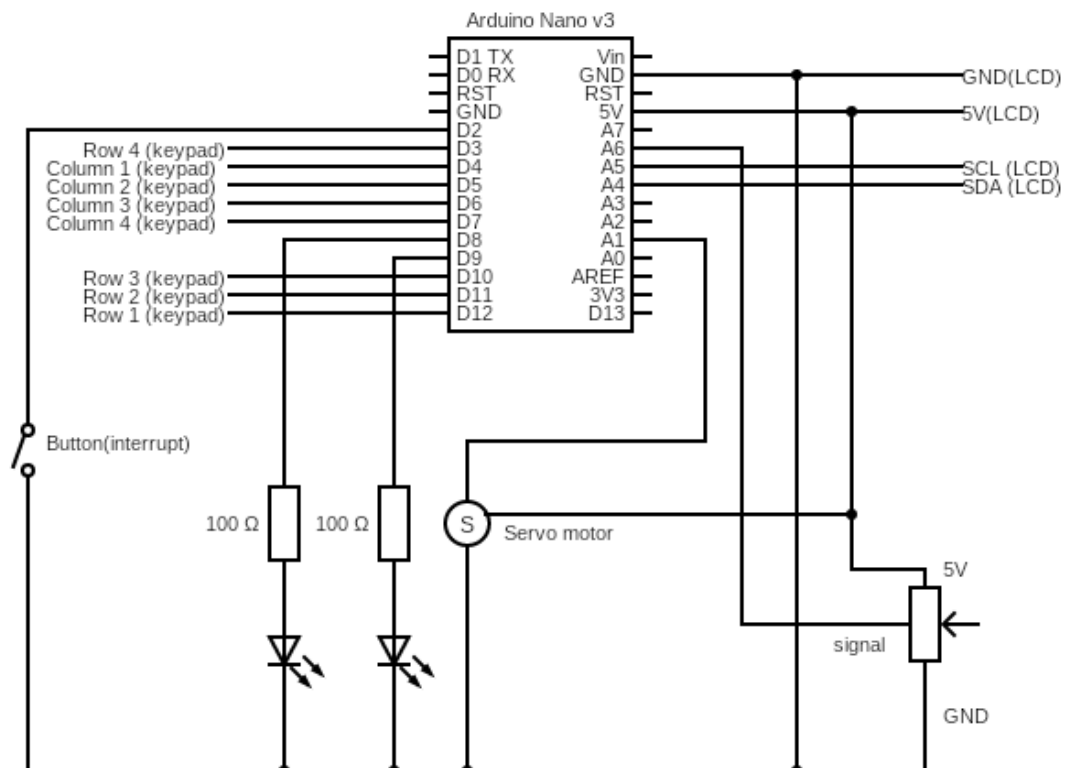
Iiro: Tinkercad-simulaatio, kytkentä, ohjelmointi (mm. näppäimistö, potentiometri ml. A/D-muunnos, nappikeskeytys, loop ml. näppäilyjen lukeminen)

Julian: kytkentä, ohjelmointi (mm. Watchdog, näyttö, servo, potentiometri ml. A/D-muunnos, nappikeskeytys)

Daniel: toimintasuunnitelma, ohjelmointi (mm. LED:it, EEPROM ml. pin-koodin lähetytys ja vertailu)

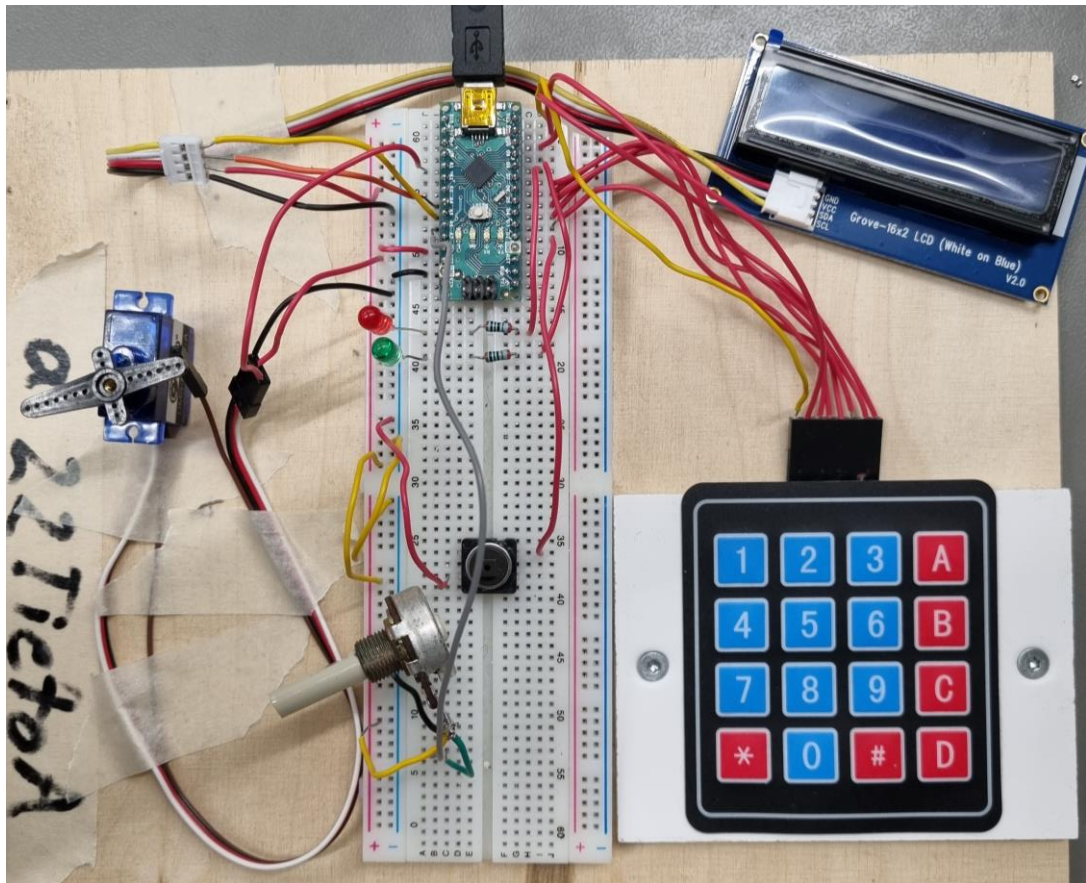
Työnjaossa oli paikoin päällekkäisyyksiä - yllä olevat merkinnät ovat suuntaa antavia.

2.3 Kytkentä kaavio



3 Toteutus

3.1 HW-kytkentä kuva



3.2 Koodi selostus

Pin-koodi alustetaan EEPROM muistiin erillisessä koodissa, jotta voidaan todistaa EEPROM muistin toiminta.

Funktiossa loadPinCode() haetaan pin-koodi EEPROMista.

Funktiossa submitPinCode() vertaillaan käyttäjän syöttämää pin-koodia ja EEPROMista saatua oikeaa pin-koodia.

Funktio buttonInterrupt() tapahtuu, kun erillistä nappia painetaan.

Funktiossa ADC_init() alustetaan AD-muunnokseen vaadittavat arvot.

Funktiossa ADC_read(uint8_t ch) tehdään AD-muunnos ja palautetaan saatu arvo.

Setupissa tehdään alustava työ, jotta ohjelma toimii normaalisti ja oikein.

Loopissa laitetaan ensin punainen valo päälle, käännetään servo lukko asentoon ja kirjoitetaan näytölle "Anna Pin:" teksti. Sen jälkeen odotetaan käyttäjältä napin painalluksia, jotka tallennetaan input muuttuun. Kun painetaan nappia #, niin

ohjelma suorittaa vertailu funktion, joka vertaa syötettyä pin-koodia ja oikeaa pin-koodia. Jos pin-koodi on oikein, näytölle tulostetaan teksti "Oikein" ja lukko aukeaa (eli servo käännetään auki asentoon) ajaksi, joka on määritelty potentiometriä kääntämällä 10-20 sekunniksi. Jos pin-koodi on väärin, niin näytölle tulostetaan teksti "Vaarin" 2 sekunniksi, jonka jälkeen se pyyhitään ja palataan koko loopin alkuun. Jos käyttäjä painaa erillistä nappia tapahtuu watchdog-reset, joka käytännössä pyyhkii käyttäjän antaman pin-koodin ja pin-koodin kirjoittaminen aloitetaan alusta. Sama tapahtuu, jos ensimmäisen näppäimistön napin painamisen jälkeen odotetaan 8 sekuntia.

4 Työnkulku

Kehitysvaiheessa joitakin yksittäisiä ominaisuuksia testattiin erikseen ennen näiden liittämistä pääohjelmaan. Pääohjelmaa osasista rakennettaessa koko järjestelmää testattiin vähän väliä lisätyillä ominaisuuksilla. Lopuksi vahvistettiin järjestelmän prosessilistan mukainen käyttäytyminen.

Vahvistimme osaamistamme suunnitella ja toteuttaa pienimuotoinen sulautettu järjestelmä. Opimme uutta komponenttien ohjaamisesta ohjelmakoodissa. Järjestelmän ominaisuuksien yhteen liittäminen oli paikoin luovaa työtä. Haasteita liittyi erityisesti siihen, miten eri lohkoja saisi ”mahdutettua” järjestelmään. Tehtävänannon kriteereitä täyttämään valittiin ominaisuuksia, jotka eivät olleet järjestelmän toiminnan kannalta välttämättömiä.

Esimerkiksi pohdimme Timer2 -lohkon käyttöä laskurina. Ongelmaksi muodostui, ettemme keksineet keinoa käynnistää laskuri hallitusti mielivaltaisella hetkellä. Sen sijaan laskuri käynnistyy heti sen asettamisen yhteydessä.

Liitteet

Pin-koodin EEPROM alustuskoodi:

```
#include <EEPROM.h>

const int PIN_LENGTH = 4;
// Tehdään oikea pin-koodi
const char pin_in[] = {'1', '2', '3', '4'};

void setup() {
    // Viedään pin-koodi EEPROM muistiin
    for (int i = 0; i < PIN_LENGTH; i++) {
        EEPROM.update(i, pin_in[i]);
    }
}

void loop() {
}
```

Koko koodi:

```
#include <Servo.h>
#include "rgb_lcd.h"
#include <EEPROM.h>
#include <Wire.h>
#include <String.h>
#include <Key.h>
#include <Keypad.h>
#include <avr/wdt.h>

const int BUTTON_PIN = 2;
int sensorPin = A6; // Potentiometrin pinni

Servo lock;
rgb_lcd lcd;

const float MAX_MS = 20000; // Lukon aukiolon maksimiaika
float ms_time; // Lukon aukioloaika
int val; // AD-muunnos arvo

const int LOCKED_DEG = 0; // servon asento asteissa, kun lukossa
const int OPEN_DEG = 180; // servon asento asteissa, kun auki
const char PIN_CODE[4]; // EEPROM:sta haettu arvo
const int PIN_LENGTH = 4;
const int ROW_NUM = 4;
const int COLUMN_NUM = 4;

// Näppäimistön alustus
```

```

char keys[ROW_NUM][COLUMN_NUM] = {
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};

// Sijoitetaan näppäimistön napit arduinon pinneihin
byte pin_rows[ROW_NUM] = {12, 11, 10, 3};
byte pin_column[COLUMN_NUM] = {4, 5, 6, 7};

// Sijoitetaan alustetut arvot muuttujaan
Keypad keypad = Keypad(makeKeymap(keys), pin_rows, pin_column, ROW_NUM, COLUMN_NUM);

String input; // Salasana, joka tulee käyttäjän syötteestä
int status = 0; // Muuttuja näppäimistön syötteen hallintaan
String pin; // Muuttuja EEPROM:ista saadun oikean pin-koodin tallentamiseen

// Haetaan oikea pin-koodi EEPROM:ista
void loadPinCode() {
    for (int i = 0; i < PIN_LENGTH; i++) {
        EEPROM.get(i, PIN_CODE[i]);
    }
}

// Pin-koodien vertaus funktio
void submitPinCode() {
    lcd.setCursor(0, 1);
    if (input == pin) {
        // Jos pin-koodit ovat samat,
        // niin kirjoitetaan näyttöön oikein ja käännetään servoa
        lcd.print("Oikein");
        lock.write(OPEN_DEG);
        PORTB=B00000010; // Sammutetaan punainen valo, vihreä päälle
        // Sammutetaan watchdog, jotta ei tapahdu resettiä delayn aikana
        wdt_disable();
        delay(ms_time); // Viivytetään niin kauan kuin halutaan lukon olevan auki
    } else {
        //Jos pin-koodit ovat erit, niin kirjoitetaan näyttöön vaarin
        // ja pyyhitaan se 2 sekunnin kuluttua
        lcd.print("Vaarin");
        delay(2000);
        lcd.setCursor(0, 1);
        lcd.print("
    ");
    }
}

// Keskeytys, joka tapahtuu, kun painetaan nappia
void buttonInterrupt() {
    wdt_enable(WDTO_15MS);
}

```

```

void ADC_init() {
    // Asetetaan vertailu arvo
    ADMUX |= (1 << REFS0);
    // Asetetaan muunnos pinnistä A6
    ADMUX |= (1 << MUX2) | (1 << MUX1) | (0 << MUX0);
    // Sallitaan AD-muunnos
    ADCSRA |= (1 << ADEN);
}

uint16_t ADC_read(uint8_t ch) {
    // Aloitetaan AD-muunnos
    ADCSRA |= (1 << ADSC);
    // Odotetaan muunnoksen valmistumista
    while (ADCSRA & (1 << ADSC)) {
        // Odotus loop
    }
    // Palautetaan AD-muunnoksesta saatu arvo
    val = ADC;
    return val;
}

void setup() {
    DDRD &= ~(1 << DDD2); // Pinni input:iksi
    PORTD |= (1 << PORTD2); // Pinni input-pullup:iksi
    // Nappikeskeytys
    attachInterrupt(digitalPinToInterrupt(BUTTON_PIN), buttonInterrupt, FALLING);
    Serial.begin(9600);
    lock.attach(A1); // Servon alustus
    DDRB=B00000011; // LED:ien pinnit output:eiksi
    // Näytön alustus
    lcd.begin(16, 2);
    lcd.setRGB(255, 0, 0);
    loadPinCode();
    pin = String(PIN_CODE); // Muutetaan EEPROM:ista saatu pin stringiksi
    Serial.println(PIN_CODE);
    Serial.println(pin);
    ADC_init(); // AD-muunnos päälle
    delay(300); // Delay, jotta AD-muunnos saadaan päälle
    uint16_t sensorValue = ADC_read(sensorPin); // AD-muunnoksesta saatu arvo
    // Vertailu, joka määrittää, kuinka kauan lukko on auki
    if(sensorValue <= 500){
        ms_time = 10000;
    } else {
        ms_time = (sensorValue * MAX_MS) / 1023.0;
    }
    Serial.print("Raw ADC Value: ");
    Serial.print(sensorValue);
    Serial.print(", Time (ms): ");
    Serial.println(ms_time);
    delay(500); // Delay, jotta kaikki saadaan alustettua oikein
}

```

```

}

void loop()
{
    PORTB=B00000001; // Punainen LED päälle, vihreä pois
    lock.write(LOCKED_DEG); // Servo lukkossa asentoon
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Anna PIN: ");

    // Kerätään nappien painallukset
    char key = keypad.getKey();
    if (key) {
        status = 1;
        lcd.print("*");
        input = key; // Painetun napin arvo tallennetaan muuttujaan
        Serial.println(key);
        // Watchdog reset tapahtuu, jos mitään nappia ei paineta 8 sekuntiin
        wdt_enable(WDTO_8S);
    }
    while (status == 1) {
        key = keypad.getKey();
        if (key) {
            Serial.println(key);
            // Jos # näppäintä painetaan, siirrytään vertailuun
            if (key == '#') {
                submitPinCode();
                status = 0;
                break; // Poistutaan loopista
            }
            input = input + key; // Lisätään painettu nappi muuttujaan
            lcd.print("*");
            Serial.println(input);
            // Restoidaan watchdog timer, joka kerta kun jotain nappia painetaan
            wdt_reset();
        }
    }
    // Sammutetaan watchdog, jotta resettejä ei tapahdu, kun nappeja ei paineta
    wdt_disable();
}

```