# **Ultraäänipohjainen Kulunvalvontajärjestelmä** NJS – Projektidokumentaatio

**Tekijät:** [Ryhmän jäsenten nimet tähän]

**Kurssi:** Sulautetut järjestelmät

**Päivämäärä:** 27.11.2025

## 1. Johdanto ja Tavoitteet

**Ultraäänipohjainen Kulunvalvontajärjestelmä NJS** on älykäs henkilöliikenteen seuranta- ja kulunvalvontalaite. Projekti toteuttaa hajautetun sulautetun järjestelmän, joka koostuu kolmesta erillisestä mikrokontrolleriyksiköstä, jotka kommunikoivat keskenään sarjaliikenteen avulla.

Laitteen päätarkoitus on laskea huoneessa olevien ihmisten määrä kahden ultraäänianturin avulla ja ohjata automaattista ovea (servo) sekä infonäyttöä reaaliajassa.

Projekti on suunniteltu täyttämään kurssin **arvosanan 5** vaatimukset hyödyntämällä edistyneitä tekniikoita, kuten suoraa rekisteriohjausta, keskeytyksiä ja kustomoitua tiedonsiirtoprotokollaa.

## 2. Järjestelmän Arkkitehtuuri ja Laitteisto

Järjestelmä on jaettu kolmeen itsenäiseen yksikköön:

1. **Master-yksikkö (Sensor Unit):** Järjestelmän keskusyksikkö. Lukee antureita ja päättelee kulkusuunnan logiikan.
2. **Slave 1 -yksikkö (Door Unit):** Toimilaiteyksikkö, joka ohjaa fyysistä ovea (servoa) ja lukee käyttäjän painonappeja.
3. **Slave 2 -yksikkö (Display Unit):** Käyttöliittymäyksikkö, joka visualisoi ihmismäärän graafisesti OLED-näytöllä.

### 2.1 Master-yksikkö (Arduino Nano)

Tämä yksikkö hoitaa mittauksen ja päätöksenteon. Se on koodattu optimoidusti käyttäen suoraa rautatason ohjausta (Direct Port Manipulation).

**Kytkentäkaavio:**

| **Komponentti** | **Arduinon Pinni** | **Rekisteri / Portti** | **Selite** |
| --- | --- | --- | --- |
| **Trig 1 (Ulko)** | D9 | PORTB 1 | Lähetyspulssi anturille 1 |
| **Echo 1 (Ulko)** | D8 | PORTB 0 | Paluupulssi (Luetaan PCINT-keskeytyksellä) |
| **Trig 2 (Sisä)** | D7 | PORTD 7 | Lähetyspulssi anturille 2 |
| **Echo 2 (Sisä)** | D6 | PORTD 6 | Paluupulssi (Luetaan PCINT-keskeytyksellä) |
| **Vihreä LED** | D3 | PORTD 3 | Indikaattori: Henkilö saapuu |
| **Punainen LED** | D4 | PORTD 4 | Indikaattori: Henkilö poistuu |
| **Buzzer** | D5 | PORTD 5 | Äänimerkki kulun yhteydessä |
| **Data Out (TX)** | D1 (TX) | UDR0 | Lähettää komennot Slave-yksiköille |
| **GND** | GND | - | Yhteinen maa kaikkien laitteiden kanssa |

### 2.2 Slave 1: Oviyksikkö (Arduino Nano)

Tämä yksikkö toimii "orjana" ja kuuntelee Masterin komentoja, mutta toimii myös itsenäisesti manuaalisilla avauspainikkeilla.

**Kytkentäkaavio:**

| **Komponentti** | **Pinni** | **Selite** |
| --- | --- | --- |
| **Servo** | D9 | Ohjaa oven asentoa (0-90 astetta) |
| **Nappi (Avaa)** | D2 | Manuaalinen avaus (Kytkin maata vasten, INPUT\_PULLUP) |
| **Nappi (Sulje)** | D3 | Manuaalinen sulku (Kytkin maata vasten, INPUT\_PULLUP) |
| **Data In (RX)** | D0 (RX) | Vastaanottaa dataa Masterilta |
| **GND** | GND | Yhteinen maa |

### 2.3 Slave 2: Näyttöyksikkö (ESP32)

Tämä yksikkö visualisoi datan käyttäjälle.

**Kytkentäkaavio:**

| **Komponentti** | **Pinni (ESP32)** | **Selite** |
| --- | --- | --- |
| **OLED SDA** | GPIO 4 | I2C Data |
| **OLED SCL** | GPIO 5 | I2C Clock |
| **Data In (RX)** | GPIO 16 | Vastaanottaa dataa Masterilta (Serial2) |
| **GND** | GND | Yhteinen maa |

## 3. Ohjelmisto ja Tekniset Ratkaisut (Arvosana 5)

Projektissa on panostettu koodin optimointiin ja rautatason ymmärrykseen. Alla on eritelty, miten vaativimmat kriteerit on täytetty.

### 3.1 Käytetyt AVR-mikrokontrollerin sisäiset lohkot (4 kpl)

1. **Timer2 (Ajastin):**
   * Konfiguroitu CTC-moodiin (Clear Timer on Compare Match).
   * Tuottaa 1 millisekunnin välein keskeytyksen (TIMER2\_COMPA\_vect).
   * **Miksi?** Korvaa Arduinon millis() ja delay() funktiot omalla kevyemmällä toteutuksella, jolloin prosessori ei jumitu odottamaan.
2. **USART (Sarjaliikenne):**
   * Alustettu ja käytetty suoraan rekistereiden kautta (UBRR0, UCSR0B, UDR0).
   * Baud rate on asetettu manuaalisesti laskemalla jakaja (prescaler).
   * Lähettää dataa muille yksiköille ilman Arduinon Serial-kirjastoa.
3. **PCINT (Pin Change Interrupt):**
   * Ultraäänianturien ECHO-signaalit luetaan keskeytyksillä (PCINT0\_vect ja PCINT2\_vect).
   * **Hyöty:** Ohjelma ei pysähdy odottamaan äänen paluuta (kuten pulseIn()-funktio tekisi), vaan mittaus tapahtuu taustalla, mikä mahdollistaa moniajon.
4. **GPIO (Portit):**
   * Pinnien tiloja (LEDit, Triggerit) ohjataan suoraan porttirekistereillä (PORTB, PORTD, DDRB).
   * Tämä on huomattavasti nopeampaa (vain muutamia kellojaksoja) verrattuna digitalWrite-funktioon.

### 3.2 Aito tiedonsiirtoyhteys

Laitteet eivät toimi yksin, vaan muodostavat verkon.

* **Protokolla:** Yksisuuntainen sarjaliikenne (UART 9600 baud, 8N1).
* **Viestit:**
  + IN = Henkilö tuli sisään.
  + OUT = Henkilö poistui.
  + COUNT:X = Synkronoi ihmismäärän (X) kaikkiin laitteisiin.

### 3.3 Kaksisuuntainen ja looginen käyttöliittymä

* **Palaute käyttäjälle:** OLED-näyttö kertoo tilanteen, LEDit (vihreä/punainen) ja Summeri antavat välittömän audiovisuaalisen palautteen.
* **Syöte käyttäjältä:** Servo-yksikön napit mahdollistavat järjestelmän manuaalisen yliajon (oven pakkoavaus), mikä lisää käyttövarmuutta.

### 3.4 Innovatiivisuus

* Järjestelmän jakaminen kolmeen eri prosessoriin simuloi oikeaa teollisuusautomaatiota (Sensoriyksikkö erikseen, toimilaitteet erikseen).
* Tilakone-logiikka (State Machine) tunnistaa kulkusuunnan pelkällä kahden anturin aikajärjestyksellä ilman monimutkaisia kirjastoja.

## 4. Ohjelmistoiminnan kuvaus (Logiikka)

### Master-yksikön tilakone

Ohjelma pyörii loopissa, joka tarkkailee kahta tilaa:

1. **IDLE (Tila 0):** Odottaa, että jompikumpi anturi aktivoituu (< 10 cm).
   * Jos anturi 1 (Ulko) aktivoituu ensin -> Siirrytään tilaan 1 ("Tulossa sisään").
   * Jos anturi 2 (Sisä) aktivoituu ensin -> Siirrytään tilaan 2 ("Menossa ulos").
2. **ACTIVE (Tila 1 tai 2):** Seurataan, aktivoituuko toinen anturi tietyn aikaikkunan (800ms) sisällä.
   * Jos toinen anturi aktivoituu -> **Tapahtuma vahvistettu** -> Lähetä data -> Palaa IDLE.
   * Jos aika loppuu -> **Virhe/Käännös** -> Palaa IDLE ilman laskentaa.

### Virheensieto (Debounce & Hystereesi)

* Kun ihminen kulkee ohi, järjestelmä pitää **500ms tauon** (dead time) laskennan jälkeen. Tämä estää sen, että yksi hitaasti kävelevä henkilö laskettaisiin kahteen kertaan.

## 5. Yhteenveto

**Ultraäänipohjainen Kulunvalvontajärjestelmä NJS** on onnistunut kokonaisuus, joka yhdistää rautason ohjelmoinnin ja hajautetun järjestelmän hallinnan. Laite toimii luotettavasti, ja sen modulaarinen rakenne mahdollistaa osien helpon vaihtamisen tai päivittämisen.

Projekti täyttää asetetut korkeimmat arvosanakriteerit osoittamalla syvällistä ymmärrystä mikrokontrollerin toiminnasta (rekisterit, keskeytykset) pelkän valmiiden kirjastojen käytön sijaan.