

Harjoitus 4

HUOM! Tutustu ensin Harjoitustehtävien palauttaminen -dokumenttiin ja -videoon, niin säästyt monelta harmilta! Dokumentin löydät Harjoitukset (3 op) -osion ohjeistustekstistä Palautuslaatikkojen yläpuolelta yhdessä **mallivastauspohjan** kanssa.

Ohjelmointitehtävät

Harjoitukset 2-5 ovat ohjelmointitehtäviä. Tehtävät vaikeutuvat loppua kohti ja niissä käsitellään sellaisia asioita, joita tarvitaan koulutusohjelman myöhemmissä vaiheissa. Tehtävät on siis laadittu yhteistyössä tulevien kurssien opettajien kanssa.

Myös ohjelmointiharkat palautetaan kukin omaan palautuslaatikkoonsa. **HUOM!** Lue tämä ohje tehtävien palauttamistaarten ennen kuin alat niitä tekemään! Näin säästyt monelta harmilta.

Tästä löydät **Mallivastauspohja**.

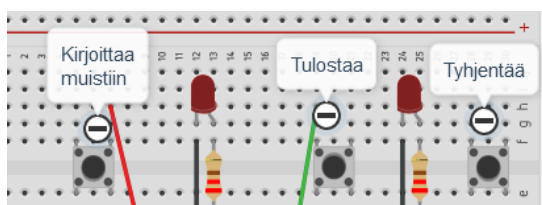
Palautuslaatikoissa on päällä nk. Lähetä arvioitavaksi -toiminto. Voit siis säilyttää halutessasi keskeneräistä vastausta palautuslaatikossa. **Muista painaa lähetä arvioitavaksi -nappia kun työ on valmis.**

Tähän harjoitukseen kuuluu **kolme tehtävää, jotka oikeuttavat eri arvosanoihin**. Ensimmäisen tehtävän tekemällä saavuttaa arvosanan 1. Ensimmäisen ja toisen tehtävän tekemällä saavuttaa arvosanan 3. Kaikki kolme tehtävää tekemällä saavuttaa arvosanan 5. Tee tehtäviä oman osaamisesi ja ajankäyttömahdollisuuksiesi mukaisesti. Kurssin arvosana muodostuu tehtävien arvosanojen keskiarvosta. Hyväksytty kurssisuoritus edellyttää arvosanan 1 saavuttamista kaikista harjoitusosioista.

Tutustumiskurssilaiset: Kurssin arvosana ei vaikuta opiskelijavalintaan (valintaa varten kurssi tulee olla kuitenkin hyväksytysti suoritettu).

Harjoitustehtävissä käytetään Harjoitus 3:n mukaista kytkentää, johon ensimmäisessä tehtävässä liitetään vielä yksi painike. **Joudut tätä tehtävää varten etsimään netistä tietoa, kuinka Arduinon EEPROM-muistiin kirjoitetaan ja kuinka sieltä luetaan.**

Tehtävissä on erilaisia painonappeja käytössä, joten merkitse selkeästi, mitä mikäkin nappi tekee. Esimerkiksi seuraavalla tavalla:



Tehtävä 1 (ARVOSANA 3): EEPROM-muistiin kirjoitus

Liitä edellisen Harjoitus 3:n kytkentään vielä kolmaskin painonappi. Tätä nappia ei voi käskyttää keskeytysrutiinilla, koska Arduino Uno käsittelee vain kahta keskeytystä. Tee siis looppiin toiminto, joka nappia painamalla kirjoittaa luvun mikrokontrollerissa olevaan 1 kilotavun sisäiseen EEPROM-muistiin.

Tee kirjoittamista varten laskuri (muuttuja), jossa säilytät muistipaikan numeroa, mihin olet kirjoittamassa. Alussa se on nolla.

Tarkastele, onko nappi painettu (kuten harjoitus 3:n tehtävässä 1). Jos nappi on painettu, kirjoita ensimmäiseen muistipaikkaan jokin luku (tässä tehtävässä ei ole väliä, minkä luvun sinne kirjoitat, myöhemmin kirjoitetaan anturilta tulevia arvoja). Kasvata tämän jälkeen muistipaikan osoittavaa laskuria, jotta seuraava luku kirjoitetaan seuraavaan muistipaikkaan. Tässä vaiheessa kannattaa pyytää ohjelmaa odottamaan hetken delay-käskyllä, jotta napin painallus kirjautuu vain yhden kerran. Mekaaninen nappi tulkitsee hyvinkin nopean painalluksen useaksi painallukseksi. Tästä pääsee siis käyttämällä delayta.

Tämän jälkeen ohjelma odottaa uutta painallusta ja kun nappia painetaan, kirjoitetaan jälleen laskurin osoittamaan muistipaikkaan, kasvatetaan muistipaikan osoittavaa laskuria ja odotetaan hetki (jotta nappi toimisi oikein).

Tässä tehtävässä ei tarvitse välittää muistin mahdollisesta täyttymisestä.

Vinkki!: voit testata toimintaa esimerkiksi niin, että samalla kun kirjoitat muistiin, niin kirjoitat seuraavalla koodirivillä myös sarjaporttiin saman arvon. Eli joka painalluksella tieto menee muistiin ja sarjaporttiin, niin näet että logiikkasi toimii.

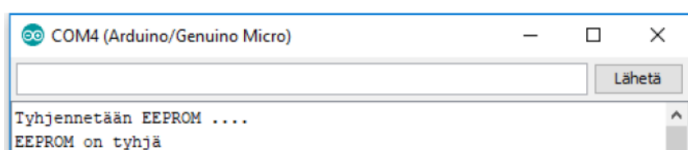
Tehtävä 2 (ARVOSANA 3): EEPROM-muistin tyhjennys ja muistin sisällön tulostus sarjaporttiin

Tehtävä: tee **keskeytysrutiinit**, jossa edellisen harjoitustehtävän kytkentäkaaviossa olevan painonapin 2 painalluksella tyhjennetään EEPROM-muistin sisältö ja painonapilla 1 tulostetaan EEPROM-muistin sisältö sarjaporttiin (lisäksi sinulla on edellisessä tehtävässä tekemäsi painonappi, jolla kirjoitetaan lukuja muistiin).

Tulostusrutiinissa tulostetaan kaikki EEPROM-muistista löytyvät luvut sarjaporttiin kukin omalle rivilleen. Koska et tiedä paljonko lukuja muistissa on, niin joudut käymään for-silmukassa koko muistin läpi ja tulostamaan epätyhjät arvot.

Tyhjennysrutiinissa muisti käytännössä kirjoitetaan täyteen nollaa. Tee tämä for-silmukassa käymällä kaikki muistipaikat läpi. Tulosta sarjaporttiin teksti ”Tyhjennetään EEPROM...” kun tyhjennys alkaa ja ”EEPROM on tyhjä.” kun tyhjennys on päättynyt (kts. tulostuskuva alla). Muuta myös muistipaikkaa osoittava muuttuja nolaksi, että kirjoittaminen alkaa jälleen muistin alusta.

Kun koodi toimii, niin tyhjennä EEPROM ensin painonapilla 2. Paina sen jälkeen tehtävä 1:n kirjoitusnappia, muutamia kertoja, jolloin muistiin pitäisi kirjoittautua painallusten lukumäärän verran lukuja. Tämän jälkeen tulostusnappia painamalla sarjaporttiin pitäisi tulostua nämä luvut.



Tehtävä 3 (ARVOSANA 5): Lämpötila-arvojen kirjoittaminen muistiin

Tässä tehtävässä voidaan unohtaa Tehtävä 1:n kirjoitusta tekevä painonappi (ja siihen liittyvä koodi loopissa). Muistiin kirjoitettava data luetaan lämpötila-anturilta. Käytä Harjoitus 3:n mukaista kytkentää.

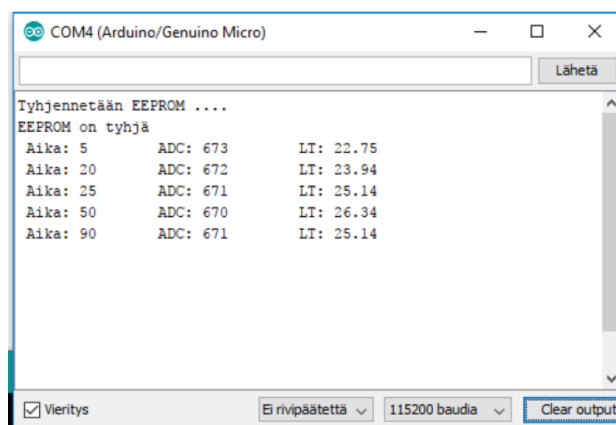
Tehtävänä on tehdä **aliohjelma, jossa tarkistetaan lämpötila-anturin antaman ADC-arvon muutos viiden sekunnin välein**. Jos ADC-arvo on muuttunut edellisestä arvosta, tallennetaan kyseinen sekunti sekä uusi ADC-arvo EEPROM-muistiin. Käytä mikrokontrollerissa olevaa 1 kilotavun sisäistä EEPROM-muistia.

- Käytännössä siis ohjelma pyörii loopissa niin kauan, että on kulunut 5s edellisestä mittauksesta, jonka jälkeen siirrytään aliohjelmaan tallentamaan lämpötilaa. Tarvitset siis timerin, joka mittaa loopissa aikaa ohjelman käynnistytksen alusta ja sen avulla osataan sitten siirtyä 5 sekunnin välein mittaavaan aliohjelmaan.
- Aliohjelmassa kirjoitat EEPROM-muistiin peräkkäisiin muistipaikkoihin ajankohdan ja ADC-arvon, **mutta vain jos ADC on eri** kuin edellisessä mittauksessa (säilytä siis edellinen mitattu ADC-arvo muuttujassa ja vertaa sitä uuteen arvoon).
- Tarvitset muistiin kirjoittamista varten muuttujan, jossa säilytät muistipaikan numeroa, mihin olet kirjoittamassa. Kirjoita ensimmäiseen muistipaikan aika ja sitä seuraavaan muistipaikkaan lämpötila ja kasvata muuttujan arvoa kahdella jne.
- Jos muisti on täynnä aloita kirjoittaminen uudelleen muistin alusta. **Joudut siis tarkistamaan ennen kirjoitusta, onko muisti täynnä** (eli onko muistipaikka, johon olet kirjoittamassa isompi kuin muistin paikkojen lukumäärä)

Lisää tähän tehtävään myös edellä tekemäsi tehtävän 2 tulostusrutiini ja muistin tyhjennysrutiini, jotka toimivat painonapeilla 1 ja 2. **Muokkaa tulostusrutiinia** niin, että siinä tulostetaan EEPROM-muistista löytyvä aika ja sitä vastaava ADC-arvo, lisäksi lasketaan ja tulostetaan ADC-arvoa vastaava lämpötila-arvo (lasketaan samalla muunnoskaavalla, kuin edellisessä Harjoitus 3:ssa). Katso mallitulostus alla. Tarkka tulostuksen ulkoasu ei ole merkityksellinen.

Kun koodi toimii, niin tyhjennä EEPROM ensin painonapilla 2 ja anna sen jälkeen laitteen mitata ADC-arvoja parin minuutin ajan (muuta lämpötilasensorin saamia arvoja klikkaamalla simulaatio tilassa anturia ja siirtämällä sen päälle ilmestyvää liukukytintä) ja kokeile tulostaa painonapilla 1.

Tulostusesimerkki: EEPROM tyhjennettiin ensin. Sitten mitattiin lämpötilaa noin pari minuuttia. Tulostettiin muistin sisältö. Tulostuksen täysin tarkka ulkomuoto ei ole olennainen, kunhan siitä ilmenee alla näkyvät asiat (muistin tyhjennys, aika, ADC ja sitä vastaava lämpötila). Huomaa, että vain muuttuneet arvot on tallennettu muistiin (esim 10s ja 15s kohdalla ei ole ollut muutosta, joten niistä ei ole merkintää muistissa)



Koskee jokaista tehtävää!

MUISTA KOMMENTOIDA KOODIIN KOODIRIVIT TAI KERRO SANALLISESTI JOKAISEN KOODILOHKON TOIMINTA

Kommentointiesimerkit:

```
int mittausarvot[10];    //varataan mittausarvoille 10 muistipaikkaa
int i = 0;               //mittausarvotaulukon indeksi
int keskiarvo, summa;    //alustetaan keskiarvo ja summa muuttujat
```

Muuttujat kannattaa nimetä selvillä nimillä. Lisäksi vielä kommentit

```
//funktiossa laskenta() nollataan ensin muuttujat keskiarvo ja summa. For silmukassa
//lasketaan mittausarvot[] taulun kaikki 10 arvoa yhteen.
//Silmukan jälkeen lasketaan keskiarvo ja tulostetaan se sarjaporttiin.
void laskenta()
```

Koodilohkojen alkuun (esim funktiot) kannattaa kirjoittaa koodin logiikkaa. Toki lohkon sisälläkin kannattaa kommentoida yksittäisiä rivejä tai tapahtumia

```
{
    keskiarvo = 0;
    summa = 0;
    for(int i=0; i<10; i++)
    {
        summa = summa + mittausarvot[i];
    }
    keskiarvo = summa/i;
    Serial.print("Keskiarvo on ");
    Serial.println(keskiarvo);
}
```