

ILMASTOTEKNOLOGIAA

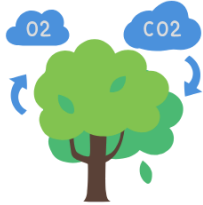
ARDUINO JA PYTHON YMPÄRISTÖTUTKIMUKSESSA



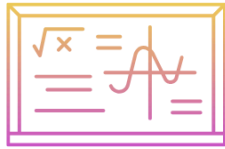
DIGIAVI-HANKE

Sisällysluettelo

Johdanto	2
Laiteversiot	3
Ohjelmointiympäristöjen asennus	4
Arduino IDE.....	4
Python.....	4
Tarvittavat komponentit	7
Kokoaminen	9
Unon ohjelmointi	11
Laitteen käyttö	14
Hiilidioksidipitoisuuden tulkinta.....	16
Tiedonkeruu	17
Unon ohjelmointi uudelleen.....	17
Tiedonkeruuohjelma Pythonilla.....	18
Mittausdatan analysointi	24
Itsenäinen laite	27
Kokoaminen.....	28
Ohjelmointi.....	29
Itsenäinen tiedonkeruulaite	33
SD-kortti ja kortin alustus.....	34
Kokoaminen.....	36
Ohjelmointi.....	38
Käyttöönotto.....	44
Kaikki komponentit	46
Kysymyksiä pohdittavaksi	48
Tutkimusideoita ja vinkkejä	50
CO ₂ -pitoisuuden ja ilmanlaadun tutkimus.....	50
CO ₂ -pitoisuus ja ihmistoiminnan vaikutukset.....	50
Lämpötilan ja kosteuden vaikutus ympäristöön.....	51
Lämpötila ja kosteuspitoisuus biologiassa ja maantiedossa.....	51
Datan analysointi ja matematiikan integrointi.....	52
Ympäristötietoisuuden kasvattaminen.....	52
Pitkän aikavälin seuranta.....	52
Ilmaston ja energiatehokkuuden tutkimus.....	53
Luonnontieteiden integrointi historiaan ja kulttuuriin.....	53
IoT ja teknologia opetuksessa.....	53
Kiinteistöautomaatio	54



BIOLOGIA



MATEMATIIKKA



OHJELMOINTI



ELEKTRONIIKKA

Johdanto

Elämme maailmassa, jossa teknologian ja ympäristön välinen yhteys korostuu yhä enemmän. Energian kulutuksen kasvu ja ilmanlaatuongelmat ovat suuria haasteita, jotka vaativat uudenlaisia ratkaisuja. Mutta entäpä jos voisimmekin käyttää teknologiaa näiden ongelmien ratkaisemiseksi? Entä jos voisimme yhdistää ohjelmoinnin, elektroniikan ja luonnontieteet ymmärtääksemme paremmin ympäristöämme ja luodaksemme kestävämpää tulevaisuutta?

Tämä opas johdattaa sinut ja oppilaasi matkalle, jossa ympäristötietoisuus ja teknologia kohtaavat. Rakennamme ja ohjelmoimme mittalaitteen, jolla pääsemme tutkimaan ympäristöä. Mittaamme ilman hiilidioksidipitoisuutta, lämpötilaa ja kosteutta sekä analysoimme mittausdataa. Opimme tekemään havaintoja ja oivalluksia, jotka voivat johtaa parempaan ymmärrykseen ympäristöstämme.

Laitteen rakentaminen on tehty mahdollisimman helpoksi. Komponenttien määrä on minimaalinen, eikä mitään tarvitse tinata. Osat vain liitetään toisiinsa painamalla ne kiinni liittimiin. Myös ohjelmointi hoituu helposti. Koodit annetaan valmiina ja tarvittavat muutokset on selostettu yksityiskohtaisesti. Halutessasi voit perehtyä syvällisemmin Arduino Unon ohjelmointiin osoitteessa bit.ly/yhlakoulu-arduino. Tarvittaessa opettaja tai IT-tuki voi asentaa tarvittavat ohjelmointiympäristöt valmiiksi kappaleen "Ohjelmointiympäristöjen asennus" ohjeilla.

Oppaan kautta saat kosketuksen ympäristöteknologiaan, ohjelmointiin, elektroniikkaan ja mittaustiedon analysointiin käytännönläheisellä tavalla. Opit, miten Arduino Uno mikro-ohjain yhdistetään mittausantureihin ja miten se ohjelmoidaan. Opit keräämään ja analysoimaan mitattua dataa. Lisäksi pääset soveltamaan luonnontieteitä ja matematiikkaa monipuolisiin ympäristötutkimuksiin. Projekti kehittää ongelmanratkaisutaitoja ja avaa oven kestävä teknologian mahdollisuuksiin. Työn tulokset voivat olla alku suuremmalle keskustelulle ympäristön hyvinvoinnista ja ihmisen vaikutuksesta siihen. Pienetkin havainnot voivat innostaa tekemään kestävämpiä valintoja ja vaikuttamaan yhteisöön positiivisella tavalla. Opimme yhdessä, miten teknologia voidaan käyttää ihmisten ja ympäristön hyväksi ja miten voimme rakentaa kestävämpää tulevaisuutta.

Laiteversiot

Materiaalissa on ohjeet kolmen eri version valmistamiseksi mittalaitteesta. Valitse niistä tarkoitukseesi parhaiten sopiva malli.

Sivuilla 7-15 on esitetty laitteen perusversio, jossa Arduinon ohjelmointiympäristö vastaanottaa ja esittää mittausdatan. Sivujen 17-23 ohjeilla koodataan Python-ohjelma vastaanottamaan mittaustiedot ja tallentamaan ne taulukkolaskentaohjelman ymmärtämään CSV-tiedostoon. Sivulla 24-26 käsitellään datan siirto taulukkolaskentaohjelmaan. Tämä laiteversio mahdollistaa sekä lyhyt- että pitkäkestoiset mittausjaksot tiedontallennuksella, mutta se vaatii tietokoneen toimiakseen.

Sivuilla 27-32 on esitetty itsenäisesti ilman tietokonetta toimiva laite, joka esittää mittaustulokset LCD-näytöllä. Tämä versio on helppo viedä haluttuun mittauspaikkaan, mutta sillä voidaan tehdä lähinnä vain hetkellisiä mittauksia, koska siinä ei ole tallennustoimintoa.

Sivuilla 33-45 on lopuksi ohjeet itsenäisesti toimivan tiedonkeruulaitteen valmistamiseksi. Laite tallentaa mittausdatan SD-muistikortille, josta se voidaan myöhemmin siirtää taulukkolaskentaohjelmaan analysoitavaksi. Tämä versio on myös helppo siirtää haluttuun mittauspaikkaan ja sillä voidaan tehdä pitkäkestoisia mittauksia.



Ohjelmointiympäristöjen asennus

Arduino IDE

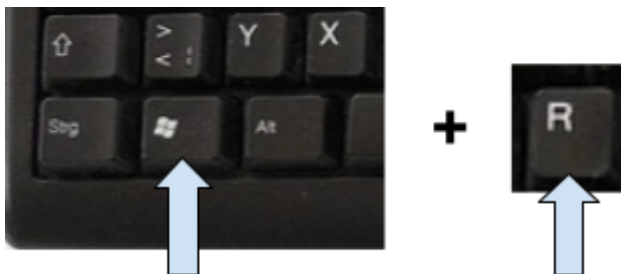
Uno mikro-ohjaimen ohjelmointiin käytettävä ohjelma on nimeltään “Arduino IDE”. Tarkista, löytyykö ohjelma käyttämästäsi tietokoneesta. Jos ohjelmaa ei löydy, voit ladata ja asentaa sen.

1. Siirry linkistä Arduinon nettisivulle: www.arduino.cc/en/software
2. Lataa ja asenna ohjelma, katso tarvittaessa [video asennuksesta](#).

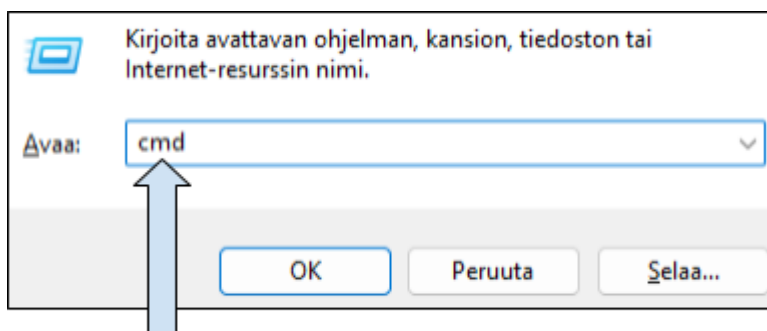
Python

Pythonia tarvitaan vasta kappaleessa “Tiedonkeruu”. Voit tehdä sen asennuksen myöhemmin. Tarkista ennen asennusta, onko Python jo asennettu tietokoneeseen. Voit halutessasi käyttää laitetta ilman Python-ohjelmaa myös muilla tavoin. Silloin sinun ei tarvitse asentaa Pythonia ollenkaan.

1. Paina **Windows**-näppäin pohjaan ja napauta näppäintä **r**.



2. Avautuvassa ikkunassa on tekstikenttä, kirjoita siihen **cmd** ja paina **Enter**.



3. Kirjoita avautuvaan komentokehotteeseen teksti “**python --version**” ja paina **Enter**. Jos näet kuvan mukaisen tulosteen (versionumero voi vaihdella), niin Python on jo asennettu tietokoneeseen. Muussa tapauksessa asenna se seuraavan sinun ohjeilla.

```
C:\Users>python --version
Python 3.13.1
```

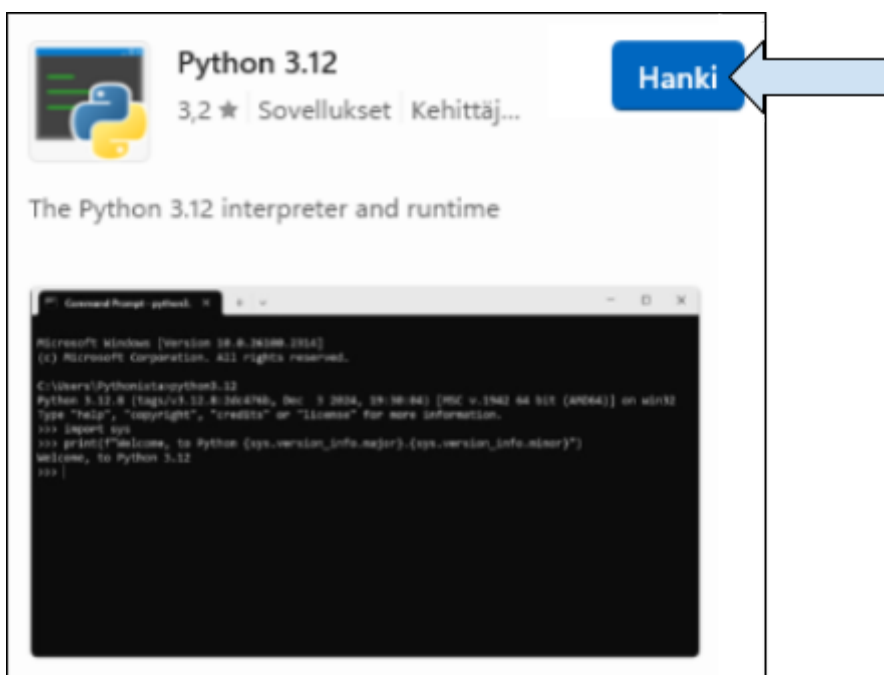
A screenshot of a terminal window. The prompt is 'C:\Users>'. The command entered is 'python --version'. The output is 'Python 3.13.1'. A blue arrow points to the output text.

Pythonin ja kirjastojen asennus

1. Avaa tietokoneesta Microsoft Store.
2. Kirjoita hakukenttään **python** ja paina **Enter**.



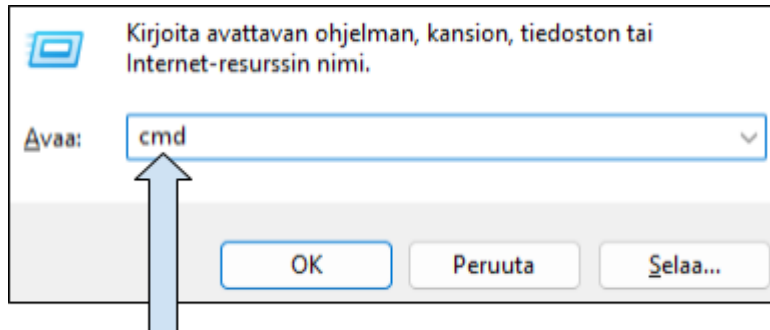
3. Etsi hakutuloksista uusin versio Pythonista ja klikkaa tuloksen oikeassa yläkulmassa olevaa painiketta "Hanki". Painike tulee näkyviin, kun siirrät hiiren hakutuloksen päälle.



4. Lataus ja asennus käynnistyy automaattisesti. Sen valmistuttua näet harmaan painikkeen "**Asennettu**". Python on nyt asennettu ja voit sulkea Microsoft Storen.

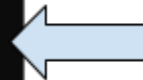


- Asennetaan seuraavaksi Pythoniin tarvittavat kirjastot. Paina uudelleen **Windows**-painike pohjaan ja napauta painiketta **r**.
- Kirjoita tekstikenttään **cmd** ja paina **Enter**.




- Kirjoita komentokehoteeseen "**pip install pyserial**" ja paina **Enter**. Tämä asentaa sarjaportin käyttöön tarvittavan kirjaston.

```
C:\Users>pip install pyserial
```




- Asennetaan vielä kirjasto keyboard, kirjoita "**pip install keyboard**" ja paina **Enter**.

```
C:\Users>pip install keyboard
```



- Kirjastot on nyt asennettu ja voit sulkea komentokehoitteen. Kirjoita **exit** ja paina **Enter**.

```
C:\Users>exit
```



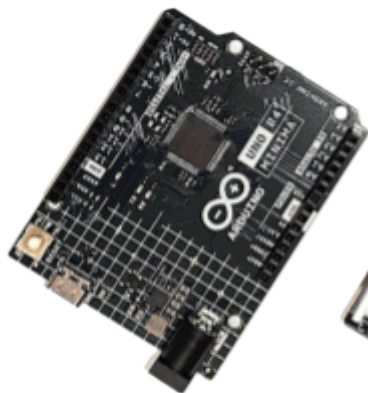
Tarvittavat komponentit

Alla olevassa taulukossa on listattu työhön tarvittavat komponentit selityksineen ja tilauslinkkeineen.

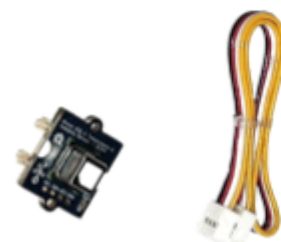
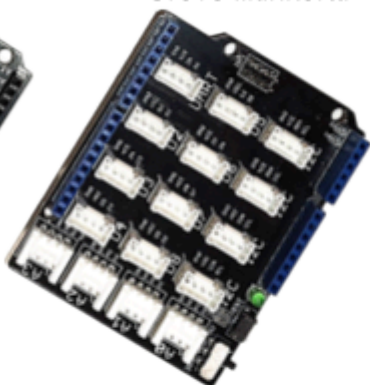
Laite / komponentti	Kuvaus ja tarkoitus	Linkit / huomautukset
Windows-tietokone, mielellään kannettava	Ohjelmointiin ja laitteen käyttöön	Käyttäjällä tulee olla oikeus asentaa tietokoneeseen ohjelmia ja muodostaa kansio C-aseman juureen
Arduino Uno mikro-ohjain esim. Uno R4 Minima	Ohjelmoidaan lukemaan anturia ja lähettämään data tietokoneelle	Partco Starelec Triopak
Liitinkortti Grove Arduino Base Shield	Unon päälle kiinnitettävä kortti, johon anturi kiinnitetään	Partco
Mittausanturi Grove CO2+temp+hum	Mittaa ympäristön CO2-pitoisuuden, lämpötilan ja kosteuden	Partco
Lukittumaton kytkentäkaapeli, tulee anturin mukana	Kaapelilla anturi kytketään liitinkorttiin	Varakaapeleita voi tilata linkistä Partco
USB-kaapeli	Arduino Unon ja tietokoneen väliin.	Tilaa käyttämäsi Unon mallin mukaan, katso kuva. Partco Radioduo

Kuvassa on esitetty työhön tarvittavat komponentit:

Arduino Uno R4 Minima



Grove liitinkortti



Grove CO2+Temp+Hum anturi ja kaapeli

USB-kaapeli Unon ja tietokoneen väliin

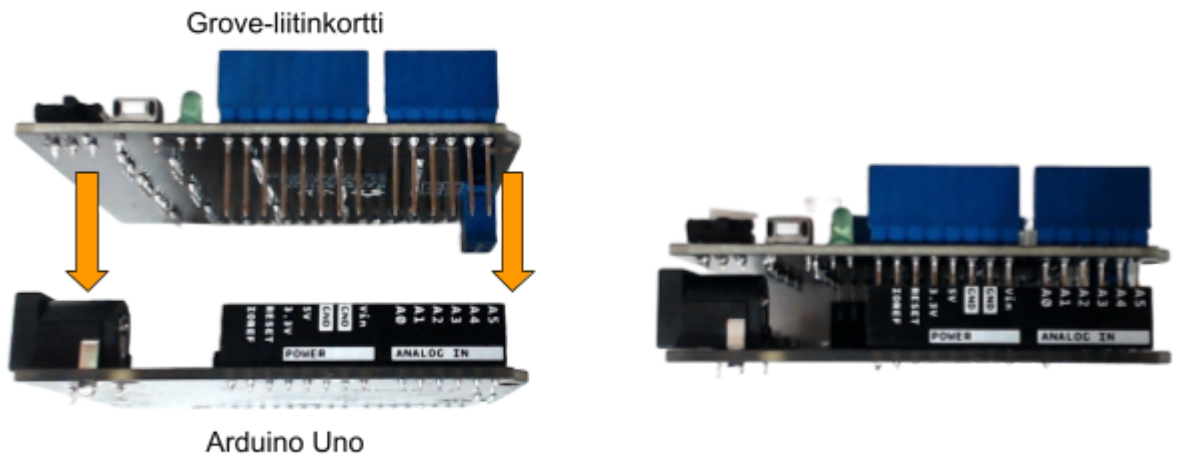


Tietokoneeseen tuleva liitin on aina USB-A uros. Unoon tuleva liitin on vanhemmissa malleissa USB-B uros, mutta uudemmassa R4-mallissa se on USB-C uros.



Kokoaminen

1. Kuvan mukaisesti Grove-liitinkortti tulee Arduino Unon päälle. Liitinkortin alapuolella on piikkejä, joiden tulee sopia Unon liittimiin. Voit joutua vähän sovitteluun. Jos piikit ovat vääntyneet, niin suorista niitä ohutkärkisillä pihdeillä. Katso että liitinkortti tulee oikein päin.



2. Liitinkortin kulmassa on pieni kytkin. Varmista, että se on kuvan mukaisesti käännetty asentoon 5V.



Kytkimen tulee olla oikealla asennossa 5V.

3. Kytke kaapeli anturiin. Kaapelin liittimessä olevat muoviset ulokkeet varmistavat, että kaapeli kytketään oikein päin. Katso, että ulokkeet menevät vastakappaleessa oleviin uriin.

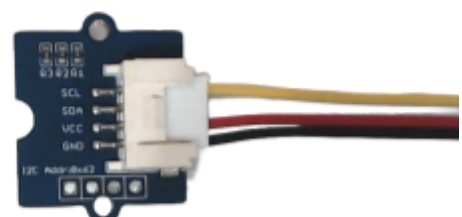
Ulokkeet liittimessä



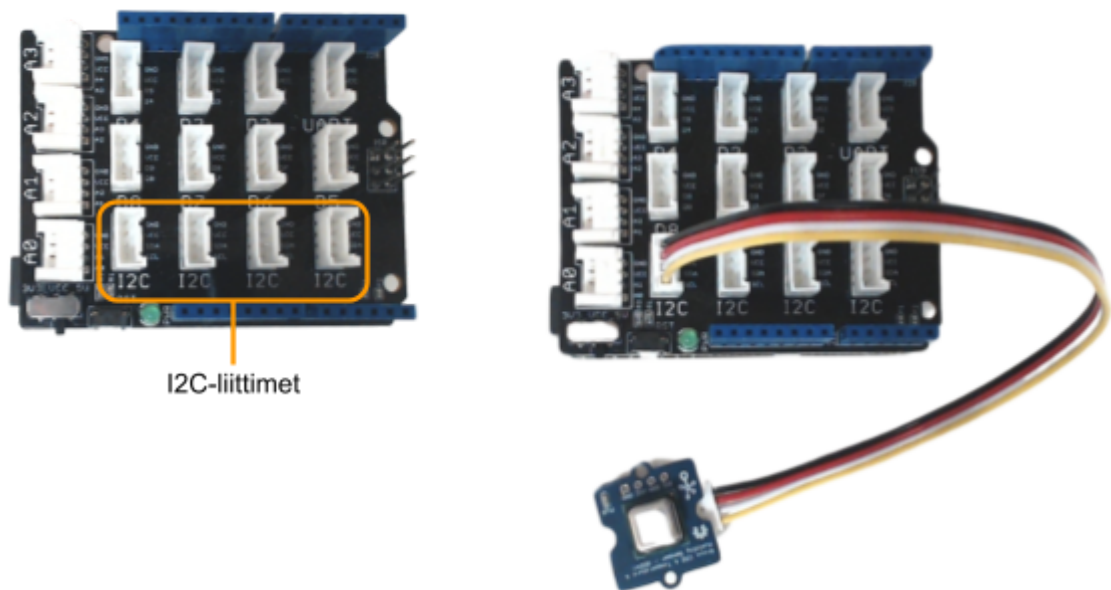
Urat vastakappaleessa



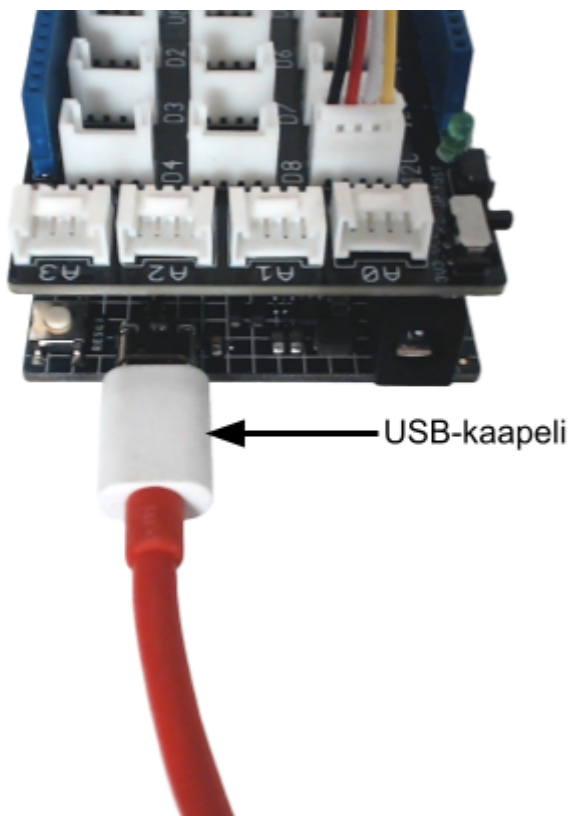
Kaapeli liitetty anturiin



4. Liitinkortissa on neljä merkinnällä **I2C** varustettua liittintä. Liitä anturista tuleva kaapeli johonkin niistä.

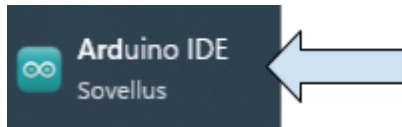


5. Liitä vielä lopuksi USB-kaapeli Unoön.

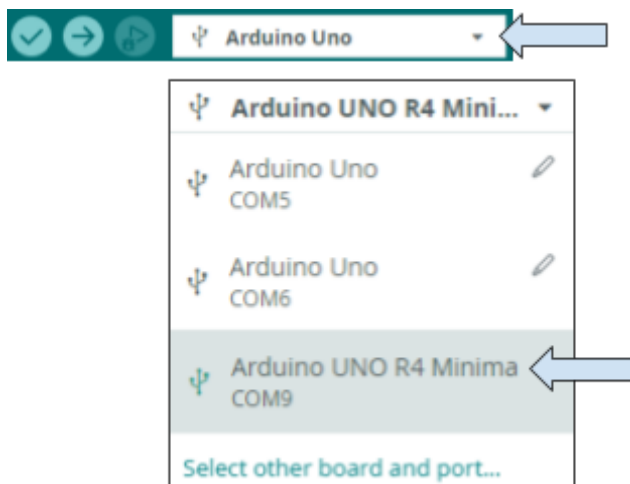


Unon ohjelmointi

1. Käynnistä tietokoneelta Arduino IDE -ohjelma. Tarvittaessa voit etsiä sen Windowsin hakutoiminnolla.



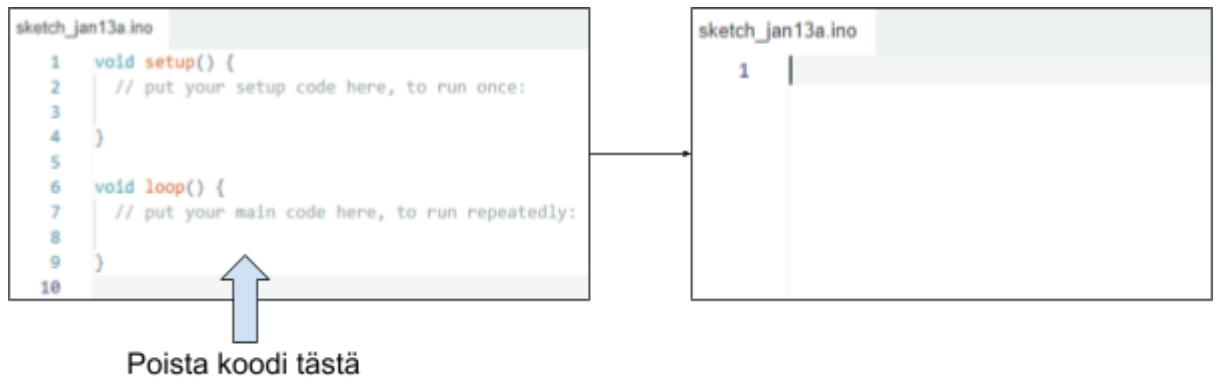
2. Liitä Unon USB-kaapeli tietokoneeseen. Jos ohjelma kysyy kirjastojen tai pakettien asennuksesta, niin asenna ne kaikki.
3. Klikkaa ohjelmointiympäristön yläpalkissa oleva valikko auki ja valitse käyttämäsi Unon malli ja sarjaliikenneportti (tässä COM9, sinulla voi olla jokin muu, yleensä väliltä COM1-COM10).



4. Anturin lukemiseen käytetään ulkoista kirjastoa. Asennetaan se seuraavaksi. Klikkaa sivupalkista kirjaston kuvaketta ja kirjoita hakukenttään kirjaston nimen alkuosa "**SparkFun_SCD4x**". Hakutulokset tulevat alapuolelle, klikkaa siitä löytyvää painiketta "**INSTALL**".



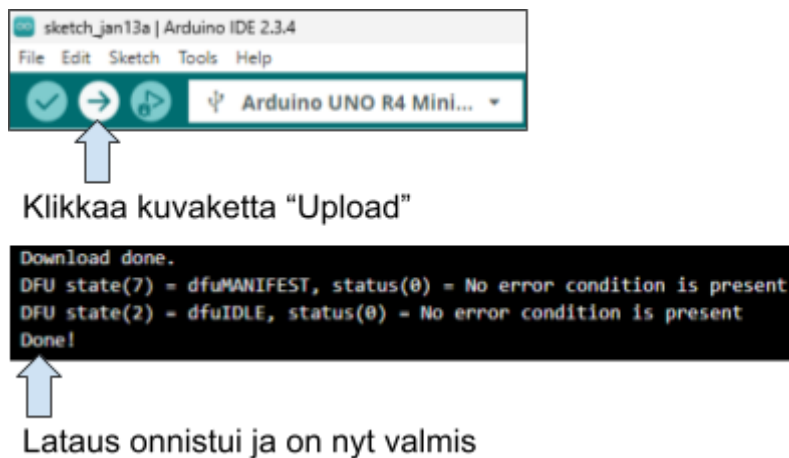
5. Poista ohjelmointiympäristöstä tyhjän ohjelman runko.



6. Maalaa ja kopioi seuraavalla sivulla oleva koodi. Liitä se sitten ohjelmointiympäristöön.



7. Klikkaa yläpalkista kuvaketta **“Upload”**. Koodi käännetään ja lähetetään Uno. Tämä kestää muutaman sekunnin. Lopuksi näet Output-ikkunassa ilmoituksen latauksen valmistumisesta.



8. Ohjelmointi on nyt valmis. Tallenna koodi haluamallasi nimellä ja siirry sitten kappaleeseen “Laitteen käyttö”

Koodi Arduino Unoon

```
#include "SparkFun_SCD4x_Arduino_Library.h"

SCD4x anturi;

void setup() {
  Serial.begin(9600);
  Wire.begin();
  // Alustetaan anturi. Jos ei onnistu, jäädään odottamaan
  if (!anturi.begin()) {
    Serial.println("Anturia ei löydy. Tarkista kytkennät.");
    while (1);
  }
  // Aloitetaan jatkuva mittaustila
  if (!anturi.startPeriodicMeasurement()) {
    Serial.println("Jatkuvan mittauksen käynnistäminen epäonnistui.");
    while (1);
  }
  delay(5000);
}

void loop() {
  // Jos mittaustiedot ovat luettavissa anturista
  if (anturi.readMeasurement()) {
    // Luetaan CO2, Temp sekä Hum anturista
    // ja tulostetaan ne sarjaporttiin
    Serial.print("CO2_ppm:");
    Serial.print(anturi.getCO2());
    Serial.print("\t");

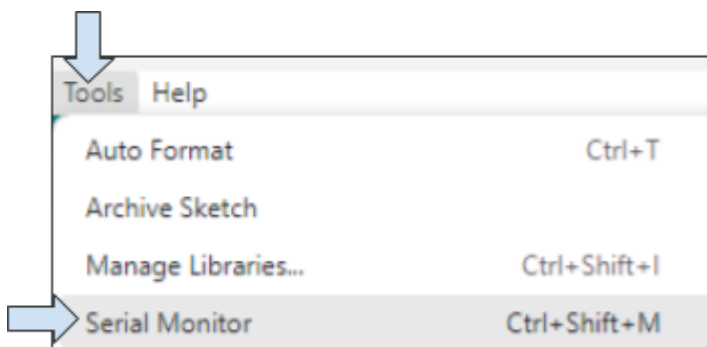
    Serial.print("Temp_C:");
    Serial.print((int)round(anturi.getTemperature()));
    Serial.print("\t");

    Serial.print("Hum_%:");
    Serial.println((int)round(anturi.getHumidity()));
  }
  delay(5000); // Mittausperiodi
}
```

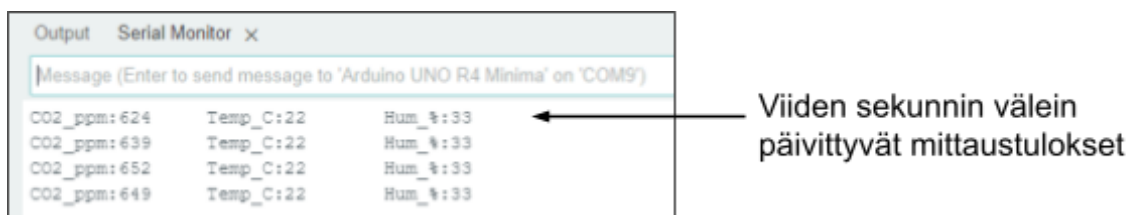
Laitteen käyttö

Laite mittaa ympäristöstä kolmea suuretta: hiilidioksidipitoisuutta [ppm], lämpötilaa [C] sekä ilman kosteusprosenttia. Mittaukset tehdään viiden sekunnin välein ja lähetetään tietokoneelle USB-kaapelin kautta. Arduino IDE-ohjelmassa on toiminto, joka mahdollistaa mittausdatan katselun.

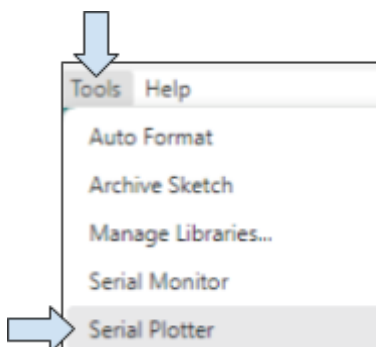
1. Klikkaa valikko **“Tools”** auki ja valitse sieltä toiminto **“Serial Monitor”**



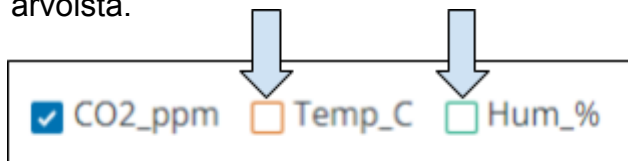
2. Serial Monitor -ikkuna aukeaa näytön alaosaan. Näet siitä laitteen lähettämät mittaustulokset.



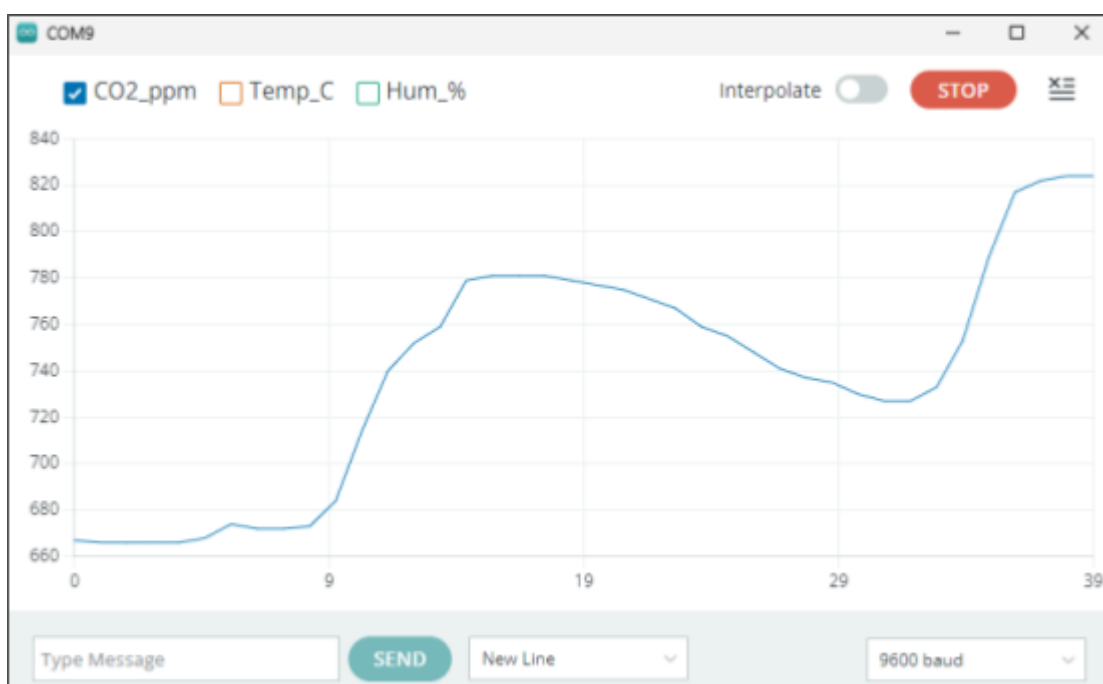
3. Mittaustuloksista saadaan myös graafinen kuvaaja. Klikkaa uudelleen valikko **“Tools”** auki ja valitse tällä kertaa **“Serial Plotter”**.



4. Serial Plotter -ikkunan avauduttua ota ruksi pois kohdista "Temp_C" ja "Hum_%". Näin plotteri piirtää kuvaajaa vain mitatuista CO2-pitoisuuden arvoista.



5. Kuvaajasta näet, miten CO2-pitoisuus muuttuu ajan funktiona. Anturi on niin tarkka, että jo yhdenkin ihmisen oleskelu sen läheisyydessä riittää nostamaan mitatun CO2-pitoisuuden arvoa (katso väli 9-19).



Voit käyttää "Serial Monitor":ia silloin, kun haluat tarkat mittaustulokset selville. "Serial Plotterilla" voit puolestaan havainnollistaa helposti CO2-pitoisuuden tai lämpötilan ja kosteuden muutoksia graafisilla kuvaajilla.

Pidempiaikaisissa mittauksissa voit jättää laitteen päälle ja käydä säännöllisin väliajoin ottamassa mittaustulokset talteen Serial Monitorista.

Laitteen käynnistyksen jälkeen anna sen olla pari minuuttia päällä, ennen kuin otat talteen mittaustuloksia. Näin tulokset ehtivät vakiintua ja tarkentua.

Hiilidioksidipitoisuuden tulkinta

Yksikkö ppm (parts per million) = pitoisuus miljoonasosina.

Sisäilma		
CO ₂ -pitoisuus ppm	Tulkinta	Vaikutus
400–600	Erinomainen	Ulkoilman taso. Hyvä ilmanvaihto, ei merkittäviä vaikutuksia.
600–1000	Hyvä	Normaali sisäilma. Sopiva ilmanvaihto useimmille tiloille.
1000–1500	Tyydyttävä	Vähäinen ilmanvaihdon puute. Saattaa aiheuttaa lievää väsymystä tai keskittymiskyvyn laskua.
1500–2000	Huono	Huono ilmanvaihto. Selkeästi havaittavaa väsymystä, päänsärkyä tai keskittymiskyvyn heikkenemistä.
Yli 2000	Erittäin huono	Erittäin huono ilmanvaihto. Mahdollisesti terveysriski pitkäaikaisessa altistuksessa.

Ulkoilma		
CO ₂ -pitoisuus ppm	Tulkinta	Vaikutus
350–450	Normaali ulkoilman taso	Luonnollinen taustapitoisuus, hyvin puhdas ilma.
450–500	Hieman kohonnut	Ilma on puhdasta, mutta ihmistoiminta, kuten liikenne ja teollisuus, vaikuttavat pitoisuuteen.
500–600	Kaupunkialueen normaali	Tyypillinen taso tiheästi asutuilla alueilla, ei merkittäviä vaikutuksia terveyteen.
600–1000	Selvästi kohonnut	Ilmansaasteiden ja ihmistoiminnan vaikutus korostuu; paikoin raskaasti liikennöidyillä alueilla.
Yli 1000	Poikkeuksellisen korkea	Esimerkiksi suljetuissa kaupunkiympäristöissä tai teollisuusalueilla. Voi lisätä ilmanlaadun heikkenemistä.

Tiedonkeruu

Unon ohjelmointi uudelleen

Edellisessä kappaleessa esitetty menetelmä sopii hyvin lyhyeen ja ajoittaiseen tiedonkeruuseen. Tässä kappaleessa koodaamme Pythonilla pitkäkestoiseen ja säännöllisiin väliajoin tehtäviin mittauksiin sopivan ohjelman. Tallennamme sillä mittaustulokset tiedostoon analysointia varten. Ihan aluksi joudumme kuitenkin muuttamaan hiukan Arduino Unossa olevaa ohjelmaa.

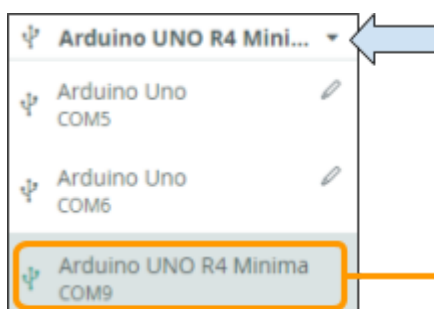
1. Käynnistä Arduino IDE ja avaa siihen aiemmin tekemäsi ohjelma (tai tarvittaessa kopioi ja liitä koodi uudelleen sivulta 13).
2. Etsi koodin lopusta kuvan mukainen rivi, jossa delay-komennolla ohjelma laitetaan odottamaan.

```
43      delay(5000); // Mittausperiodi
```

3. Sulussa oleva luku 5000 määrittää mittausperiodin. Eli millaisin aikavälein laite tekee mittaukset ja lähettää tiedot. Luku on millisekunteina, eli tässä mittausperiodi on viisi sekuntia. Pidempikestoisessa mittausjaksossa ei ole mielekäästä kerätä tuloksia näin lyhyin väliajoin. Voimme helposti kasvattaa mittausperiodin minuuttiin, eli kirjoittaa sulkuihin luvuksi 60000 (1 minuutti = 60 000 millisekuntia). Halutessasi voit käyttää vieläkin suurempaa odotusaikaa.

```
43      delay(60000); // Mittausperiodi
```

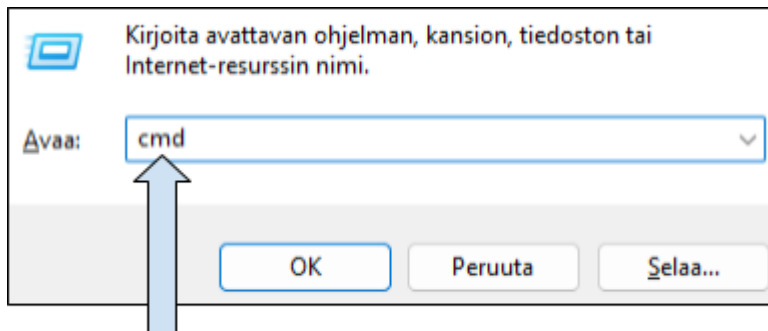
4. Tee muutos, liitä Uno USB-kaapelilla tietokoneeseen, valitse valikosta Unon käyttämä sarjaportti (COM X) ja lataa muutettu koodi Unoan. Katso tarvittaessa latausohjeet kappaleesta "Unon ohjelmointi".
5. Ota itsellesi talteen tieto Unon käyttämästä sarjaportista. Näet sen klikkaamalla yläpalkin valikon auki kuvan mukaisesti,



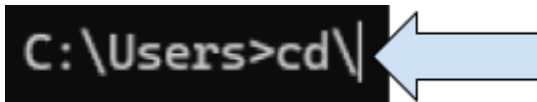
Tässä portti on COM9, mutta se voi olla myös jokin muu, yleensä väliltä COM1 - COM11.

Tiedonkeruuohjelma Pythonilla

1. Paina **Windows**-näppäin pohjaan ja napauta näppäintä **r**.
2. Kirjoita tekstikenttään **cmd** ja paina **Enter**.



3. Kirjoita komentokehoteeseen teksti **cd** ja paina **Enter**.

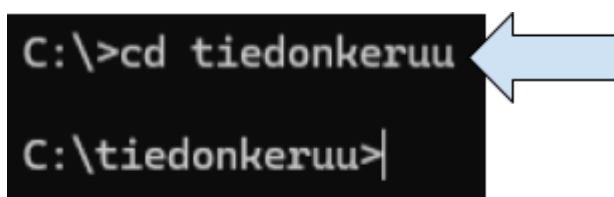


4. Siirryt C-aseman päähakemistoon. Tehdään seuraavaksi kansio Python-koodille. Kirjoita **md tiedonkeruu** ja paina **Enter**.



Huom! Jos tietokoneen käyttöoikeudet estävät kansion muodostuksen C-aseman juureen, voit käyttää asemana USB-tikkua. Tällaisen aseman päähakemistoon siirrytään kirjoittamalla <aseman tunnus>: ja painamalla Enter (esim. F: ja Enter). Korvaa ohjeissa oleva C: USB-tikkusi asematunnuksella.

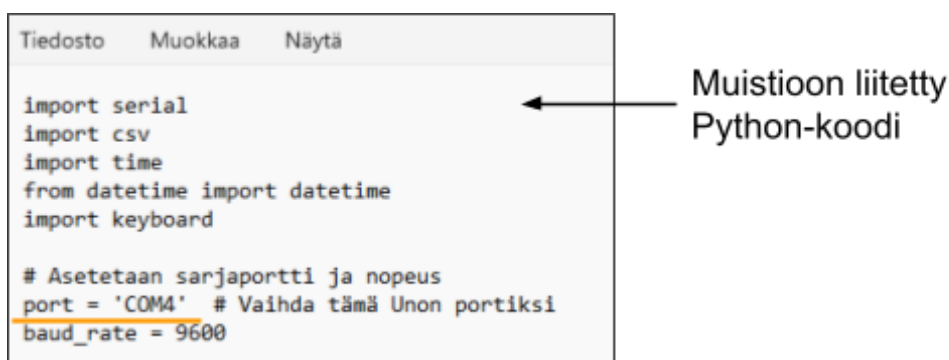
5. Siirry kansioon kirjoittamalla **cd tiedonkeruu** ja paina **Enter**.



6. Etsi tietokoneesta Muistio-sovellus ja käynnistä se. Voit esimerkiksi kirjoittaa Windowsin hakukenttään **muistio** ja käynnistää ohjelman hakutuloksesta.

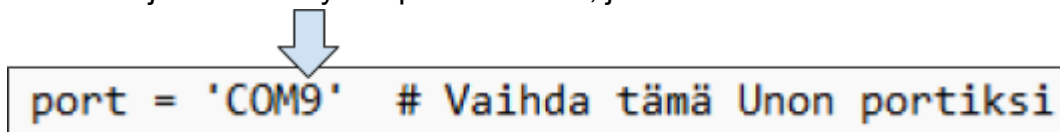


7. Maalaa ja kopioi sivuilla 22-23 oleva Python-koodi. Siirry takaisin Muistioon ja liitä koodi siihen.

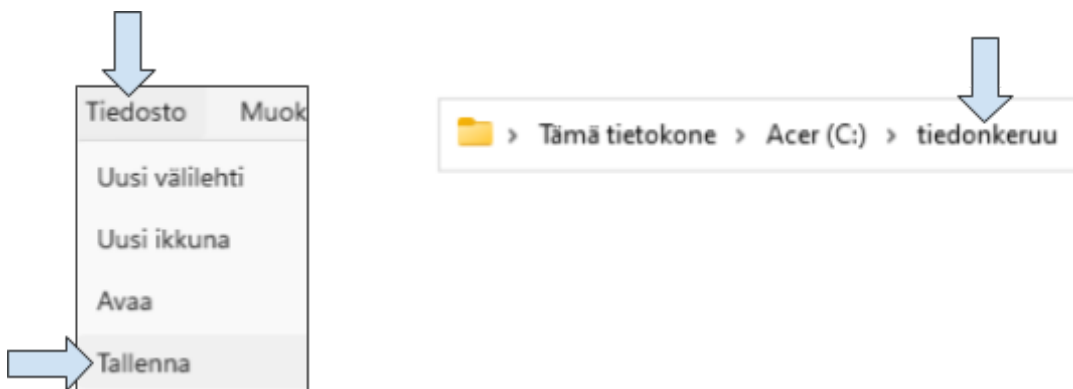


8. Muistio ei osaa tehdä koodiin syntaksikorostusta, joten se voi näyttää hieman sekavalta. Sinun tarvitsee tehdä siihen kuitenkin vain yksi muutos. Mene riville, joka alkaa näin “**port = 'COM4'**”
9. **Numeron 4** tilalle tulee vaihtaa **sinun Unon käyttämä porttinumero**. Otit sen aiemmin talteen sivun 17 kohdan 5 ohjeen mukaisesti.

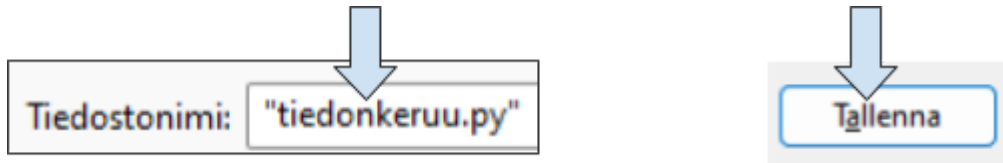
Tämän ohjeen Uno käyttää porttia **COM9**, joten koodi muutetaan näin:



10. Tallenna koodi seuraavaksi. Avaa **Tiedosto**-valikko ja valitse **Tallenna**. Valitse kohteeksi “**Tämä tietokone / C: / tiedonkeruu**”.



11. Kenttään **tiedostonimi** kirjoita lainausmerkkien sisään "**tiedonkeruu.py**" ja klikkaa painiketta **Tallenna**.



12. Varmista asetuksista, että tietokone ei mene lepotilaan tai sammuta itseään tiedonkeruun aikana. Avaa asetukset näppäinyhdistelmällä **Windows näppäin** ja **i-kirjain**. Valitse valikosta "**Järjestelmä**" ja "**Virta ja akku**" tai "**Virta ja lepotila**" ja tee tarvittaessa muutokset asetuksiin.

13. Liitä Uno USB-kaapelilla tietokoneeseen.

14. Siirry komentokehoteeseen, jät siinä muodostamaasi kansioon tiedonkeruu. Kirjoita komento **dir** ja paina **Enter**. Näet kansiossa tallentamasi kooditiedoston tiedonkeruu.py.

```
C:\tiedonkeruu>dir
Volume in drive C is Acer
Volume Serial Number is 687A-19E2

Directory of C:\tiedonkeruu

14.01.2025  20.04    <DIR>          .
14.01.2025  20.00             1 996 tiedonkeruu.py
               1 File(s)                1 996 bytes
               1 Dir(s)  94 312 615 936 bytes free

C:\tiedonkeruu>
```

15. Kaikki siis kunnossa, nyt voimme ajaa ohjelman. Kirjoita **python tiedonkeruu.py** ja paina **Enter**. Ohjelma käynnistyy, vastaanottaa, näyttää ja tallentaa mittausdatan.

```
C:\tiedonkeruu>python tiedonkeruu.py

C:\tiedonkeruu>python tiedonkeruu.py
Ohjelma käynnistetty. Paina välilyönti lopettaaksesi.
Viimeisin mittaus: 20:11, CO2: 683 ppm, Lämpötila: 23 °C, Kosteus: 30 %
Viimeisin mittaus: 20:12, CO2: 682 ppm, Lämpötila: 23 °C, Kosteus: 32 %
```

16. Ohjelma muodostaa tiedoston **data.csv** ja tallentaa mittaustulokset siihen. Jätä laitteisto keräämään haluamasi määrä dataa. Kun sitä on riittävästi, voit lopettaa ohjelman toiminnan painamalla **välilyöntiä**. Varmuuden vuoksi, älä tee tiedonkeruun aikana tietokoneella mitään muuta, koska välilyönnin painaminen missä tahansa sovelluksessa lopettaa ohjelman toiminnan. Sulje lopuksi komentokehoite kirjoittamalla komento **exit** ja paina **Enter**.



Python-koodi

```
import serial
import csv
import time
from datetime import datetime
import keyboard

# Asetetaan sarjaportti ja nopeus
port = 'COM4' # Vaihda tämä Unon portiksi
baud_rate = 9600

try:
    # Avataan sarjaportti
    ser = serial.Serial(port, baud_rate, timeout=0.1)

    # Luodaan CSV-tiedosto
    with open('data.csv', mode='w', newline='') as file:
        writer = csv.writer(file)
        writer.writerow(["Aika", "CO2 ppm", "Temp C", "Hum %"])

    print("Ohjelma käynnistetty. Paina välilyönti
lopettaaksesi.")

    while True:
        # Tarkistetaan, onko välilyönti painettu
        if keyboard.is_pressed('space'):
            print("Välilyönti painettu, lopetetaan ohjelma.")
            break

        # Luetaan data sarjaportista
        line = ser.readline().decode('utf-8').strip()
        if line:
            try:
                # Jaetaan mittaustulokset muuttujiin
                values = line.split('\t')
                co2 = values[0].split(':')[1]
                temp = values[1].split(':')[1]
                hum = values[2].split(':')[1]

                # Lisätään aikaleima muodossa HH:MM
                timestamp = datetime.now().strftime('%H:%M')
```

```
# Kirjoitetaan rivi CSV-tiedostoon
writer.writerow([timestamp, co2, temp, hum])

# Näytetään ruudulla mittauksen tiedot
print(f"Viimeisin mittaus: {timestamp}, CO2:
{co2} ppm, Lämpötila: {temp} °C, Kosteus: {hum} %")
except (IndexError, ValueError):
    print("Virhe datan käsittelyssä, ohitetaan
rivi.")

except serial.SerialException as e:
    print(f"Virhe sarjaporttiin yhdistämisessä: {e}")
finally:
    # Sulje sarjaportti turvallisesti
    if 'ser' in locals() and ser.is_open:
        ser.close()
    print("Yhteys sarjaporttiin suljettu.")
```


Mittausdatan analysointi

Kuten ja aiemmin todettiin, tiedonkeruu.py-ohjelma muodostaa tiedoston **data.csv** ja tallentaa siihen kaikki mittaustulokset. Tiedoston ensimmäinen rivi on otsikkorivi, se kertoo mitä arvoja alapuolella on. Sen jälkeen jokaisesta mittauksesta tulee uusi rivi tiedostoon. Kuvan mukaisesti rivissä on mittausaika, CO2-pitoisuus [ppm], lämpötila [C] ja ilman kosteusprosentti [%]. Tulokset on erotettu pilkulla toisistaan.

Timestamp	CO2_ppm	Temp_C	Hum_%
20:11	683	23	30
20:12	682	23	32
20:13	657	22	33

← Otsikkorivi

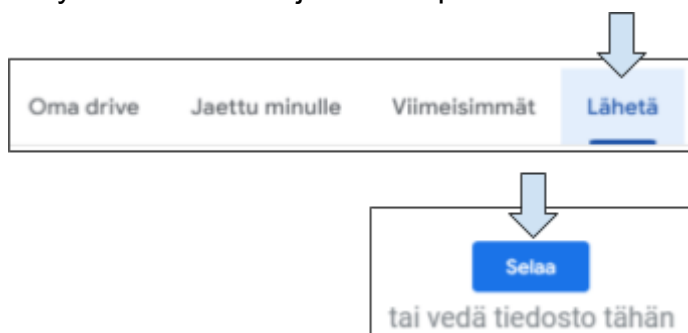
← Mittaustuloksia

data.csv-tiedosto on tekstimuotoinen, joten sen voi avata vaikkapa Muistio-ohjelmaan katselemista varten. Tarkempi analysointi on kuitenkin parempi tehdä taulukkolaskentaohjelmassa. Taulukkolaskennassa mittaustulokset ovat kaikki omissa soluissaan ja voit käsitellä niitä matemaattisilla kaavoilla ja piirtää niistä kuvaajia. Katsotaan, miten tämä tehdään.

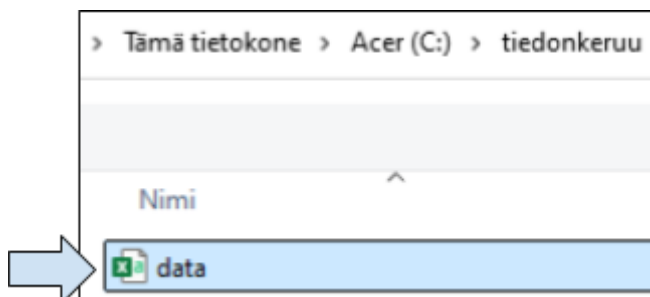
1. Kerää haluamasi määrä **mittaustuloksia data.csv**-tiedostoon. Lopeta tämän jälkeen tiedonkeruu.py-ohjelman toiminta painamalla välilyöntiä.
2. Avaa **Goole Sheets** -taulukkolaskenta.
3. Avaa **Tiedosto**-valikko ja valitse **Tuo**.



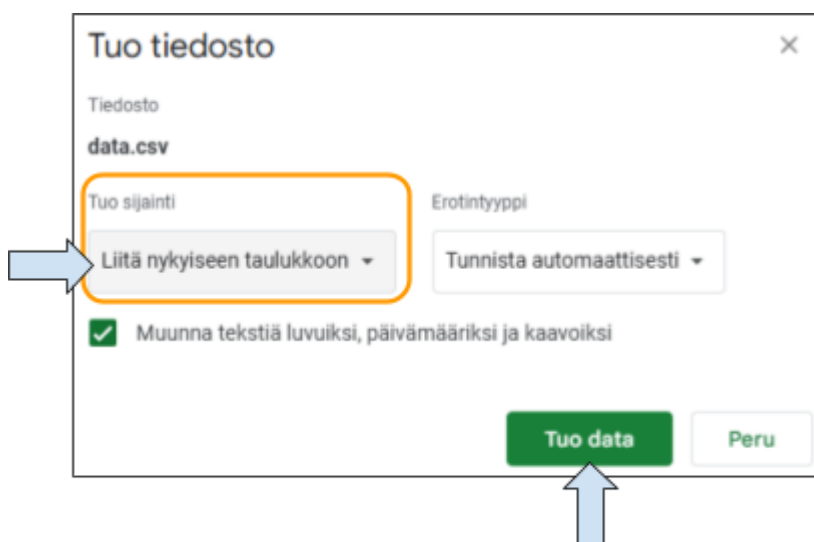
4. Siirry osioon **Lähetä** ja klikkaa painiketta **Selaa**.



5. Valitse tietokoneeltasi kansioista “C:\tiedonkeruu\” Python-ohjelman muodostama tiedosto **data.csv**.



6. Avaa valikko kohdasta “Tuo sijainti” ja valitse valinta “Liitä nykyiseen taulukkoon”. Klikkaa sitten painiketta “Tuo data”.



7. Mittaustulokset on nyt tuotu taulukkolaskentaan. Voit käsitellä tuloksia matemaattisilla kaavoilla ja piirtää niistä kuvaajia.

	A	B	C	D
1	Timestamp	CO2_ppm	Temp_C	Hum_%
2	14:14	740	23	28
3	14:15	749	23	28
4	14:16	756	23	28
5	14:17	765	23	28
6	14:18	768	23	28
7	14:19	773	23	28
8	14:20	784	23	28
9	14:21	787	23	28
10	14:22	790	23	28
11	14:23	786	23	28



Taulukkolaskentaohjelmassa kerättyjä CO₂-, lämpötila- ja kosteusmittaustuloksia voidaan analysoida monipuolisesti.

1. Keskiarvot ja hajonta:

- Laske mittausjakson keskiarvot ja hajonnat jokaiselle suurelle. Tämä auttaa ymmärtämään vaihteluväliä ja yleistä tasoa.

2. Kuvaajien laatiminen:

- Piirrä kuvaajia, jotka esittävät mittaustulosten muutoksia ajan suhteen. Näin voidaan havaita trendejä, kuten CO₂-pitoisuuden nousu luokkahuoneessa päivän aikana.

3. Korrelointi:

- Analysoi, onko CO₂-pitoisuuden, lämpötilan ja kosteuden välillä yhteyksiä. Voisiko esimerkiksi lämpötilan nousu vaikuttaa CO₂-pitoisuuden mittaamiseen?

4. Huippuarvot ja minimiarvot:

- Etsi korkeimmat ja matalimmat mittausarvot ja pohdi, mistä ne voisivat johtua. Yritä selvittää esimerkiksi, miksi CO₂-pitoisuus on vaihdellut. Mikä on aiheuttanut pitoisuuden laskua ja nousua tiettyinä aikoina?

5. Vertailut:

- Vertaa eri päivien tai paikkojen mittaustuloksia. Eroavatko arvot luokkahuoneessa ja ulkona? Entä aamun ja iltapäivän välillä?

6. Trendien tunnistaminen:

- Tarkista pitkän aikavälin mittaustuloksia trendejä varten. Nouseeko CO₂-pitoisuus tiettyyn aikaan päivällä, kun huoneessa on enemmän ihmisiä?

7. Suositusarvojen vertailu:

- Vertaa mittaustuloksia oppaassa esitettyihin CO₂-pitoisuuden suositusarvoihin. Tämä auttaa arvioimaan ympäristön tilaa.

8. Hypoteesien testaaminen:

- Oppilaat voivat tehdä hypoteeseja, kuten: "Korkea lämpötila korreloi korkeampaan CO₂-pitoisuuteen" ja käyttää dataa hypoteesien testaamiseen.

9. Pitkän aikavälin analyysi:

- Jos dataa kerätään useiden päivien tai viikkojen ajalta, analysoi, näkyykö vuodenaikojen tai sään vaikutuksia.

10. Raportointi:

- Oppilaat voivat käyttää analyysiään luodakseen raportteja ja esityksiä, joissa he havainnollistavat tuloksiaan graafien ja tilastojen avulla. Raporteista voidaan tehdä graafisesti näyttäviä infograafeja Adobe Expressillä. Tulokset voidaan esittää myös animaationa tai videona.

Itsenäinen laite

Laitteen helpompi siirrettävyys voi toisinaan olla tarpeellista. Ohjelmoinnin jälkeen Uno ei tarvitse toimintaansa tietokonetta, kunhan virta syötetään jollain muulla tavalla. Voimme käyttää esimerkiksi 9 Voltin paristoa. Lisätään laitteeseen LCD-näyttö, josta mittaustulokset ovat luettavissa. Työhön tarvitaan aiemmin rakennetun laitteen lisäksi:

Komponentti	Kuvaus ja tarkoitus	Linkit
Grove 2x16 LCD-näyttö	Kaksi riviä, 16 merkkiä / rivi. Mittaustulosten esitys.	Partco
Lukittumaton Grove liitinkaapeli	Kytkee LCD-näytön Grove-liitinkorttiin	Tulee näytön mukana.
Virtajohto Unolle 9 Voltin paristoliittimellä	Itsenäisessä käytössä virta otetaan paristosta	Partco
9 Voltin paristo		

Laitteen uudet komponentit:



Grove LCD-näyttö tulosten esittämiseen



Kaapeli näytön liittämistä varten



Virtajohto 9 Voltin paristoliittimellä



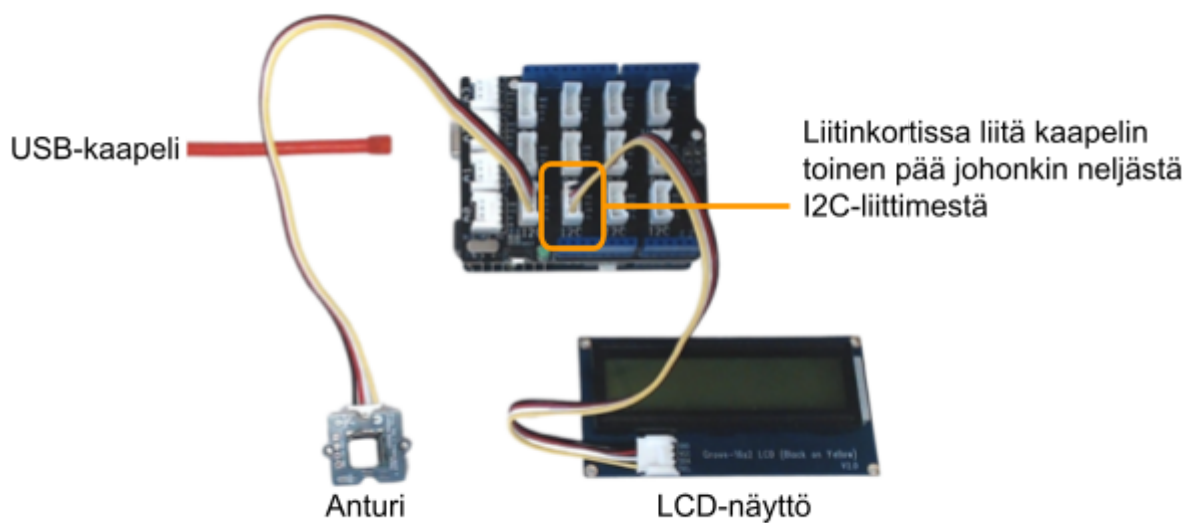
9 Voltin paristo

Kokoaminen

1. Liitä kytkentäkaapeli LCD-näyttöön. Liittimet menevät vain oikein päin kiinni. Katso aiemmasta ohjeesta sivulta 9.



2. Liitä kaapelin toinen pää Grove liitinkortin **I2C-liittimeen**.

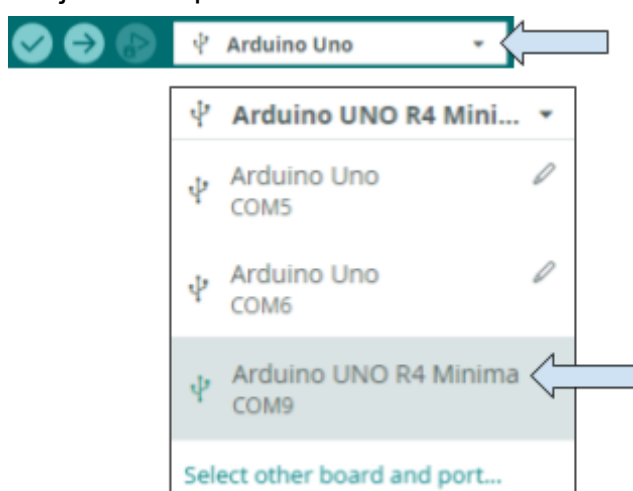


3. Kokoamisen osalta laite on valmis. Seuraavaksi Uno tulee ohjelmoida uudelleen.

Ohjelmointi

LCD-näyttöä varten Uno on tulella ladata uusi ohjelma, joka esittää mittaustulokset näytöllä. Tämäkin koodi annetaan valmiina. Jo aiemmin käsitellyt ohjelmointiin liittyvät vaiheet esitetään tässä varsin lyhyesti. Katso tarvittaessa tarkemmat ohjeet sivuilta 11-12.

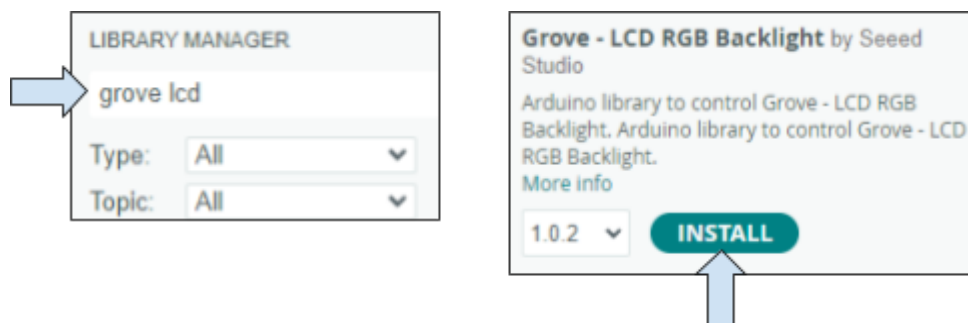
1. Käynnistä tietokoneelta Arduino IDE -ohjelma.
2. Liitä Uno USB-kaapelilla tietokoneeseen.
3. Arduino IDE -ohjelmassa, valitse yläpalkin valikosta käyttämäsi Unon malli ja sarjaliikenneportti.



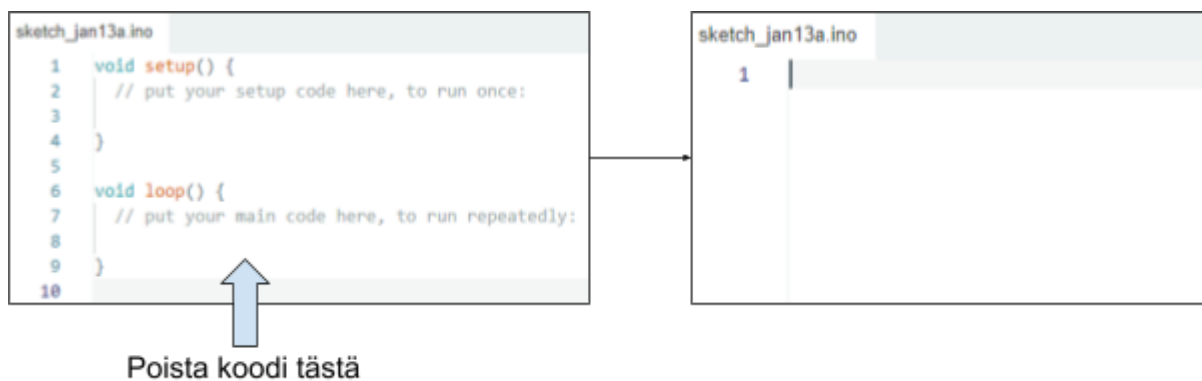
4. Jos et ole aikaisemmin asentanut anturin lukemiseen käytettävää ulkoista kirjastoa, niin asenna se seuraavaksi. Jos kirjasto on jo asennettu, saat ilmoituksen yrittäessäsi asentaa sen uudelleen. Klikkaa sivupalkista kirjaston kuvaketta ja kirjoita hakukenttään kirjaston nimen alkuosa **“SparkFun_SCD4x”**. Hakutulos tulee alapuolelle, klikkaa siitä löytyvää painiketta **“INSTALL”**.



5. Asennetaan vielä LCD-näytön käyttöön tarvittava kirjasto. Kirjoita hakukenttään **“grove lcd”** ja asenna hakutuloksista kirjasto **“Grove - LCD RGB Backlight”**.



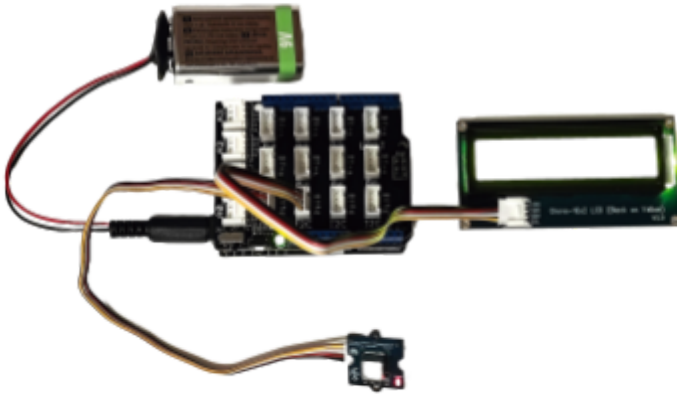
6. Poista ohjelmointiympäristöstä tyhjän ohjelman runko.



7. Maalaa, kopioi ja liitä sivulta 32 alkava koodi Arduino IDEen.
8. Käännä ja lataa ohjelma Unoon.
9. Latauksen jälkeen laite kykenee toimimaan itsenäisesti. Irrota USB-kaapeli tietokoneesta, kytke paristo virtajohtoon ja liitä se Unoon.
10. Laitteen liikuttamisen helpottamiseksi voit kiinnittää komponentit teipillä tai kaksipuolisilla tarroilla sopivan kokoisen pahvin tai vaneripalan päälle.

Alla olevissa kuvissa on esitetty mittalaite komponentteineen sekä LCD-näytöllä näkyvät mittaustulokset.

Itsenäisesti toimiva mittalaite



Ilman CO2-pitoisuus



Lämpötila

Kosteusprosentti



Arduino Uno koodi:

```
#include "Wire.h"
#include "SparkFun_SCD4x_Arduino_Library.h"
#include "rgb_lcd.h"

SCD4x anturi;
rgb_lcd lcd;

void setup() {
  Wire.begin();
  // Alustetaan anturi. Jos ei onnistu, jäädään odottamaan
  if (!anturi.begin()) while (1);

  // Aloitetaan jatkuva mittaustila
  if (!anturi.startPeriodicMeasurement()) while (1);

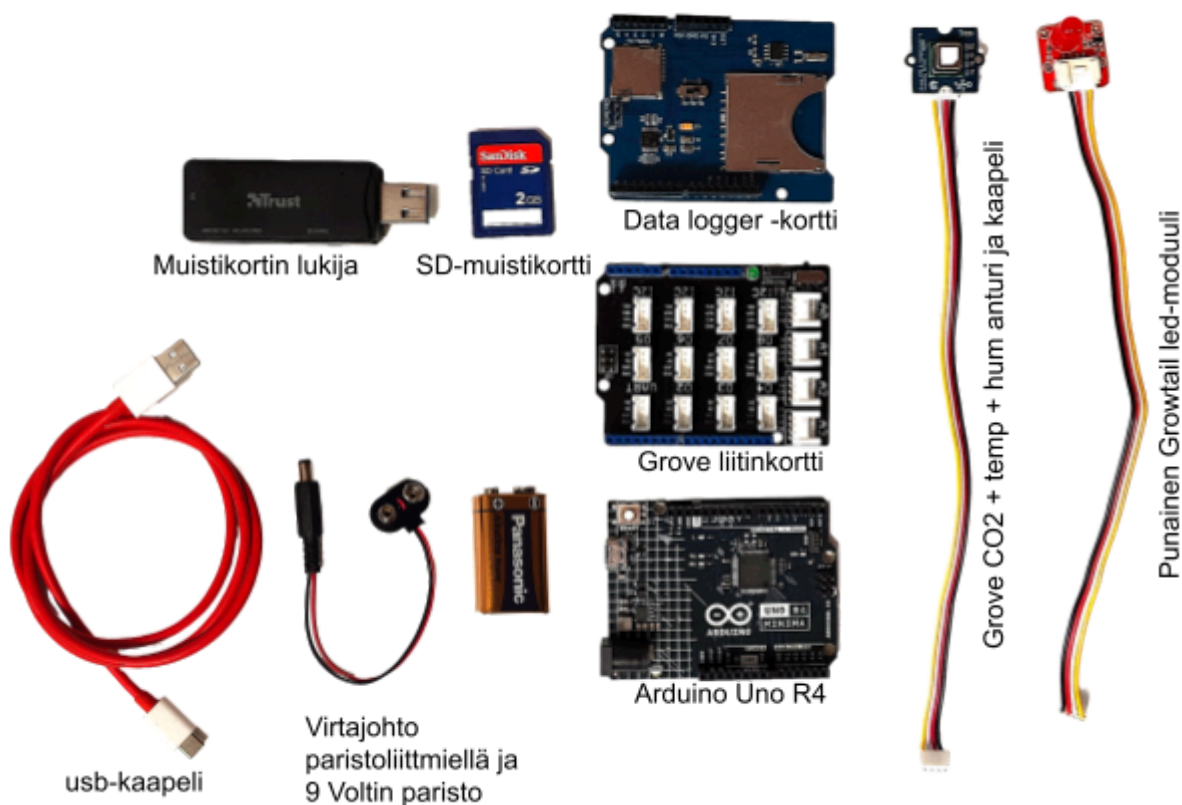
  // Alustetaan ja tyhjäetään LCD-näyttö
  lcd.begin(16, 2);
  lcd.clear();
  delay(5000);
}

void loop() {
  // Jos mittaustiedot ovat luettavissa anturista
  if (anturi.readMeasurement()) {
    // Luetaan CO2, Temp sekä Hum ja tallennetaan muuttujiin
    uint16_t co2 = anturi.getCO2();
    int itemp = (int)round(anturi.getTemperature());
    int ihum = (int)round(anturi.getHumidity());

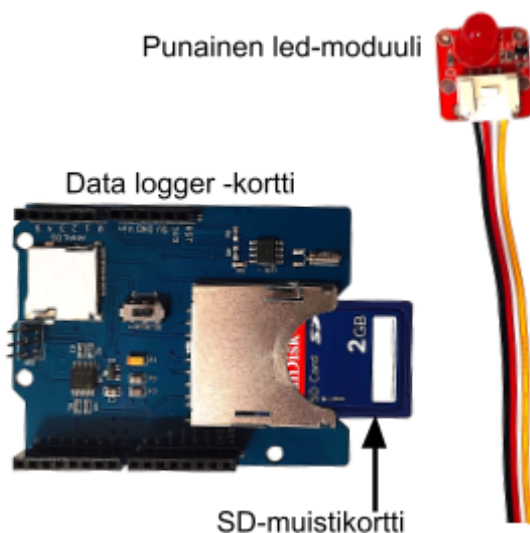
    // Tulostetaan arvot LCD-näytölle
    lcd.clear();
    lcd.print("CO2:");
    lcd.print(co2);
    lcd.print(" ppm");
    lcd.setCursor(0,1);
    lcd.print("T:");
    lcd.print(itemp);
    lcd.print(" C | H:");
    lcd.print(ihum);
    lcd.print(" %");
  }
  delay(5000); // Mittausperiodi
}
```

Itsenäinen tiedonkeruulaite

Nimensä mukaisesti itsenäinen tiedonkeruulaite tekee mittaukset ja tallentaa ne täysin itsenäisesti. Mittaukset tallentuvat muistikortille tiedostoon, joka mittausjakson loppuun siirretään taulukkolaskentaohjelmaan analysoitavaksi. Alla on esitetty kaikki laitteessa käytetyt komponentit. Laite sopii erityisen hyvin esimerkiksi ulkona tehtäviin pitkäkestoisin mittauksiin (älä jätä suojaamattomana sateeseen).



Uusia, aiemmin käyttämättömiä komponentteja laitteeseen ovat data logger-kortti, SD-muistikortti ja punainen led-moduuli.



Komponentti	Kuvaus ja tarkoitus	Linkit
Data logger -kortti	Mittaustulosten tallennus SD-muistikortille. Sisältää lisäksi kellon (RTC).	Partco
Growtail led-moduuli, punainen	Toimii laitteen merkkivalona	Partco
SD-muistikortti, kapasiteetti 2-4 Gt Voi olla täysikokoinen tai microSD.	Mittaustietojen tallennukseen	Muistikauppa
SD-muistikortin lukija (jos tietokoneessa ei ole valmiina korttipaikkaa)		Yleisesti myynnissä

SD-kortti ja kortin alustus

Muistikorttina kannattaa käyttää 2-4 gigatavun tallennuskapasiteetin omaavaa mallia. Tällaiset kortit on tarkoitettu toimimaan vanhemmissa laitteissa ja toimivat myös tässä tiedonkeruulaitteessa varmin.

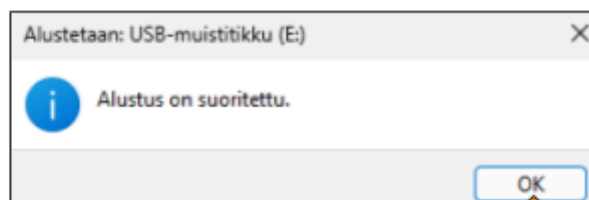
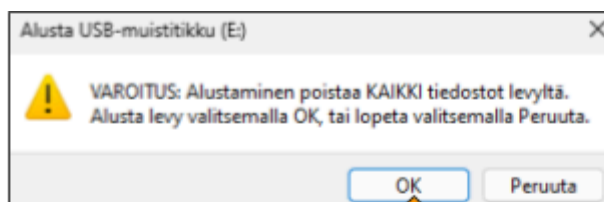
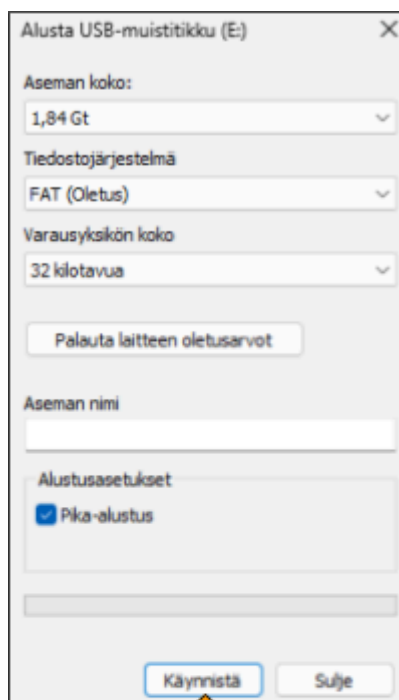
Muistikortti tulee alustaa aina ennen kuin se liitetään korttipaikkaan. Myös jokaisen uuden mittausjakson alussa.

1. Liitä kortti muistikortinlukijalla tietokoneeseen.
2. Näet sen yhtenä tietokoneen resurssina. Klikkaa aseman päällä hiiren oikeaa painiketta ja valitse valikosta valinta **“Alusta”**.



Alusta muistikortti jokaisen mittausjakson alussa.

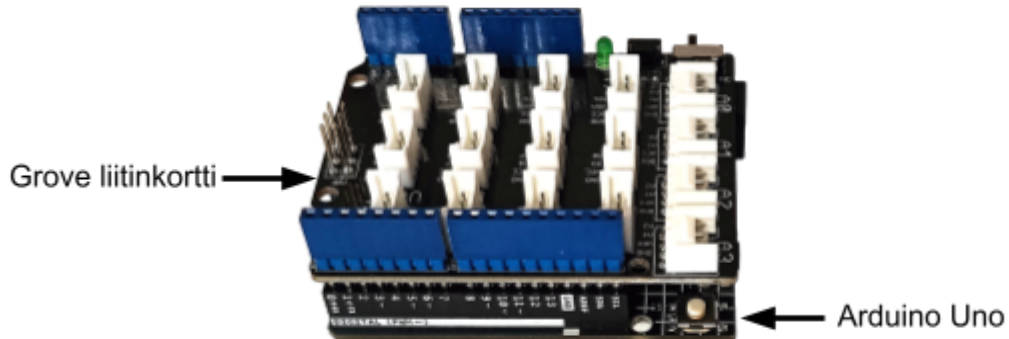
3. Varmista, että olet valinnut oikean aseman alustettavaksi. Klikkaa sitten painiketta **“Käynnistä”** ja kahteen kertaan painiketta **“OK”**. Muistikortti on nyt alustettu, poista se turvallisesti.



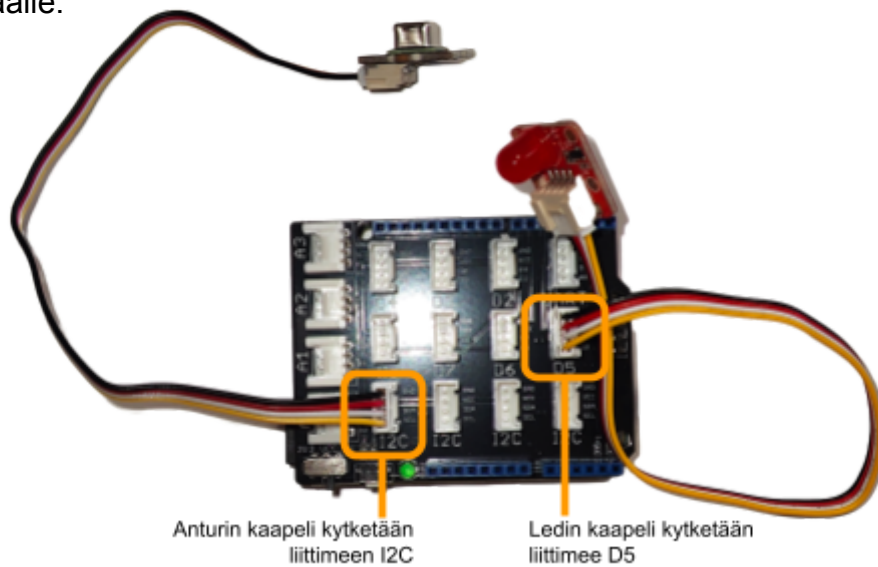
4. Älä laita muistikorttia vielä data loggeriin, vaikka kuvissa se esiintyykin havainnollistamiseksi laitteessa. Muistikortti asetetaan paikoilleen vasta käyttöönoton yhteydessä.

Kokoaminen

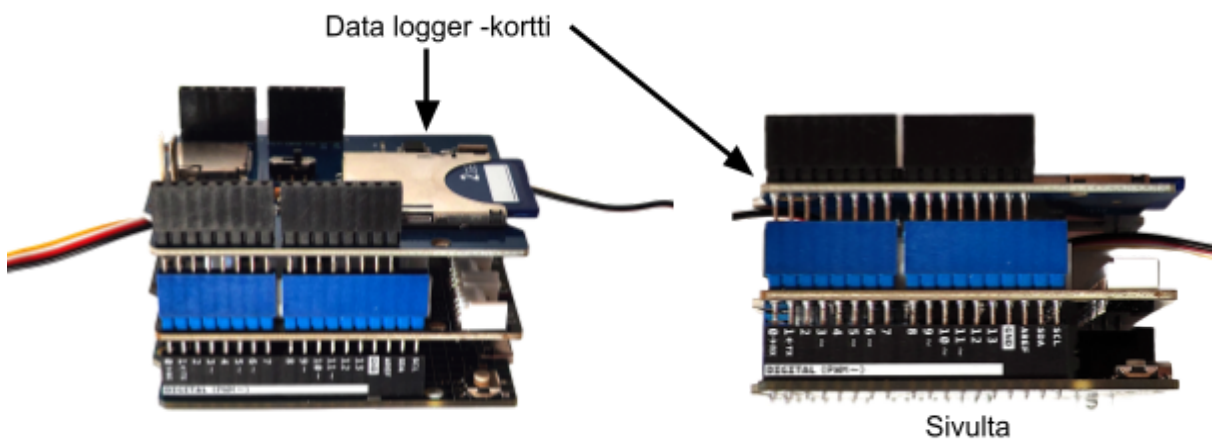
1. Paina ensimmäiseksi Grove liitinkortti Unon päälle kiinni.



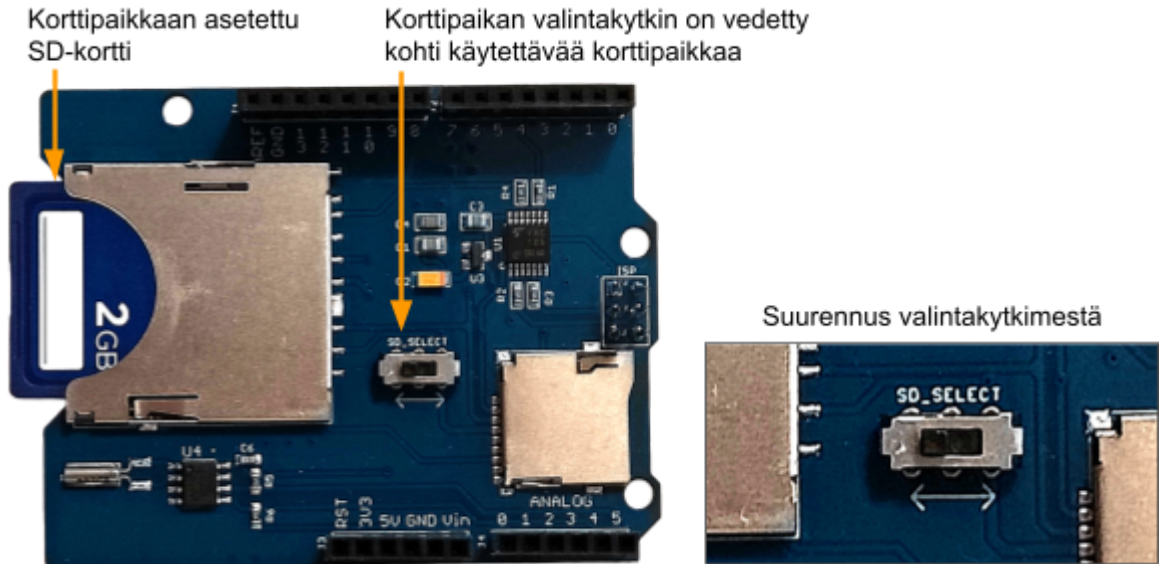
2. Kytke sitten anturin kaapeli johonkin **I2C-liittimeen** ja punaisen ledin kaapeli liittimeen **D5**. Taivuta kaapeleita sivulle päin, jotta data logger -kortti mahtuu päälle.



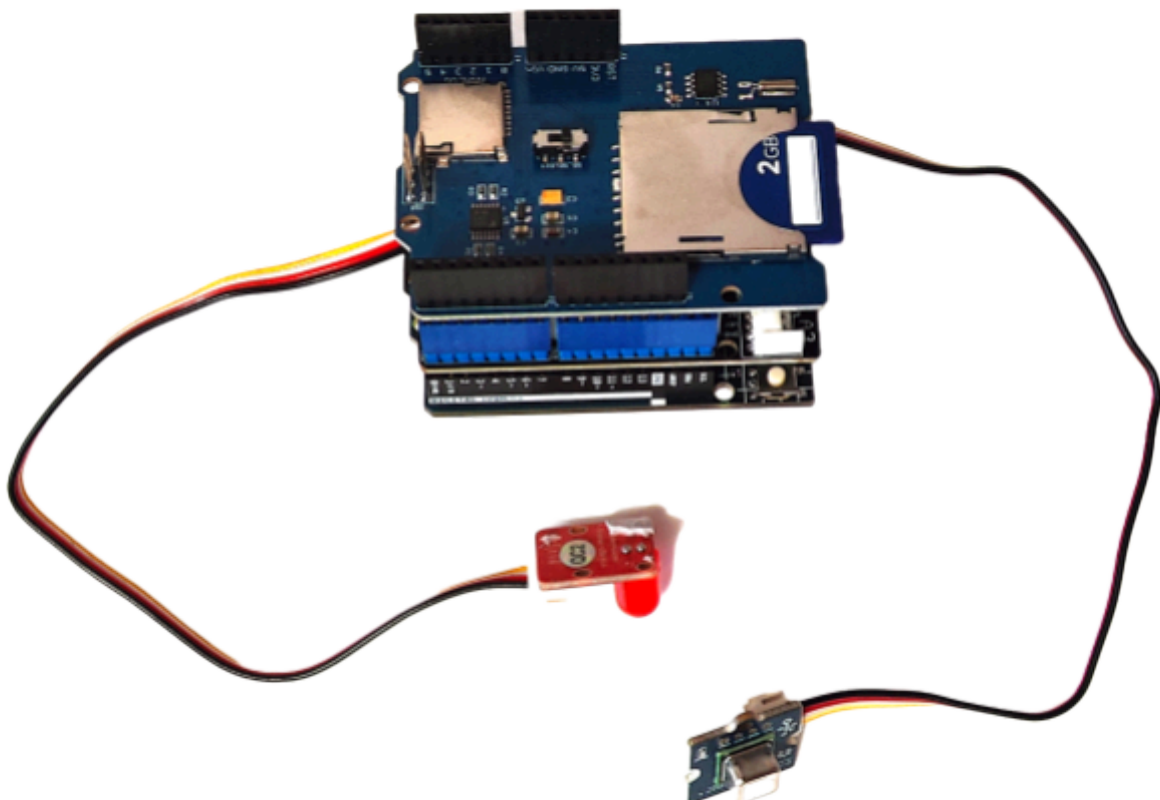
3. Paina Data logger -kortti kiinni liittinkortin päälle kuvan mukaisesti. Taivuta ledin kaapeli kiertämään pinnit.



4. Data logger -kortissa on paikat vakiokokoiselle SD-kortille sekä microSD-kortille. Aseta käyttämäsi kortti sen koon mukaiseen paikkaan. Kortin päällä on kytkin (**SD_SELECT**), sillä valitaan käytettävä korttipaikka. Vedä kytkin käytössä olevaa korttipaikkaa kohti.

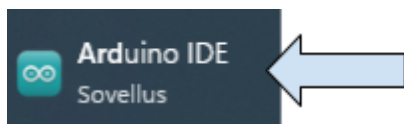


5. Itsenäisesti toimiva tiedonkeruulaite on nyt koottu. Seuraavaksi se tulee ohjelmoida.

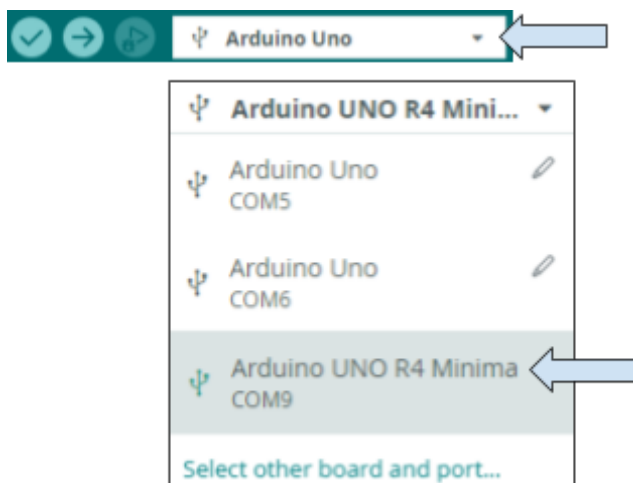


Ohjelmointi

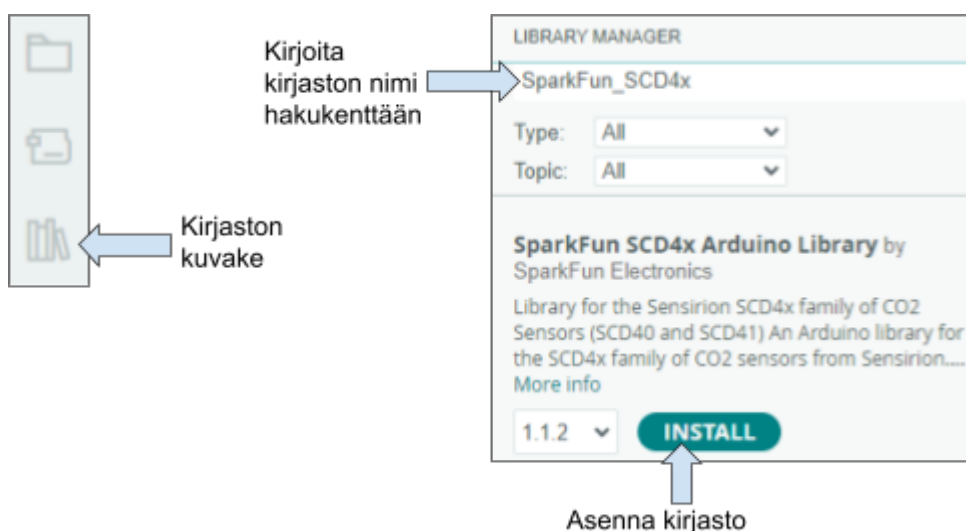
1. Käynnistä tietokoneelta Arduino IDE -ohjelma. Tarvittaessa voit etsiä sen Windowsin hakutoiminnolla.



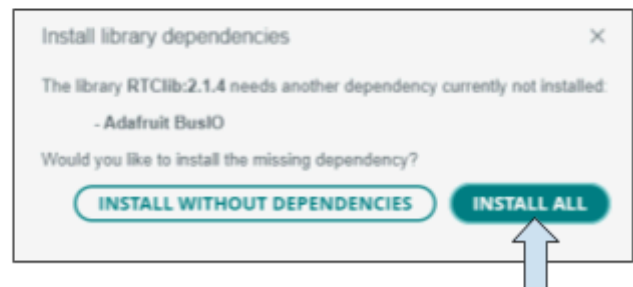
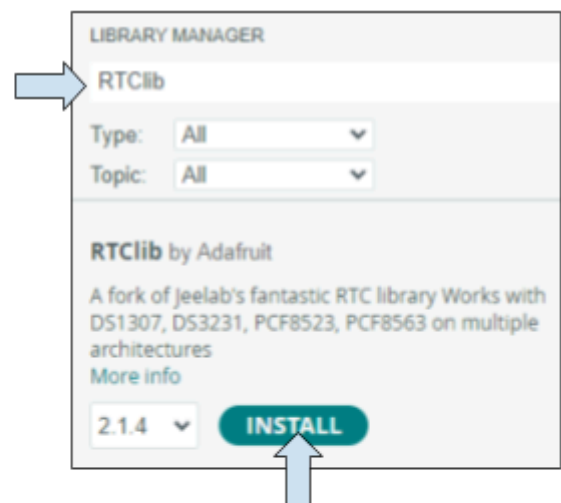
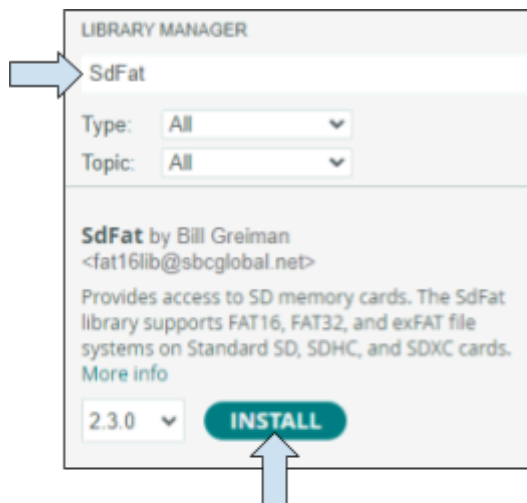
2. Liitä Uno USB-kaapeli tietokoneeseen. Jos ohjelma kysyy kirjastojen tai pakettien asennuksesta, niin asenna ne kaikki.
3. Klikkaa ohjelmointiympäristön yläpalkissa oleva valikko auki ja valitse käyttämäsi Unon malli ja sarjaliikenneportti (tässä COM9, sinulla voi olla jokin muu, yleensä väliltä COM1-COM10).



9. Jos et ole aikaisemmin asentanut anturin lukemiseen käytettävää ulkoista kirjastoa, niin asenna se seuraavaksi. Jos kirjasto on jo asennettu, saat ilmoituksen yrittäessäsi asentaa sen uudelleen. Klikkaa sivupalkista kirjaston kuvaketta ja kirjoita hakukenttään kirjaston nimen alkuosa "**SparkFun_SCD4x**". Hakutulos tulee alapuolelle, klikkaa siitä löytyvää painiketta "**INSTALL**".



10. Data logger -kortin käyttöön tarvitaan kaksi uutta kirjastoa. Hae ja asenna kirjastot nimillä **"SdFat"** sekä **"RTClib"**.



Asenna myös kaikki RTClib:in riippuvat kirjastot

11. Poista ohjelmointiympäristöstä tyhjän ohjelman runko.

```

sketch_jan13a.ino
1 void setup() {
2   // put your setup code here, to run once:
3 }
4
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8 }
9
10
  
```

Poista koodi tästä

```

sketch_jan13a.ino
1
  
```


12. Maalaa ja kopioi sivulta 42 alkava koodi. Liitä se sitten ohjelmointiympäristöön.

```
itsenainen_tiedonkeruu.ino
1  #include <SPI.h>
2  #include <SdFat.h> // Muistikortin kirjasto
3  #include <RTCLib.h> // RTC-kirjasto
4  #include "SparkFun_SCD4x_Arduino_Library.h"
5
6  // SD-korttipaikan piirivalinta (CS) -pin
7  const int chipSelect = 4;
8  SdFat SD;
9  RTC_DS3231 rtc; // RTC-objekti
10 const String fileName = "data.csv";
11
```

Liitetty koodi

13. Data logger -kortissa on myös kello. Se on erittäin arvokas ominaisuus, sillä se mahdollistaa mittaustulosten sitomisen kiinni aikaan. Tiedämme, mitä arvot ovat olleet milläkin ajan hetkellä.

Koodissa oleva rivi asettaa päivämäärän ja ajan aina Unon käynnistyksen yhteydessä. Etsi koodista kuvan mukainen rivi ja muuta siihen se ajankohta, jolloin olet suunnitellut käynnistäväsi tiedonkeruun laitteella.

```
16  rtc.adjust(DateTime(2025, 1, 24, 17, 52, 0));
```

↑ ↑ ↑ ↑ ↑
vuosi, kuukausi, päivä tunnit, minuutit

Toinen kohta mahdollinen muutettava kohta on mittausperiodi, eli jakso jonka välein mittaukset tehdään. Koodissa on oletuksena mittaukset yhden minuutin välein, mutta etenkin hyvin pitkäkestoisessa mittauksessa periodi voi olla pidempi. Etsi koodista kuvan mukainen rivi ja kirjoita numeron 1 tilalle haluamasi tauko minuuhteina mittausten välissä.

```
115 // Tehdään mittaus minuutin välein
116 delayMinutes(1);
```

Vaihda tarvittaessa tilalle haluamasi luku. Se määrittää, kuinka monen minuutin tauko mittausten välissä pidetään.

14. **Varmista, että muistikortti ei ole laitteessa.** Se laitetaan vasta käyttöönoton yhteydessä.

15. Koodi on valmis ladattavaksi, klikkaa yläpalkista kuvaketta "Upload". Koodi käännetään ja lähetetään Unoon. Tämä kestää muutaman sekunnin. Lopuksi näet Output-ikkunassa ilmoituksen latauksen valmistumisesta.



Klikkaa kuvaketta "Upload"

```
Download done.  
DFU state(7) = dfuMANIFEST, status(0) = No error condition is present  
DFU state(2) = dfuIDLE, status(0) = No error condition is present  
Done!
```

Lataus onnistui ja on nyt valmis

16. **Irrota USB-kaapeli Unosta.** Tallenna koodi haluamallasi nimellä ja siirry sitten kappaleeseen "Käyttöönotto".



Arduino Uno koodi:

```

#include <SPI.h> // Muistikortin tiedonsiirto
#include <SdFat.h> // Tiedostojärjestelmän kirjasto
#include <RTCLib.h> // Reaaliaikakellon kirjasto
#include "SparkFun_SCD4x_Arduino_Library.h" // Anturin kirjasto

const int chipSelect = 4; // SD-korttipaikan (CS) -pin
const int ledPin = 5; // Led kytketty pinniin 5
SdFat SD;
RTC_DS3231 rtc; // RTC-objekti
const String fileName = "data.csv";
SCD4x anturi;

void setup() {
  // Käynnistetään kello, asetetaan pvm ja aika
  if (!rtc.begin()) while (true);
  rtc.adjust(DateTime(2025, 1, 24, 17, 52, 0));
  // Alustetaan SD-kortti SdFat-kirjastolla
  if (!SD.begin(chipSelect, SD_SCK_MHZ(10))) while (true);
  // Luodaan tai alustetaan "data.csv" tiedosto
  File dataFile = SD.open(fileName.c_str(), O_WRONLY | O_CREAT | O_TRUNC);
  if (dataFile) {
    dataFile.println("PVM,AIKA,CO2,TEMP,HUM"); // Otsikkorivi
    dataFile.close();
  } else while (true);
  // Alustetaan anturi
  if (!anturi.begin()) while (true);
  if (!anturi.startPeriodicMeasurement()) while (true);
  // Ledin ohjaus
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  delay(7000);
}

String formatTwoDigits(int value) {
  if (value < 10) {
    return "0" + String(value);
  }
  return String(value);
}

String formatDate(int day, int month) {
  return String(day) + "." + String(month) + ".";
}

```

```
void delayMinutes(int minutes) {
  for (int i = 0; i < minutes; i++) {
    delay(60000); // Viive 60 000 millisekuntia (1 minuutti)
  }
}

void loop() {
  if (anturi.readMeasurement()) {
    // Luetaan CO2, Temp sekä Hum ja tallennetaan muuttujiin
    uint16_t co2 = anturi.getCO2();
    int temp = (int)round(anturi.getTemperature());
    int hum = (int)round(anturi.getHumidity());
    // Haetaan nykyinen aika RTC:stä
    DateTime now = rtc.now();
    // Muodostetaan päivämäärä ja kellonaika oikeassa muodossa
    String date = formatDate(now.day(), now.month());
    String time = formatTwoDigits(now.hour()) + ":" +
formatTwoDigits(now.minute());
    // Avaa tiedosto lisäystä varten ja tallennetaan data
    File dataFile = SD.open(fileName.c_str(), O_WRONLY | O_APPEND);
    if (dataFile) {
      digitalWrite(5, HIGH);
      dataFile.print(date);
      dataFile.print(",");
      dataFile.print(time);
      dataFile.print(",");
      dataFile.print(co2);
      dataFile.print(",");
      dataFile.print(temp);
      dataFile.print(",");
      dataFile.println(hum);
      dataFile.flush();
      dataFile.close();
      delay(2000);
      digitalWrite(5, LOW);
    }
  }
  delayMinutes(1); // Tehdään mittaus minuutin välein
}
```

Käyttöönotto

1. Irrota USB-kaapeli Arduino Unosta.
2. Aseta muistikortti data loggerissa olevaan korttipaikkaan.
3. Vie tiedonkeruulaite haluamaasi mittauspaikkaan.
4. Liitä 9 Voltin paristo virtajohtoon ja kytke se Unoon. Pyri käynnistämään laite koodissa määrittämäsi aikaan.

```
16 | rtc.adjust(DateTime(2025, 1, 24, 17, 52, 0));
```

Diagram illustrating the parameters of the `DateTime` function in the code snippet above. Orange arrows point from the labels below to the corresponding values in the code:

- 2025: vuosi
- 1: kuukausi
- 24: päivä
- 17: tunnit
- 52: minuutit

5. Noin seitsemän sekunnin kuluttua virtajohtoon kytkemisestä punaisen ledin pitäisi syttyä kahdeksi sekunniksi. Ledi syttyy aina laitteen tallentaessa tietoa muistikortille.

Jos ledi ei vilkahda, irrota virtajohto, odota hetki ja kytke se uudelleen. Toista tarvittaessa muutaman kerran.

6. Kerää haluamasi määrä mittaustuloksia.
7. **Älä irrota muistikorttia laitteesta, kun paristo on kytkettynä.** Odota, että näet punaisen ledin syttyvän ja sammuvan. **Ledin sammumisen jälkeen viimeisin tallennus on suoritettu ja voit irrottaa virtajohtoon Unosta.** Tämän jälkeen voit poistaa muistikortin laitteesta.

8. Liitä muistikortti tietokoneeseen. Mittaustulokset on tallennettu kortilla olevaan "data.csv"-tiedostoon. Kuvassa on esitetty tiedoston alkuosa. Ensimmäinen rivi on otsikkorivi. Sen alapuolella mittaustulokset minuutin välein.

```
PVM, AIKA, CO2, TEMP, HUM
21.1., 16:30, 987, 24, 24
21.1., 16:31, 866, 24, 27
21.1., 16:32, 792, 24, 27
21.1., 16:33, 762, 24, 27
21.1., 16:34, 758, 24, 27
```

CO2-arvojen osalta anturi vaatii pari minuuttia aikaa, ennen kuin mittaustulokset vakiintuvat. Heti käynnistyksen jälkeen ne ovat hiukan todellisia arvoja suurempia.

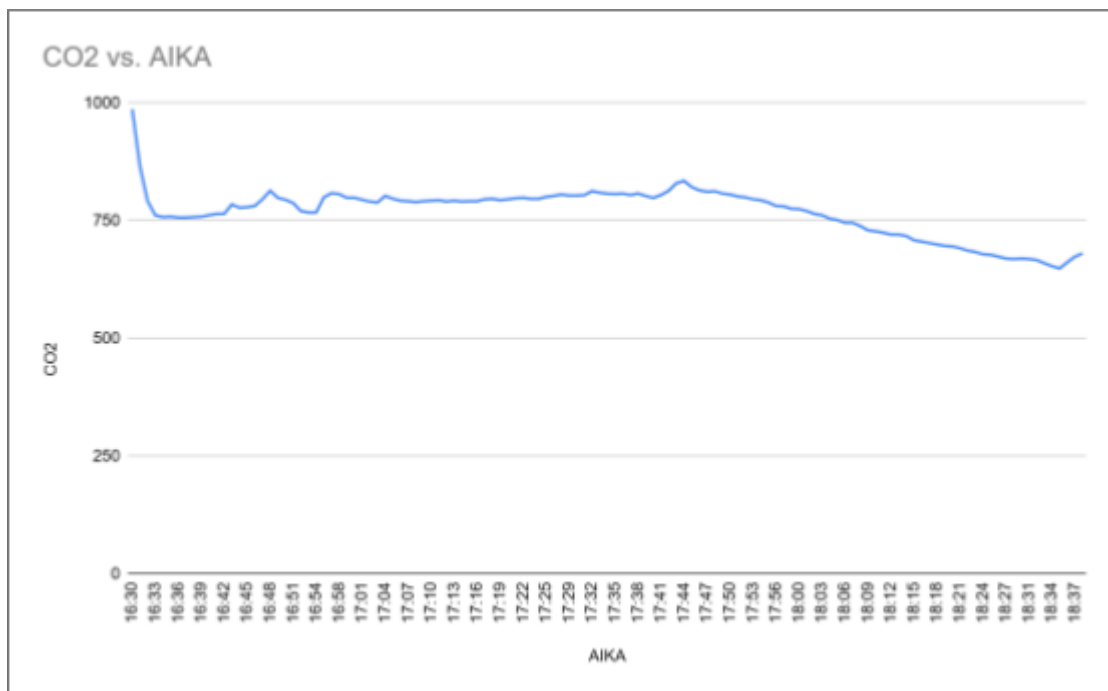
Mittaustulokset data.csv -tiedostossa

9. Vie tiedosto Google Sheetsiin (tai Exceliin) analysoitavaksi. Katso tarvittaessa ohjeet sivulta 24-26.

Mittaustulokset Google Sheetissä:

	A	B	C	D	E
1	PVM	AIKA	CO2	TEMP	HUM
2	21.1.	16:30	987	24	24
3	21.1.	16:31	866	24	27
4	21.1.	16:32	792	24	27
5	21.1.	16:33	762	24	27
6	21.1.	16:34	758	24	27
7	21.1.	16:35	759	24	26

CO2-pitoisuudesta piirretty kuvaaja:



Kaikki komponentit

Taulukossa on esitetty kaikki mittauslaitteen eri versioissa käytetyt komponentit tilauslinkkeineen.

Laite / komponentti	Kuvaus ja tarkoitus	Linkit / huomautukset
Windows-tietokone, mielellään kannettava	Ohjelmointiin ja laitteen käyttöön	Käyttäjällä tulee olla oikeus asentaa tietokoneeseen ohjelmia ja muodostaa kansio C-aseman juureen
Arduino Uno mikro-ohjain esim. Uno R4 Minima	Ohjelmoidaan lukemaan anturia ja lähettämään data tietokoneelle tai LCD-näytölle	Partco Starelec Triopak
Liitinkortti Grove Arduino Base Shield	Unon päälle kiinnitettävä kortti, johon anturi ja LCD-näyttö kiinnitetään	Partco
Mittausanturi Grove CO2+temp+hum	Mittaa ympäristön CO2-pitoisuuden sekä lämpötilan ja kosteuden	Partco
Lukittumaton kytkentäkaapeli, tulee anturin mukana	Kaapelilla anturi kytketään liitinkorttiin	Varakaapeleita voi tilata linkistä Partco
USB-kaapeli	Arduino Unon ja tietokoneen väliin.	Tilaa käyttämäsi Unon mallin mukaan, katso kuva. Partco Radioduo
Grove 2x16 LCD-näyttö	Kaksi riviä, 16 merkkiä / rivi. Mittaustulosten esitys.	Partco
Lukittumaton Grove liitinkaapeli	Kytkee LCD-näytön Grove-liitinkorttiin	Tulee näytön mukana.
Virtajohto Unolle 9 Voltin paristoliittimellä	Itsenäisessä käytössä virta otetaan paristosta	Partco

Komponentti	Kuvaus ja tarkoitus	Linkit
9 Voltin paristo		
Data logger -kortti	Mittaustulosten tallennus SD-muistikortille. Sisältää lisäksi kellon (RTC).	Partco
Growtail led-moduuli, punainen	Toimii laitteen merkkivalona	Partco
SD-muistikortti, kapasiteetti 2-4 Gt Voi olla täysikokoinen tai microSD.	Mittaustietojen tallennukseen	Muistikauppa
SD-muistikortin lukija (jos tietokoneessa ei ole valmiina korttipaikkaa)		Yleisesti myynnissä

Kysymyksiä pohdittavaksi

CO₂-pitoisuus

1. Tutki mittausdataa ja analysoi kuvaajaa. Miksi CO₂-pitoisuus nousee ja laskee joinakin ajanjaksoina? Selitä, mistä muutokset johtuvan.
2. Mitä erilaisia keinoja on vähentää hiilidioksidipitoisuutta luokkahuoneessa? Mikä näistä on mielestäsi tehokkain?
3. Mitä ulkoiset tekijät, kuten liikenne tai teollisuus, voivat kertoa ympäristön hiilidioksidipitoisuudesta?

Lämpötila

4. Miten ilman lämpötila vaikuttaa ihmisten mukavuuteen eri tilanteissa (esim. kotona, koulussa, urheilukentällä)?
5. Voiko liian korkea tai matala lämpötila vaikuttaa laitteen mittaustuloksiin? Jos voi, miten?
6. Pohdi, miten lämpötilan vaihtelut vaikuttavat eri ekosysteemeihin, kuten metsäalueisiin tai kaupunkialueisiin.

Ilmankosteus

7. Miksi ilmankosteus voi tuntua epämukavalta, jos se on hyvin korkea tai matala?
8. Kuinka ilman kosteus vaikuttaa kasveihin? Voisiko laitteella seurata kasvihuoneen olosuhteita?
9. Miten kosteuden hallinta on tärkeää rakennetussa ympäristössä, kuten koulurakennuksissa?

Mittausdatan analysointi

10. Voisiko mittaustuloksia käyttää todistamaan ilmanvaihdon puutteellisuutta tietyssä tilassa? Miten esittäisit havaintosi?
11. Miten voisit yhdistää laitteen keräämät tiedot sääennusteisiin? Mitä lisäarvoa tämä voisi tuoda?
12. Millä muilla tavoilla voisit käyttää mittausdataa ympäristön tilan arvioimiseen?

Ympäristö ja ilmastonmuutos

13. Mitä voisit oppia mittaamalla CO₂-pitoisuuksia pitkän aikavälin ajan? Voisitko havaita trendejä tai säännönmukaisuuksia?
14. Kuinka tällaisia mittauksia voitaisiin hyödyntää ilmastonmuutoksen torjunnassa?
15. Miten voit omalla toiminnallasi vaikuttaa CO₂-pitoisuuteen ja ympäristön hyvinvointiin?

Tutkimusideoita ja vinkkejä

CO₂-pitoisuuden ja ilmanlaadun tutkimus

- **Tutkimuskysymyksiä:**
 - Miten CO₂-pitoisuus muuttuu päivän aikana luokkahuoneessa tai ulkona?
 - Miten ilmanvaihto vaikuttaa CO₂-pitoisuuteen?
 - Onko CO₂-pitoisuus korkeampi, kun huoneessa on enemmän ihmisiä?
- **Käytännön tehtävät:**
 - Oppilaat mittaavat CO₂-pitoisuuksia eri tilanteissa (esim. välitunnilla, oppitunnilla, ja tyhjässä huoneessa).
 - Tuloksia verrataan suositusarvoihin (esim. sisäilman CO₂-suositus < 1000 ppm).
- **Opetusmahdollisuudet:**
 - Keskustelkaa, miksi korkeat CO₂-tasot voivat vaikuttaa oppimiseen ja vireyteen.
 - Johdattele oppilaat ymmärtämään ilmanlaadun parantamisen keinoja (ilmanvaihto, tuulettaminen, ilmanpuhdistimet).

CO₂-pitoisuus ja ihmistoiminnan vaikutukset

- **Tutkimuskysymyksiä:**
 - Miten liikenteen määrä vaikuttaa ulkoilman CO₂-tasoon eri vuorokaudenaikoina?
 - Onko ulkoilman CO₂-taso korkeampi teiden läheisyydessä verrattuna puistoihin?
- **Käytännön tehtävät:**
 - Oppilaat mittaavat ulkoilman CO₂-pitoisuuksia eri ympäristöissä (esim. koulun parkkipaikka vs. metsä).
 - Vertailu eri kaupunkien tai asuinalueiden ilmanlaadun välillä yhteistyössä toisen koulun kanssa.
- **Opetusmahdollisuudet:**
 - Keskustelu kaupunkisuunnittelun merkityksestä ilmanlaadun parantamisessa.
 - Ilmastopäästöjen lähteiden tunnistaminen ja analysointi.

Lämpötilan ja kosteuden vaikutus ympäristöön

- **Tutkimuskysymyksiä:**
 - Miten lämpötila ja kosteus muuttuvat eri vuorokauden- tai vuodenaikoina?
 - Miten kosteusprosentti muuttuu sateisina päivinä verrattuna aurinkoisiin päiviin?
- **Käytännön tehtävät:**
 - Oppilaat sijoittavat laitteen eri ympäristöihin (esim. luokkahuone, ulkoilma, kasvihuone) ja seuraavat mittaustuloksia.
 - Tuloksia käytetään kuvaajien ja keskiarvojen laskemiseen Google Sheetsissä.
- **Opetusmahdollisuudet:**
 - Ymmärrys kosteuden ja lämpötilan vaikutuksesta hyvinvointiin (esim. liian kuiva tai kostea ilma).
 - Ilmastomuutoksen vaikutusten havainnollistaminen.

Lämpötila ja kosteus pitoisuus biologiassa ja maantiedossa

- **Tutkimuskysymyksiä:**
 - Miten kasvihuoneen ilmanlaatu muuttuu päivän aikana?
 - Miten kasvien määrä vaikuttaa huoneen CO₂-pitoisuuteen ja kosteuteen?
- **Käytännön tehtävät:**
 - Oppilaat sijoittavat laitteen kasvihuoneeseen ja vertaavat mittauksia kasvien lähellä ja kauempana.
 - Analysoidaan, miten kasvien yhteyttäminen näkyy CO₂-pitoisuuden laskuna päivän aikana.
- **Opetusmahdollisuudet:**
 - Kasvien merkitys hiilen kiertokulussa ja ilmastonmuutoksen torjunnassa.
 - Ilmaston ja sääolosuhteiden vaikutus kasvillisuuteen.

Datan analysointi ja matematiikan integrointi

- **Matemaattiset taidot:**
 - Lasketaan keskiarvoja, huippuarvoja ja hajontoja CO₂-, lämpötila- ja kosteusmittauksista.
 - Harjoitellaan kuvaajien laatimista ja niiden tulkitsemista.
- **Opetusmahdollisuudet:**
 - Visualisointi: Oppilaat oppivat, miten numeeriset arvot muutetaan helposti ymmärrettäviksi kuvaajiksi.
 - Tilastollinen päättely: Analysoidaan, ovatko erot tilastollisesti merkittäviä.

Ympäristötietoisuuden kasvattaminen

- **Projektityöt:**
 - Oppilaat voivat tehdä tutkimusprojekteja ilmanlaadusta.
 - He voivat esittää johtopäätöksiä ja ehdottaa parannuksia esimerkiksi koulun ilmanvaihtoon tai energiansäästöön.
- **Keskustelut:**
 - Miten erilaiset ihmisen toiminnat, kuten liikenne tai energiankulutus, vaikuttavat ympäristöön?
 - Voidaanko datan avulla tehdä kestävämpiä päätöksiä?

Pitkän aikavälin seuranta

- **Tutkimuskysymyksiä:**
 - Miten eri vuodenajat vaikuttavat CO₂-tasoihin ulkona ja sisällä?
 - Voidaanko lämpötilasta ja kosteudesta löytää säännöllisiä trendejä?
- **Käytännön tehtävät:**
 - Oppilaat keräävät dataa viikkojen tai kuukausien ajan ja analysoivat pitkäaikaisia trendejä.
 - He voivat verrata omia mittauksiaan säähavaintoihin.
- **Opetusmahdollisuudet:**
 - Johdattele oppilaat ilmastotieteeseen ja säähavaintojen tulkitsemiseen.
 - Keskustelkaa siitä, miten ilmastomallinnus perustuu pitkäaikaisiin havaintoihin.

Ilmaston ja energiatehokkuuden tutkimus

- **Tutkimuskysymyksiä:**
 - Miten erilaiset lämmitysratkaisut vaikuttavat sisäilman lämpötilaan ja kosteuteen?
 - Kuinka energiatehokkuus ja ilmanvaihto voidaan optimoida?
- **Käytännön tehtävät:**
 - Oppilaat vertaavat erilaisten huoneiden lämmitystehokkuutta (esim. hyvin eristetty vs. huonosti eristetty tila).
 - Simuloidaan energiahävikkiä avaamalla ikkuna tai käyttämällä tuuletinta.
- **Opetusmahdollisuudet:**
 - Johdatus kestävään energiankäyttöön ja sen vaikutuksiin sisäilman laatuun.

Luonnontieteiden integrointi historiaan ja kulttuuriin

- **Käytännön tehtävät:**
 - Tutkitaan, miten ilmanlaatu on muuttunut historiallisesti (esim. teollistumisen aikana).
 - Luodaan oppilaille kulttuuriprojekti, jossa he analysoivat ilmaston ja ihmistoiminnan välistä vuorovaikutusta taiteen tai kirjallisuuden kautta.
- **Opetusmahdollisuudet:**
 - Yhdistä ilmastotutkimus laajempaan kontekstiin, kuten historian suuriin tapahtumiin.

IoT ja teknologia opetuksessa

- **Johdatus ohjelmointiin ja elektroniikkaan:**
 - Oppilaat voivat tutustua Arduinin ohjelmointiin ja muokata koodia lisäämään esimerkiksi uusia antureita.
 - Keskustelkaa, miten IoT (esineiden internet) auttaa ympäristön monitoroinnissa.

Kiinteistöautomaatio

Automaatio tarkoittaa järjestelmien ja prosessien itsenäistä ohjaamista ja valvontaa. Kiinteistöautomaatioissa tavoitteena on optimoida rakennuksen toimintoja, kuten lämmitystä, ilmanvaihtoa ja valaistusta, jotta käyttömukavuus paranee ja energian kulutus pienenee. Automaatiojärjestelmä perustuu antureista saatavaan dataan, joka analysoidaan ja jonka perusteella tehdään päätöksiä automaattisesti.

Automaatiojärjestelmän keskeiset osat ovat:

1. **Anturit:** Mittaavat ympäristön muuttujia, kuten lämpötilaa, kosteutta ja CO₂-pitoisuutta.
2. **Ohjausyksikkö:** Analysoi anturien keräämän tiedon ja tekee päätökset ohjelmoitujen sääntöjen mukaan.
3. **Toimilaitteet:** Toteuttavat ohjausyksikön päätökset, kuten kytkevät lämmityksen päälle tai muuttavat ilmanvaihdon tehoa.

Ilmanvaihdon ohjaus CO₂-pitoisuuden perusteella

CO₂-pitoisuuden mittaus antaa tärkeää tietoa ilmanlaadusta. Sisäilman hiilidioksidipitoisuus voi nousta liian korkeaksi, kun huoneessa on paljon ihmisiä tai ilmanvaihto ei ole riittävää. Tämä voi heikentää vireystilaa ja aiheuttaa epämukavuutta.

- Automaatiojärjestelmä voi säätää ilmanvaihtoa CO₂-pitoisuuden perusteella. Kun pitoisuus nousee yli suositusarvon (esim. 1000 ppm), järjestelmä lisää ilmanvaihdon tehoa. Kun huone tyhjenee ja CO₂-pitoisuus laskee, ilmanvaihto voidaan säätää takaisin pienemmälle, mikä säästää energiaa.

Lämmityksen optimointi lämpötilan ja kosteuden perusteella

Lämpötilan ja kosteuden mittaukset auttavat ylläpitämään mukavaa sisäilmaa energiatehokkaasti.

- Jos huonelämpötila laskee alle mukavuusrajan (esim. 20 °C), järjestelmä voi kytkeä lämmityksen päälle. Kun lämpötila nousee halutulle tasolle, lämmitys voidaan katkaista.
- Kosteuden hallinta on tärkeää sekä mukavuuden että rakennuksen kunnossapidon kannalta. Jos kosteus nousee liian korkeaksi (esim. yli 60 %), järjestelmä voi lisätä ilmanvaihtoa tai säätää lämmitystä kosteuden hallitsemiseksi.

Mukavuusalueen ylläpito

Yhdistämällä CO₂-pitoisuuden, lämpötilan ja kosteuden mittaukset automaatiojärjestelmä voi luoda ihanteelliset olosuhteet pienellä energian kulutuksella.

- Esimerkiksi talvella ilmanvaihdon lisääminen CO₂-pitoisuuden takia nostaa samalla lämmityskustannuksia, joten järjestelmä voi optimoida nämä molemmat tarpeet tasapainoisesti.
- Automaattinen ohjaus varmistaa, että sisäilmasto pysyy mukavana ilman jatkuvaa käyttäjän säätämistä.

Kiinteistöautomaatiojärjestelmän käytännön hyödyt

1. **Energiatehokkuus:** Optimointi vähentää energiankulutusta, koska järjestelmä toimii vain tarpeen mukaan.
2. **Parantunut sisäilman laatu:** CO₂-pitoisuuden, kosteuden ja lämpötilan seuranta varmistaa mukavan ja terveellisen ympäristön.
3. **Käyttäjäturvallisuus:** Automaatiojärjestelmä toimii itsenäisesti, mikä vähentää käyttäjän tarvetta säätää järjestelmää manuaalisesti.

Opetukselliset mahdollisuudet

- Oppilaat voivat analysoida, miten automaatio vaikuttaa energiansäästöön ja asumismukavuuteen.
- Projektit, joissa tutkitaan sisäilman laatua ja automaation vaikutuksia, voivat innostaa oppilaita ymmärtämään teknologian merkitystä kestävä kehityksen tavoitteiden saavuttamisessa.
- Automaatio tarjoaa konkreettisen esimerkin siitä, miten luonnontieteitä, matematiikkaa ja ohjelmointia voidaan yhdistää käytännön sovelluksiin.