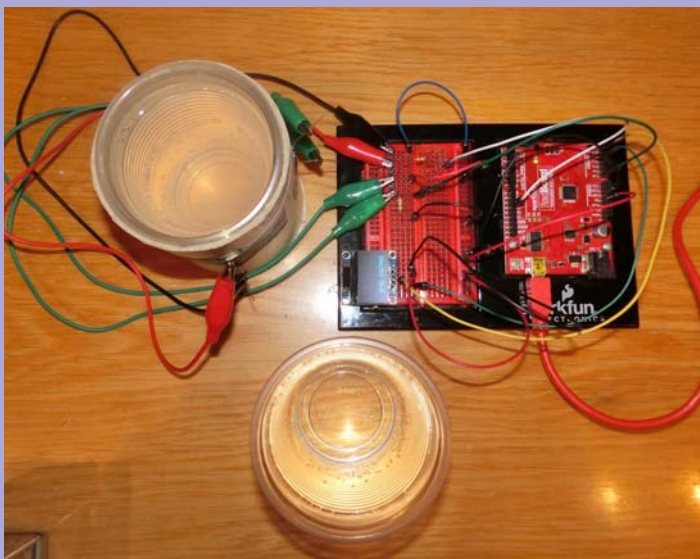


*Nils Kr. Rossing, Erik Løkketangen*

# Måling av partikkeltetthet i væske med Arduino – Lærerveiledning



NTNU



Trondheim

Institutt for  
fysikk

Skolelaboratoriet  
for matematikk, naturfag  
og teknologi

November 2019



# Måling av partikkeltetthet i væske med Arduino LÆRERVEILEDNING

Nils Kr. Rossing, Erik Løkketangen

## **Måling av partikkeltetthet i væske med Arduino – Lærerveiledning**

Trondheim 2019

Layout og redigering: Nils Kr. Rossing, Skolelaboratoriet ved NTNU

Trykk: NTNU Grafisk senter

Tekst og bilder: Nils Kr. Rossing, Skolelaboratoriet ved NTNU

Korrektur og utprøving: Erik Løkketangen, Skolelaboratoriet/Brundalen skole

Faglige spørsmål rettes til:

**Skolelaboratoriet for matematikk, naturfag og teknologi**

**Institutt for fysikk**

v/ Nils Kr. Rossing [nils.rossing@ntnu.no](mailto:nils.rossing@ntnu.no)

Skolelaboratoriet ved NTNU

Realfagbygget,

Høgskoleringen 5,

7491 Trondheim

Telefon: 73 55 11 91

<http://www.ntnu.no/skolelab/>

Rev 1.2 – 20.11.19



---

## Forord

Hftet ble utarbeidet i forbindelse med fagdag 29. nov. 2019 og beskriver bygging og programmering av et turbidimeter for måling av partikkeltetthet i væsker.

Ønsket var å utvikle et undervisningsopplegg hvor man kunne gjøre målinger og anvende matematikk til å behandle resultatene. Videre skulle oppgaven være slik at elevene selve kunne lage og koble opp måleutstyret samt skrive programmet.

Turbidimeteret egner seg primært til å måle relativ partikkelkonsentrasjon, men burde kunne la seg kalibrere dersom man har kalibrerte oppløsninger eller et kalibrert måleinstrument som f.eks. Pasco turbidimeter.

Ideen til oppgaven er opprinnelig hentet fra Inst. for informatikk ved Universitetet ved Oslo, hvor det ble laget en lignende oppgave for elever i videregående skole. Ideen ble overbrakt av Førsteamanuensis Roger Antonsen.

En tilsvarende oppgave ble benyttet under Science Camp 2019 som ble utviklet på oppdrag fra Trøndelag fylkeskommune (TrFK) våren 2019, hvor oppgaven var å måle partikkelkonsentrasjonen i avløpsvann fra Trondheim kommune.

Erik Løkketangen har testet ut og kvalitetssikret oppgaven før gjennomføringen på fagdagen.

Skolelaboratoriet ved NTNU  
November 2019  
Nils Kr. Rossing  
Erik Løkketangen





## Innhold

<b>1 Innledning .....</b>	<b>9</b>
1.1 Åpen oppgave eller lukket oppgave .....	9
<b>2 Montering og oppkobling for trykkmåling .....</b>	<b>11</b>
2.1 Prinsipiell beskrivelse .....	11
2.2 Oppdrag .....	11
2.2.1 Oppdrag 1 – Montering av lysdiode og lyssensor i kyvettehuset	12
2.2.2 Oppdrag 2 – Montering av turbidimeteret: .....	13
2.3 Kort innføring i bruk av Arduino editoren (IDE) .....	14
2.4 Lag programmet .....	17
2.4.1 Oppdrag 3 – Program for tenne lysdioden og å avlese lysstyrken	17
2.4.2 Gjør målinger på 10 beger og sett dem i rett rekkefølge .....	22
2.4.3 Koble opp et eksternt display .....	23
2.4.4 Skrive ut antall dråper .....	27
2.4.5 Kalibrering av måleinstrumentet .....	28
<b>3 Referanser .....</b>	<b>31</b>
<b>Vedlegg A Blokkdiagram for koden .....</b>	<b>32</b>
<b>Vedlegg B Testing og måling med turbidimeteret .....</b>	<b>34</b>
<b>Vedlegg C Turbiditet .....</b>	<b>37</b>
C.1 Krav til turbiditet i drikkevann .....	39
C.2 Krav til turbiditet i avløpsvann .....	40
<b>Vedlegg D Utstysrliste .....</b>	<b>41</b>







# 1 Innledning

Mikrokontrollerkortet Arduino, med alle sine varianter, gir rike muligheter til å koble til sensorer og lagringsenheter av mange slag. Arduino UNO er kanskje den mest brukte i familien av kontrollerkort, men det finnes også mange andre, f.eks. Arduino Nano som er en god kandidat dersom det er viktig med små dimensjoner. En av fordelene med å bruke Arduino UNO er at det finnes et rikt utvalg av “shield”-kort som lett kan plugges sammen med UNO’en, det samme gjelder sensorer med tilhørende programbiblioteker.

Ved bruk av mikrokontrollere og sensorer kan mange fagområder knyttes sammen. Det er f.eks. lett å knytte sensorteknologi til fysikk og kjemi (ev. materialteknologi), programmering og matematikk for beregning og kalibrering av målte størrelser, og i dette tilfellet datalogging med påfølgende databehandling.

Notatet beskriver oppbyggingen, kalibrering og bruk av et hjemmelaget turbidimeter. Tanken er at instrumentet skal kunne anvendes til måling av partikkeltetthet i avløpsvann og drikkevann o.a..

Et turbidimeter er et instrument som måler spredningen av lys fra partikler i en væske. Instrumentet er om mulig ment å kunne kalibreres mot et profesjonelt instrument. Hovedhensikten er imidlertid at deltagerne skal få en forståelse av hvordan instrumentet fungerer. Instrumentet kan også fungere som et kolorimeter (måling av absorbans), ved montasje av en ekstra lysdiode rett overfor lyssensoren.

En kan se for seg følgende organisering av undervisningsopplegget:

- Kort introduksjon til Arduino og bruk av ulike sensorer, gjerne knyttet til å måle partikler i væsker.
- Turbidimetri og spredning av lys som følge av mikropartikler oppløst i væske, og hva dette anvendes til i industrien (se vedlegg C).
- Oppbygging av måleutrustning med lyskilde og lyssensor, Arduino og display (presentasjon med lab.)
- Utføre målinger av lysspredning i vann tilsatt ulike mengder melk og skrive ut resultatet på display
- Gjøre mange målinger og finne middelveidien for å redusere innflytelsen av tilfeldig støy
- Kalibrere det hjemmebygde apparatet ved hjelp av kalibrerte oppløsninger.
- Diskusjon og anvendelighet

Opplegget med bygging og måling med turbidimeteret vil kunne gjennomføres i løpet av en 2 – 3 timers økt for elever som har litt kjennskap til programmering av Arduino.

## 1.1 Åpen oppgave eller lukket oppgave

Avhengig av kunnskapene hos elevene og hvor mye tid man har til rådighet, kan man enten velge å gjøre undervisningsopplegget til en åpen oppgave eller en mer oppskriftsmessig oppgave som beskrevet i kapittel 2.

For elever som har tilstrekkelige kunnskaper kan man la oppgaven være relativt åpen med fokus på måleoppdraget som beskrevet i oppgaveeksempelet under

### Forslag til åpen oppgave

Lag et måleinstrument som ved hjelp av sensorer, aktuatorer og en Arduino UNO klarere å skille 10 plastbeger tilsatt fra 1 – 10 dråper med melk. Begrene skal etter utført måling, plasseres i rekkefølge fra beger tilsatt 1 dråpe og opp til begeret tilsatt 10 dråper i stigende rekkefølge.

Tilgjengelig utstyr er:

- Et lystett kammer med lokk
- Et begerglass i plast
- Litt melk
- En pipette
- En hvit lysdiode
- En lyssensor (LDR)
- Div kabler for oppkobling
- Arduino kit med mikrokontrollerkort, jumpere, motstander og koblingsbrett

Lag en hensiktsmessig oppkobling, skriv et program og gjennomfør målinger. Beskriv framgangsmåten og resultatet av målingene. Reflekter over resultatet og foreslå eventuelle forbedringer av instrumentet.



## 2 Montering og oppkobling for trykkmåling

I vårt måleoppsett har vi valgt å benytte en Arduino UNO i tillegg til et SSD 1306 grafisk display.

### 2.1 Prinsipiell beskrivelse

Et turbidimeter måler *spredningen* av lyset som sendes inn i væsken, derfor må lyskilden og lyssensoren plasseres på vegger som står vinkelrett på hverandre. Graden av spredning avhenger av størrelsen og konsentrasjonen av partiklene.

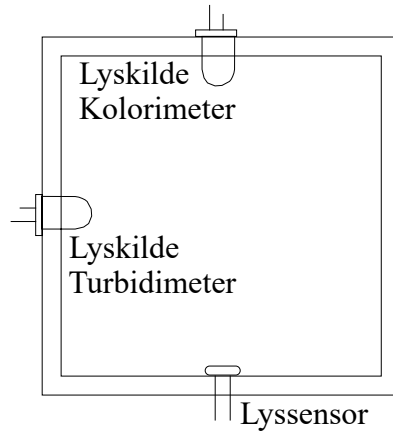
Samtidig kan vi måle *absorbansen* til væskeblandingen, dvs. hvor mye av det utsendte lyset som absorberes i væsken. For å måle absorbans plasseres lyskilden rett overfor sensoren. Dette kaller vi et *kolorimeter*.

Både absorbans og spredning vil være avhengig av fargen til lyset som sendes ut. Det er ikke uvanlig å bruke IR-LED som kilde for turbidimetermålinger, og ulike farger tilpasset absorpsjonsspekteret til oppløsningen. Her har vi foreløpig valgt å bruke hvitt lys.

For å hindre strølys fra å forstyrre målingene våre, monteres lyskilder og sensor i et lystett kammer. Kammeret er så stort at det kan romme et plastkrus med plass til 1,5 dL vann. På sikt kan man også tenke seg et mindre oppsett tilpasset *dransglass* eller lignende.

Turbidimeteret kan også brukes til å måle innholdet av partikler i drikke- og avløpsvann o.a. Det er også mulig å sammenligne målinger gjort med vårt hjemmelagde instrument med et profesjonelt måleinstrument og om mulig kalibrere turbidimeteret vårt. Vurdering av kvaliteten til instrumentet og ev. forslag til forbedringer vil være et mål med utforskningen.

For å utføre målingene bruker vi en Arduino UNO mikrokontroller som styrer lyskildene, leser av sensorverdiene og skriver ut måleverdiene på et lite display.



### 2.2 Oppdrag

Oppdraget krever at det bygges et hensiktsmessig kyvettehus for plastkoppen. For den som arbeider fort er det også mulig å kalibrere turbidimeteret for å utføre kalibrerte målinger.

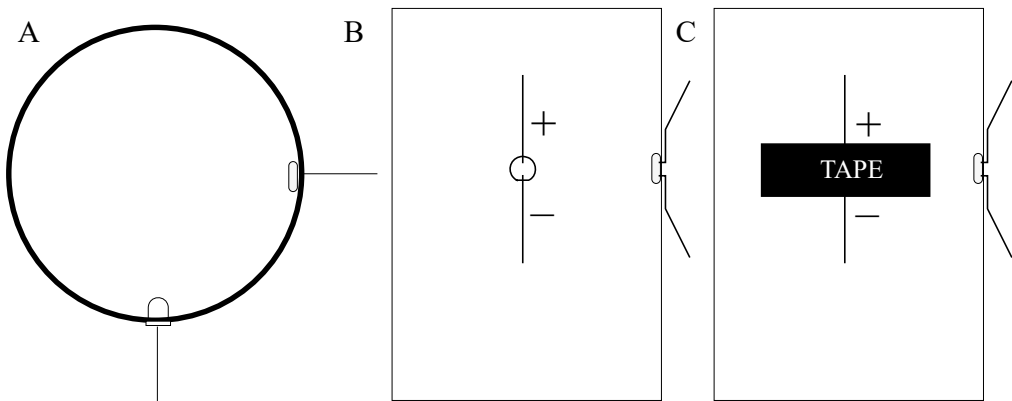
Til dette oppdraget trenger vi følgende komponenter:

- Pappsyylinder med div hull
- Lokk som holder lyse ute

- Krympestrømpe for å avgrense lysstrålen
- To lysdioder for hvitt lys
- En lysfølsom motstand (LDR)
- Tilpasset stiftlist
- Fire labledninger med isolerte krokodilleklemmer
- Arduino Inventor kit + motstand 220kOhm (rød – rød – gul)

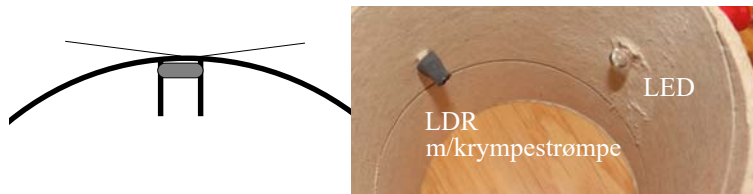
### 2.2.1 Oppdrag 1 – Montering av lysdiode og lyssensor i kyvettehuset

1. **Monter lysdioden og den lysfølsomme motstanden** som vist på figuren under. Vær gjerne konsekvent ved å monter plussledningen LED'en opp. Det spiller ingen rolle hvilken vei den lysfølsomme motstanden (LDR) monteres. Bøy ledningene slik at de peker henholdsvis oppover og nedover som antydnet på figuren under (B). NB! Det er viktig å sette en svart tape tvers over baksiden av lysdioden og lyssensoren (C) for å hindre lyslekkasje.



2. **Monter krympestrømpe på lyssensoren.**

Siden vi kun er interessert i lys som er reflektert fra partiklene må vi sørge for at direkte lys fra dioden blir fjernet. Dette gjør vi ved å skyve en ca. 7 mm krympestrømpe inn på lyssensoren før vi monterer den i boksen.

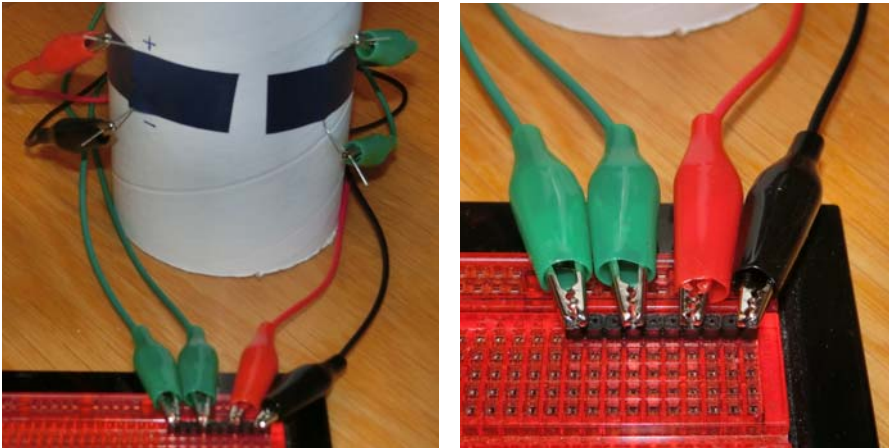


3. **Monter stiftlisten og krokodilleledningene**

Stiftlisten monteres ytterst til høyre på koblingsbrettet som vist på bildet under. Bruk krokodilleledninger til å koble opp forbindelsen mellom kyvettehuset og koblingsbrettet som vist.

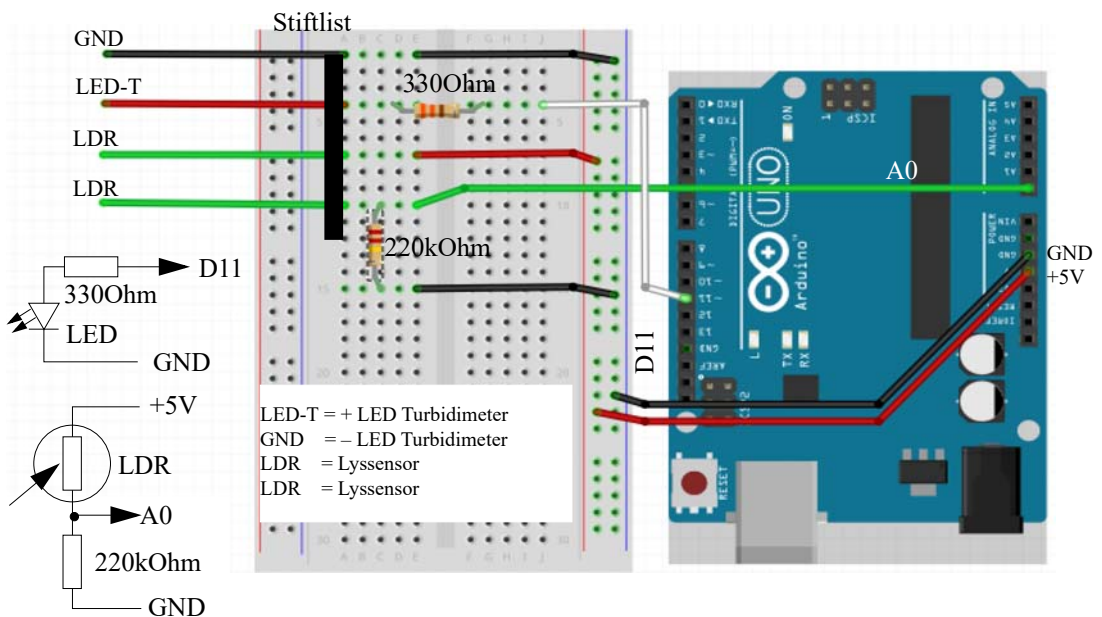


Om tilgjengelig bruk rød ledning til + på (anoden) lysdioden og sort til – (katoden). Bruk en annen farge til å koble opp lyssensoren.



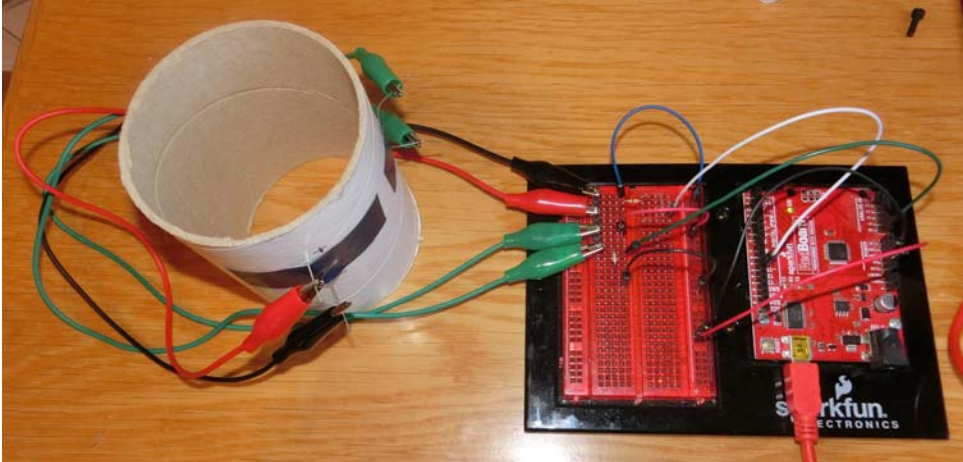
## 2.2.2 Oppdrag 2 – Montering av turbidimeteret:

1. Koble lysdiode (LED) og lyssensor (LDR) til Arduinoen. Bruk *Sparkfun Inventors kit* pluss tilbehør for å koble opp kretsen som vist på figuren under. Se koblingskjema nede til venstre.



Vi bruker to motstander 330 Ohm (oransje – oransje – brun) i serie med lysdioden og 220 kOhm (rød – rød – gul) i serie med lyssensoren (LED).

Bildet under viser hvordan det kan bli seende ut.



Vi er nå klar til å begynne å bygge opp programmet. Men la oss først ta en titt på bruken av Arduino editoren. De som har installert Arduino IDE og kjenner programvaren, kan hoppe over neste avsnitt og gå direkte til avsnitt 2.4 på side 17.

## 2.3 Kort innføring i bruk av Arduino editoren (IDE)

Dette er en meget kortfattet innføring i bruk av editoren til Arduino.

### 1. Installasjon av programvaren

Det er normalt ikke nødvendig å installere en ny utgave dersom en gammel er installert i maskinen fra før. Eldre versjoner vil sannsynligvis fungere godt.

Programvaren hentes fra:

<https://www.arduino.cc/en/Main/Software>

Download the Arduino IDE

Velg: *Windows installer* (For Mac – *Mac OS X*)

Velg: *Just Download*



Velg: *Kjør*



Velg: *Følg prosedyren i installasjonen*

Velg: *Å installere drivere*

## 2. Start programmet og installer tre biblioteker

Klikk på ikonet til Arduino og start programmet.



Du vil forhåpentlig se følgende programvindu åpne seg etter en liten stund.



For å kunne kommunisere med displayet (SSD1306) og trykkmåleren (BMP180), så må vi installere tre biblioteker. Disse bibliotekene er komprimert i zip-filer og kan lastes ned fra:

<https://www.ntnu.no/skolelab/bla-hefteserie>

under fanen: *Måling av lufttrykk, høyde og posisjon*

*Adafruit\_SSD1306-master.zip*

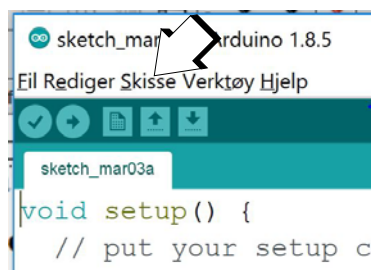
*Adafruit-GFX-Library-master.zip*

*BMP180\_Breakout\_Arduino\_Library-master.zip*

Lagre bibliotekene i nedlastingskatalogen eller et annet sted der du finner dem igjen.

Bibliotekene installeres på følgende måte:

Åpen Arduino-editoren



Velg: *Skisse*

Velg: *Inkluder bibliotek*

Velg: *Legg til .zip bibliotek* (øverst i nedtrekksmenyen)

Finn biblioteket i katalogen Nedlasting

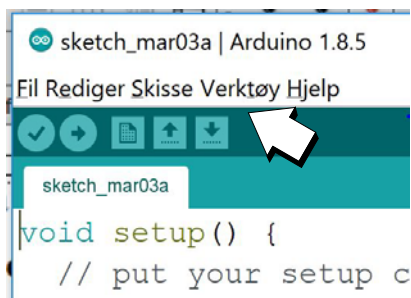
Velg: *Open*

... og biblioteket installeres

Gjenta det samme for alle tre bibliotekene.

### 3. Start programmet og koble Arduinoen til en USB-kontakt

Se på menylinjen øverst til venstre.



Velg: *Verktøy*

Velg: *Kort*

Velg: *Arduino/Genuino UNO*

Velg: *Verktøy*

Velg: *Port*

Velg: Det høyeste COM nummeret, eller velg porten merket med Arduino/Genuino UNO

Det skal nå være opprettet kontakt mellom programeditoren (IDE) og Arduinokortet

### 4. Kort brukerveiledning for program-editoren





De viktigste kommandoene for å betjene bruken av editoren er oppsummert under:



Sjekk at koden er fri for skrivefeil (syntaksfeil)



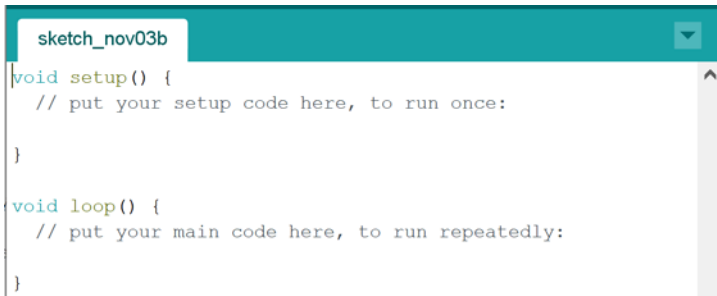


-  Kompiler og send koden til kortet
-  Start en ny skisse med blanke ark
-  Last opp en programskisse fra disken
-  Lagre programskisse på disk

## 2.4 Lag programmet

### 2.4.1 Oppdrag 3 – Program for å tenne lysdioden og for å avlese lyssensoren

I dette avsnittet skal vi bygge opp programmet gradvis. La oss først teste at lysdioden fungerer. Når man åpner en ny skisse (sketch) vil man se følgende bilde:



```
sketch_nov03b
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Her ser man to tomme funksjoner. *Kroppen* til funksjonen er det som er plassert mellom klammeparentesene:

- **void setup() { }** – Det som står mellom klammeparentesene vil kjøre en gang idet programmet starter. Her legger vi inn alle kommandoer som skal kjøres ved oppstart.
- **void loop() { }** – Det som står mellom klammeparentesene i loop-funksjonen vil kjøre om og om igjen så lenge mikrokontrolleren har seppning. Her legges alle kommandoer som vi vil skal gjenta seg. Det gjelder tenning av lysdioden, avlesning av lyssensoren, skriving av resultatene osv.

I fortsetningen vil vi referere til disse to funksjonene når vi skal skrive programmet.

#### 1. Test lysdioden: Tenn og slukk lysdioden

Vi har koblet lysdioden til utgang D11. Vi legger da inn følgende kommandoer:

- Vi setter navnet “pinHvitTurbi” på port 11 som er koblet til lysdioden. Det gjør det lettere å huske hva denne utgangen brukes til:

```
int pinHvitTurbi = 11; // Kilde for måling av turbiditet, port 11
Plasseres helt øverst i programmet (global variabel) før setup() funksjonen
```

- **Definerer at porten skal være en utgang**

```
pinMode(pinHvitTurbi, OUTPUT); // Definerer port 11 som en utgang
Plasseres i setup() funksjonen, kjøres altså en gang ved oppstart
```

- **Tenner lysdioden:**

```
digitalWrite(pinHvitTurbi, HIGH); // Slå på lyskilden (LED)
Plasseres i loop() funksjonen siden vi vil slå lysdioden av og på mange ganger.
```

- **Programmet venter i 0,5 sek (500 millisekunder):**

```
delay(500);
Denne legges rette etter at lysdioden tennes
```

- **Slukker lysdioden:**

```
digitalWrite(pinHvitTurbi, LOW); // Slå av lyskilde
Plasseres i loop() funksjonen siden denne kommandoen skal gjentas mange ganger
```

- **Holder lysdioden slukket i 0,5 sek (500 millisekunder):**


```
delay(500);
Denne legges rette etter at lysdioden slukkes
```

Programmet som slår på og slår av lysdioden blir dermed seende slik ut. Skriv det inn i editoren:

```
int pinHvitTurbi = 11;      // Kilde for måling av turbiditet, port 11

void setup() {
  pinMode(pinHvitTurbi, OUTPUT);
}

void loop() {
  digitalWrite(pinHvitTurbi, HIGH); // Slå på lyskilde
  delay(500);
  digitalWrite(pinHvitTurbi, LOW); // Slå av lyskilde'
  delay(500);
}
```

Trykk på knappen for å compilere og overføre  og kjør programmet:

Lysdioden skal nå blinke 0,5 sek. på og 0,5 sek. av, og som gjentas helt til strømmen brytes eller vi legger inn et nytt program. Om det ikke er tilfelle, sjekk oppkoblingen og se om det er en feil i programmet.

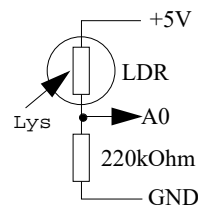
**NB!** Under uttestingen av turbidimeteret har vi funnet ut at lysdioden bruker tid til å slå seg på, dersom tiden fra påslag til måling blir for kort (f.eks. 100 msek) så vil dette påvirke målingen i betydelig grad. Undersøk gjerne dette.

Vurder derfor gjerne å la lysdioden være på hele tiden.



## 2. Les av sensoren og skriv verdien til monitoren

Figuren til høyre viser hvordan sensoren er koblet opp. En lysfølsom motstand fungerer slik at motstandsverdien (resistansen) reduseres med økende lysstyrke. Spenningen mellom A0 og jord (GND) vil derfor øke når lysstyrken øker og minke når lysterken avtar.



Vi kan nå lese av spenningen over den 220kOhms motstanden siden denne er koblet til den analog inngangen A0 på Arduino-kortet.

Når programmet leser av spenning på inngang A0 leser den ikke av spenningen direkte, men et tall mellom 0 – 1023 hvor 0 tilsvarer 0V og 1023 tilsvarer 5 V. Dersom vi ønsker å regne om fra tall til spenning må vi benytte følgende omregningsformel:

$$\text{Avlest spenning} = \text{Avlest måleverdi} * 5V / 1023$$

Hvor *Måleverdien* er det tallet vi leser av.

Strengt tatt trenger vi ikke å finne spenningen, men kun holde oss til måleverdien. Jo høyere måleverdi jo sterkere lys.

- **Deklarer en variabel som holder måleverdien.** Dette kan f.eks. gjøres slik:  

```
int lysIntensitet = 0; // Variabel som holder måleverdi for lysintensitet
```

Plasseres øverst i programmet (global variabel) før setup() funksjonen  
Denne er en heltallsverdi (int) og gis startverdien 0.
- **Måleverdien på inngang A0 leses av med følgende kommando:**  

```
lysIntensitet = analogRead(A0);
```

Kommandoen legger den avleste måleverdien inn i variabelen `lysIntensitet`.  
Avlesningen må gjøres i loopen mens lysdioden er påslått.

For at vi skal bli istand til å se den avleste verdien så må den skrives til monitoren i editoren.

## 3. Skriv til monitoren

**Monitoren** er et vindu som viser informasjon som sendes fra programmet som kjører i Arduino'en og tilbake til PC'en. Monitoren åpner vi etter at programmet er lastet over i Arduino'en. Når du åpner monitoren startes programmet fra begynnelsen.



- **Initialisering av monitor:**  
Før programmet kan begynne å skrive tilbake til PC'en må vi initialisere monitorfunksjonen og fortelle programmet hvilken hastighet dataene skal sende med:  

```
Serial.begin(9600); // Sett kommunikasjons-hastigheten til 9600 tegn i sekundet
```

Denne kommandoen plasseres i setup()-funksjonen.

For å kunne skrive informasjon tilbake til PC'en må det legges inn *skrivekommandoer* i programmet.

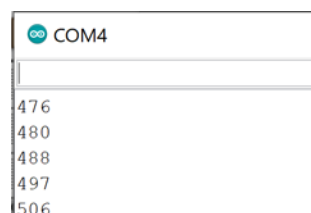
- Skriv måleverdi til monitor:  

```
Serial.println(lysIntensitet); // Skriver lysintensitet til monitoren
```

Plasseres i loopen etter at avlesningen er gjort

- Skriv inn kommandoene og test programmet.

Når du nå trykker på monitor-knappen skal du se verdier som kommer fram på skjermen en i sekundet. Vi ser at verdien varier en del.



- Hold fingeren foran lyssensoren og se hva som skjer med verdien.
- Du kan også legge på lokket på kyvettehuset og se hva som skjer med verdien

#### 4. Testing med tomt og fylt beger

Sett et tomt beger ned i kammeret, sett lokket på og les av verdien på displayet. Du vil sannsynligvis få en verdi i nærheten av 2 – 300. Fullstendig mørke vil være ca. 0 og fullt opplyst ca. 1000. Siden du sender lyset vinkelrett inn i kammeret i forhold til sensoren, (LDR) måler du nå lys som er spredt i vannet og som treffer sensoren. Gjør følgende forsøk:

- Les av verdien med tomt beger og med lokket på: \_\_\_\_\_
- Ta av lokket og les av verdien: \_\_\_\_\_
- Ta ut begeret og sett på lokket og les av verdien: \_\_\_\_\_
- Fyll begeret opp til streken med vann og vent noen minutter til vannet har klarnet og roet seg. Sett på lokket og les av verdien: \_\_\_\_\_

Drøfte resultatene av de fire første målingene: \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

- Drei begeret 90° og 180° rundt, sett på lokket å se om målingene er avhengig av hvordan begeret er orientert:

Hva ble resultatet: \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

#### 5. Gjør målinger av to vannprøve

Vi skal nå teste måleinstrumentet med to vannprøver:

- Fyll to plastkopper med rent vann. Sørg for at de ikke bobler som svever i vannet eller at det er bobler langs innsiden av koppene. Bruk gjerne en pipette til å fjerne luftbobler på innsiden.
- Bruk pipetten til å ha i 5 dråper lettmeik i den ene koppen og rør godt rundt
- Senk koppen med det rene vannet ned i kyvettehuset og legg på lokket.  
Les av måleverdien: \_\_\_\_\_
- Senk koppen med 5 dråper melk ned i kyvettehuset og legg på lokket.  
Les av måleverdien: \_\_\_\_\_



- Forskjellen mellom de to målingene er typisk: \_\_\_\_\_

Legg merke til at gjentatte målinger varierer noe, kanskje inntil  $\pm 5$ , kanskje mer.

Anslå forskjellen i måleverdi mellom de to målingene:

Variasjonen i målingene på hver av koppene skyldes hovedsakelig støy. Vi ønsker å redusere variasjonen ved å gjøre en midling på 500 målinger.

## 6. Fjern målestøy ved hjelp av midling og bruk av *for-loop*.

Siden måling og addisjon av verdier går særdeles fort med en mikrokontroller vil vi knapt merke det dersom vi midler over f.eks. 500 målinger før vi går videre i programmet. Til det bruker vi en *for-loop*. Dvs. en løkke som gjentas et visst antall ganger. I vårt tilfelle må denne løkken inneholde avlesning og akkumulering av måleverdier på inngang A0.

```
      Deklarasjon og nullstill tellevariabel
      ↓
      Betingelse for å utføre innholdet mellom klammene
      ↓
for(int i = 0; i < 500; i++) ← Øke tellevariabelen, en for hver gang
{
  // Innhold som skal gjentas
}
```

Mellom klammeparentesene gjøres en måling og verdien akkumuleres:

```
sumIntensitet = lysIntensitet + analogRead(A0);
// Avleser analog inngang A0
```

(Merk at her har vi laget oss en ny variabel med et navn som gir mening: *sumIntensitet*)

Dernest gjentas dette forløpet 500 ganger etter hverandre. Tellevariabelen økes med 1 (*i++*) for hver runde og stopper ikke før den er kommet til 499 (*i < 500*) som er den siste gangen målingen utføres.

Etter at alle 500 verdiene er akkumulert (*i = 0* til 499), deler vi den akkumulerte verdien på antallet målinger for å bestemme et gjennomsnitt:

```
midletIntensitet = sumIntensitet / 500;
// Midle ved å dele på antallet akkumulerte verdier
```

Man må dessuten huske på å sette *sumIntensitet* lik null før man kjører en ny runde med målinger og akkumulering.

```
sumIntensitet = 0; // Sett U_DT lik null slik at en ikke tar med seg verdier fra tidligere
```

**NB!** Husk at avlesningen av porten og akkumuleringen må gjøres inne i *for-loop*en. Mens midlingen skjer rett utenfor. Lyset kan slås på før *for-loop*en og slukkes etter.

**NB!** Videre må de to nye variablene deklarerer før *setup()*-funksjonen:

```
float sumIntensitet = 0; // Variabelen som holder den akkumulerte verdien
float midletIntensitet = 0; // Variabelen som holder middelverdien
```

Vi deklarerer disse verdiene som *float*, dvs. de vil bli behandlet som desimaltall som tillater store verdier. Vi setter startverdien til begge lik 0.

**NB!** Det kan være lurt å legge inn et delay(100) etter at lysdioden er slått på

- Skriv programmet, kompiler, overfør til Arduino'en og test ut.

### 7. Gjør målinger på to vannprøve med midling

Vi skal nå teste måleinstrumentet med våre to vannprøver med midling av 500 målinger:

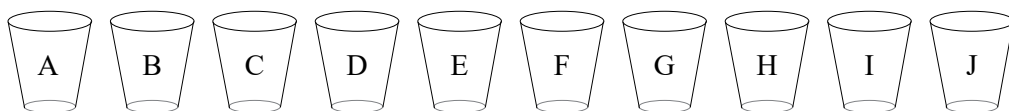
- Bruk de samme vannprøvene som tidligere:
- Senk koppen med det rene vannet ned i kyvettehuset og legg på lokket. La målingene stabilisere seg i ca. 30 sek. og les av måleverdien: \_\_\_\_\_
- Senk koppen med 5 dråper melk ned i kyvettehuset og legg på lokket. La målingene stabilisere seg i ca. 30 sek. og les av måleverdien: \_\_\_\_\_

Ble målingene mer stabile? \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

### 2.4.2 Gjør målinger på 10 beger og sett dem i rett rekkefølge

#### Oppdrag 4 – Bestem rekkefølgen til begrene

Det er satt frem 10 plastbeget merket A til J. I tillegg til 1,5dL vann inneholder disse fra 1 til 10 dråper melk.



- Bruk instrumentet som *turbidimeter* og bestem rekkefølgen til begrene slik at begeret med én dråpe står lengst til venstre og det med ti står lengst til høyre. Før måleverdiene inn i tabellen:

**Tabell 2.1 Målinger av turbiditet**

Målinger	A	B	C	D	E	F	G	H	I	J
Verdi										
Rekkefølge										



Tegn grafene for turbiditet med antall dråper langs x-aksen og de avleste måleverdiene langs y-aksen. Skaler aksene slik at de passer til måleserien.


0    1    2    3    4    5    6    7    8    9    10    Antall dråper

- Legg inn en best tilpasset kurve og bestem følsomheten til instrumentet, dvs. endring i måleverdi som funksjon av antall dråper melk i oppløsningen for turbidimeteret.

Foreslå mulige forbedringer av måleinstrumentet: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**NB!** Vær oppmerksom og unngå å komme bort i lysdioden eller lyssensoren når begeret settes ned i kyvetten slik at betingelsene ikke endres fra måling til måling. Det er også viktig at begeret settes ned omtrent på samme måte for hver måling.

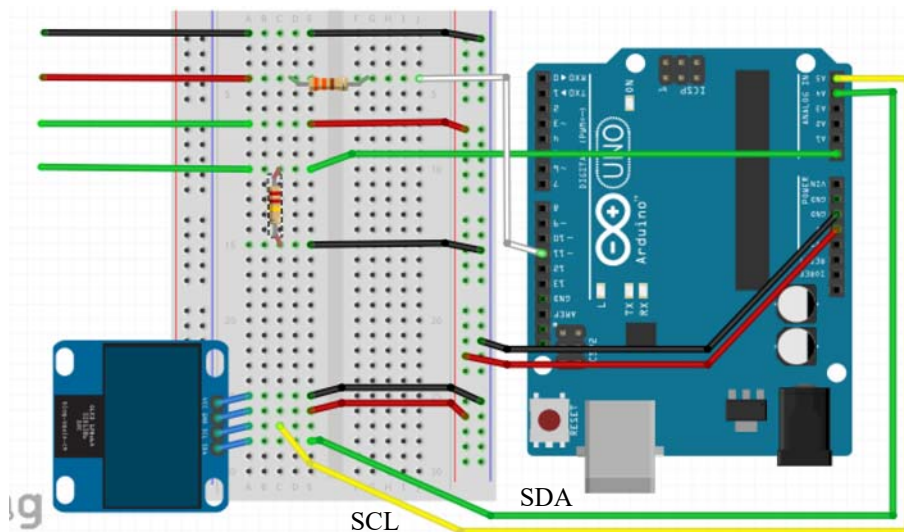
### 2.4.3 Koble opp et eksternt display

Så langt har vi brukt monitoren på PC'en for å vise resultatet. Vi ønsker nå å vise måleresultatet på et lite display. Dette er veien å gå dersom man ønsker å lage et mobilt måleinstrument som er uavhengig av PC'en

Koble Arduino-kortet fra PC'en før du kobler opp displayet.

## 1. Monter displayet

Begynn med å montere displayet SSD1306 som vist på figuren under. Displayet har fire pinner, to for spenning og jord (GND) og to for seriell kommunikasjon SCL og SDA.



Kontroller en ekstra gang at oppkoblingen er riktig.

## 2. Installer biblioteker

For å kunne kommunisere med displayet (SSD1306), må vi installere to biblioteker. Biblioteker er en samling av kommandoer (funksjoner) som gjør det lettere å bruke komponenter som f.eks. dette displayet. Bibliotekene er ofte komprimert i zip-filer og kan i dette tilfelle lastes ned fra:

<https://www.ntnu.no/skolelab/bla-hefteserie>

under fanen: *Grunnkurs programmering Arduino - (CanSat)*

*Adafruit\_SSD1306-master.zip*

*Adafruit-GFX-Library-master.zip*

Lagre bibliotekene i nedlastingskatalogen eller et annet sted der du finner dem igjen. Programeditoren IDE vil automatisk pakke ut bibliotekene når de installeres.

Bibliotekene installeres på følgende måte:

Åpen Arduino-editoren

Velg: *Skisse*

Velg: *Inkluder bibliotek*

Velg: *Legg til .zip bibliotek* (øverst i nedtrekksmenyen)







Finn biblioteket i katalogen Nedlasting på PC'en (eller der du la det)

Velg: *Open*

... og biblioteket installeres

Gjenta for begge bibliotekene.

### 3. *Kommandoer for skriving til display*

- ***Inkluder bibliotekene i programmet***

Øverst i programmet inkluderer man de to bibliotekene i tillegg til et par til. Dermed får programmet adgang til alle funksjonene som trengs for å skrive til displayet.

// Inkludering av biblioteker

```
#include <SPI.h> // Inkluder bibliotek for kommunikasjon på serie linjer
#include <Wire.h> // Inkluder bibliotek for kommunikasjon via I2C
#include <Adafruit_GFX.h> // Inkluder bibliotek for å skrive til display
#include <Adafruit_SSD1306.h> // Inkluder bibliotek for å skrive til display
```

Disse legges inn helt i starten av programmet før deklarasjon av variabler og setup().

To av bibliotekene er alt inkludert i editoren når den leveres (SPI.h og Wire.h)

- ***Deklarasjon av displayet***

Dernest må vi deklare, dvs. gi displayet et navn og en type, i prinsippet kan vi ha flere displayer med forskjellige navn og adresser.

```
Adafruit_SSD1306 display(4); // Deklarasjon av enheten "display" av typen
Adafruit_SSD1306
```

Dette settes inn sammen med andre deklarasjoner av variabler før setup()-funksjonen.

- ***Sett adressen til displayet***

Displayet har en adresse som programmet må vite om, denne er 3C (60) i heksadesimal kode og er normalt satt av fabrikken.

```
display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // Initialiser display med adresse
0x3C
```

Initialiseringen med adressen skal plasseres i setup()-funksjonen.

Vi skal nå se hvordan vi kan skrive tekst og verdier til displayet:

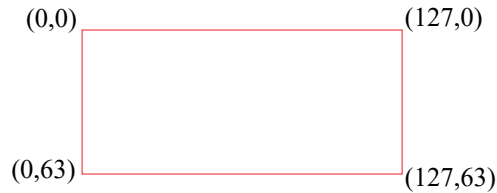
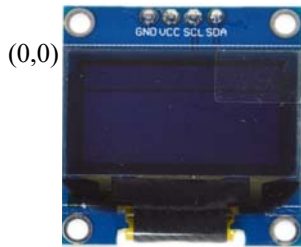
- ***Posisjoner markør, tøm displayet og skriv ut tekst og variabler***

Først må vi tømme displayet før vi begynner å skrive noe nytt. Vi må også bestemme størrelsen på teksten og om vi skal skrive hvit tekst på svart bakgrunn eller omvendt. Disse kommandoene skrives inn i programmet rett før selve utskriften skal skje.

// Klargjør display for utskrift

```
display.clearDisplay(); // Slett informasjon på display
display.setTextSize(1); // Sett størrelse på tekst (2 gir større tekst enn 1)
display.setTextColor(WHITE); // Hvit tekst på mørk bakgrunn
```

Dernest må vi plassere markøren der vi ønsker at teksten skal begynne og skrive:



Koordinater for plassering av markør

// Skriver ut midlere lysintensitet:

```
display.setCursor(0,0); // Plasser markør øverst til venstre (x, y)
display.print("Intensitet: "); // Skriv ut ordet "Tekst: "
display.println(midletIntensitet); // Skriv ut variabelen i og skift linje (display.println)
```

Til slutt sendes all informasjonen over til displayet med kommandoen:

```
display.display(); // Overfør informasjonen til displayet og vis
```

Dersom vi ønsker å skrive på en annen linje på displayet plasser markøren i ønsket posisjon før skriving begynner:

```
display.setCursor(0,10); // Flytt markør til linje to
display.setCursor(0,20); // Flytt markør til linje tre
```

Legg merke til at med skriftstørrelse 1 så er det greit å flytte markøren 10 bildepunkter ned for hvert linjeskift. Merk at koordinatene til markøren tar utgangspunkt i øverste venstre hjørne i karakterruta (origo) som vist på figuren over.

#### 4. Skriv programmet og test ut

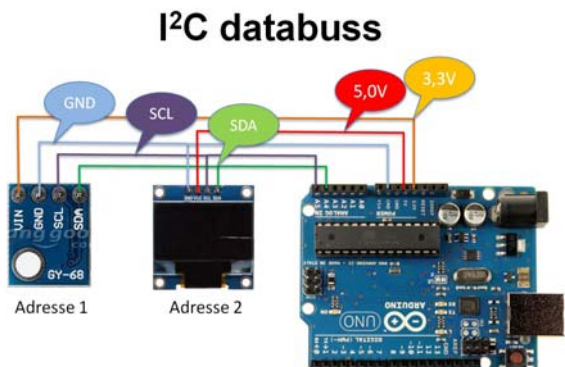
Skriv ut den midlere intensiteten til displayet.

La oss tilslutt i dette avsnittet se litt nærmere på hvordan I<sup>2</sup>C bussen virker.

#### 5. IIC (I<sup>2</sup>C) buss

Dette en måte å kommunisere mellom integrerte kretser (Inter IC, I<sup>2</sup>C) ved hjelp av to ledninger (SDA, SCL). Dersom hver enhet (IC) har forskjellig adresse, kan man adressere informasjonen slik at den kommer til riktig enhet.

På figuren til høyre ser vi at to enheter er koblet på de samme to kommunikasjonslinjene (SDA (data), SCL (klokke)). Dette er displayet vårt og i





dette eksempelet en trykksensor. I tillegg er de forbundet til spenningskildene 5,0V og 3,3V og jord (GND). Det er også vanlig å koble en motstand mellom SDA og SCL og opp til 5V, men det går også ofte bra uten.

I vårt tilfelle har vi bare en enhet koblet til bussen, nemlig displayet. Dersom hver enhet på bussen har forskjellig adresse så kan vi koble mange sensorer og display på de to linjene og vi får en særdeles enkel og oversiktlig oppkobling.

## 6. Lagre programmet

Husk å lagre programmet før du går over til neste oppgave. Husk og bruk norsk tegnsett i navnet – f.eks. Oving 1.

### 2.4.4 Skrive ut antall dråper

I dette avsnittet skal vi bruke *if-kommandoen* for å skrive ut antall dråper i det begeret vi måler på. Dvs. vi må gjøre en omregning slik at måleverdien gjøres om til antall dråper.

#### 1. Finn grenseverdiene for ulike antall grenseverdier

Ta utgangspunkt i tabell 2.1, side 22 og finne grenseverdiene for intensitetsmålingene for ulike antall dråper.

Tabell 1: Grenseverdier for ulike antall dråper melk

# dråper	0	1	2	3	4	5	6	7	8	9	10
Grenseverdier											

Avhengig av den målte verdien skal vi nå skrive ut antall dråper i glasset på displayet. Til dette kan vi bruke *if else* – kommandoen.

#### 2. *if else-kommandoen*

Under er vist et eksempel på hvordan *if else () {}* kan brukes for å velge antall dråper.:

```
if (var > 100 && var < 199)
{
// Skriv antall dråper til display
}
else if (var > 200 && var < 299)
{
// Skriv antall dråper til display
}
:
else if (var > 900 && var < 1000)
{
```

```
// Skriv antall dråper til display
}
```

Her har vi valgt grenseverdiene 100, 200, 300 osv. Dere må finne de riktige grenseverdiene å sette in disse i betingelsene for if-kommandoene.

### 3. *Skriv og test ut programmet*

Ta utgangspunkt i programmet du alt har laget og skriv inn tilstrekkelig antall if else kommandoer slik at du får ut antall dråper i begeret.

Beskriv hvilke utfordringer dere møtte: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Forslag til løsning på utfordringer: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

## 2.5 Kalibrering av måleinstrumentet

Som det framgår av vedlegg C, side 37 så finnes det en måleenhet for turbiditet som kalles NTU (Nephelometric Turbidity Unit).

For å kalibrere instrumentet vårt opp mot denne enheten må vi bruke en kalibrert oppløsning eller et profesjonelt instrument som er kalibrert. Vi velger å gjøre det siste, vi bruker Pasco's turbidimeter som vi kobler til en PC. Vedlegg A, side 32 beskriver kalibrering og bruk av Pasco turbidimeter. Andre tilsvarende turbidimeter kan selvfølgelig også brukes.

### 1. *Kalibrering av hjemmelaget turbidimeter*

Om vi ønsker kan vi bruke det kalibrerte instrumentet til å kalibrere vårt hjemmelagede turbidimeter:

Fyll væsken fra noen utvalgte begre med melkedråper over i den tomme kyvetten som følger med Pasco-instrumentet vi velger begrene med 1 dråpe, 5 dråper og 10 dråper:

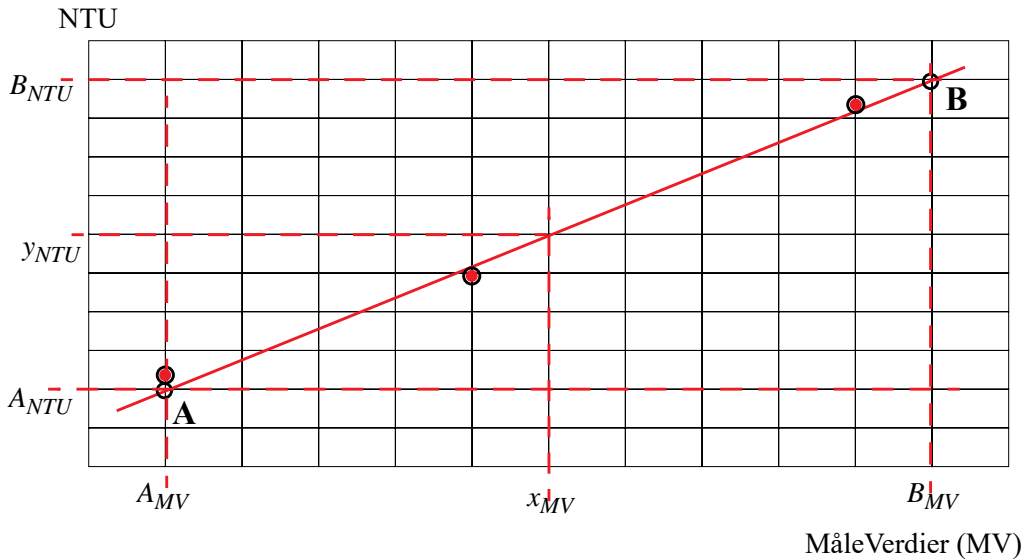
- Fyll den tomme kyvetten med blandingen av 1 dråpe melk og mål NTU-verdien  
1 dråpe melk gir: \_\_\_\_\_ NTC. Tøm prøven tilbake i riktig beger og rengjør kyvetten.
- Fyll den tomme kyvetten med blandingen av 5 dråper melk og mål NTU-verdien  
5 dråpe melk gir: \_\_\_\_\_ NTC. Tøm prøven tilbake i riktig beger og rengjør kyvetten.
- Fyll den tomme kyvetten med blandingen av 10 dråper melk og mål NTU-verdien  
10 dråpe melk gir: \_\_\_\_\_ NTC. Tøm prøven tilbake i riktig beger. og rengjør kyvetten

Bruk disse tre målte verdiene til å justere måleverdiene for vårt hjemmelagede turbidimeter.



## 2. Linearisering:

Legg en best tilpasset lineær kurve gjennom målepunktene (bruk gjerne øyemål), og gjør en omregning fra måleverdiene (MV) til standardverdiene NTU ved å velge to punkter langs linjen (A og B).



Vi kan da sette opp topunktsligningen:

$$\frac{(y_{NTU} - A_{NTU})}{(x_{MV} - A_{MV})} = \frac{(B_{NTU} - A_{NTU})}{(B_{MV} - A_{MV})} \quad (2.1)$$

Denne kan vi så snu om på slik at vi får:

$$y_{NTU} = \frac{(B_{NTU} - A_{NTU})}{(B_{MV} - A_{MV})}(x_{MV} - A_{MV}) + A_{NTU} \quad (2.2)$$

Dersom vi har bestemt de to kalibreringspunktene ( $A_{MV}$   $A_{NTU}$ ) og ( $B_{MV}$   $B_{NTU}$ ) og leser av måleverdien  $x_{MV}$  så kan vi lett beregne målepunktets NTU-verdi lik  $y_{NTU}$ .

- Legg inn formelen og kalibreringsverdiene og beregn NTU-verdien

Med dette ligningssettet inkludert skulle vårt hjemmelagde turbidimeter kunne kalibreres og være klart til bruk.

### 3. *Skriv programmet og test ut*

Husk å deklarer de nye variablene, det kan være lurt å deklarerere dem som float, dvs. desimalverdier:

```
float A_MV = 0; // Her legges kalibreringsverdiene fra diagrammet inn
float B_MV = 0; // Her legges kalibreringsverdiene fra diagrammet inn
float A_NTU = 0; // Her legges kalibreringsverdiene fra diagrammet inn
float B_NTU = 0; // Her legges kalibreringsverdiene fra diagrammet inn

float x_MV = 0; // Avlest måleverdi
float y_NTU = 0; // Beregnet kalibrert NTU-verdi
```

Sammenlign verdier på det profesjonelle måleinstrumentet og det hjemmelagde instrumentet.

Beskriv hva du ser: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_



---

### 3 Referanser

[1]

## Vedlegg A Blokkdiagram for koden

Til høyre er vist et blokkdiagram over programmet slik at det skal være lettere å forstå koden.

### Start

Programmet starter på nytt hver gang strømmen slås på eller man trykker på RESET-knappen på kortet. Den starter også på nytt når man slår på monitoren i program-editoren.

### Biblioteker og deklarasjoner

De fleste av de kommandoene vi bruker når vi koder er funksjoner som kalles opp fra biblioteker. For å bruke display og kommunikasjon på linjene mellom kretsene ( $I^2C$ ), så kreves skreddersydde funksjoner, disse er samlet i biblioteker som må inkluderes i starten av programmet.

Variabler som skal kunne brukes i alle funksjoner (globale variable) må deklarerer utenfor funksjonene.

Disse deklarerer ofte her i starten.

### Setup () – funksjonen

Setup() - funksjonen kjøres bare en gang, hver gang programmet starter. Her ligger initialiseringer som ikke trengs og gjentas mens programmet kjører.

Dette gjelder initialisering av kommunikasjon mellom kortet og monitoren i PC'en (*Serial.begin(9600);*).

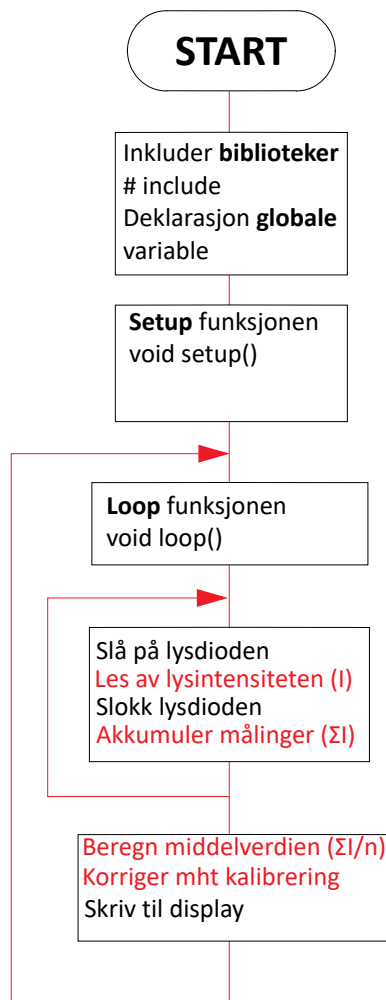
Videre gjelder det initialisering av displayet (*display.begin()*).

### Loop () – funksjonen

Denne gjentar seg så lenge kortet har spenning. Når en kommer til slutten, så begynner programmet fra toppen av loop()-funksjonen igjen. For hver runde avleses lysintensiteten. Hvor ofte det gjøres målinger er hovedsakelig bestemt av hvor lang tid det tar å gå gjennom loopen. Kommandoen *delay(x)*; på slutten av loopen legger inn en pause på x millisekunder. Det er hovedsakelig denne som bestemmer hvor ofte det tas målinger.

### Avlesing av lysintensitet

Intensiteten til lyset som reflekteres av partiklene i væsken registreres ved hjelp av en lysfølsom motstand (LDR). Den varierende motstandsverdien omformes til en spenning før den leses av på en av de analoge inngangene som en spenning mellom 0 – 5,0 V.







---

### *Midling*

Siden målingene er befengt med støy vil det være lurt å gjøre mange målinger som summeres for så å beregne en middelværdi. Dette gjør at tilfeldig støy kanselleres. Dette kan vi lett gjøre med en *for-loop*.

### *Display resultater*

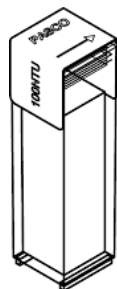
De neste blokkene inneholder kommandoer som skriver data til displayet. Dette er ulike kommandoer av typen, *display.print* og *display.println o.l.*

## Vedlegg B Testing og måling med turbidimeteret

Dere skal nå bruke et “profesjonelt” måleinstrument for å kalibrere instrumentet brukt som turbidimeter slik at måleverdiene kommer ut i en standardisert enhet.

Turbiditet måles i NTU (Nephelometric Turbidity Unit). Siden vi ikke har noen kalibrert oppløsning som kan brukes direkte til vårt turbidimeter gjør vi en sammenligning mellom vårt hjemmelagde turbidimeter og et profesjonelt instrument, f.eks. ved hjelp av en *Turbidity sensor fra Vernier*, eller *Pasco's SE-3215*. I vårt tilfelle velger vi Pasco's SE-3215.

Til tross for at Instrumentet er trådløst så velger vi å koble det til PC'en med en USB-kabel.



Kalibrert oppløsning  
100NTU

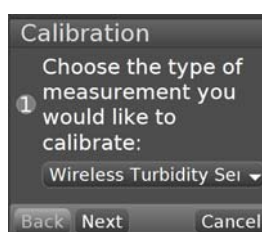
Med instrumentet følger en standard på 100 NTU som passer perfekt i kammeret. I tillegg har vi tomme kyvetthus. Legg merke til at vi bruker de blanke som følger med.

Gjør følgende:

1. Koble instrumentet til PC'ens USB-port
2. Klikk på programikonet og programmet starter.

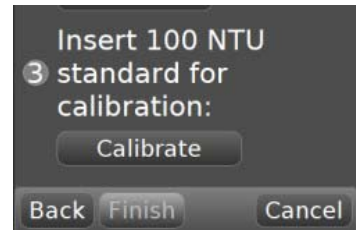


3. Når programmet starter og instrumentet er plugget inn, får man spørsmål om man ønsker å kalibrere instrumentet (under til venstre), velg “Calibrate”. Velg deretter “Wireless Turbidity Sensor” fra menyen (under til høyre) og trykk deretter “Next”.

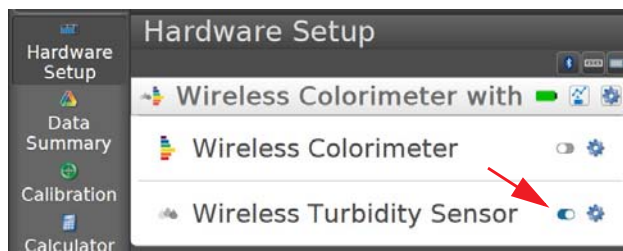




4. Kalibreringsmenyen kommer opp og man blir bedt om å sette en helt tom og *blank* kyvette ned i kammeret (funder til venstre). Tørk godt av den før den settes ned i kammeret, sett på lokket, trykk "Calibrate" og deretter "Next". Ta så ut den tomme kyvetten.



5. Du blir nå bedt om å sette inn en 100 NTU standard (over til høyre). Ta kyvetten med 100 NTU standarden, vend den om fir-fem ganger (ikke rist) og tørk av fingeravtrykk på utsiden. Sett den inn i kammeret og trykk "Calibrate" og deretter "Finish". Instrumentet skal nå være kalibrert.
6. Fra menyen helt til venstre velg "Hardware Setup". Du får da opp valget mellom "Wireless Colorimeter" og "Wireless Turbidity Sensor". Bruk bryteren til høyre for å velge "Wireless Turbidity Sensor". Slå samtidig av "Wireless Colorimeter", som vist på figuren under.



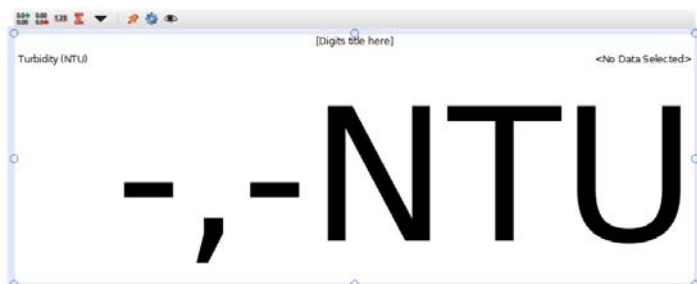
At du har valgt riktig kan du se midt på menylinjen nederst på skjermen (figuren under). Her kan du også stille inn hvor ofte målingene skal tas. I vårt tilfelle tas det en måling hvert sekund (1 Hz).



7. Det neste vi må gjøre er å velge hvilket display vi ønsker å bruke. Siden vi kun skal måle enkeltverdier kan vi velge displayet "Digits" fra menyen til høyre (under til venstre). Bruk musa og dra "Digits-ikonet" ut på arbeidsflata slik at du får et bilde omtrent som vist til høyre på figuren under.



8. Displayet skal så knyttes til et måleinstrument. Dette gjør vi ved å trykke "Select Measurement". Velg "Turbidimeter" fra nedtrekksmeny. Dermed har vi knyttet displayet til turbidimeteret og displayet får benevnningen NTU som vist på figuren under.



9. Målingen startes ved å velge "Record" til venstre på menylinja nederst (til venstre under).



10. Dermed skulle målingen begynne og displayet viser den målte turbiditeten for den ukjente væsken.



Instrumentet skal nå være kalibrert og kan brukes til å måle turbiditet.



## Vedlegg C Turbiditet

Stoffet som presenteres her er en forkortet utgave av [tu.no/artikler/vaeskeanalyse-turbiditet/218521](https://tu.no/artikler/vaeskeanalyse-turbiditet/218521) og er forfattet av Knut Erik Boland, Inline Prosess<sup>1</sup>.

Turbiditetsmåling brukes ofte i forbindelse med kontroll på inn- og utløp på vannverk og rensningsanlegg, grunnvannsmålinger fra steinbrudd og tunnelbygging o.l.

Turbiditet er ofte en kvalitetsparameter, hvor det ferdige produktet ofte skal inneholde minst mulig partikler.

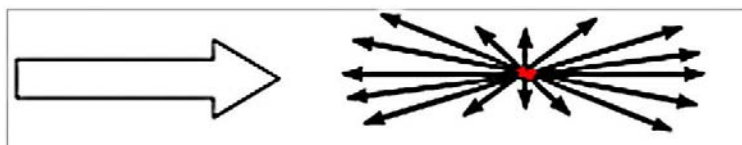
### Partikkelstørrelse og form avgjør

Turbiditet er fenomenet som oppstår når en bestemt mengde lys passerer gjennom en væskebasert prøve og dette lyset blir reflektert i ulike retninger av de partiklene som ikke er oppløst i væsken.

Refleksjonsmønsteret er bestemt av både størrelse og form på partiklene i væsken, og man kan dele det opp i tre kategorier som går på størrelse. I dag benyttes ofte en lysdiode (LED) som lyskilde i et turbidimeter, noe som gir lang levetid (typisk ti år), har helt stabil lysenergi og krever liten effekt. Typisk bølgelengde på en LED/NIR-lampe (nær infrarød) er 0,7 til 5 micron (micron = en milliondel av en).

### Får ulik lysspredning

Partikler som er mindre enn en tidel av bølgelengden på lyset som benyttes (under 0,5 micron), gir en symmetrisk spredning av lyset ut fra partikkelen. Det blir like mye lys reflektert som mengden som trenger gjennom væsken.



Figur 3.1 De minste partiklene (under 0,5 micron) gir en symmetrisk spredning av lyset.

---

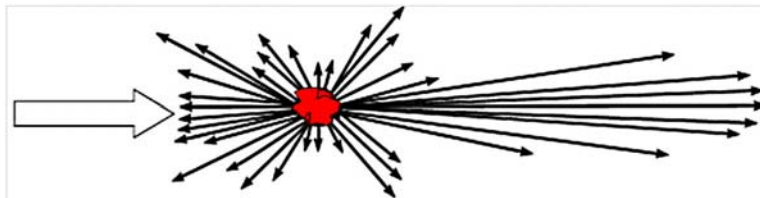
1. Artikkelen bygger også på: [https://www.isanorcal.org/download/tech2007\\_presentations/turbidity.pdf](https://www.isanorcal.org/download/tech2007_presentations/turbidity.pdf) (by John Daly, ISA)

Partikler som er rundt en firedel av bølglengden på lyset som benyttes (ca. 1 micron) gir en økning av lysmengden som slipper gjennom væsken. Det vil si at over halvparten av lyset når fram til mottakeren.



Figur 3.2 Mellomstore partikler (rundt 1 micron) slipper gjennom mer lys enn det som blir reflektert mot lyskilden.

Partikler som er større enn bølglengden på lyset som benyttes (over 1 micron), gir en ytterligere økning i lyset som slipper gjennom. I tillegg oppstår også uregelmessige økninger av lysrefleksjoner i alle retninger.



Figur 3.3 De største partiklene (over 1 micron) slipper gjennom mer lys enn det som blir reflektert mot lyskilden, og gir i tillegg uregelmessige refleksjoner i alle retninger.

### Partikkeldefinisjon

En partikkel kan være hva som helst som danner en uensartet overflate (heterogen) og som danner en brytningsindeks av lys som er ulik brytningsindeksen for mediet den befinner seg i.

Eksempler kan være:

- Faste partikler i væske, som også kalles en *suspensjon*.
- Oljedråper i vann, eller vanddråper i olje, som kalles en *emulsjon*.
- Gassbobler i væske eller luft, for eksempel *skum*.
- Støvpartikler i luft, som oppfattes som *røyk*.
- Dråpepartikler i luft, som kan oppfattes som *tåke*.

### Påvirker refleksjonen

Det er mer enn partikkelstørrelse og form som påvirker lysrefleksjonen som f.eks.:

- Hvilken *type partikkel* det gjelder (som nevnt ovenfor), og dens evne til å absorbere og reflektere lys.
- *Konsentrasjonen* av partikler, hvor mange, vil naturlig nok påvirke mengden reflektert lys.



- *Bølgelengden* på det utsendte lyset vil påvirke absorpsjons- og refleksjonsegenskapene til partiklene.
- *Vinkelen*, med plassering av detektoren eller fotocellen i forhold til lampekilden, som skal fange opp det reflekterte lyset.

### Lysdifferanse

Når man måler på oppløste partikler i en væskeprøve, måler man konsentrasjonen av partiklene, eller suspendert stoff (tørrestoff). Måleenheten er vanligvis mg/l eller ppm (parts per million). Absorbans av utsendt lys er differansen mellom utsendt lys og det som blir absorbert i væskeprøven.

Turbiditetsmåling måler mengden av reflektert lys fra de partiklene som ikke er oppløst i prøven. Detektoren eller fotocellen som fanger opp det reflekterte lyset er ofte plassert 90°, 120° eller 180° i forhold til lampekilden.

Det finnes sensorer som benytter tilbakereflektert lys (på engelsk kalt "back scattering"), og sensorer som har mer enn én detektor for nettopp å fange opp både store og små partikler.

Turbiditeten øker i takt med mengden lys som væsken absorberer. En typisk måler består av en lyskilde, et målekammer og en mottaker som kalkulerer signalene og angir måleresultatet.

### Måleenheten NTU (

Det er mange ulike måter å angi turbiditet NTU er en av dem:

NTU (Nephelometric Turbidity Unit) Denne enheten er basert på at lyskilden og lysmåleren står plassert 90° på hverandre. Standarden som man ofte tar utgangspunkt i er en oppløsning av Formasin-løsning  $C_2H_4N_2$ , som har en turbiditet lik 4000 NTU (FTU - Formazine Turbidity Unit hvor 1 FTU = 1 NTU).

## C.1 Krav til turbiditet i drikkevann

Krav til turbiditet i drikkevann:

**Tabell 3.1 Noen krav til turbiditet av drikkevann, ut fra anlegg**

Standard	Anbefalt	Maks.
USA	0,3 NTU	1 NTU
Europa		4 NTU
Norge (Mattilsynet) <sup>a</sup>		1 NTU
Verdens helseorganisasjon	< 1 NTU	5 NTU

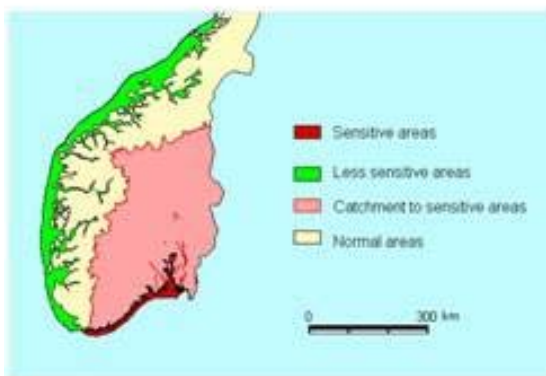
a. <https://lovdata.no/dokument/SF/forskrift/2016-12-22-1868>

## C.2 Krav til turbiditet i avløpsvann

I utgangspunktet stilles det ikke krav til turbiditet i avløpsvann, men siden redusert turbiditet er knyttet til fjerning av andre stoffer vil det være naturlig at denne reduseres med fjerning av de uønskede stoffene. Derfor er det satt krav til turbiditet.

Som det framgår av kartet til høyre så varierer kravene til rensing i Norge. Dette skyldes at noen områder er mer sårbare for forurensning. Som vi ser så er kravene i Trøndelag ikke så strenge som på Østlandet og langs sørlandskysten øst for Kristiansand.

### Sensitivity of the recipient



Man er spesielt oppmerksomme på følgende stoffer:

- **BOF<sub>5</sub>**: Biologisk Oksygen Forbruk  
Mål for biologisk materialer i avløpsvannet som forbruker oksygen ved nedbrytning
- **KOF**: Kjemisk Oksygen Forbruk  
Mål for kjemisk materialer i avløpsvannet som forbruker oksygen ved nedbrytning
- **ToT-P**: Totalt innhold av Fosfor
- **ToT-N**: Totalt innhold av Nitrogen
- **SS**: Suspendert stoff (partikler i vannet som fanges opp av filter med poreåpning < 0,45 mikrometer (0,045mm))

Under er vist et sett med renskrav til avløpsvann. Som nevnt kan kravene kan variere noe fra sted til sted. Vi velger her å holde oss til disse:

- Reduksjon av BOF<sub>5</sub>-innhold 70 – 80%, utslippsgrense lik 25-35 mg pr. liter O<sub>2</sub>
- Reduksjon av KOF-innhold 70 – 80%, utslippsgrense lik 90-120 mg pr. liter O<sub>2</sub>
- Reduksjon av Fosfor-innhold 93 – 94%, utslippsgrense lik 0,25 mg Fosfor pr. liter
- Reduksjon av SS (partikler) på 90 – 92%, med en utslippsgrense på 10 – 25 mg pr. liter

Forholdet mellom NTU og suspenderte faste stoffer er som følger: 1 mg / l (ppm) tilsvarer 3 NTU.

Kravet til rensset avløpsvann blir da:

$$25 \text{ mg / l gir SS} = 75 \text{ NTU}$$





## Vedlegg D Utstysrliste

Utstysrliste for hvert sett

2 stk. drikkebeget, plast, klarplast,	REMA 1000
1 stk. Plastpipette 1.0 mL	VWR/Apotek
1 stk. 220kOhms motstand	ELFA
1 stk. LDR 5mm GL5528 (fotomotstand) <a href="https://www.kultogbillig.no">https://www.kultogbillig.no</a> (skriv LDR i søkefeltet)	Kult og Billig
1 stk. stiftlist (10 pin hvor 6 pin er fjernet) <a href="https://www.kultogbillig.no/index.php?_route_=40-pins-2-54mm-Stiftlist-Header&amp;keyword=stiftlist">https://www.kultogbillig.no/index.php?_route_=40-pins-2-54mm-Stiftlist-Header&amp;keyword=stiftlist</a>	Kult og Billig
4 stk. Test-kabler med alligator-klyper <a href="https://www.kultogbillig.no/index.php?_route_=alligator-test-kabler&amp;keyword=kabler">https://www.kultogbillig.no/index.php?_route_=alligator-test-kabler&amp;keyword=kabler</a>	Kult og Billig
8 cm Svart elektrertape	Biltema/Cals Ohlson/Andre
9,5 cm x Ø 70 mm Pappør	Staples
1 stk. Lokk til pappør, ev. en bok	
1 stk. Arduino sett med koblingsbrett og jumpere eller lignende <a href="https://www.elfa.se/sv/arduino-starter-kit-classroom-pack-arduino-k000007-6p/p/30163026">https://www.elfa.se/sv/arduino-starter-kit-classroom-pack-arduino-k000007-6p/p/30163026</a>	ELFA eller SparkFun

### **Eventuelt også:**

1 stk. OLED display 0.96 Inch I2C IIC Serial 128 x 64 <a href="https://www.kultogbillig.no">https://www.kultogbillig.no</a> (skriv OLED i søkefeltet)	Kult og billig
--	----------------

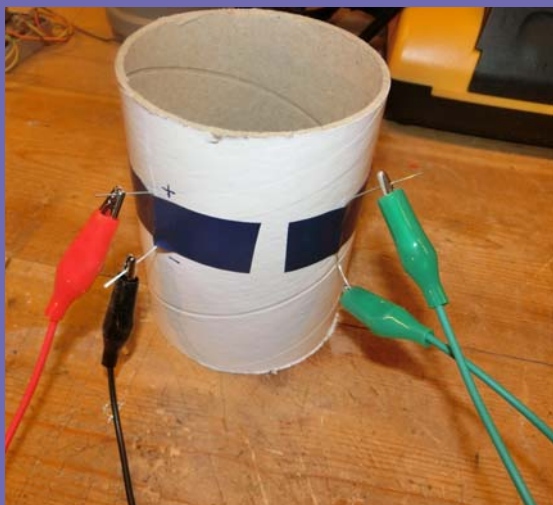
Stiftlist:

Stiftlistene fås i lengder på 40 pinner. For å kunne koble seg til koblingsbrettet med test-kablene med alligator-klemmer kan en benytte en stiftlist hvor noen av pinnene er fjernet slik at avstanden blir større, se figur under. Pinnen kan dras ut ved hjelp av en sideavbiter. Dessuten kan en de skyves gjennom plastholderen slik at lengden blir lik på begge sider.









Heftet ble utarbeidet i forbindelse med fagdag 29. nov. 2019 og beskriver bygging og programmering av et turbidimeter for måling av partikkeltetthet i væsker.

Ønsket var å utvikle et undervisningsopplegg hvor man kunne gjøre målinger og anvende matematikk til å behandle resultatene. Videre skulle oppgaven være slik at elevene selve kunne lage og koble opp måleutstyret samt skrive programmet.

Turbidimeteret egner seg primært til å måle relativ partikkelkonsentrasjon, men burde kunne la seg kalibrere dersom man har kalibrerte oppløsninger eller et kalibrert måleinstrument som f.eks. Pasco turbidimeter.

### ***Nils Kr. Rossing***

Dosent ved Skolelaboratoriet

E-post: [nils.rossing@ntnu.no](mailto:nils.rossing@ntnu.no)

Prosjektleder ved Vitensenteret

E-post: [nkr@vitensenteret.com](mailto:nkr@vitensenteret.com)

Erik Løkketangen

Lærer ved Brundalen skole i Trondheim

E-post:

[erik.lokketangen@ou.trondheim.kommune.no](mailto:erik.lokketangen@ou.trondheim.kommune.no)

NTNU



Trondheim

Institutt for  
fysikk

Skolelaboriet  
for matematikk, naturfag  
og teknologi

Tlf. 73 55 11 43

<https://www.ntnu.no/skolelab>