## Seriemonitoren (Serial Monitor)

**Hva er Seriemonitoren?** Den er et vindu du kan åpne i Arduino IDE på PC’en din, og som gir mulighet for kommunikasjon mellom Arduinoen og PC’en:

* Der kan du kan se data/resultater som programmet på Arduinoen produserer.
* Og du kan styre Arduinoen på enkelte måter ved å skrive tekst i Seriemonitoren.

Seriemonitoren har fått sitt navn fordi kommunikasjonen mellom PC og Arduino-kort er av typen seriell kommunikasjon. Dvs. at informasjonen som på vanlig datamaskin-vis består av bits (nullere og enere: 1001101011001…) sendes serielt – bit for bit i en rekke, eller serie som vi også kan kalle det. Og den er en monitor – derav navnet Seriemonitoren.

**Hvor finner du Seriemonitoren?** For å bruke den må Arduinoen være koblet til datamaskinen med USB-kabelen. Ved å trykke på forstørrelsesglass-knappen som er markert i bildet av Arduino IDE nedenfor åpner du Seriemonitor-vinduet:

|  |  |
| --- | --- |
| C:\Users\andreln\OneDrive\Sommerjobb NTNU\Thora Storm\Laboratieøvinger\Forslag Moduler\Pictures\Arduino Åpne Serial Monitor.png |  |

Arduino IDE Seriemonitoren

**Kommunikasjonen må klargjøres (settes opp)**. For at Seriemonitoren skal fungere må du ha satt opp kommunikasjonen mellom Arduino-kortet og PC’en, og det gjøres i ***void setup*** i programmet ditt med kommandoen ***Serial*.*begin****(9600);*

Og så legger du kanskje merke til at en annen Serial-kommando (***Serial***.***println***) er brukt i ***void loop***, og det er den som produserer teksten du ser vist i Seriemonitoren.

## Feilsøking vha Seriemonitoren

**To typer feil i programmet.** Når du kompilerer/verifiserer et program du nettopp har skrevet, eller gjort endringer i, vil det utføres en syntakssjekk – dvs sjekkes om du har skrevet alle kommandoer grammatikalsk korrekt, om alle semikolon og andre nødvendige tegn er på plass, osv. Bommer du på noe av det så er det en type feil vi kaller syntaksfeil, og du får beskjed om disse i informasjonsvinduet helt nede i Arduino IDE. De rapporterte feilene fikser du så, kjører ny kompilering/verifisering av koden, og holder på med det til du har null syntaksfeil rapportert.

Den andre typen feil, hvor kompilatoren har ingen ting å utsette på koden din, men resultatet av kjøringen av programmet blir ikke som du hadde tenkt, kaller vi logiske feil. Du har da tenkt feil et eller annet sted i programmet ditt, men det er ikke alltid så lett å finne slike feil.

**Feilsøking etter logiske feil**. Her kommer faktisk Seriemonitoren inn som et verktøy du kan ta i bruk. Ved å sette inn visse typer Serial-kommandoer (typisk ***Serial***.***println***) i utvalgte steder i koden din, kan vi få tilbakemelding i Seriemonitoren fra Arduinoen.

Tilbakemeldingene du genererer Serial-kommandoer i programmet, som så skrives til Seriemonitoren, vil kunne si noe om statusen i kjøringen av programmet akkurat der de står. F.eks. så kan én si noe om hva verdien til utvalgte variabler er akkurat der og da, mens en annen kan plasseres et sted for å skrive til Seriemonitoren «nå kom programkjøring hit».

Kort sagt: Seriemonitoren kan brukes slik at vi enkelt kan få tilbakemelding når noe er galt, når vi lurer på om koden vår fungerer slik vi tror eller når vi trenger informasjon om status på variabler.

Denne linjen med kode kan puttes hvor som helst i koden vår og bør brukes ofte.

## Eksempel på feilsøking med Seriemonitor

I eksempelet under har vi en if-løkke som kjører dersom variabelen x har verdien 2. Tenker vi oss at x fikk startverdien 1 da vi deklarerte den, vil if-løkka aldri kunne kjøre uten at x blir endret til 2 et sted i programmet.

**Kjøres if-løkka?** Anta at programmet ditt (altså i void loop) endrer verdien på x flere ganger i løpet av kjøringen, og at du er usikker på om den noen gang blir endret til 2 slik at if-løkka kan kjøre. Ved å bruke Seriemonitoren og et par Serial-kommandoer kan du enkelt å finne ut om programmet kommer seg inn i if-løkka i det hele tatt:

|  |  |
| --- | --- |
|  |  |

**Hvilken verdi har x?** Så kan det være at du i feilsøkingen din også ønsker å finne ut hvilken verdi x faktisk har, fordi programkoden kan være så vidt komplisert at du ikke uten videre kan resonnere deg fram til det.

Du kjenner allerede til hvordan du legger inn en kommentarlinje i programmet ditt med to skråstreker: //

En annen bruk av kommentar-skråstrekene er å «kommentere bort kodelinjer». Dvs at ved å sette inn // i starten av en kodelinje, og kompilere/verifisere programmet på nytt, unntar du kodelinjen fra kjøring. Dette benyttes ofte når vi ved feilsøking legger inn Serial-kommandoer rundt omkring i programmet vårt. Vi «aktiverer» en eller flere Serial-kommandoer, og kanskje «deaktiverer» (kommenterer bort) andre, mens vi søker etter feil.

I eksempelet under har vi kommentert bort de to linjene med Serial-kommandoer fra forrige eksempel, og lagt inn en ny Serial.println. Denne gangen ber vi imidlertid ikke om å få skrevet tekst til Seriemonitoren, men i stedet verdien til x.

Legg merke til forskjellen på syntaks når vi skriver tekst og en variabel-verdi til Seriemonitor.

Etter disse endringene kan vi nå se i Seriemonitoren at x alltid har verdien 1, og det vil være en viktig pekepinn i feilsøkingen din.

Slik kan du legge inn Serial-kommandoer flere steder i komplekse programmer du har skrevet, og benytte dem for å finne ut av logiske feil.

|  |  |
| --- | --- |
|  |  |

## Serial-kommandoene Serial.print og Serial.println

Serial.print er en funksjon som skriver ut tekst, bokstaver eller variabler uten linjeskift.

|  |  |
| --- | --- |
|  |  |

## Serial.println

Uttales «serial printline» er en funksjon som skriver ut tekst, bokstaver eller variabler med linjeskift. Denne kommandoen bruker vi oftest til å skrive ut informasjon fra Arduinoen.

|  |  |
| --- | --- |
|  |  |