

# in1060: Bruksorientert design



---

## ukeoppgaver i Arduino - del 1 vår 2020

### Innhold

<b>1. Elektrisitet</b>	2
1.1. Kretsbygging	2
1.2. Komponenter	2
1.3. Dårlige kretser	3
1.4. Analoge og Digitale signaler	4
1.5. Likestrøm og vekselstrøm	4
<b>2. Kodeoppgaver</b>	4
2.1. Setup() og loop()	4
2.2. Noen viktige funksjoner	4
2.3. Sende signaler	5
2.4. Loop og blink	5
2.5. LED, knapp og Arduino-kode	6
a) Uten programmering	6
b) Med programmering	6
2.6. SOS signalsystem	7
2.7. Blink level 2	8
<b>3. Nøtt</b>	9

# 1. Elektrisitet

En mye brukt analogi for å forklare elektrisitet er hvordan vann oppfører seg i en vannkran. Analogien kan en blant annet lese om [her](#). Bruk denne analogien for å forklare begrepene strøm, spenning og motstand.





1. Hvordan representeres strøm?
2. Hvordan representeres motstand?
3. Hvordan representeres spenning?
4. Hvordan henger strøm, spenning og motstand sammen?

## 1.1. Kretsbygging

I Arduinosettet hører det med et breadboard. Hvorfor er dette et nyttig verktøy når dere prototyper og bygger kretser? Ikke skriv mer enn et par setninger.

## 1.2. Komponenter

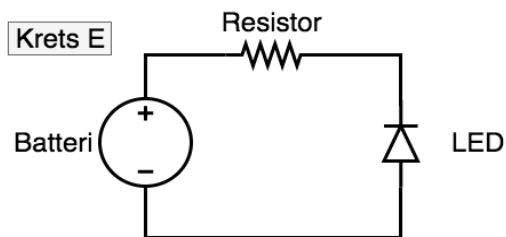
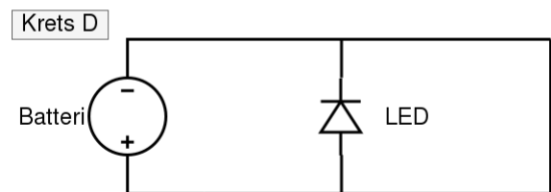
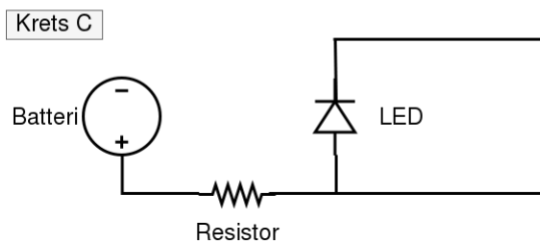
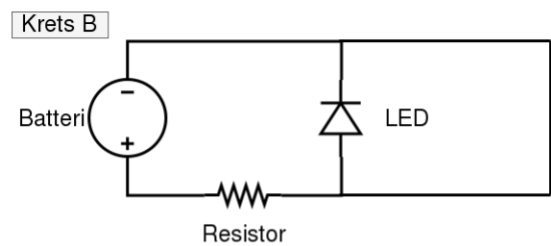
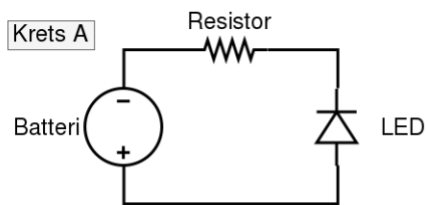
Før en forstår kretser er det nødvendig å kjenne til noen komponenter. Ikke skriv mer enn et par setninger.

1. Hva er en LED?
2. Hva er et batteri?
3. Hva er en resistor?
4. Hva er en bryter?
5. Kombiner komponentene LED() , bryter() , batteri() , og resistor () i en krets som kan skru av og på LED-pæren. Her trenger du ikke spesifisere volt, strøm, Ohm, osv.

### 1.3. Dårlige kretser

Kortslutninger og åpne kretser er typiske problemer dere kan møte i arbeid med Arduino. Særlig om en endrer på en krets mens elektrisitet fortsatt er koblet til.

1. Hva er en kortslutning? Ikke skriv mer enn et par setninger.
2. Hva er en åpen krets? Ikke skriv mer enn et par setninger.
3. Vil LED-pæren lyse i følgende kretser? Begrunn svaret kort.



## 1.4. Analoge og Digitale signaler

Komponenter krever forskjellig elektrisitet for å operere/fungere. En må for eksempel ta hensyn til forskjellen på analoge og digitale signaler.

1. Forklar forskjellen på analoge og digitale signaler. Ikke skriv mer enn et par setninger.
2. Gi et eksempel på noen komponenter som sender/mottar/kan brukes med digitale signaler, og gi et eksempel på noen komponenter som sender/mottar/kan brukes med analoge signaler.

## 1.5. Likestrøm og vekselstrøm

Forskjellen på AC/DC (likestrøm og vekselstrøm) er også viktig å vite.

1. Forklar forskjellen på likestrøm og vekselstrøm. Ikke skriv mer enn et par setninger.
2. Kommer det likestrøm eller vekselstrøm fra Arduino sin "5V" og "3.3V" port?

# 2. Kodeoppgaver

## 2.1. Setup() og loop()

Når vi programmerer en Arduino må vi forholde oss til to viktige funksjoner: `setup()` og `loop()`. Ikke skriv mer enn et par setninger.

1. Når kjører disse funksjonene, og hvor mange ganger kjører de?
2. Hvilken del av koden er hensiktsmessig å skrive i `setup()` funksjonen?
3. Hvilken del av koden er det hensiktsmessig å skrive i `loop()` funksjonen?

## 2.2. Noen viktige funksjoner

1. Hva gjør funksjonen `digitalRead()`?
2. Hva gjør funksjonen `digitalWrite()`?
3. Hva gjør funksjonen `delay()`?
4. Hva gjør funksjonen `pinMode()`?

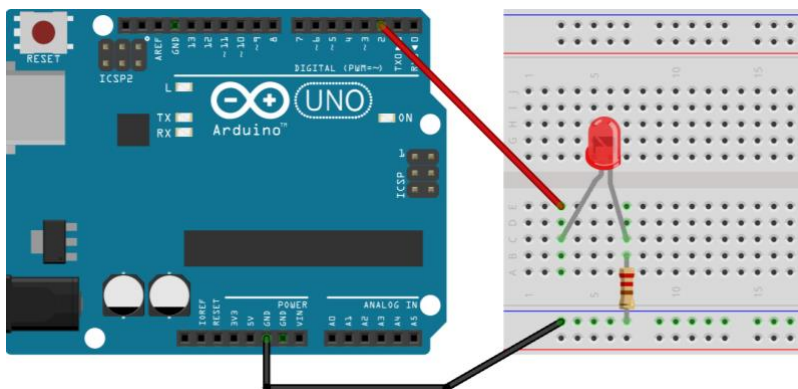
## 2.3. Sende signaler

Arduino-Uno beskytter alle sine pins fra kortslutning ved å “stenge de av” ved oppstart. Dette er veldig nyttig ettersom kortslutninger ville skadet Arduinoen. Men dette gjør også at vi må sette pins i riktig modus før vi kan bruke de til diverse funksjoner.

- Se for deg at du ønsker å sende strøm til en LED som er koblet til port 2 på Arduinoen (slik som i kretsen nedenfor). Hva må du skrive i koden for å få til dette?

## 2.4. Loop og blink

Anta at port 2 er koblet til en LED slik som i figuren nedenfor.



Hvordan vil lampen oppføre seg når disse kodesnuttene kjører? Begrunn svaret kort.

1.

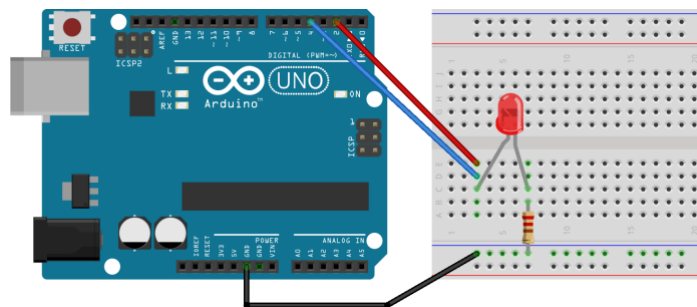
```
7 void loop() {
8     digitalWrite(2, HIGH);
9     delay(1000);
10    digitalWrite(2, LOW);
11 }
```

2.

```
7 void loop() {
8     digitalWrite(2, HIGH);
9     delay(1000);
10    digitalWrite(2, LOW);
11    delay(1000);
12 }
```

3. Følgende kode, med følgende krets

```
5 void setup() {  
6   pinMode(2, OUTPUT);  
7   pinMode(4, INPUT);  
8 }  
9  
10 void loop() {  
11   digitalWrite(2, !digitalRead(4));  
12   delay(1000);  
13 }
```

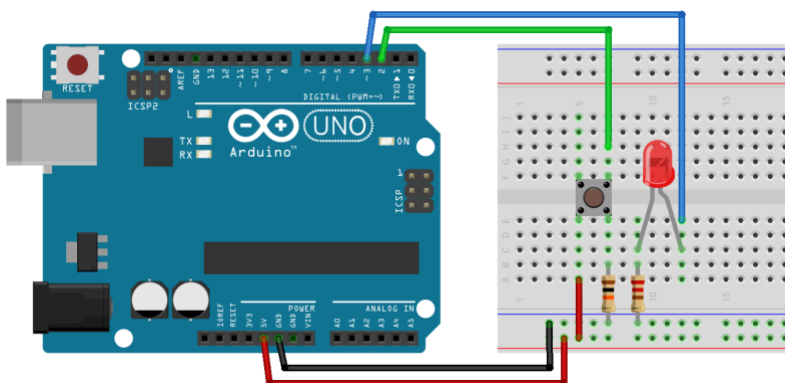


## 2.5. LED, knapp og Arduino-kode

a) Uten programmering

Koble opp en krets som gjør at en LED lyser når du trykker på en knapp. Her skal du ikke skrive kode, du skal kun benytte Arduino som strømkilde.

b) Med programmering



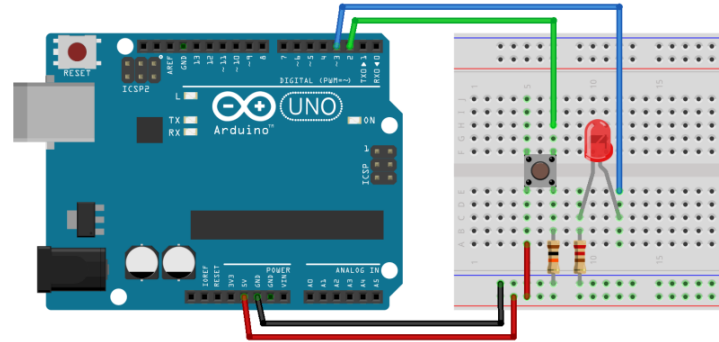
For å finne ut om en knapp er trykket ned eller ikke, kan vi “lytte” på den med en av de digitale portene på Arduinoen. Prøv å koble til en knapp og en LED til et breadboard. Koble deretter knappen til port 2 og LED-pæren til port 3 på Arduinoen som i kretsen ovenfor.

- Skriv kode som gjør at LED-pæren lyser når knappen trykkes.

## 2.6. SOS signalsystem

Ta utgangspunkt i forrige oppgave og lag et SOS signalsystem. Når knappen trykkes ned skal LED-pæren blinke i et SOS-mønster (tre korte blink, tre lange blink, og så tre korte blink). Gjør dette på følgende måte:

1. Koble opp en krets som i forrige oppgave (den samme du ser til høyre her).
2. Bruk funksjonen “pinMode()” til å sette opp en pin som digital output, og en pin som digital input i funksjonen “setup()”.
3. I funksjonen “loop()”: Skriv kode som lytter til om knappen er trykket ned eller ikke. Gjør dette gjennom den digitale input pin-en du satte opp i “setup()” funksjonen.
4. For å gjøre programmet ditt enklere å lese, lag en ny void funksjon med navn “sos()”. Når knappen trykkes skal funksjonen “sos()” kalles.
5. Skriv kode som gjør at LED-pæren blinker i mønsteret til en sos-beskjed i funksjonen “sos()”.



Hint: Her kan det lønne seg å skrive flere funksjoner. F.eks en funksjon ”blinkI(milliSec)” som du kaller tre ganger med et lavt tall, tre ganger med et høyt tall, og så tre ganger med et lavt tall igjen. Eller metodene “kortBlink()” og “langtBlink()” hvor du kaller “langtBlink()” tre ganger, men “kortBlink()” tre ganger før og etter det.

6. Er det mulig å lage et slikt SOS signalsystem uten arduino, men med bare et batteri og komponentene fra arduino-starter-kit? Dette kan være en veldig vanskelig oppgave. Ikke bruk mye tid på den, men prøv og tenk gjennom noen muligheter.

## 2.7. Blink level 2

Anta at port 2 er koblet til og styrer en LED slik som i oppgave 2.4. Hvordan vil lampen oppføre seg når disse kodesnuttene kjører? Begrunn svaret kort.

1.

```
6 int state = HIGH;
7 void loop() {
8     digitalWrite(2, state);
9     state = !state;
10    delay(1000);
11 }
```

2.

```
7 int i;
8 void loop() {
9     for (i = 100; i < 500; i = i + 20) {
10        digitalWrite(2, !digitalRead(2));
11        delay(i);
12    }
13
14    for (i = 500; i > 100; i = i - 20) {
15        digitalWrite(2, !digitalRead(2));
16        delay(i);
17    }
18 }
```

3.

```
13 int ledState = HIGH;
14 void loop() {
15     if (ledState == HIGH) {
16         digitalWrite(2, LOW);
17     } else if (ledState == LOW) {
18         digitalWrite(2, HIGH);
19     }
20 }
```



### 3. Nøtt

Sett opp en krets hvor en kan lese om en knapp er trykket ned, og i tillegg tre LED-pærer slik at du kan styre dem individuelt. Oppgaven er å få LED-pærene til å lyse opp etter tur hver gang knappen trykkes ned. Figuren nedenfor viser hva som skal skje når du trykker knappen fire ganger. Når den siste LED-pæren i rekken lyser og knappen trykkes igjen skal lyset begynne fra starten av rekken igjen.

