

Arduinoøving 1

IELET1002 - Datateknikk

Gunnar Myhre, BIELEKTRO

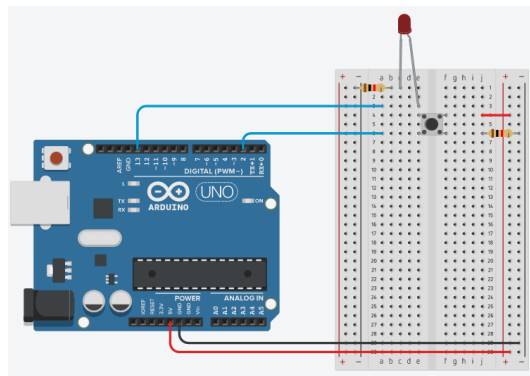
18. november 2021

1 Oppgave 1

a) Pull-down-knapp

- brytar åpen → LED på
- brytar lukka → LED av

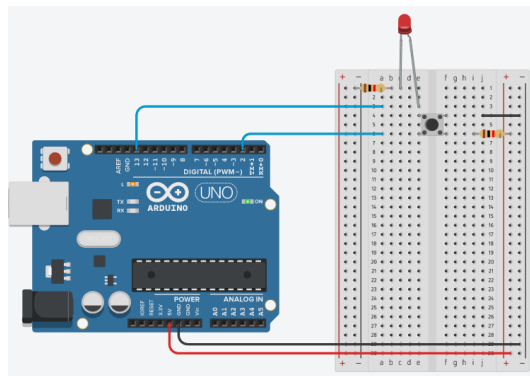
```
1 const int buttonPin = 2;
2 const int ledPin = 13;
3 const int delayTimeMs = 10;
4
5 bool buttonState = 0;
6
7 void setup() {
8   pinMode(ledPin, OUTPUT);
9   pinMode(buttonPin, INPUT);
10 }
11
12 void loop() {
13   buttonState = digitalRead(buttonPin);
14   if (buttonState == 0) {
15     digitalWrite(ledPin, LOW);
16   } else {
17     digitalWrite(ledPin, HIGH);
18   }
19   delay(delayTimeMs);
20 }
```



b) Pull-up-knapp

- brytar åpen \rightarrow LED av
- brytar lukka \rightarrow LED på

```
1 const int buttonPin = 2;  
2 const int ledPin = 13;  
3 const int delayTimeMs = 10;  
4  
5 bool buttonState = 0;  
6  
7 void setup() {  
8   pinMode(ledPin, OUTPUT);  
9   pinMode(buttonPin, INPUT);  
10 }  
11  
12 void loop() {  
13   buttonState = digitalRead(buttonPin);  
14   if (buttonState == 1) {  
15     digitalWrite(ledPin, LOW);  
16   } else {  
17     digitalWrite(ledPin, HIGH);  
18   }  
19   delay(delayTimeMs);  
20 }
```



c) Pull-down-knapp med blinkande LED

- brytar åpen → LED på
- brytar lukka → LED blinker

```
1 const int buttonPin = 2;
2 const int ledPin = 13;
3 const int delayTimeMs = 10;
4
5 bool buttonState = 0;
6 bool ledState = 0;
7
8 void setup() {
9   pinMode(ledPin, OUTPUT);
10  pinMode(buttonPin, INPUT);
11 }
12
13 void loop() {
14   buttonState = digitalRead(buttonPin);
15   if (buttonState == 0) {
16     // Blink LED
17     digitalWrite(ledPin, ledState);
18
19     // Toggle ledState
20     ledState = !ledState;
21
22     delay(100);
23   } else {
24     digitalWrite(ledPin, HIGH);
25   }
26   delay(delayTimeMs);
27 }
```

d) Pull-down-knapp med blinkande LED

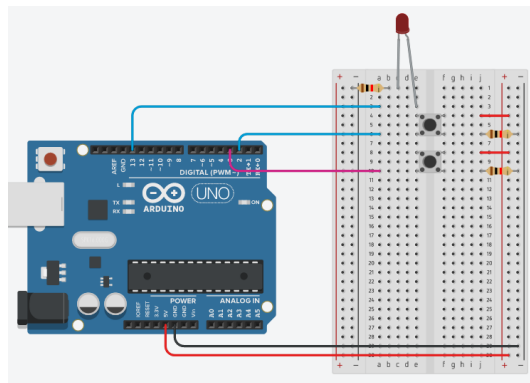
- brytar åpen → LED på
- brytar lukka → LED blinker
- brytar åpen igen → LED av

```
1 const int buttonPin = 2;
2 const int ledPin = 13;
3 const int delayTimeMs = 10;
4
5 bool buttonState = 0;
6 bool buttonInhibit = 0;
7 bool ledState = 0;
8
9 void setup() {
10  pinMode(ledPin, OUTPUT);
11  pinMode(buttonPin, INPUT);
12 }
13
14 void loop() {
15   buttonState = digitalRead(buttonPin);
16   if (buttonState == 1) {
17     buttonInhibit = 1;
18     // Blink LED
19     digitalWrite(ledPin, ledState);
20
21     // Toggle ledState
22     ledState = !ledState;
23
24     delay(100);
25   } else {
26     if (buttonInhibit) {
27       digitalWrite(ledPin, LOW);
28     } else {
29       digitalWrite(ledPin, HIGH);
30     }
31   }
32   delay(delayTimeMs);
33 }
```

e) Pull-down-knapp med blinkande LED

- brytar 1 åpen → LED på
- brytar 1 lukka → LED blinker
- brytar 1 åpen igen → LED av
- brytar 2 lukka → LED på igen

```
1 const int blinkButtonPin = 2;
2 const int resetButtonPin = 3;
3 const int ledPin = 13;
4 const int delayTimeMs = 10;
5
6 bool blinkButtonState = 0;
7 bool resetButtonState = 0;
8 bool ledInhibit = 0;
9 bool ledState = 0;
10
11 void setup() {
12   pinMode(ledPin, OUTPUT);
13   pinMode(blinkButtonPin, INPUT);
14   pinMode(resetButtonPin, INPUT);
15 }
16
17 void loop() {
18   blinkButtonState = digitalRead(blinkButtonPin);
19   resetButtonState = digitalRead(resetButtonPin);
20
21   if (blinkButtonState == 1) {
22     ledInhibit = 1;
23     // Blink LED
24     digitalWrite(ledPin, ledState);
25
26     // Toggle ledState
27     ledState = !ledState;
28
29     delay(100);
30   } else {
31     if (ledInhibit) {
32       //Inhibit the LED
33       digitalWrite(ledPin, LOW);
34     } else {
35       digitalWrite(ledPin, HIGH);
36     }
37   }
38   if (resetButtonState == 1) {
39     //Remove the inhibitor so the LED can be activated again
40     ledInhibit = 0;
41   }
42   delay(delayTimeMs);
43 }
```

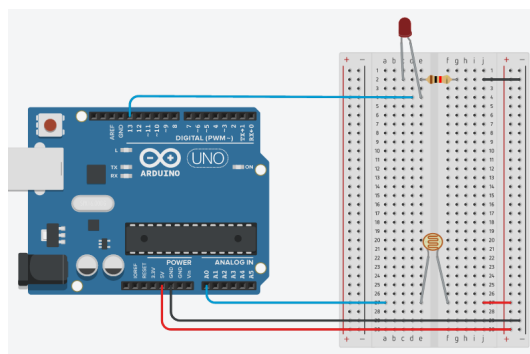


2 Oppgave 2

a) Fotoristor med indikator-LED for to terskelverdier

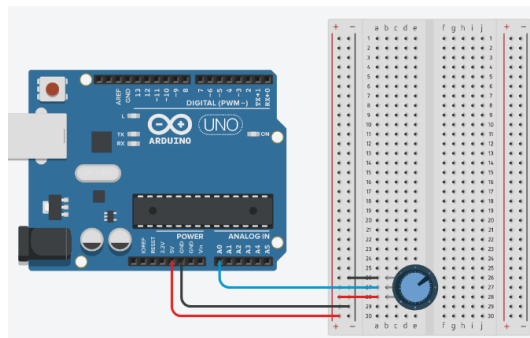
- $lysverdi < terskel_1 \rightarrow$ LED av
- $terskel_1 < lysverdi < terskel_2 \rightarrow$ LED på
- $lysverdi > terskel_2 \rightarrow$ LED blinker

```
1 const int photoresistorPin = A0;
2 const int ledPin = 13;
3 const int blinkDelayMs = 200;
4
5 float threshold_1 = 1000;
6 float threshold_2 = 1500;
7 float lightSensorReading = 0;
8 int ledState = 0;
9
10 void setup() {
11   pinMode(ledPin, OUTPUT);
12   pinMode(photoresistorPin, INPUT);
13 }
14
15 void loop() {
16   lightSensorReading = analogRead(photoresistorPin);
17
18   if (lightSensorReading >= threshold_1) {
19     if (lightSensorReading >= threshold_2) {
20
21       //If sensor value is above threshold 2, blink the LED
22       ledState = !ledState;
23       digitalWrite(ledPin, ledState);
24
25     } else {
26       //If sensor value is between threshold 1 and 2, LED on
27       digitalWrite(ledPin, HIGH);
28     }
29   } else {
30     //If sensor value is under threshold 1, LED off
31     digitalWrite(ledPin, LOW);
32   }
33   delay(blinkDelayMs);
34 }
```

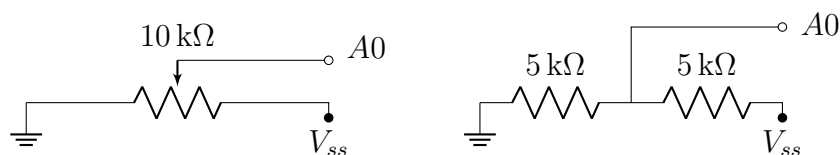


b) Potensiometerkrets

```
1 // Deklarerer variablar
2 const int potMeterPin = A0;
3 int potMeterReading;
4
5 void setup() {
6   //Deklarererodus for potmeter-pinnen
7   pinMode(potMeterPin, INPUT);
8
9   //Initialiserer seriellmonitoren over 9600 baud
10  Serial.begin(9600);
11 }
12
13 void loop() {
14   //Gjer ei måling på potmeter-pinnen
15   potMeterReading = analogRead(potMeterPin);
16
17   //Printer målingsverdien over seriellmonitoren
18   Serial.println(potMeterReading);
19
20   //Venter eitt millisekund
21   delay(1);
22 }
```



Når programmet køyrer vert verdier avlest frå A0 kontinuerleg printa til seriellmonitoren. Eg observerer at verdiane går ifrå 0 på jordsida til 1023 på kildesida. Dette er eit verdispenn på 1024, altså 2^{10} . Dette kan skyldast måten Arduino UNO sin ADC (analog-til-digital-konverterar) er implementert.



Kretsteikningane over viser eit potmeter og ein spenningsdelar. Desse to kretsane er ekvivalente dersom eit 10kΩ-potensiometer står i 50%-posisjon. Eg gjennomfører testen frå oppgåveteksten.

- Kopler av kilden, nå står R_{pot} mellom GND og A0. Det skal ikkje gå nokon straum igjennom R_{pot} og spenninga skal då vere $0V = GND$. Dette ser vi også i seriellmonitoren som printer 0.

- Kopler av GND. Nå står R_{pot} mellom A0 og 5V. Det skal ikkje gå nokon straum inn i ADC-en, og derfor vil spenningsfallet vere 0. Spenninga som står i A0 skal derfor vere 5V. Verdien som skrivast til seriellmonitoren er 1023.

Sidan den simulerte arduinoen i Tinkercad opererer ideelt (den er m.a. i stand til å forsyne kretsen med 125kW) vil ikkje verdiane i testen avvike frå verdiane når potmeteret er kopla mellom GND og 5V. På ein faktisk arduino kan verdiane avvike pga arduinoens straumforsyning som må forsyne tilstrekkelig effekt.

$$P = \frac{v^2}{R} = \frac{(5V)^2}{10k\Omega} = 2,5mW \quad (1)$$

c)

For å måle faktiske spenningsverdiar mellom 0V og 5V må vi ta hensyn til at målingas verdispenn er 1024.

$$1023k = 5 \Rightarrow k = \frac{5}{1023} = 0,004887586 \quad (2)$$

I arduino vert dette

```
1 //Rekner ut spenning i A0 og printer til Serial
2 pinVoltage = potMeterReading * 0.004887586;
3 Serial.println(pinVoltage);
```

Det vi måler her er spenningsfallet over den delen av R_{pot} som til ein kvar tid befinner seg mellom A0 og 5V.

d)

For å finne motstandsverdiane kan vi bruke formel for spenningsdeling

$$v_{A0} = \frac{R_1}{R_{pot}} v_s \rightarrow R_1 = \frac{v_f R_{pot}}{v_s} \quad (3)$$

Dette er motstanden mellom A0 og GND. Vi subtraherer frå R_{pot} for å finne verdien mellom A0 og 5V. I arduino kan vi skrive det slik:

```
1 //Rekner ut motstand mellom A0 og Vss og printer til Serial
2 potMeterResistance = 10000 - (pinVoltage * 10000) / 5;
3 Serial.println(potMeterResistance);
```

Heile koden for **c)** og **d)** ser slik ut:

```
1 // Deklarerer variablar
2 const int potMeterPin = A0;
3 int potMeterReading;
4 float pinVoltage;
5 float potMeterResistance;
6
7 void setup() {
8     //Deklarerer modus for potmeter-pinnen
9     pinMode(potMeterPin, INPUT);
10
11     //Initialiserer seriellmonitoren over 9600 baud
12     Serial.begin(9600);
13 }
14
15 void loop() {
16     //Gjer ei måling på potmeter-pinnen
17     potMeterReading = analogRead(potMeterPin);
18
19     //Rekner ut spenning i A0
20     pinVoltage = potMeterReading * 0.004887586;
21
22     //Rekner ut motstanden mellom A0 og Vss
23     potMeterResistance = 10000 - (pinVoltage * 10000) / 5;
24
25     //Printer målingsverdiane over seriemonitoren
26     Serial.print(" voltage: ");
27     Serial.print(pinVoltage);
28     Serial.print(" resistance: ");
29     Serial.println(potMeterResistance);
30
31     //Venter eitt millisekund
32     delay(1);
33 }
```

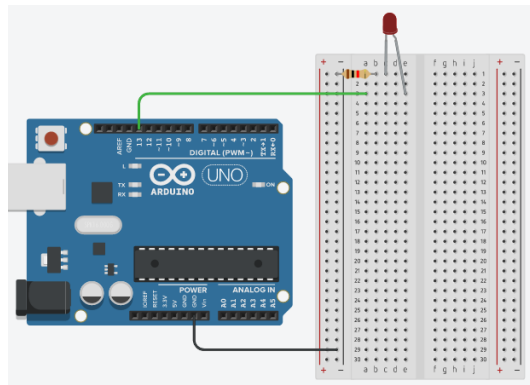

3 Oppgave 3

a) Blinkkrets med stadig aukande frekvens, $1 - x^2$. Krav: ingen while-løkke (skjønt arduinos loop er ei while-løkke).

```
1 //Constants
2 const int ledPin = 13;
3
4 //Variables
5 int ledState = 0;
6 int blinkInterval;
7 int blinkIterator;
8 float maxDelay = 1000;
9 int minDelay = 50;
10 int stepSize = 1;
11
12 void setup() {
13     pinMode(ledPin, OUTPUT);
14 }
15
16 void loop() {
17     //If delay interval passes minDelay, reset it and the iterator
18     if (blinkInterval < minDelay) {
19         blinkInterval = maxDelay;
20         blinkIterator = 0;
21     }
22
23     //Quadratic function: 1 - x**2
24     blinkInterval = maxDelay - pow(blinkIterator, 2);
25
26     //Blink the LED
27     digitalWrite(ledPin, HIGH);
28     delay(blinkInterval);
29     digitalWrite(ledPin, LOW);
30     delay(blinkInterval);
31
32     //Increment the iterator
33     blinkIterator += stepSize;
34 }
```

b) Blinkekrets med stadig lengre oppholdsintervall, x^2 . Her er det enkelt å bytte om på grensene og starte grafen frå null i staden for 1 s. Eg bytter óg plass på dei to blokkene slik at eit potensielt kjempehøgt oppholdsintervall vert luka ut i if-setninga.

```
1 //Quadratic function: x**2
2 blinkInterval = pow(blinkIterator , 2);
3
4 //If delay surpasses maxDelay, reset it and the iterator
5 if (blinkInterval > maxDelay) {
6     blinkInterval = minDelay;
7     blinkIterator = 0;
8 }
```



Oppkopling av arduino for oppg. 3.