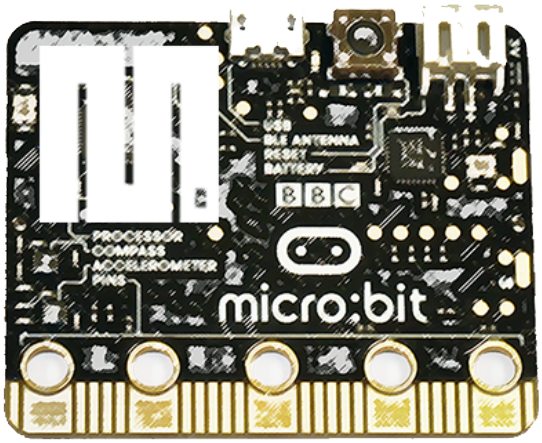
FYS1210 ELEKTRONIKK LAB 1



Introduksjon

I Lab 1 skal du gjøre målinger på en enkel krets og få en kort innføring i BBC micro:bit. micro:bit er et utviklingsbrett (developement board) laget for barn som også ungdommer og voksne kan ha glede og læringsutbytte av. For å programmere micro:bit skal vi i FYS1210 bruke MicroPython. Lab 1 har to hovedoppgaver:

- Oppgave 1: Måling av spenning, strøm og motstand med multimeter
- Oppgave 2: micro:bit innføring
- Oppgave 3: Kitronik Inventor's Kit experiments for BBC micro:bit

BBC micro:bit

> You can use your BBC micro:bit for all sorts of cool creations, from robots to musical instruments – the possibilities are endless. This little device has an awful lot of features, like 25 red LED lights that can flash messages. There are two programmable buttons that can be used to control games or pause and skip songs on a playlist. Your BBC micro:bit can detect motion and tell you which direction you're heading in, and it can use a low energy Bluetooth connection to interact with other devices and the Internet – clever!

MicroPython

> MicroPython is a lean and efficient implementation of the Python 3 programming language that includes a small subset of the Python standard library and is optimised to run on microcontrollers and in constrained environments.

Jupyter Notebook

I FYS1210 skal alle labene gjøres i Jupyter Notebook (http://jupyter.org/).

> The Jupyter Notebook is a web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, machine learning and much more.
>
> Project Jupyter was born out of the IPython Project in 2014 as it evolved to support interactive data science and scientific computing across all programming languages.

Kjør Python kode!

I Jupyter kan du altså kjøre Python kode. Jupyter skiller seg ut fra kjøring av Python på vanlig vis ved bruk av celler. Teksten du leser nå er i en `Markdown`-celle mens cellen under er en `Code`-celle. For å kjøre koden under; merk cellen og trykk `SHIFT+ENTER`.

In [ ]:
```
heiverden = "Hei verden"
print(heiverden)

fys = 1210
print("FYS%d" % fys)
```

Variabelene i cellen over kan brukes videre i Notebooken.

In [ ]:
```
en_liste_med_tall = [fys, 1010, 1110]
print(en_liste_med_tall)
```

Du kan også importere pakker som du (forhåpentligvis) har brukt i INF1100, MAT-INF1100 eller andre kurs.

In [ ]:
```
from numpy import *
en_numpy_array = asarray(en_liste_med_tall)
print(en_numpy_array, type(en_numpy_array))

import sys
print(sys.version)
```

Aritmetikk

I denne laben skal du kun gjøre enkle utregninger og printe disse slik som i eksempelet under.

In [ ]:
```
a = 1
b = 2

addisjon = a + b
subtraksjon = a - b
multiplikasjon = a*b
divisjon = a/b

print("Addisjon: %g" % addisjon)
print("Subtraksjon: %g" % subtraksjon)
print("Multiplikasjon: %g" % multiplikasjon)
print("Divisjon: %g" % divisjon)
```

Divisjon vil gi forskjellig svar avhengig av om en bruker Python 2 eller Python 3. Python 2 vil gi 0 mens Python 3 vil gi 0.5. For Python 2 kan det løses ved å:

> 1. Gjøre om minst ett av tallene til type float
> 2. Importere divisjon fra fremtid

In [ ]:
```
# 1
print("Alternativ 1: %g" % (float(a)/b))
# 2
from __future__ import division
print("Alternativ 2: %g" % (a/b))
```

Hvis du har kjørt cellen over trenger du ikke lenger tenke på divisjon for resten av laben. Prøv å kjøre cellen under og se om du får svaret du forventer.

In [ ]:
```
print("Test av divisjon: %g" % (1/2))
```

Formatere print

Det kan være greit å formatere print for lesbarhet. En kommer veldig langt med `%g`, men av og til kan det være greit å bestemme selv. For detaljer se Python 2 Docs: String Formatting Operations (https://docs.python.org/2/library/stdtypes.html#string-formatting-operations).

In [ ]:
```
print("Stort tall:")
stort_tall = 1234567.89
print("f, Floating decimal: %f" % stort_tall)
print("g, Floating decimal som 'tenker for deg': %g" % stort_tall)
print("e, Floating exponential: %e" % stort_tall)
print("d, Integer decimal: %d" % stort_tall)

print("\nLite tall:")
lite_tall = 0.0001234
print("f, Floating decimal: %f" % lite_tall)
print("g, Floating decimal som 'tenker for deg': %g" % lite_tall)
print("e, Floating exponential: %e" % lite_tall)
print("d, Integer decimal: %d" % lite_tall)

desimaler = 1.23456
print("\nDesimaler")
print(".0f, %.0f" % desimaler)
print(".1f, %.1f" % desimaler)
print(".2f, %.2f" % desimaler)
print("osv...")
```

Det er viktig å huske på at FYS1210 er et kurs i elektronikk og ikke programmering. Programmering med Python er ment som et verktøy som kan hjelpe deg å jobbe med og forstå elektronikk bedre. Det vil ikke være nødvendig å kunne avansert programmering for å jobbe med labene og det vil bli gitt ferdig kode der hvor vi mener at det vil være for vanskelig eller ta for mye tid å programmere selv for en lab med begrenset tid.

Oppgave 1: Måling av spenning, strøm og motstand

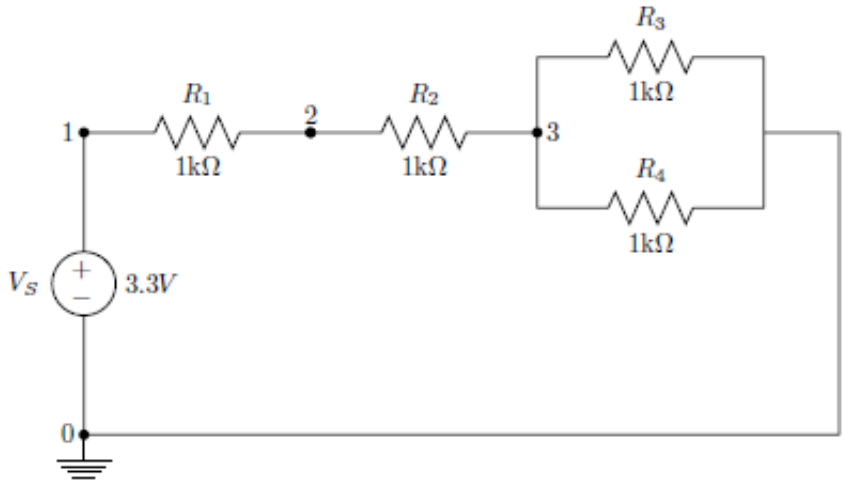
Utstyr

På plassen skal følgende utstyr være tilgjengelig:

1. Micro:bit, 1 stk
2. micro USB-kabel, 1 stk
3. Multimeter Tecpel 127, 1 stk
4. Motstand 1 kOhm, 4 stk
5. Ledning med gripeklo, 2 stk (rød og svart)
6. Brødbrett (på et Kitronikbrett), 1 stk
7. Jumper wire male-to-female, 5 stk
8. Kitronik Inventor's Kit, 1 stk

Kretsen og oppgavene

I oppgave 1 skal du for kretsen under først regne ut spenning, strøm og motstand med påtrykte verdier og deretter gjøre målinger med multimeteret.



Deloppgavene for oppgave 1 er følgende:

(a) Regne ut spenning, strøm og motstand for påtrykte verdier
(b) Måle motstandene
(c) Fargekode og toleranse
(d) Koble krets på brødbrett
(e) Måle strøm
(f) Måle spenning over motstandene
(g) Måle spenning i punkter (multimeter)
(h) Regn ut strømmen i kretsen
(i) Sammenlign utregning og målingene

Formler til oppgave 1

Ohms lov

$$I = \frac{V}{R}$$

Motstander i serie og parallell

- Serie:

$$R_s = R_1 + R_2 + \cdots + R_n$$

- Parallell (2 motstander):

$$R_p = \left(\frac{1}{R_1} + \frac{1}{R_2}\right)^{-1} = \frac{R_1 R_2}{R_1 + R_2}$$

Kirchhoffs spenningslov (KVL)

Summen av spenningene i en lukket sløyfe er null:

$$\sum_{j=1}^{N} V_j = 0$$

Kirchhoffs strømlov (KCL)

Summen av strømmen inn i et punkt er lik summen av strømmen ut av punktet:

$$\sum_{j=1}^{N} I_j = 0$$

Spenningsdelerformelen (2 motstander)

$$V_{R_1} = \frac{R_1}{R_1 + R_2} V_s$$

Krets og formler: lett tilgjengelig ved bruk av Magic

I Jupyter Notebook kan du bruke en type funksjonalitet som kalles magics (line and cell magics). Vi skal bruke to forskjellige line magics i Lab 1:

1. %lab, en custom line magic for å åpne et nytt Google Chorme vindu for å gjøre informasjon du ofte kan ha bruk for i labene lett tilgjengelig.
2. %less, en standard line magic for å lese innholdet i en fil (oppgave 3).

Kjør cellen under for å se alle tilgjengelige magics.

In [2]:
```
%lsmagic
```

Out[2]:
```
Available line magics:
%alias %alias_magic %autocall %automagic %autosave %bookmark %
cd %clear %cls %colors %config %connect_info %copy %ddir %de
bug %dhist %dirs %doctest_mode %echo %ed %edit %env %gui %h
ist %history %killbgscripts %lab %ldir %less %load %load_ext
%loadpy %logoff %logon %logstart %logstate %logstop %ls %lsma
gic %macro %magic %matplotlib %mkdir %more %notebook %page %
pastebin %pdb %pdef %pdoc %pfile %pinfo %pinfo2 %popd %pprin
t %precision %profile %prun %psearch %psource %pushd %pwd %p
ycat %pylab %qtconsole %quickref %recall %rehashx %reload_ext
%ren %rep %rerun %reset %reset_selective %rmdir %run %save %
sc %set_env %store %sx %system %tb %time %timeit %unalias %
unload_ext %who %who_ls %whos %xdel %xmode

Available cell magics:
%%! %%HTML %%SVG %%bash %%capture %%cmd %%debug %%file %%htm
l %%javascript %%js %%latex %%perl %%prun %%pypy %%python %%
python2 %%python3 %%ruby %%script %%sh %%svg %%sx %%system %
%time %%timeit %%writefile

Automagic is ON, % prefix IS NOT needed for line magics.
```

Her ser du at magic %less og %lab (forhåpentligvis) er listet. Du kan bruke %lab på alle laber ved å skrive `%lab <labnummer>` slik som i cellen under. Kjør cellen, plasser og juster det nye vinduet på siden av Notebooken slik at begge er synlig.

In [4]:
```
%lab 1
```

(a) Utregning med påtrykte verdier

Bruk formlene og de påtrykte verdiene til å regne ut:

1. Total motstand i kretsen.
2. Total strøm som går gjennom kretsen.
3. Spenningen over motstandene. Kall spenningene over motstandene for V_{R_1}, V_{R_2} og $V_{R_{34}}$.
4. Spenningen i punktene 1, 2 og 3. Kall spenningene i punktene for V_1, V_2 og V_3.

micro:bit har $V_S = 3.3\,\mathrm{V}$ selv om det står $3\,\mathrm{V}$ på Kitronik-brettet.

In [ ]:
```
VS = 3.3
R1 = R2 = R3 = R4 = 1e3

### Din utregning her ###
```

(b) Måling av motstandene

Bruk multimeteret til å måle motstandene.

1. Koble ledningene med gripeklo til inngangene COM og $\mathrm{V}\Omega$.
2. Sett bryter til Ω.
3. Mål motstandene.
4. Regn ut den totale motstanden.

In [ ]:
```
### Din måling og utregning her ###
```

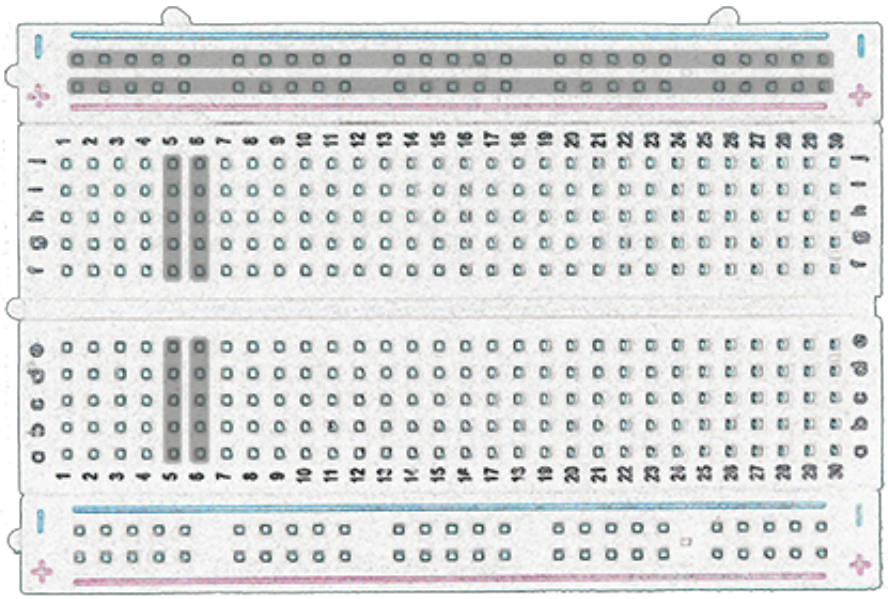
(c) Fargekode og toleranse

1. Bruk fargekodetabell til å lese av verdien og toleransen på motstandene.
2. Regn ut forholdet mellom målt og påtrykt verdi.

In [ ]:
```
### Din utregning her ###
```

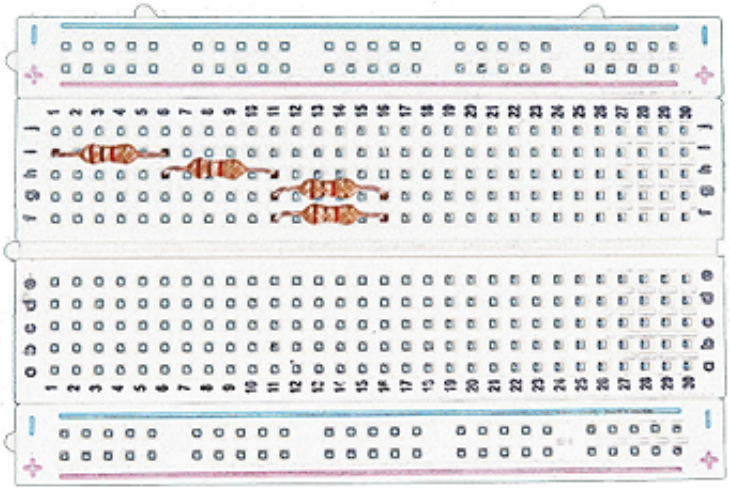
(d) Kobling av krets på brødbrett

Du skal nå koble opp kretsen på brødbrettet til Kitronik Inventor's Kit. Punktene i brødbrettet som følger med Inventor's Kit er koblet sammen slik som de grå feltene viser på bildet under.



Altså er punktene i de to lange radene oppe og nede koblet sammen horisontalt, mens de korte kolonnene er koblet sammen vertikalt. De lange radene oppe og nede blir brukt til spenning (rød/+) og jord (blå/-) hvis nødvendig. For kretsen i Lab 1 er det letteste å koble spenning og jord direkte. Øvre og nedre område med kolonner blir brukt til komponenter. Mellomrommet mellom de to områdene gjør det mulig å bruke integrerte kretser (IC; svart brikke med ben på sidene) på brødbrettet samt å utnytte arealet på brødbrettet bedre. Hvis mellomrommet ikke hadde vært der, eller hvis du setter hele IC'en i et av kolonnefeltene, vil den kortsluttes. For kretsen i Lab 1 er det enkleste å koble motstandene i kun ett av områdene med kolonner.

1. Sett micro:bit inn i Kitronikbrettet hvis det ikke allerede står der. Pass på at det er satt inn riktig vei (LED-matrisen er synlig).
2. Koble micro USB-kabelen fra micro:bit til en av USB-kontaktene på fremsiden av datamaskinen.
3. Plasser motstandene på brødbrettet omtrent slik som vist på bildet til høyre.



4. Koble svart jumper wire fra `0V` på Kitronikbrettet til et punkt på brødbrettet.
5. Koble rød jumper wire fra `3V` på Kitronikbrettet til et punkt på brødbrettet.

Hvis du har koblet riktig går det nå en strøm gjennom kretsen som du skal måle i neste oppgave. I FYS1210 bruker vi elektronstrøm som positiv retning. Altså at strømmen (og elektronene) går fra lavt (`0V`) til høyt potensiale (`3.3V`).

(e) Måling av strømmen i kretsen

For å måle strømmen må du koble multimeteret i serie med strømmen som skal måles.

1. Koble ledningene med gripeklo til COM og mA på multimeteret.
2. Sett bryteren til $20\ \mathrm{mA}$.
3. Ta ut rød jumper wire fra brettet og fest hann-kontakten på rød gripeklo.
4. Fest svart gripeklo til motstanden.

Hvis du har gjort det riktig skal multimeteret nå være koblet i serie med kretsen og du skal kunne lese av strømmen på multimeteret.

1. Les av verdien og skriv den inn i cellen under.
2. Fjern multimeteret fra kretsen og sett tilbake rød og svart jumper wire på brødbrettet.

In [ ]:
```
### Din måling her ###
```

(f) Måling av spenningen over motstandene

I forrige oppgave måtte du koble inn multimeteret i kretsen for å måle strømmen. Dette er ikke nødvendig når du måler spenning hvor du kan koble gripekloene i parallell med kompenentene du skal måle.

1. Koble ledningene med gripeklo til inngangene COM og $\mathrm{V}\Omega$.
2. Sett bryteren til V.
3. Koble ledningene med gripeklo over motstandene og mål spenningene (V_{R_1} , V_{R_2} , $V_{R_{3||4}}$).

In [ ]:
```
### Din måling her ###
```

(g) Måling av spenningen i punktene med multimeteret

Når du måler spenning i et punkt er det alltid i forhold til punktet som er definert som jord i kretsen. Bruken av ordet jord kan være litt forvirrende fordi jord har flere betydninger. I dette tilfellet er jord en nullpunktsreferanse i kretsen og ikke "fysisk-til-jord" slik som stikkontaktene på laben har.

Koble svart ledning til jord (på kretsen) og mål spenningen i punktene 1, 2 og 3 med rød ledning.

In [ ]:
```
### Din måling her ###
```

(h) Strømmen i kretsen

Bruk målte verdier til å regne ut strømmen.

In [ ]:
```
### Din utregning her ###
```

(i) Sammenligning av utregning og målingene

:

> Skriv en kort kommentar om verdiene du regnet ut og målinger med multimeteret. Hvis det er en forskjell, hvilken måling er det som er mest forskjellig fra påtrykt verdi?

Oppgave 2: micro:bit innføring

I oppgave 2 skal du gå gjennom en mini-tutorial for micro:bit og gjøre samme måling som i (f) med micro:bit i stedet for med multimeter.

Deloppgavene for oppgave 2 er følgende:

> (a) micro:bit innføring
> (b) Måling i punktene ved bruk av micro:bit
> (c) Sammenligning av målingene med multimeteret og micro:bit

(a) micro:bit innføring

Litt om måling med micro:bit før du begynner på tutorialen.

> micro:bit har en Analog-til-Digital konverter (ADC) med en oppløsning (resolution) på 10 bits. Det vil si at ADC'en kan måle $2^{10} = 1024$ nivåer. micro:bit vil returnere et heltall på mellom 0 og 1023 (1024 nivåer). Heltallet N må altså gjøres om til spenning ved

$$V = \frac{N}{1023} \cdot \mathrm{V}_{\mathrm{ref}},$$

> hvor N er heltallet micro:bit returnerer og $V_{\mathrm{ref}} = V_1$ som du har målt i (f).

Start en Jupyter Notebook med micro:bit kernel i en ny tab ved å trykke her (ubit/Lab1_microbit.ipynb) og gå gjennom tutorialen. Returner til denne Notebooken når du er ferdig og fullfør deloppgave (b).

(b) Måling i punktene ved bruk av micro:bit

> 1. Skriv inn heltallene du fikk i mini-tutroial i cellen under.
> 2. Gjør om målingene til Volt.

In [5]:
```
### Din måling og utregning her ###
```

(c) Sammenligning av målingene med multimeteret og micro:bit

> Skriv en kort kommentar som sammenligner målingene du har gjort med multimeteret og micro:bit.

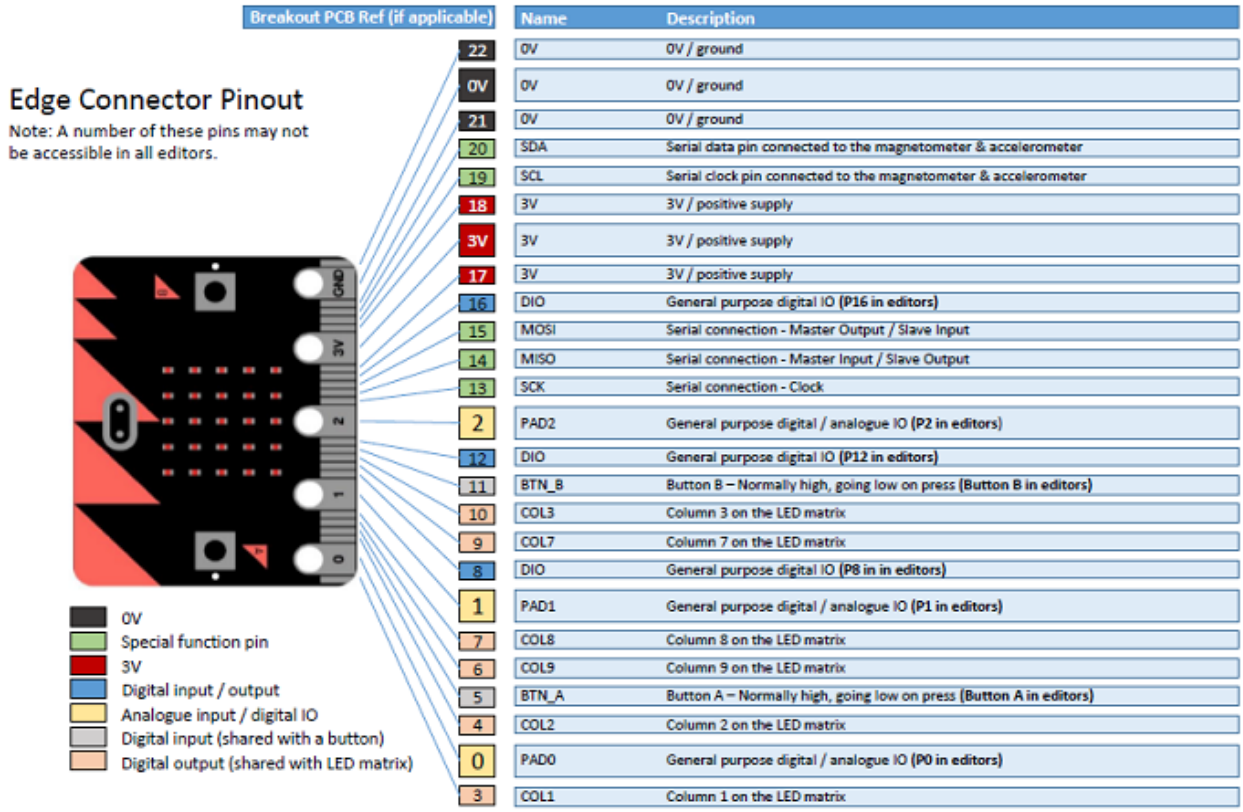
In [6]:
```
### Din kommentar her ###
```

Oppgave 3: Inventor's kit experiments

Gjør så mange av følgende eksperimenter i Kitronik Inventor's Kit du rekker. Gangen i alle eksperimentene er den samme:

1. Bruk den trykte veiledningen som følger med Inventor's Kit til å koble opp eksperimentet.
2. Se på filen som skal flashes til micro:bit ved bruk av magic-kommandoen %less
3. Flash filen til micro:bit
4. Les i veiledningen for hva som skal gjøres med eksperimentet.
5. Trykk på link ved behov.

Illustrasjonen under viser en oversikt over pins på micro:bit i forhold til Kitronikbrettet, men alt du trenger for å koble står i veiledningen.



Breakout PCB Ref (if applicable)	Name	Description
22	0V	0V / ground
0V	0V	0V / ground
21	0V	0V / ground
20	SDA	Serial data pin connected to the magnetometer & accelerometer
19	SCL	Serial clock pin connected to the magnetometer & accelerometer
18	3V	3V / positive supply
3V	3V	3V / positive supply
17	3V	3V / positive supply
16	DIO	General purpose digital IO **(P16 in editors)**
15	MOSI	Serial connection - Master Output / Slave Input
14	MISO	Serial connection - Master Input / Slave Output
13	SCK	Serial connection - Clock
2	PAD2	General purpose digital / analogue IO **(P2 in editors)**
12	DIO	General purpose digital IO **(P12 in editors)**
11	BTN_B	Button B – Normally high, going low on press **(Button B in editors)**
10	COL3	Column 3 on the LED matrix
9	COL7	Column 7 on the LED matrix
8	DIO	General purpose digital IO **(P8 in in editors)**
1	PAD1	General purpose digital / analogue IO **(P1 in editors)**
7	COL8	Column 8 on the LED matrix
6	COL9	Column 9 on the LED matrix
5	BTN_A	Button A – Normally high, going low on press **(Button A in editors)**
4	COL2	Column 2 on the LED matrix
0	PAD0	General purpose digital / analogue IO **(P0 in editors)**
3	COL1	Column 1 on the LED matrix

(a) Experiment One: Say "hello" to the micro:bit!

In [ ]:
```
%less ubit/ubit_experiment1.py
```

In [ ]:
```
!uflash ubit/ubit_experiment1.py
```

Experiment One: Further Help (https://www.kitronik.co.uk/blog/inventors-kit-experiment-1-help)

(b) Experiment Ten: Using an RGB LED

In [ ]:
```
%less ubit/ubit_experiment10.py
```

In [ ]:
```
!uflash ubit/ubit_experiment10.py
```

Experiment Ten: Further Help (https://www.kitronik.co.uk/blog/inventors-kit-experiment-10-help)

(c) Experiment Eight: Making a game using the compass

In [ ]:
```
%less ubit/ubit_experiment8.py
```

In [ ]:
```
!uflash ubit/ubit_experiment8.py
```

Experiment Eight: Further Help (https://www.kitronik.co.uk/blog/inventors-kit-experiment-8-further-help)

Appendiks

BBC micro:bit (http://microbit.org/)
MicroPython (https://micropython.org/)

In [ ]:
```

```