

#### Forløb med Micro:Bit – LKM0:

Formål: Lære om sensorer med Micro:Bit (MB) og MicroPython som udgangspunkt.

Mål: At beherske Micro:Bit's sensorer via MicroPython, at kunne bygge et produkt med MB, og kunne argumentere for konstruktion og hvilke fordele/ulempes der er med den valgte konstruktion og argumentere om forbedringer, herunder specifikt hvilken sensor funktionalitet der kunne forbedre produktet.

Knytter an til kernestof i fysik omkring sensorer, men også som byggesten til teknologifags projekter, hvor der evt. bygges større ting. Ud over PC/MAC og Micro:Bit som udleveres til alle elever henvises til:

<https://www.w3schools.com/python/default.asp>

<https://microbit-micropython.readthedocs.io/en/v1.0.1/index.html>

som opslagsværker.

Der vil være:

- En introduktion til Micro:Bit og til Python/MicroPython, samt udviklingsmiljøet – Thonny.
- Gennemgang af sensorer, radio og styring af højttaler med hands-on kodning på klassen.
  - 1) Løbende opgaver til at undersøge funktion selvstændigt.
- Opgave i grupper, hvor der skal bygges mindst 2 produkter – En bunden opgave og én fri der begge afslutningsvis skal fremlægges og forklares som nævnt i indledning.
  - 1) Lav en aktiv udgave af Corona badgen, som giver en advarsel med lyd og lys når der kommer nogen for tæt på.
  - 2) Lav et andet produkt efter eget valg. Det kan f.eks. være en leg, et måleinstrument eller et forsøg med nudging. For eksempel om man kan få flere til at gå til højre end til venstre når man går ind ad en dør eller tage trappen - Se:  
<https://www.youtube.com/watch?v=2lXh2n0aPyw&feature=youtu.be>  
Begge produkt skal testes af, og test dokumenteres og bruges i argumentation ved fremlæggelse.

Forløb:

- 1) Introduktion til Micro:Bit (MB) og udviklingsmiljø. Mål: 'Hul igennem hos alle'  
(I får en MB og USB kabel/mellemstik så MB kan kobles til jeres MAC/PC)
  - a. Intro MB.
    - i. Sensorer
    - ii. basis SW/OS (forskel fra fabrik – blokprogrammering og MicroPython)
  - b. Intro Thonny/MicroPython – hul igennem til MB på alles PC/MAC
    - i. Forskellige fortolkere (interpreter) i Thonny's Run menu.
    - ii. Dokumentation i 2 dele. Tutorials og API.
  - c. Syntax og termer.
    - i. Termer:
      1. biblioteker (libraries), moduler, packages – giver ekstra funktionalitet.
      2. Objekter – Metoder - funktioner
    - ii. Syntax: Forskel på store og små bogstaver, kolon ':' og indrykninger.
  - d. Interaktion med MB via Shell/editor i Thonny. Bibliotek/modul import.

- e. Første program på MB – Gemme på PC/MAC
  - i. `print("Hello world")` – virker både på MAC/PC/MB.
  - ii. `display.show("Hello world")` eller med et billede. Se:  
<https://microbit-micropython.readthedocs.io/en/v1.0.1/tutorials/images.html>  
(virker kun på MB, og efter import af microbit bibliotek)
  - iii. Løkke: Se: [https://www.w3schools.com/python/python\\_while\\_loops.asp](https://www.w3schools.com/python/python_while_loops.asp)  
`while <betingelse>:`  
    <gør dit>  
    <gør dat>
  - iv. Betinget udførelse: Se:  
[https://www.w3schools.com/python/python\\_conditions.asp](https://www.w3schools.com/python/python_conditions.asp)  
`,if <betingelse 1>:`  
    <gør dit når betingelse 1 er opfyldt>  
    <gør dat når betingelse 1 er opfyldt>  
`elif <betingelse 2>:`  
    <gør noget andet dit når betingelse 2 er opfyldt>  
    <gør noget tredje når betingelse 2 er opfyldt>  
`else:`  
    <gør noget femte når hverken betingelse 1 eller 2 er opfyldt>
  - v. funktionerne: `button_a.is_pressed()`, `display.show()` fra microbit biblioteket  
Kopier kode eksempel fra (bunden af siden):  
<https://microbit-micropython.readthedocs.io/en/v1.0.1/tutorials/buttons.html>
- 2) Introduktion til lyd på MB. (herunder fortælling om PIN's) (I får et par kabler med krokodillenæb)
  - a. PIN's.  
Tutoriel afsnit: <https://microbit-micropython.readthedocs.io/en/v1.0.1/tutorials/io.html>  
API afsnit: <https://microbit-micropython.readthedocs.io/en/v1.0.1/pin.html>  
Vi bruger kun GND/PIN0,1 og 2! Nu: Hovedtelefon på GND og P0.
  - b. Bibliotek/Modul: `speech` – funktion: `speech.say("Hi ...")`  
Åbn et nyt program/fane – og lav enkelt program der siger en sætning efter eget valg.  
<https://microbit-micropython.readthedocs.io/en/v1.0.1/tutorials/speech.html>  
<https://microbit-micropython.readthedocs.io/en/v1.0.1/speech.html>
  - c. Bibliotek/Modul: `music` – funktion: `music.pitch(frekvens,duration in ms)`  
Udbyg program fra trin 1 med lyd på et knap tryk. (prøv med duration sat til -1)  
Tilføj `music.stop()` til afslutning af program for at rydde op på PINs – ellers kan det drille når man senere vil bruge pin's til noget andet!  
<https://microbit-micropython.readthedocs.io/en/v1.0.1/tutorials/music.html>  
<https://microbit-micropython.readthedocs.io/en/v1.0.1/music.html>
  - d. Opgave med at undersøge hvilke frekvensværdier der duer:  
Hint: [https://www.w3schools.com/python/python\\_for\\_loops.asp](https://www.w3schools.com/python/python_for_loops.asp)  
Ekstra spg.: Hvad sker der med en negativ frekvens angivelse.
  - e. Når man har løst opgaven, kan man evt. selv eksperimentere med `display.scroll()` og/eller `display.show()` funktionerne – evt. lave en animation - Indtil der er samling i klassen og alle har løst opgaven. Man kan også hjælpe hinanden!

3) Introduktion til lys sensor

- a. `display.read_light_level()`.

Brug search funktionen i "read the docs"!

Bemærk forskel på at importere bibliotek via: "import microbit" versus:

"from microbit import \*". Kan I forklare forskellen?

- b. Vi laver et lille program der udskriver lys niveau i shell med `print()`.

- c. Opgave: Konverter lys niveau til en frekvens og afspil den.

Eksperimenter/Læs dokumentation: Hvilke værdier kan læses mht. lys niveauet?

Hvilke værdier kunne man bruge som frekvens i `music.pitch()`?

Hvordan kan man konvertere mellem de 2?

Lav en while løkke hvor I løbende læser lys-niveau, konverterer det til frekvens og afspiller det som lyd. Indsæt en `sleep(x)` i løkken.

4) Introduktion til magnetometer. (I får en magnet)

- a. `compass.get_field_strength()`, eller `compass.get_x()`, eller `get_y()` eller `get_z()`

Brug search function til at finde dokumentation.

Hvis ikke I kan finde dokumentation for hvilke værdier I kan få så lav et lille eksperiment med `print(værdi)` så I kan finde ud af hvilke værdier der fås ved læsning af magnetfeltet.

- b. Opgave: Omsæt til lyd (Hint: Division og funktion til at omsætte tal fra float til integer)

Som hjemmeopgave inden så læs om "Python Arithmetic Operators" i:

[https://www.w3schools.com/python/python\\_operators.asp](https://www.w3schools.com/python/python_operators.asp)

5) Introduktion til accelerometer.

- a. `accelerometer.get_values()`, eller `accelerometer.get_x()`, eller `get_y()` eller `get_z()`

<https://microbit-micropython.readthedocs.io/en/v1.0.1/accelerometer.html>

- b. Opgave: Omsæt til lyd (Hint: find funktion til at omsætte neg til positivt tal)

- c. Til hjemmelæsning - Baggrundsinfo MEMS:

<https://www.youtube.com/watch?v=iPGpoUN29zk>

[https://en.wikipedia.org/wiki/Microelectromechanical\\_systems](https://en.wikipedia.org/wiki/Microelectromechanical_systems)

6) Introduktion til radio

- a. `radio.on()`, `radio.config(channel=x)`, `radio.send("Besked")`, `besked = radio.receive()`.

Bemærk at læse funktionen (`radio.receive()`) returnerer None, hvis der ikke er modtaget nogen besked. None svarer til False!

<https://microbit-micropython.readthedocs.io/en/v1.0.1/radio.html>

- b. Opgave: Del accelerometer aftastning og lyd generering fra opgave 5b i 2 MBs og send værdier fra accelerometer aftastning fra den ene til den anden, som så laver lyden.

- c. Opgave: Der sendes på en udvalgt kanal – Kan I finde beskeden/kanalen?

- d. Beskeden som blev sendt gav en ny opgave som I skal kikke på.

- e. Hjemmeopgave: I får program som I skal kikke på og forklare hvordan virker næste i næste lektion.

- 7) Introduktion til at bruge PIN's som input, så man kan bygge egne 'knapper'.  
(I får et par ekstra kabler med krokodillenæb og noget kobbertape)
  - a. `pin0.is_touched()`, `pin1.is_touched()`, `pin3.is_touched()`  
<https://microbit-micropython.readthedocs.io/en/v1.0.1/pin.html>
  - b. Opgave: Lav 3 knapper der afspiller hver deres lyd.
  - c. Opgave: Hvordan kan man med 3 pin's lave 7 forskellige inputs? Udvid program til 7 forskellige og en ottende til at stoppe lyden.  
Læs om "Python Logical Operators" som hjemmeopgave inden! Findes på:  
[https://www.w3schools.com/python/python\\_operators.asp](https://www.w3schools.com/python/python_operators.asp)
- 8) Snak/opsamling om sensorer og hvordan de kan bruges til detektion af forskelligt.
  - a. Åben/lukket med magnetometer
  - b. Tælle omgange med magnetometer
  - c. Detektion af nogen der går ind med lys
  - d. Etc.

Undervejs og når alt er gennemgået arbejdes med de 2 overordnede opgaver i grupperne.