

Her er 3 forløb kort beskrevet. De 3 forløb er samlet blevet prøvet af og kan gennemføres på ca. 10 lektioner. Men for at give større fleksibilitet er de nu delt op, og man kan eksempelvis nøjes med det første forløb for at klæde elever på til at kunne gå videre med Spor 2. Beskrivelsen suppleres af en slide-stak. Både denne beskrivelse og slide-stak er tiltænkt lærer, som kan tilpasse efter behov og evt. få det ind i en større sammenhæng/forløb.

Modul a er det basale 'værkstedskørekort' til LYD-Kit, og kan gennemføres på 2-3 lektioner.

Modul b er en option, og kan eksempelvis i forbindelse med Teknologifag være et værkstedsmodul for dem som vil vide mere og når først når behovet opstår. Det kan også bruges som niveaudeling ifm. fysik og sensorer.

Modul c er en case, som i forbindelse med Teknologifag kan bruges som en tidlig Case til at få tydeliggjort et projekts forskellige faser, og fagligheder.

Basis forløb med Micro:Bit (værkstedskørekort) – Spor 1 – Modul a (LKS1-Ma):

Formål: Lære de grundlæggende ting omkring udviklingsmiljø, MicroPython og de indbyggede sensorer i Micro:Bit (MB). At have den grundlæggende viden til at kunne skifte kode i Micro:bit uden nødvendigvis at skrive den selv – hvilket er grundlaget for at kunne bruge Spor 2 og Spor 4, som lukket bog (uden at kende til software i dybden)

Mål: At kunne benytte Micro:Bitten's indbyggede sensorer via MicroPython, og at kunne bruge udviklingsmiljøet – Thonny. Have tilstrækkelig viden til at bygge med færdige software byggeklodser i Spor 2 og Spor 4.

Knytter an til kernestof i fysik omkring sensorer, men også som byggesten til teknologifags projekter, hvor der evt. bygges større ting.

Materialer: En Micro:Bit udleveret til alle elever, samt elevernes egne PC/MAC hvorpå der installeres Thonny. Se: "Notat om at komme i gang med Thonny".

Det er en fordel at sikre installation på elevernes MAC/PC inden der startes med undervisning. Derudover bør man have nogle magneter (en pr elev), og en lyskilde – men der kan eleverne typisk bruge deres telefoner. Kabel til at forbinde Micro:Bit med PC/MAC – BEMÆRK at standard kabel til Micro:bit har USB-A stik, så hvis PC/MAC ikke har det – men måske kun et USB-C stik, skal man have et andet kabel, eller en adapter.

Ud over elevernes egne PC/MAC (med Thonny installeret) og en Micro:Bit som tænkes udleveret 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.

Forløb:

- 1) Introduktion til Micro:Bit (MB) og udviklingsmiljø. Mål: 'Hul igennem hos alle' (IMB og USB kabel/mellemstik så MB kan kobles til jeres MAC/PC til alle)
 - a. Intro Thonny/MicroPython – hul igennem til MB på alles PC/MAC
 - i. Forskellige fortolkere (interpreter) i Thonny's Run menu.
 - ii. Dokumentationen i 2 dele. Tutorials og API.
 - b. 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.
 - c. Intro MB.
 - i. Sensorer
 - ii. basis SW/OS (forskel fra fabrik – blokprogrammering og MicroPython)
 - d. Interaktion med MB via Shell/editor i Thonny. Bibliotek/modul import.
 - e. Første program på MB – Gemme på PC/MAC
- 2) Introduktion til lyd på MB.
 - a. Bibliotek/Modul: speech – function: 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>
 - b. Bibliotek/Modul: music – function: 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>
 - c. Evt. opgave med at undersøge hvilke frekvensværdier der duer:
Hint: https://www.w3schools.com/python/python_for_loops.asp
Ekstra spg.: Hvad sker der med en negativ frekvens angivelse.
 - d. 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?

- 4) Introduktion til magnetometer. (udlever 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
- 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)
- 6) Introduktion til mikrofon (Micro:Bit version 2)
 - a. `microphone.sound_level()`
[Microphone V2 — BBC micro:bit MicroPython 1.0.1 documentation \(microbit-micropython.readthedocs.io\)](https://microbit-micropython.readthedocs.io/en/v1.0.1/microphone.html).
 - b. Opgave: eksperimenter med støjniveauer. Konverter til display visning på microbitten.

Udvidet forløb med Micro:Bit - Spor 1 – Modul b (LKS1-Mb):

Formål: At få grundlaget for at kunne bygge eksterne komponenter på Micro:Bit (MB), og at kommunikere mellem Micro:Bits. At have den grundlæggende viden til at kunne kikke ind i og forstå den færdige kode som ligger i Spor 2.

Mål: At kunne bygge eksterne sensorer på Micro:Bit, og aflæse dem via MicroPython. At kunne bygge et selvstændigt produkt med Micro:Bit. At have tilstrækkelig viden til at kunne modificere i de færdige softwarebyggekodser i Spor 2.

Knytter an til kernestof i fysik omkring sensorer, men også som byggesten til teknologifags projekter, hvor der arbejdes med større projekter.

Materialer: Som til Modul a, samt kabler med krokodillenæb (3 per elev), kobbertape, og optionelt. komponenter som potentiometer, PIR sensor eller lignende. Eleverne skal her sættes i grupper, så de er mindst 2 i hver gruppe. (2 Micro:bits' per gruppe)

1) Introduktion til MEMS

- a. Til hjemmelæsning - Baggrundsinfo MEMS:

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

- https://en.wikipedia.org/wiki/Microelectromechanical_systems

2) Introduktion til radio

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

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

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

- b. Modtag besked fra lærers MB.

- c. Fordel kanal numre, og se man kun kan modtage på den kanal.

- d. Opgave:

- i. Se udleveret kode "LKS1Mb-anaPitch-Modtager.py" og: "LKS1Mb-AccToPitch-Sender.py"

- ii. Brug udleveret kode, med gruppens kanal nummer. Forklar hvad `nof_nones` bruges til i modtager koden.

- iii. Modificer sender til at bruge anden sensor til at sende værdier til modtager.

3) Introduktion til at bruge PIN's som input, så man kan bygge egne 'knapper'.

- a. (Udlever ekstra kabler med krokodillenæb og noget kobbertape)

- b. Opgave: Lav 3 knapper der afspiller hver deres lyd. Brug udleveret kode: "LKS1Mb-DigPitch-Modtager.py" og "LKS1Mb-3Key-Sender.py"

- 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.

- i. Læs om "Python Logical Operators" som hjemmeopgave inden! Findes på:

- ii. https://www.w3schools.com/python/python_operators.asp

- d. Ekstra opgave: Udvid til 2 sendere og 15 lyde i modtager.

- e. Ekstra Opgave: Lav mekaniske knapper/kontakter.

- 4) 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.
- 5) Optioner – Byg kredsløb. (Se slide stak). Kan knytte an til kernestof omkring el-lære i fysik
- 6) Tilføj evt. selvstændig opgave om at bygge en funktionsmodel af et produkt efter elevernes eget valg, hvor de skal bruge sensor input fra Micro:bit til at lave funktionaliteten.

Aktiv corona badge - Case - Spor 1 – Modul c (LKS1-Mc):

Formål: At kunne forholde sig kritisk til en given funktionalitet i et interaktivt digitalt produkt. At kunne teste funktionalitet op mod en given brugssituation. At kunne forholde sig til forskellen på en tidlig funktionsmodel og et færdigt produkt, og kunne argumentere for hvilken udvikling der skal til for at gøre produktet modent.

Mål: At bibringe eleverne en forståelse for at der er mange forskellige fagligheder i spil i et udviklingsprojekt. At kunne argumentere for konstruktion og hvilke fordele/ulemper der er med den valgte konstruktion og argumentere om forbedringer, herunder specifikt hvilken sensor funktionalitet der kunne forbedre produktet.

Knytter an til kernestof i teknologifags projekter.

Case:

Forestil jer at I er ansat i et konsulenthus, som giver råd og vejledning i forbindelse med produktudviklingsopgaver. I er blevet kontaktet af en kunde, som har fået en ide til at lave et aktivt corona badge, som giver en advarsel med lyd og lys når der kommer nogen for tæt på. De har lavet en mock-up (tidlig funktionsmodel) vha. Micro:bits. Der udleveres noget kode der kan gemmes på Micro:Bit.

Opgaven I bliver stillet er:

- Forhold jer til funktionaliteten af Corona Badge i forhold til konkret brug og anbefalinger om afstandskrav – fra første bølge.
- Lav et oplæg til fysik design (Industrielt design) af et endeligt produkt. Gerne med forslag til bedre signalering (lyd/lys)
- Kom med en liste over afklaringer/udviklingspunkter med hensyn til elektronik og software for at produkt kan blive til et salgbart produkt, og en overordnet plan for et udviklingsprojekt.
- Kom med samlet anbefaling om at gå videre med produktideen, med tilhørende argumentation.