

LYD-Kit - Modul 1 efterår 2019

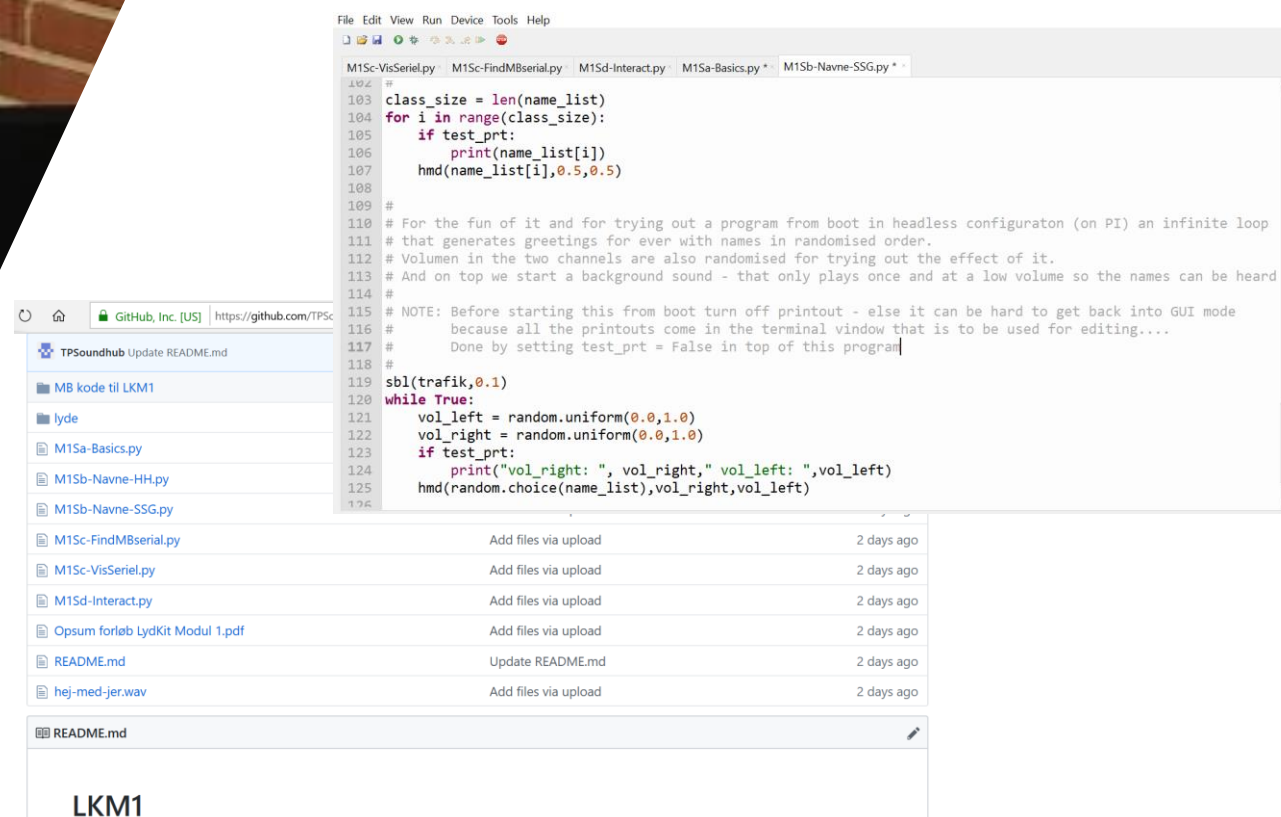
Projekt støttet af:

midt
regionmidtjylland

Integreret udviklingsmiljø (IDE) - Thonny
Python og biblioteker

- Pygame, Time, Serial, Random

Micro:Bit med kode til input events
PI , Beocreate boards samt et par højtalere



```
File Edit View Run Device Tools Help
M1Sc-VisSerial.py M1Sc-FindMBserial.py M1Sd-Interact.py M1Sa-Basics.py * M1Sb-Navne-SSG.py *
103 class_size = len(name_list)
104 for i in range(class_size):
105     if test_prt:
106         print(name_list[i])
107         hmd(name_list[i],0.5,0.5)
108
109 #
110 # For the fun of it and for trying out a program from boot in headless configuraton (on PI) an infinite loop
111 # that generates greetings for ever with names in randomised order.
112 # Volumen in the two channels are also randomised for trying out the effect of it.
113 # And on top we start a background sound - that only plays once and at a low volume so the names can be heard
114 #
115 # NOTE: Before starting this from boot turn off printout - else it can be hard to get back into GUI mode
116 # because all the printouts come in the terminal window that is to be used for editing...
117 # Done by setting test_prt = False in top of this program
118
119 sbl(trafik,0.1)
120 while True:
121     vol_left = random.uniform(0.0,1.0)
122     vol_right = random.uniform(0.0,1.0)
123     if test_prt:
124         print("vol_right: ", vol_right," vol_left: ",vol_left)
125     hmd(random.choice(name_list),vol_right,vol_left)
126
```

TPSoundhub Update README.md

MB kode til LKM1

lyde

- M1Sa-Basics.py
- M1Sb-Navne-HH.py
- M1Sb-Navne-SSG.py
- M1Sc-FindMBserial.py
- M1Sc-VisSerial.py
- M1Sd-Interact.py
- Opsum forløb LydKit Modul 1.pdf
- README.md
- hej-med-jer.wav

Add files via upload	2 days ago
Add files via upload	2 days ago
Add files via upload	2 days ago
Add files via upload	2 days ago
Update README.md	2 days ago
Add files via upload	2 days ago

README.md

LKM1

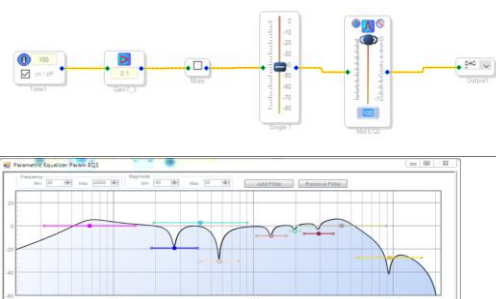
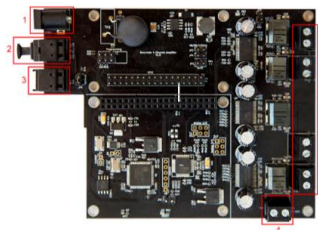
Materiallesæt

P1 – efteråret 2019

Projekt støttet af:

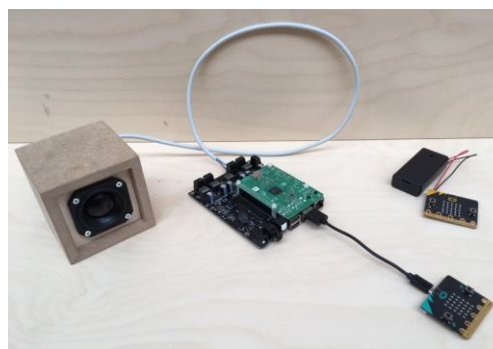
midt
regionmidtjylland

Forstærker modul med DSP



Visuel programmering af DSP

Højttaler og Micro:Bit



Raspberry PI med general purpose uP



```
File Edit Format Run Options Window Help
# Import af biblioteker med ekstra funktioner
# Afspil af "Hello World" som lyd

import pygame
import time

Helloworld = "helloworld.wav"

# initialisering af mixer
pygame.mixer.init()
# spil lyden med bestemt volumen
pygame.mixer.music.set_volume(0.1)
pygame.mixer.music.load(Helloworld)
pygame.mixer.music.play(0)

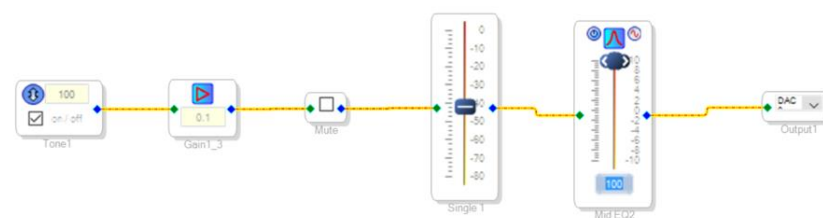
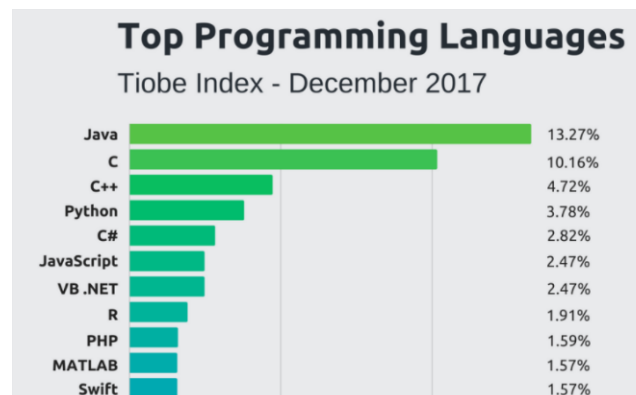
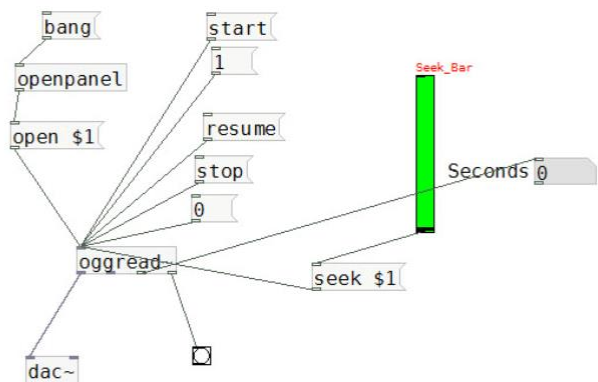
# Vent til lyden er afspillet.
time.sleep(5)

# Ryd op i mixer
pygame.quit()
```

Tekst baseret programmering med lyd

Programmeringssprog

Projekt støttet af:



text file named HelloWorld.java

```

public class HelloWorld {
    public static void main(String[] args) {
        // Prints "Hello, World" in the terminal window.
        System.out.print("Hello, World");
    }
}
    
```

Annotations in the image:

- name**: points to `HelloWorld`
- main() method**: points to `main`
- statements**: points to the code inside the `main` method block.
- body**: points to the `System.out.print` statement.

LKM1: Hands-on

<https://thonny.org/>

<https://github.com/TPSoundhub/LKM1>

Mere om Python kan I finde her:

<https://docs.python.org/3/>

Generelt så kan I få meget ud af at stille 'Google' spørgsmål på engelsk. Ex.:

"How to make loop in python".

"How to generate random number in python"

Etc....

Så kommer der ofte gode eksempler i diverse links. Kan dog anbefale et par steder til introduktion:

<https://www.learnpython.org/en/>

<https://wiki.python.org/moin/>

<https://realpython.com/>

LKM1: Første program

Kører og ser at der kommer "hej med jer" ud i shell

Første lille funktion som så kan kaldes flere gange og gøre koden 'effektiv' Kaldes med pn("xxx") flere gange

En anden funktion med tal i stedet for tekst.
Introduktion til betingelse if/elif/else

":" og indrykning vigtige for at 'python'/Thonny kan forstå det!! Det som er indrykket hører til funktion/betingelse

```
print ("hej med jer")

def pn(t):
    print("Hej med dig: ")
    print(t)
    print("Godt at se dig")
    print("")

def lms(n):
    if n<10:
        print("tal under 10")
    elif n == 10:
        print("tal er 10")
    else:
        print("tal over 10")
```

LKM1: Introduktion til lister/løkker

Køre og se at man kan pille et tegn ud af en streng

At man kan lave en liste med strenge

Og se at man kan udpege en enkelt i listen

Fra 0 til 3 når der er 4 i listen.

Bruger funktionen "pn" som blev defineret i forrige trin

At man kan lave en for-løkke som går igennem alle elementerne i en liste og udføre en funktion derpå

```
jens = "Jens"
print(jens[2])

nl = ["Jens", "Peter", "Lotte", "Marie"]

print(nl[3])

pn(nl[1])

for i in range(4):    # da vi ved listen har 4 indgange.
    pn(nl[i])         # Kunne også bruge len(nl) som er mere generel.
```

LKM1: Introduktion til biblioteker

Der findes masser af foruddefinerede funktioner i biblioteker man kan importere og bruge i sin egen kode.
Her time og random til at lave stop/pauser i program samt til at udvælge/generere tilfældigt

```
import random  
import time
```

En while-løkke som fortsætter så længe betingelsen er sand.
Her sat til True så den aldrig stopper!
Bruger random.choice til at vælge en tilfældig indgang i tabel fra forrige trin.
og funktion pn fra første trin til at udskrive.
Bruger time.sleep(5) til at holde en pause på 5 sekunder mellem hvert kald af funktionen pn

```
while True:  
    pn(random.choice(nl))  
    time.sleep(5)
```


LKM1: Introduktion til pygame(sound)

```
import pygame

pygame.mixer.init()
pygame.mixer.music.set_volume(0.5)
pygame.mixer.music.load("hej-med-jer.wav") # works with the file in same directory as the program else use full path name
pygame.mixer.music.play()
```

Bruger mixer i pygame biblioteket til at starte afspilning af en længere lyd-fil med halv volume.
Kan ex. bruges til at starte en baggrunds lyd.

Pygame.mixer.init() skal bare kaldes én gang i begyndelsen af programmet.
Resten kan bruges flere gange til at starte flere lyde. Man kan evt. lave en funktion til det

BEMÆRK:

At når lydfilen er i samme direktorier som det program man skriver, så er der ikke behov for sti-navnet. Ligger lyd filen et andet sted skal man have udpeget lydfilen med det fuld sti navn.

Så er der problemer med at finde den rigtige sti så kan man 'bare' gemme lyd filer samme sted som programmerne.

LKM1: Introduktion til pygame(sound)

Definerer en funktion der bruger mixer i pygame til at :

- Finde en kanal (i stereo)
- Sætte volume i hhv. højre og venstre udgang for kanal
- Spiller én lyd udpeget med "hejmeddig" som lydobjekt (der kan være flere samtidigt og 'oveni' det som spiller i 'music' fra forrige trin)
- Venter 1 sek til lyd er færdig
(Da vi ved at det er en kort lyd 😊. Kan laves smartere ved at spørge til længden af lyden)
- Gentager samme historie med en lyd der kommer til funktionen som parameter.

```
import pygame
```

```
def hmd(n,vl,vr):  
    channel = pygame.mixer.find_channel()  
    channel.set_volume(vl,vr)  
    channel.play(pygame.mixer.Sound(hejmeddig))  
    time.sleep(1)  
    channel = pygame.mixer.find_channel()  
    channel.set_volume(vr,vl)  
    channel.play(pygame.mixer.Sound(n))  
    time.sleep(1)
```

LKM1: Introduktion Micro:Bit/Serial

Åbn den serielle kanal som MB er på!
Find den med **M1Sc-VisSeriel.py**

Funktion til at læse tegn fra den serielle.
Med return afleverer funktionen tegn modtaget fra Micro:Bit
Der modtages en hel tekst streng som også indeholder navn.

Uendelig løkke der læser tegn fra Micro:Bit og udskriver den

```
import serial

ser = serial.Serial()

ser.baudrate = 115200
ser.port = "COM6" # på MAC: "/dev/tty.usbmodem14102"
                # eller  "/dev/cu.usbmodem14102"

ser.open()
```

```
test_print = True

def receive_char():
    microbitdata = str(ser.readline())
    if test_print:
        print(microbitdata)
    return microbitdata[2]
```

```
while True:
    rc = receive_char()
    if test_print:
        print("Modtaget tegn: ",rc)
    ...
```

LKM1: Input's fra Micro:Bit

```
# Microbit functionality in short:
#
# Send string "Hello world ..." on serial USB and make Happy face on LEDs when powerup
# Send characters on serial USB when sensors/buttons are triggered
#
# - "A", "B" when a/b key is pressed - plus "A" "B" on local display - Once pr press
# - "C" when pin 0 is activated - repetitive as long as pin 0 is active with 500ms spacing
# - "O" - "G" randomly selected when pin 1 is active - repetitive 500ms spacing
# - "V" - when tilted to left - repetitive 500ms spacing
# - "H" - When tilted to right - repetitive 500ms spacing
# - "F" - when tilted forward - repetitive 500ms spacing
# - "T" - when tilted backwards - repetitive 500ms spacing
# - "D" - when light intensity goes from outside range into range between 40 and 180. (Skift til Dag)
# - "N" - when light intensity goes from above to below 40. (Skift til Nat)
# - "S" - when light intensity goes from below to above 180. (Skift til Solskin>Dag)
#
# - Switch mode between sending on radio vs sending on serial when pin 2 is active
# - small square followed by big square -> Radio transmission from remote location
# - big square followed by small square -> Serial transmission local connected via USB
# If transmission on radio then feedback on LED are in small letters "a", "b" etc..
# - and so is the character transmitted.
# - Digits not transmitted - so no action on pin 1 when in radio transmission mode.
#
# If anything is received on radio the received character is sent on serial USB. No feedback on LEDs
# Feedback is given on the remote MicroBit
#
# All characters are followed by name the one having the Micro:Bit
```

LKM1: BYG

Kombiner og byg -

Lav kode der afspiller lyde afhængigt af input fra Micro:Bitten

Definer lydene – konstanter med reference til lydfilerne inklusiv fuld sti med mindre de ligger samme sted som programmet I laver
Brug funktionerne `receive_char` og `play_sound` til at lave noget som reagerer på input fra Micro:Bitten.

```
def play_sound(n,vl,vr):  
    channel = pygame.mixer.find_channel()  
    channel.set_volume(vl,vr)  
    channel.play(pygame.mixer.Sound(n))  
  
def receive_char():  
    microbitdata =str(ser.readline())  
    if test_print:  
        print(microbitdata)  
    return microbitdata[2]
```

```
pygame.mixer.init()  
music_volume = 0.1  
  
pygame.mixer.music.load(baggrundslyd)  
pygame.mixer.music.set_volume(music_volume)  
pygame.mixer.music.play(-1)
```

```
while True:  
    rc = receive_char()  
    print(rc)  
    if rc == "A":  
        play_sound(lyd1,0.1,0.7)  
    elif rc == "B":  
        ....
```

. Byg videre