

## Opsummering af undervisningsforløb 15/11 – KITpilot – uden illustrationer

### **Forudsætning:**

x Undervisnings stationer etableret som beskrevet i 'lærer kagebog'.

Elever er fordelt i grupper, som har arbejdet med projektopgave og formulering inden de introduceres til værktøjs gennemgang. Roller fordelt med hensyn til hvem der skal sidde ved tastaturet på opstillingerne.

### **Mål:**

Gennem et styret forløb med underviser på 4 lektioner etableres en interaktiv opstilling på alle gruppers undervisningsstationer. Knapper forbindes til en Beocreate/Raspberry PI. Der laves programmer til at reagere på knap tryk og medfører at forskellige lyde mixes og afspilles.

Undervejs introduceres:

- Basale programmerings teknikker i Python, og et integreret udviklingsmiljø (IDLE) der kan bruges på PC/MAC smat Raspberry PI i opstillingerne.
- At kunne køre programmer uden tilsluttet skærm og tastatur (headless), samt at tilgå opstilling via LAN.
- Elementerne i en højttaler og digital signalbehandling.

Efter forløbet kan man selv arbejde videre med opstillingen og bruge den til projektopgaven. I kan gå dybere i programmering hvis I har tid og lyst. Dels på egne PC/MAC derhjemme og dels på opstillingerne i skolen.

### **Trin:**

1. Undervisnings stationerne indeholder:
  - a. 2 programmerbare moduler.
    - i. Raspberry PI, som kan sammenlignes med en PC når der er koblet skærm og tastatur til. Man kan koble den til routere, gå på nettet osv.
    - ii. Beocreate board, som er en forstærker med 4 kanaler ud og en digital signalprocessor (DSP) on-board.
  - b. Man kan koble forskellige dimser til - ex. knapper, som kan give input til programmerne via hvad man kalder General purpose I/O (GPIO's) - Mere om det senere. I opstillingen er der 3 trykknapper.
  - c. Man kan også koble 1-4 højttaler enheder til så man kan lave forskellige kombinationer. I opstillingen er der en enkelt enhed og et kabinet. Baggrundsinformation kan findes på nedenstående links, men i dette projekt bør I ikke gå dybt i det.  
<https://www.hifiberry.com/beocreate/beocreate-doc/beocreate-use-sigmastudio/>  
<https://www.raspberrypi.org/>

2. Python er et ud af mange programmeringssprog. Se hjemmesiden for dokumentation og tutorials:  
<https://docs.python.org/3/>  
Download Python på egen MAC/PC:  
<https://www.python.org/downloads/>
3. Hello world - Første lille program, funktioner og introduktion til debugging. (som i "HelloW.py")
4. Biblioteker med funktioner som man kan bygge videre på, så man ikke skal lave alt fra bunden hver gang.  
Installation af Pygame bibliotek på MAC/PC via cmd linje på PC (MAC ??).  
Indtast følgende i kommando linje:  
**"python3 -m pip install -U pygame --user"** - På PC evt. "py -m pip install -U pygame --user" istedet  
reference: <https://www.pygame.org/wiki/GettingStarted>  
Dokumentation/manualerfor biblioteket findes: <https://www.pygame.org/docs/>
5. Hello world - Nu med Lyd. Reference: <https://www.pygame.org/docs/ref/mixer.html>
  - a. Eksempler fra Github udpakkes til PC/MAC i 'rod direktorie'  
<https://github.com/TPSoundhub/KITpilot>
  - b. Sørg for at få "Hej-1sek.wav", "Baggrund.wav", "Lyd1.wav" og "Lyd2.wav" lagt op under KITPilot-master
  - c. Program med lyd mix sammen på klassen. (som HelloW2.py)
6. Introduktion til Knapper og PINs og skift til opstillingerne.
  - a. Bemærk at PINs på PI ikke er direkte tilgængelige, men nogle er via Beocreate modulet!

PIN nummerering på Raspberry PI - se: <https://pinout.xyz/#>

PIN nummerering på Beocreate modulet se : <https://www.hifiberry.com/beocreate/beocreate-doc/beocreate-gpios/>

**BRUG følgende PIN's på BeoCreate modulet ifm GPIO til Raspberry PI modulet og SW i Python:**

**PIN 10 for 3.3V**

**PIN 2 eller 34 for GND**

**PIN 31 til 'første knap' (samme pin nummer på RPI (31) - den generelle GPIO6)**

- Ved reference til PIN nummer i SW så er det '31' og samme i HW - ingen forvirring!

**PIN 29 til 'anden knap' (også samme pin nummer på RPI ('29' - den generelle GPIO5)**

**Dernæst:**

**PIN 26 - GPIO23 (PIN 16 på RPI)**

- Ved reference i SW til PIN nummer og ikke GPIO så skal '16' bruges men fysisk på Beocreate er det PIN 26!!

**PIN 27 - GPIO24 (PIN 18 på RPI)**

- Ved reference i SW til PIN nummer og ikke GPIO så skal '18' bruges men fysisk på Beocreate er det PIN 27!!

**PAS PÅ MED AT BRUGE FORKERTE PIN's, da man kan komme til at ødelægge elektronikken ved forkert brug!**

- b. Lave knap 'fumble' board med 3 knapper og pull down modstand. Ref: [Raspberry Pi: Using GPIO Inputs](#)
  - c. Tilslut knap board op til Beocreate modul.
  - d. Tænd for opstilling
7. Programmer med knapper på Raspberry PI.
  - a. Rasbian GUI med samme IDLE og Python muligheder som på PC/MAC, men GPIO delen er særlig for PI da PC/MAC ikke har den samme PIN HW.
  - b. Rpi.GPIO bibliotek, knap programmering - forskel på polling og interrupt/event styret tilgang. (Hoveddøren som eksempel)  
Reference: <https://raspberrypi.hq.com/use-a-push-button-with-raspberry-pi-gpio/>
  - c. Fælles programmering af poll (som i "Key\_polling.py")
  - d. Fælles programmering af event (som i "Key\_event.py")
  - e. Kombiner lyd og knapper til endeligt eksempel (som i "WithHK.py")
8. Headless, fjern adgang og start program fra boot.
  - a. Køre program fra kommando linje på PI (og man kan det samme fra PC/MAC via Putty/SSH koblet på samme net/wifi).
  - b. Indsæt program/script "HelloW2.py" i filen "rc.local" som er placeret i folderen "Etc"  
Man editere med simpel editor kaldet nano. (Brug "sudo nano rc.local" i etc direktoriet). Husk "&" tegn til sidst i indsat tekst streng. Det som skal indsættes i bunden af rc.local er (alternativt program navn hvis I har gemt et):  
"python /home/pi/KITpilot-master/WithHK.py &" ( " - tegnene skal ikke med!!)
  - c. Gå ind i config (med kommandoen "sudo raspi-config") og sæt PI op til at starte uden GUI med bruger PI.
  - d. Genstart PI. Knap programmet "WithHK.py" kører fra Boot og virker uden at skærm og tastatur skal bruges.
  - e. Modsat udkommenter/fjern linjen fra rc.local og reboot med GUI
9. "Keyboard.py" er et eksempel på program/script som kan køre både på PC/MAC og PI, hvor knapper på tastatur bliver brugt i stedet for hjemmebygget HW. Det kan I bruge/lege videre med derhjemme. Programmet bruger pygame biblioteket, samt det med lydene fra foregående eksempler. Se <https://www.pygame.org/docs/ref/key.html>  
På PC/MAC skal I ikke have sti til lydfile defineret som på PI - så det skal lige rettes til!  
Funktionen er :
  - a. Baggrundslyd spiller fra start - Kan genstartes med tast "P"
  - b. Lyd1 spiller når tast "1" aktiveres.
  - c. Lyd2 spiller når tast "2" aktiveres.
  - d. Baggrundslyd pauses/genstartes med mellemrums tast.
  - e. Lydniveau på Baggrundslyd kan justeres med gentagne tryk på pil op og pil ned.
  - f. Program stoppes med tast "Esc"
10. Program/Script "Hentlyd.py" henter "Baggrund.wav", "lyd1.wav" og "lyd2.wav" fra Github og placerer dem i defineret sti. Så med skrive adgang til Github kan man den vej rundt skifte lydene ud som bruges. Så med en headless PI, der ved fjern adgang eller ved en reboot henter nye lyde, og med sensorer i stedet for knapper, så har I samlet set de 'stumper' der skal til for at lave noget som svarer til lydens bæk i lydens by. I kan evt. lave en egen Github og eksperimentere, men det er ikke nødvendigt for projektopgaven.

11. introduktion til signal behandling og HT trimning

- a. Spil lyd fra eget program eller alternativt fra web browser Dr.dk eller youtube
- b. Forskel i lyd med højtaler enhed udenfor og indenfor kabinet. (Højtaler enhed monteres i kabinet undervejs)
- c. Start sigma studio på PC på nettet og load eksempel på DSP program.
- d. Parametre ændres så der kan høres forskel. Det kan bruges til at trimme højtaler til kabinet, omgivelser og materiale, men det går vi ikke dybere i nu, og er heller ikke tænkt til at skulle bruges i projekt. Men på et senere semester :-)