

De første skridt

Åbn Mu-editoren og tilslut M5StickC via det medfølgende usb-kabel. Mu skulle gerne selv opdage M5StickC. Nederst i højre hjørne skal der stå: **ESP MicroPython** .

Når det er på plads er du klar til at gå videre.

Første program

Slet den linje hvor der står **#Write your code here :-)** og skriv i stedet følgende linjer:

```
1  from m5stack import lcd
2
3  lcd.orient(lcd.LANDSCAPE)
4  lcd.clear(0xFF0000)
```

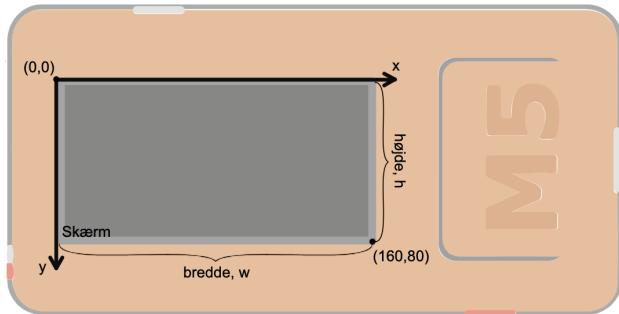
tryk på **Run**  og se hvad der sker.

Prøv at erstatte det der står i parantesen på linje 4 med 0x00FF00 og tryk på **Run** 
 Prøv det samme med 0x0000FF og 0xFFFF00

Kan du få skærmen til at blive lilla? Hvad med sort?

TIP: søger på "HEX color picker" på Google

Tekst på skærmen



For at placere tekst på skærmen skal vi give koordinaterne til hvor teksten skal placeres, sammen med den ønskede tekst. M5StickCs koordinatsystem vender lidt på hovedet i forhold til dem I kender fra matematik. Punktet (0,0) er i øverste venstre hjørne, og y-aksen vokser nedad. Men principippet er det samme; et punkt har et x og et y koordinat.

Tilføj en 5. linje til din kode hvor der står

```
5  lcd.text(10, 10, "Hello!")
```

Øvelse

Skriv dit navn på midten af skærmen!

Gem den kode du har lavet ved at trykke **Save** . Giv din kode navnet "lcd.py"

Tegning

Man kan også tegne figurer og linjer på skærmen.

Klik på **New**  for at begynde en ny kode.

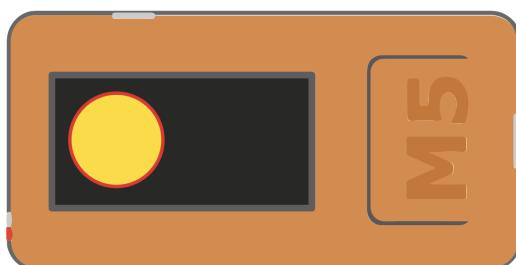
Slet den linje hvor der står *#Write your code here :-)* og skriv i stedet følgende linjer:

```
1  from m5stack import lcd
2
3  lcd.orient(lcd.LANDSCAPE)
4  lcd.clear(0xFFFFFFF)
5
6  lcd.circle(40, 50, 30, fillcolor=0xffffcf00)
7  lcd.line(25,60,55,60,color=0x000000)
8
9  lcd.circle(30, 40, 10, fillcolor=0xFFFFFFFF)
10 lcd.circle(50, 40, 10, fillcolor=0xFFFFFFFF)
11 lcd.circle(30, 43, 4, fillcolor=0x000000)
12 lcd.circle(50, 43, 4, fillcolor=0x000000)
```

Tryk på **Run** 

Man kan farve sine figurer på med **color** og **fillcolor**. **Color** er farven man vil have stregen, eller omridset skal have. **Fillcolor** er hvad figuren skal fyldes med.

Hvis man f.eks. skrev: *lcd.circle(40, 40, 30, color=0xFF0000, fillcolor=0xFFCF00)* ville man få en gul cirkel med et rødt omruds.



Hvis man ikke skriver nogen fillcolor, får man bare et omruds.

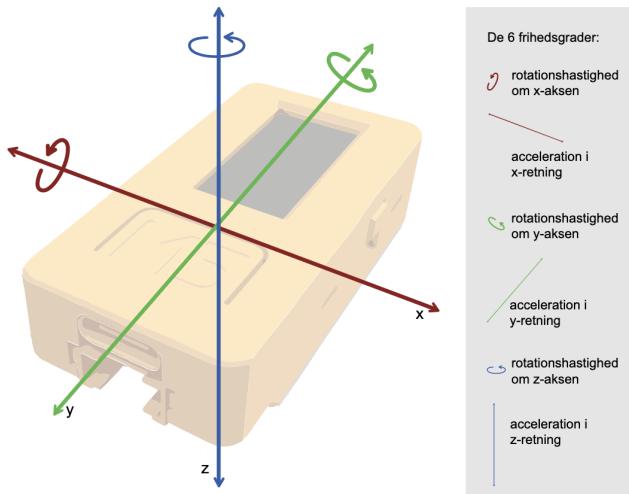
Øvelse

Tegn endnu et ansigt i højre side af skærmen.

Gem den kode du har lavet ved at trykke **Save**  . Giv din kode navnet "emoji.py"

Tip: På <https://m5guide.readthedocs.io/da/latest/tegne.html> kan I finde hjælp til at tegne figurer og farve dem.

Bevægelsessensor



M5StickC har en indbygget bevægelsessensor, kaldet en **IMU**, og den skal vi bruge.

Klik på **New**  for at begynde en ny kode.

Slet den linje hvor der står *#Write your code here :-)* og skriv i stedet følgende:

```
1 import imu
2 sensor = imu.IMU()
3
4 while True:
5     print(sensor.acceleration)
```

Klik på **Run**  og derefter på **Plotter** .

Øvelse - acceleration

Prøv at lade den ligge stille, fladt på bordet. Prøv at vende den så den står på højkant og på siden. Bevæg den lidt. Kan du regne ud hvad den mäter? Skriv dit bedste bud herunder.

(Undgå at skrive det åbenlyse: "acceleration" - overvej hvad acceleration egentlig betyder)

sensor.acceleration mäter: _____

Øvelse - gyroskop

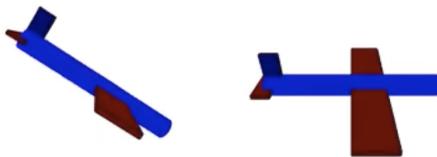
Erstat linje 5 med:

```
5     print(sensor.gyro)
```

Klik på **Run**  og derefter på **Plotter**. Prøv at bevæge M5StickC og se om du kan regne ud hvad den mäter? Skriv dit bedste bud herunder.

sensor.gyro mäter: _____

Pitch og Roll



Pitch: -33°

Roll: -33°

De rå data fra IMU'en er lidt svære at aflæse, men hvis man kombinere accelerometerdata med gyroskopodata, kan man regne det der hedder *pitch* og *roll* ud.
Hvis man tænker på et fly, så er pitch et udtryk for hvad vej spidsen af flyet peger, mens roll er et udtryk for hvor meget flyet har roteret.

Klik på New  for at begynde en ny kode og skriv:

```

1  from m5stack import lcd
2  import imu
3  import fusion
4
5  myIMU = imu.IMU()
6  filter = fusion.MahonyFilter()
7  lcd.orient(lcd.LANDSCAPE)
8
9  while True:
10     filter.update(myIMU.acceleration, myIMU.gyro)
11     pitch = str(int(filter.pitch))
12     roll = str(int(filter.roll))
13     print("pitch: " + pitch + " roll: " + roll)

```

Klik på Run 

Lige nu bliver pitch og roll skrevet ud i Mu - men vi vil gerne kunne aflæse det på M5StickC.

Opgave



Forsøg ved hjælp af `lcd.text`, om du kan skrive noget kode, så man kan aflæse pitch og roll på uret.

Du kan starte på linje 14, sådan her:

```
14      lcd.text(10, 10, "Pitch: ")
```

Højtalere



helnode A, frekvens = 440, nodeværdi = 1/1



halvnode A, frekvens = 440, nodeværdi = 1/2



4.delsnode A, frekvens = 440, nodeværdi = 1/4



8.delsnode A, frekvens = 440, nodeværdi = 1/8



16.delsnode A, frekvens = 440, nodeværdi = 1/16



32.delsnode A, frekvens = 440, nodeværdi = 1/32

Klik på New

```

1 from m5stack import lcd
2 from flowlib import hat
3 import time
4
5 spk = hat.get(hat.SPEAKER)
6
7 spk.sing(440, 1/4)
8 time.sleep_ms(100)
9 spk.sing(440, 1/4)

```

Klik på Run

Den tone du hører er kammertonen "A". Tonens frekvens er 440 Hz, og nodeværdien er $\frac{1}{4}$ og det er netop de to ting vi har brug for at bestemme, for at få afspillet en tone.

Opgave

Starten af Lille Peter Edderkop kan spilles ved hjælp af 3 toner; et **C** et **D** og et **E**.

Den starter på **C**.

Tonen C har en frekvens på 523 Hz, D har en frekvens på 587 Hz og E har en frekvens på 659 Hz.

Hvor langt kan du komme i sangen?

```

C   C   C D   E E   E
Lille Peter Edderkop
D   D   D   E   C C
Kravled' op ad muren

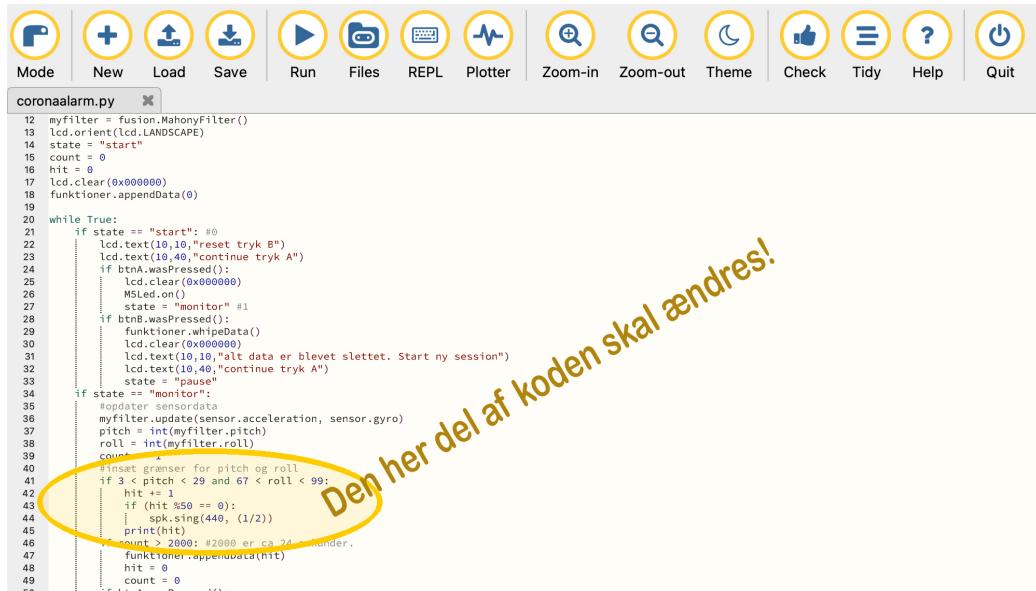
```

Lyst til mere musik? Du kan finde en oversigt over alle tonernes frekvenser her:
<https://m5guide.readthedocs.io/da/latest/hat.html#frek>

Kode til overvågning

Gå ind på <https://github.com/Majahh/smartwatch/blob/main/coronaalarm.py>

Marker **al** koden og kopier ved at trykke **ctrl+c**. Åben en ny fil i Mu **New**  og indsæt alt det du har kopieret ved at trykke **ctrl+v**



```

12 myfilter = fusion.MahonyFilter()
13 lcd.orient(lcd.LANDSCAPE)
14 state = "start"
15 count = 0
16 hit = 0
17 lcd.clear(0x000000)
18 funktioner.appendData(0)
19
20 while True:
21     if state == "start": #0
22         lcd.text(10,10,"reset tryk B")
23         lcd.text(10,40,"continue tryk A")
24         if btnA.wasPressed():
25             lcd.clear(0x000000)
26             M5Led.on()
27             state = "monitor" #1
28         if btnB.wasPressed():
29             funktioner.whipeData()
30             lcd.clear(0x000000)
31             lcd.text(10,10,"Alt data er blevet slettet. Start ny session")
32             lcd.text(10,40,"continue tryk A")
33             state = "pause"
34     if state == "monitor":
35         #opdater sensordata
36         #filter
37         #acceleration, sensor.gyro
38         pitch = int(myfilter.pitch)
39         roll = int(myfilter.roll)
40         count = 0
41         #Indsat grænser for pitch og roll
42         if 3 < pitch < 29 and 67 < roll < 99:
43             hit += 1
44             if (hit %50 == 0):
45                 spk.sing(440, (1/2))
46             count > 2000: #2000 er ca 24 minutter.
47             funktioner.appendData(hit)
48             hit = 0
49             count = 0

```

Jeres egne fundne grænsen for pitch og roll sættes ind på den linje hvor der står :

`if 3 < pitch < 29 and 67 < roll < 99:`

Øvelse

Læs koden igennem og finde de steder I kan personliggøre koden - I kan ændre i den tekst der bliver skrevet på skærmen eller på farverne. Kig efter `lcd.text` og Hexidecimaltal ala `0xF00FFF`

Husk at trykke **Run** , hver gang I har ændret i koden for at tjekke at det stadig kan køres (og for at se om jeres ændring virkede).

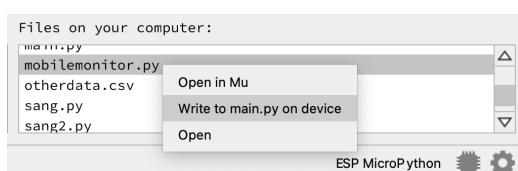
Overfør koden til uret

For at kunne køre koden selvstændig på uret, uden det er tilsluttet computer, skal det overføres til uret.

Først skal du gemme din kode - tryk **Save** . Giv din kode navnet "mobilemonitor.py".

Sørg for at REPL er lukket ved at klikke på **REPL** . Nu kan du åbne for filerne, der ligger på din M5StickC. Klik på **Files** 

Til højre ser du en liste med filer gemt på computeren - højreklik på filen "mobilemonitor.py" og vælg "Write to main.py on device".



Forberedelse

Her er de 5 skridt I skal tage for at sætte det hele igang:

- Oplad uret
- Lad forsøgspersonen læse bagsiden af dette ark.
- Påsæt uret på den hånd I har lavet programmet til og sørge for at det sidder stramt
- Start uret - den røde led lampe skal lyse.
- Sæt alarm/tidtagning til en time på din mobil

Udførelse

Når uret sidder på personen, skal du prøve at undgå at blande dig, så personen ikke føler sig direkte overvåget

Hvis uret løber tør for strøm undervejs (den røde led slukker), så stop tidtagningen og lad uret op. Næste gang uret startes vælges "continue" og eksperimentet kan færdiggøres. Noter tiderne på arket

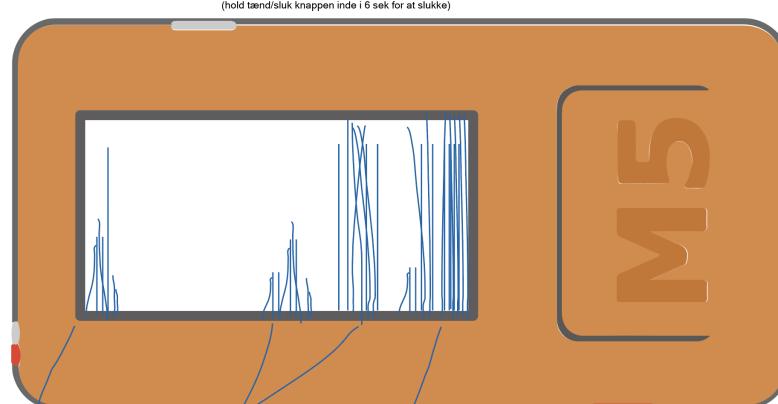
Når tiden er gået: tag gerne et billede af M5StickCs display med din mobil, så I har en backup, hvis uret løbet tør for strøm, eller I kommer til at slette data.

Aflæsning af opsamlede data

Brug nu det udleverede ark til at få gang i en diskussion om data. Udfyld de ting der skal udfyldes og overvej, for hvert spørgsmål, hvad der taler for/imod det der svares.

Tegn så godt du kan, det billede der vises på M5StickCs display. Alle de steder hvor der er meget grønt, skal I i følelseskab - blive enige om hvilken aktivitet der blev lavet. Noter på arket.

Uret påsat: 20:14 Uret stoppet: 21:30
(hold tænd/sluk knappen inde i 6 sek for at slukke)



1) kiggede på uret på telefonen	4) tjekkede mails	Ville du gå med uret hvis det kun var dig selv der kunne se dine data?
2) vaskede op?	5)	Ville du gå med uret hvis dine data kunne ses af din familie?
3) på toilet	6)	Ville du gå med uret hvis dine data ikke kunne ses af din familie, men kun af nogen der ikke kender dig?
Ville du gå med uret hvis dataopsamlingen var anonymiseret?		

Hvis noget driller eller ikke vil virke, så ta' det helt roligt og prøv eventuelt at gentage eksperimentet. Hvis din forsøgsperson ikke synes det er sjovt at være med, så skal du også bare stoppe.

Når I er færdige med opgaven lægger I det udfyldte ark i tasken - så har I det automatisk med næste gang vi ses!

Til forsøgspersonen:

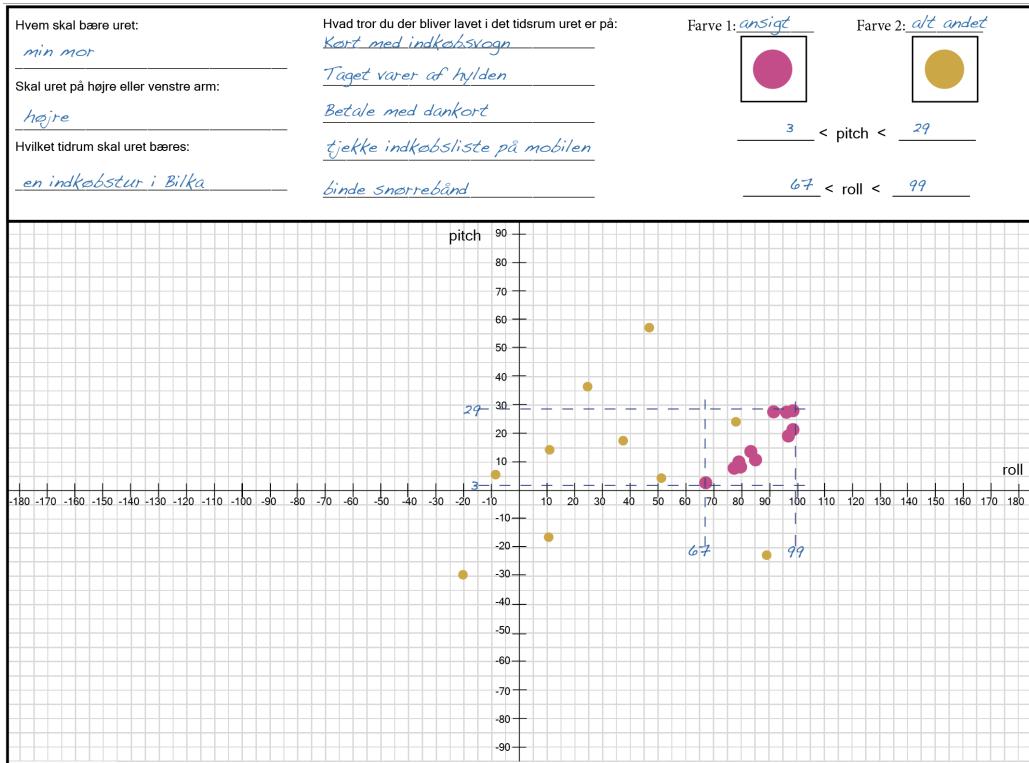
Dette er et forsøgsprojekt på Datalogisk Institut på Københavns Universitet der sætte fokus på data, overvågning og etik - med det mere specifikke formål at finde en relevant og vedkommende måde at undervise folkeskoleelever i emnet.

De data der bliver indsamlet her bliver ikke gemt nogen steder.

Det er meningen eksperimentet skal sætte gang i en debat mellem forældre og unge om dataovervågning og etik. Hvis det undervejs føles ubehageligt at blive overvåget, skal I bare stoppe eksperimentet og i stedet snakke om hvorfor I stoppede det.

I er velkomne til at kontakte Maja Hvidtfeldt Håkansson på mhv@di.ku.dk ved spørgsmål

Aflæs koordinaterne



Tag det udleverede ark med koordinatsystemet og to tusher i hver sin farve.

Snak om hvem der skal have uret på når det skal med hjem - og prøve at regne ud om uret skal sidde på venstre eller højre hånd for at få det mest brugbare resultat.

En af jer skal nu have uret på og lade som om hen kigger på sin mobil. Den anden notere i koordinatsystemet på arket ved at sætte en prik. Prøv lidt forskellige håndstillinger og noter mindst 10 punkter.

Tag den anden farve og gør det samme, men forstil jer at I laver alt muligt andet end at kigge på en mobil. Feks, skriver på computer, drikker af en kop, tømmer vaskemaskinen, eller andre aktiviteter I forstiller jer at jeres testperson kunne udføre. Hvis nogen af de prikker I sætter her, kommer tæt på de prikker I satte for mobilbrug, så lav gerne en lille note, så I kan huske hvad den aktivitet var.

Byt om, så den der ikke havde uret på før, har uret på og den anden skriver ned.

Når I begge har et udfyldt koordinatsystem, skal I ramme tallene for mobil-kiggeriet ind og forlænge linjerne til de møder x og y-aksen. I kan nu aflæse det interval jeres mobil-forbrug-data ligger indefor. Noter det på arket ved hjælp af uligheder.