

# Python med M5Stack Core 2

Af: Michael Hansen og Jarl Friis, Coding Pirates Furesø, 2021, version 0.91

Dokumentet her: <https://github.com/mhfalken/m5stack/blob/main/m5stack-python.pdf>



Formålet er at lære at programmering i Python, hvor man bruger en M5Stack Core 2 som platform. Selve dokumentet er baseret på 'learning by doing', dvs der er ikke meget Python undervisning i selve dokumentet. Jeg har lavet et præsentationssæt som gennemgår de vigtigste ting i Python og som også indeholder lidt generelt viden omkring embedded systemer.

Før første brug er det vigtigt at man har installeret den rigtige firmware på M5Stack'en. Se kapitel 10.

1	Installation af editor på desktop .....	3
1.1	Visual Studio Code editor .....	3
2	Hello world .....	4
3	Tekst og grafik.....	5
3.1	Tekst .....	5
3.2	Grafik .....	5
3.3	Touchskærm .....	6
4	Gyro .....	6
5	SD kort, lyd og billeder .....	6
6	Ekstern hardware (Grove) .....	7
7	Labyrint spil .....	7
7.1	Flyt kuglen .....	7
7.2	Lav labyrint .....	7
7.3	Selve spillet.....	8
8	Ideer.....	8
9	M5Stack hardware overblik.....	9
9.1	M5Stack Core 2.....	9
10	Installering af M5stack Core 2 .....	9
10.1	M5Burner .....	10
11	API reference .....	10
11.1	Generelle tips .....	10
11.2	Problemer .....	10
11.3	Funktions overview .....	11
12	Løsninger .....	12
13	Links .....	13

# 1 Installation af editor på desktop

For at programmere M5stack'en i Python er det nødvendigt at installere Visual Studio Code (VSC) editor på sin computer.

## 1.1 Visual Studio Code editor

### Windows & Mac

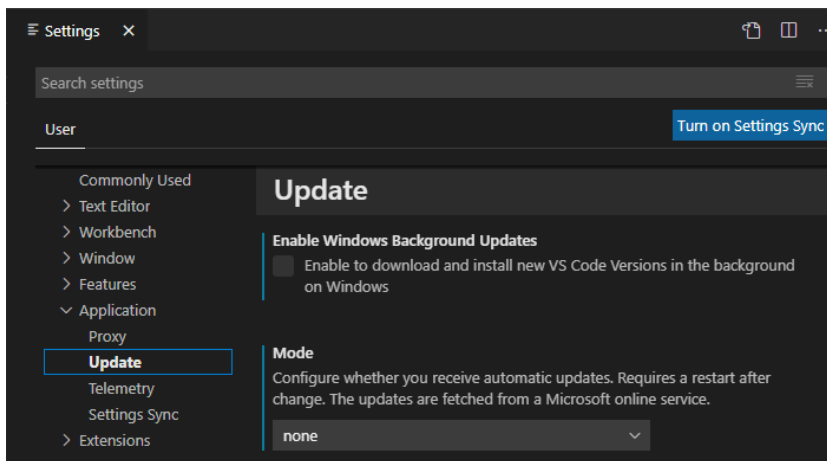
Følg instrukserne for at installere Visual Studio Code på <https://code.visualstudio.com/Download>

**OBS:** I mange tilfælde så virker M5Stack plug'in ikke (se senere) og det kan derfor være nødvendigt at installere en ældre version af VSC. Dette gøres på følgende måde:

Man skal installere version 1.52.1: [https://code.visualstudio.com/updates/v1\\_52](https://code.visualstudio.com/updates/v1_52) (User)

Derefter er det vigtigt at man forhindrer den i at opdatere sig selv. Start VSC og tryk på **File->Preferences->Settings**

Fjern 'Enable Windows Background Update' og set Mode til 'None' (se billede).



### Linux



Hent den seneste debian pakke på <https://code.visualstudio.com/sha/download?build=stable&os=linux-deb-x64> og installer den (den sørger også for en ny software kilde).

Den installeres med

```
$ sudo dpkg -i code_<version>.deb
```

#### 1.1.1 M5Stack udvidelsen

Efter installation af VSC skal man installere en M5 stack plugin som hedder `vscode-m5stack-mpy`.

Start VSC editoren tryk på  ude til venstre og skriv **M5Stack** og installer plug-in'en. Efter installeringen, så kommer der en knap frem i bunden af editoren: **Add M5Stack**. Tilslut M5Stack til USB porten og tryk på knappen **Add M5Stack** og vælg USB porten (COM eller /dev/ttyUSBx). Herefter kan man se filerne på enheden ved vælge Explorer (CTRL+SHIFT+E) eller .

Nu er der forbindelse og vi kan oprette filer på enheden og flytte filer til enheden.

### 1.1.2 USB drivers (kun Windows)

For at kommunikere med M5stack er det muligvis nødvendigt at installere de rigtige USB-drivere.

Windows kan have disse installeret allerede, men ellers kan de findes her:

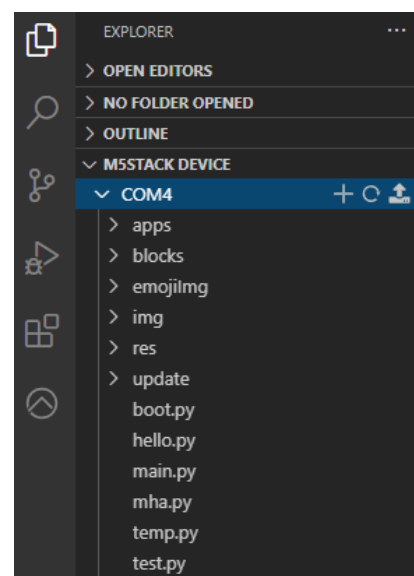
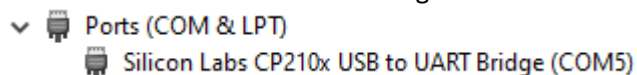
**USB driver:**

[http://www.ftdichip.com/Drivers/CDM/CDM21228\\_Setup.zip](http://www.ftdichip.com/Drivers/CDM/CDM21228_Setup.zip). Filen pakkes ud og programmet køres for at installere driveren.

**USB bridge:**

[https://www.silabs.com/documents/public/software/CP210x\\_Universal\\_Windows\\_Driver.zip](https://www.silabs.com/documents/public/software/CP210x_Universal_Windows_Driver.zip). Files pakkes ud og filen CP210xVCPInstaller\_x64.exe køres for at installere bridgen.

Det skal se sådan ud i Device Manageren:



### 1.1.3 UIFlow

Dette program er ikke nødvendigt, men kun omtalt, da der kan være godt til at 'gætte' hvordan man programmere hardwaren. Det er ikke yderligere omtalt i dette dokument.

UIFlow desktop IDE (til visuel programmering)

UIFlow desktop er et program til at programmere M5stack på en mere intuitiv måde med visual blocks programmering, som det er kendt fra Scratch og Microbit. Det hentes her:

<https://m5stack.com/pages/download>. Det er en zip-fil som blot pakkes ud, og så kører man programmet som det er, altså ingen egentlig installation så der kræves ikke super-bruger privilegier.

(Den nuværende version af UIFlow er ikke opdateret til den nyeste Python version som findes til Core 2! – og de er desværre ikke helt kompatible...!)

## 2 Hello world

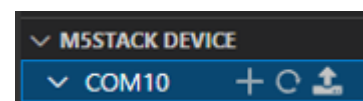
Start VSC editoren og sørg for at der er forbindelse til M5Stack'en.


Tryk på + (ud for M5Stack-forbindelsens COM port) for at tilføje en fil på enheden og kald den `my_demo.py`.

(Når man har oprettet en ny fil på enheden, så har den en tendens til at 'falde af USB forbindelsen'. Så skal man bare re-konnekte den igen ved at trykke på Add M5Stack i bunden af VSC, og derefter åbne filen.)

Skriv følgende i filen:

```
from m5stack import lcd
lcd.print('Hello world!', 50, 50, 0xff0000)
```



Kør det ved at trykke på  (den hvide) i toppen til højre af editoren. Enheden skulle nu gerne skrive "Hello World!" på skærmen.

Tip: Hvis Python koden ikke 'afslutter', så kan man ikke køre koden igen, før man har resat enheden. Se 11.3.1.

Når man gemmer sin kode, så gemmes den på selve M5Stack Core 2 enheden, dvs, at hvis man næste gang får en anden enhed, eller hvis den er blevet slettet, så er al ens kode væk. Det er derfor vigtigt at man også gemmer sine filer på sin egen laptop!

## 3 Tekst og grafik

### 3.1 Tekst

Displayet er 320x240 pixels, med (0, 0) i øverste venstre hjørne (hvilket er helt normalt).

Man skriver en tekst på følgende funktion (se eksemplet ovenfor):

```
lcd.print("Test", x, y, textColor)
```

Man kan ændre font'en på følgende måde:

```
lcd.font(lcd.FONT_DejaVu18)
```

hvor '18' angiver størrelsen på bogstaverne og kan være 18, 24, 40, 56 og 72.

Farven styres generelt med en 'farvekode' med følgende format: 0xRRGGBB, hvor RR er rød, GG er grøn og BB er blå, og 0x angiver at det er angivet i HEX. Grunden til at der er to bogstaver er, at det skal skrives i HEX hvor 2 tegn angiver en værdi mellem 0 og 255 (0-FF).

```
lcd.print('Test', 50, 50, 0xff8000)
```

'Gode' HEX værdier er: 00, 40, 80, c0, f0, ff. (Det er frit valg om man bruger store eller små bogstaver)

Man kan slette hele skærmen med følgende kommando:

```
lcd.clear()
```

Man kan ikke skrive 'æøå'.

Hvis man skriver noget som M5Stack'en ikke kan forstå skriver den typisk 'Invalid syntax' Man kan så trykke på skærmen (detail) for at se hvad den ikke kan lide. Her vil typisk være en fejl kode og et linjenummer, hvor det er gået galt.

### Opgave

1. Skriv tekster med forskellige størrelser og farver

Gem som: tekst-size-color.py.

### 3.2 Grafik

Følgende funktioner laver streger og figurer på skærmen:

```
lcd.line(x0, y0, x1, y1, lineColor)
lcd.rect(x, y, dx, dy, lineColor, fillColor)
lcd.circle(x, y, r, lineColor, fillColor)
```

Parametrene er lidt ulogiske. Linjen angives med de på endepunkter, mens firkanten angives med et punkt og en afstand (dx, dy). Cirklen angives med centrum og en radius.

### Opgave

1. Tegn nogle figurer på skærmen

Gem som grafik-test.py.

## 3.3 Touchskærm

M5Stack'en har ingen fysiske knapper, men en touchskærm. Man kan aflæse hvor man har trykket sidst med følgende funktion.

```
from m5stack import touch
(x, y) = touch.read()
```

Hvis man vil vide om der er trykket på skærmen, bruges følgende funktion:

```
touch.status()
```

### Opgave

1. Lav et program som venter på at man trykker på skærmen og så skriver positionen ud

Gem som touch-pos.py.

## 4 Gyro

M5Stack'en har en 3 akse gyro og en 3 akse accelerometer. Gyroen kan måle retningsforandringer, mens accelerometeret kan måle "kraften" i forhold til jorden (tyngdekraften).

De kan aflæses på følgende måde og returnerer et tal mellem -1 og 1 (kommatal).

```
import imu
imu0 = imu.IMU() # Skal kun kaldes een gang

(x, y, z) = imu0.acceleration
(x, y, z) = imu0.gyro
```

### Opgave

1. Tegn en cirkel midt på skærmen, men en radius på 10.
2. Flyt cirklen rundt på skærmen vha. accelerometeret, så den tegner en streg.
3. Ændre nu koden så den flytter cirklen, uden at der kommer en streg efter den.

### Hint

Husk at man skal slette den 'gamle' cirkel inden man tegner den 'nye'.

Gem som cirkel-gyro.py.

## 5 SD kort, lyd og billeder

TBD

Vibrator,

## 6 Ekstern hardware (Grove)

M5Stack'en har et stik på siden (GROVE), så man kan sætte nogle eksterne sensorer til den. Man tilslutter sensorerne vha. et lille kable.

Vi har forskellige sensorer at vælge i mellem og deres API er lidt forskellige alt afhængig af hvad de kan.

**Lyssensor** (kan måle hvor lyst det er).

```
import unit
light = unit.get(unit.LIGHT, (32,33))
light.analogValue # 0-1023
```

**RGB sensor** (kan måle hvor meget rød/grøn/blå farve lyset har)

```
import unit
color = unit.get(unit.COLOR, (32,33))
color.red # 0-255
color.green # 0-255
color.blue # 0-255
```

**Barometer** (kan måle lufttrykket og temperaturen. Den er meget følsom og kan derfor måle hvor højt den er oppe).

```
import unit
bps0 = unit.get(unit.BPS, unit.PORTA)
bps0.pressure() # hPa
bps0.temperature() # C
```

### Opgave

1. Vælg en sensor og lav et program som kan skrive værdien/erne ud på skærmen.
2. Prøv at få værdien til at forandre sig, ved at 'bruge' sensoren.

## 7 Labyrint spil

Her er en vejledning i, hvordan man selv laver et simpelt kugle labyrint spil på en M5Stack'en.

Forudsætninger: Man skal have kendskab til Python og vid hvad en liste er. Der ud over skal man bruge "API reference" fra dette dokument.

### 7.1 Flyt kuglen

Tegn en kugle (cirkel) på skærmen med en radius på 5.

Aflæs accelerometeret og flyt kuglen. Brug det til at tegne en streg på skærmen.

Ændre nu koden så kuglen flyttes, dvs at den ikke tegner en streg efter sig. Flyt nu kuglen i alle retninger for at se om det virker.

(Tip: hvis kuglen blinker for meget, så husk at man skal slette kugle lige inden den tegnes igen, og have et lille delay efter den er tegnet. Slet gammel kugle, tegn ny kugle, delay)

### 7.2 Lav labyrint

Lav en datastruktur til en labyrint. Den skal helst opfylde følgende krav:

Let at kode i Python

Let at printe på skærmen

Let at finde rundt i

Det kunne se sådan ud:

```
labyrint = [
    "XXXXXXXXX",
    "X        X",
    "X XXXX X",
    "X XE   X",
    "X X XX X",
    "X    X X",
    "XXXXXXXXX"]
```

Dette er en simpelt dobbelt liste (liste af linjer og liste af tegn). Det er nemt at tegne i Python, let at printe og let at finde rundt i.)

Tilgang til listen er: `labyrint[y][x]`

Tegn labyrinten på skærmen, og brug flere pixels pr. felt, 14x14 (tegn som et rektangel).

Hint: man kan læse listen på følgende måde:

```
for line in labyrint:
    for c in line:
```

## 7.3 Selve spillet

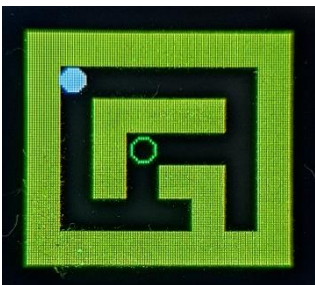
Sæt kuglen ind i labyrinten.

Flyt kuglen som før, men check om den rammer væggen inden den flyttes. Dette er lidt tricky, da man skal checke alle 4 'hjørner' af kuglen. Lav en funktion som gør det.

Hint: husk labyrinten er 14 pixels bred og kuglen har en radius på 5. Så man finder indekset i labyrinten på følgende måde for det ENE 'hjørne':

```
x = int((kugleX-5)/14)
y = int((kugleY-5)/14)
```

Hvis den 'rammer' 'E', så er man færdig med spillet. Jeg har tegnet 'E' som en grøn cirkel.



Den fulde kode kan findes her: <https://github.com/mhfalken/m5stack/blob/main/labyrinth-simple.py>

Spillet kan udvides med flere features, som fx:

- Større bane
- Tidtagning
- Ekstra 'huller' som man ikke må ramme
- Lydeffekter
- Inerti i kuglen (svært)

## 8 Ideer

Ping-pong spil – spil 'bold' op af 'mur' og bevæg 'bat' med gyro.



## 9 M5Stack hardware overblik

M5Stack findes i flere varianter, som har mange ligheder, men også en del forskelle. De fleste er baseret på en CPU som hedder ESP32, som er en dual core 32 bit 240MHz processor med indbygget WiFi og bluetooth. De har USB-C interface og indbygget batteri, samt en generisk 4 pin connector kaldet Grove.



Her er nogle eksempler:

- De mindste hedder M5StickC og M5StickC+ og har en 0.96" skærm, 1 knap, gyro og en extension connector på enden.
- Den næste i rækken er M5Stack core basic, som har en 2" skærm og 3 knapper, SD kort, højttaler og en extension connector på siderne.
- Den mest avanceret er M5Stack core 2, som har en 2" touch skærm, gyro, SD kort, mikrofon, højttaler og en extension connector på undersiden.



Der ud over så findes der også nogle, som er til specifikke formål og derfor er noget anderledes og derfor ikke yderligere omtalt her.

Vær opmærksom på at M5Stack basic og m5stack core 2 IKKE har de samme tilføjelse muligheder. På trods af samme udseende og dimensioner, så er det kun via de porte (Sd-kort, 4 pin connector) som er i siden, at de to moduler kan arbejde med samme hardware og sensorer. De moduler som kan "stackes" er ikke kompatible imellem de to versioner, og køber du derfor et stackmodul, så virker det kun til den ene version! Hvordan skelner du så imellem hvad der er kompatibel - du skal gætte. Der er pt. ikke nogen markør for hvilke dele som er kompatible.

Der findes mange eksterne enheder som kan tilsluttes den fælles 4 pin konnektor, såsom temperatur sensorer, neopixels mv. Her er de fysiske ben desværre heller ikke de samme og det er ikke alle sensorer som virker på M5Stack basic, da de har valgt de 'forkerte' ben. Det er blevet løst på fx. M5Stack core 2. Koden kan derfor ikke flyttes mellem de forskellige enheder uden at den bliver rettet en smule. Dette dokument forudsætter at man bruger en M5Stack Core 2.

### 9.1 M5Stack Core 2

Der er en beskrivelse af hardwaren på undersiden af enheden.

- ESP32
- 2" touch skærm (320 x 240)
- 6x Gyro, højttaler, mikrofon, SD kort
- Grove (pin 32, 33) og extension konnektor
- USB C stik



Power-on	Tryk på <b>Power</b> knappen
Power-off	Tryk på <b>Power</b> knappen i 6 sekunder
Reset	Tryk på <b>Reset</b> knappen – der går lidt tid inden der 'sker' noget.

## 10 Installering af M5stack Core 2

M5stack'en er baseret på en ESP32 microcontroller, og den forstår egentlig kun maskinkode. For at køre programmer skrevet i Python er der brug for at der er installeret en Python fortolker på M5Stacken. På M5Stack'en ligger der typisk noget software i forvejen. Det er normalt enten noget demo software eller UIFlow. UIFlow bruges til Python og UIFlow programmering.

## 10.1 M5Burner

M5Burner er et stykke software som bruges til at “brænde” software over på M5stack-hardwaren. Dette software skal bruges uanset hvordan man vil bruge/programmerer M5stack. Det hentes her: <https://m5stack.com/pages/download>. Det er en zip-fil som blot pakkes ud, og så kører man programmet som det er, altså ingen egentlig installation så der kræves ikke super-bruger privilegier.

### Windows

Udpak zip-filen et passende sted.

Herefter startes programmet som `M5Burner.exe`.

### Linux

Udpak filerne med

```
$ unzip -d M5Burner_Linux{,.zip}
```

Herefter startes programmet som `M5Burner_Linux/bin/m5burner`.

Med enheden sat i USB-porten og ved at starte programmet `M5Burner` kan man se en stribe muligheder for at installere allerede udviklede firmware. Firmware er specifik for en bestemt enhed, så du skal være opmærksom på hvilken enhed du har. Her tager vi udgangspunkt i **M5Stack Core 2**.

Man vælger enhed i venstre side, vi vælger **Core**.

Både til Python programmering og til visuel programmering bruger vi den firmware som hedder UIFlow. Se billede.

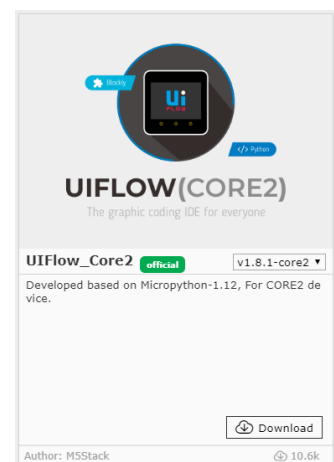
Efter man har valgt version (i billedets tilfælde 1.8.1-en, så klikkes der “Download”, hvorefter firmwaren hentes til din arbejdscomputer og der fremkommer 3 nye knapper.

Så vælger man “Burn” og til Wifi-billedet som kommer frem klikker man blot “Start”. Så går der noget tid (ca. 30 sekunder), hvorefter firmware er brændt på enheden. Nu er enheden klar til at fortolke Python-programmer.

M5Stack’en skal sættes i USB mode, før den kan bruges. Dette gøres lidt forskellige afhængig af M5Stack type. Her er Core 2 beskrevet:

Tænd for enheden (eller sæt USB stikket i). Tryk på Flow og vælg USB.

Enheden er nu i USB mode og husker selv denne tilstand.



## 11 API reference

### 11.1 Generelle tips

Der er nærmest ingen dokumentation på hvordan Python interfacet til M5Stack hardwaren ser ud. Det lettest er at bruge UIFlow, vælge de blokke man vil vide noget om og så trykke på Python knappen for at se hvordan Python koden ser ud. Det kan man så kopiere direkte ind i sin egen kode.

Tip: når kan vælge en specifik blok, så kommer der lidt yderlige hjælp frem ude til højre på skærmen.

### 11.2 Problemer

Hvis man skifter ekstern sensor (Grove) ‘on the fly’, så er det ikke sikkert at koden virker - det har de ikke support for. (Dette er ellers brugbart, hvis man laver et Escaperoom spil - se senere) Det er et rent software problem, da hardwaren ikke har noget problem.

## 11.3 Funktions overview

Her er en list over de mest gængse Python API funktioner til at interface med M5Stack hardwaren.

I de følgende eksempler, skal der bruges diverse imports. Det er ikke sikkert alle er nødvendige hver gang, men her er en kort list over hvilke man kan prøve (eller bare inkludere altid).

```
from m5stack import *
from m5ui import *
from uiflow import *
```

### Tekst & grafik

```
from m5stack import lcd
color = 0xffffffff # RRGGBB

lcd.clear()
lcd.orient(lcd.LANDSCAPE_FLIP) # If needed
lcd.font(lcd.FONT_DejaVu18) #18, 24, 40, 56, 72
# lcd.FONT_Ubuntu, lcd.FONT_7seg, lcd.FONT_Comic
lcd.print("Test", x, y, textColor)

lcd.line(x0, y0, x1, y1, lineColor)
lcd.rect(x, y, dx, dy, lineColor, fillColor)
lcd.circle(x, y, r, lineColor, fillColor)
```

### Touch UIFlow metode:

```
from m5stack_ui import *

touch_button0 = M5Btn(text='Button', x=60, y=50, w=70, h=30,
bg_c=0xFFFFFFFF, text_c=0x000000, font=FONT_MONT_14, parent=None)
touch_button0.set_pos(50, 50)
touch_button0.get_state()

def touch_button0_pressed():
    # Event pressed
    pass
touch_button0.pressed(touch_button0_pressed)
```

### Touch raw metode:

```
from m5stack import touch
(x, y) = touch.read()
touch.status() # Touched=True
```

### Lyd:

```
speaker.playWAV('/sd/tadar.wav', 44100, speaker.F16B) # WAV file
speaker.playTone(220, 2) # Tone: frekvens (Hz), tid (s)
```

### Vibrator:

```
power.setVibrationIntensity(100) # 0-100
power.setVibrationEnable(True)
power.setVibrationEnable(False)
```

### Gyros

```
import imu
imu0 = imu.IMU()
(x, y, z) = imu0.acceleration
(x, y, z) = imu0.gyro
```

x, y og z er et tal mellem -1 og 1.

Batteri

```
v = power.getBatVoltage()
```

### 11.3.1 Koden afslutter

Det er vigtigt at koden afslutter, da man ellers ikke kan downloade en ny version. Så hvis man ikke vil trykke på reset hele tiden og vente på at enheden booter, så man kan med fordel sørge for at en 'knap' får koden til at afslutte.

For M5Stack core 2, kunne det se sådan ud:

I stedet for

```
while True:
```

så kan man skrive:

```
while not touch.status(): # Debug
```

Så når man trykker på skærmen, så kommer man ud af løkken.

## 12 Løsninger

**text-size-color.py**

```
from m5stack import lcd

lcd.clear()

lcd.font(lcd.FONT_DejaVu18)
lcd.print("Test", 10, 10, 0xff0000)
lcd.font(lcd.FONT_DejaVu24)
lcd.print("Hej", 10, 50, 0x00ff00)
lcd.font(lcd.FONT_DejaVu40)
lcd.print("Dav", 10, 100, 0x0000ff)
```

**grafik-test.py**

```
from m5stack import lcd

lcd.clear()

lcd.line(10, 10, 100, 20, 0xff0000)
lcd.rect(150, 100, 80, 25, 0xffff00, 0x0000ff)
lcd.circle(100, 150, 40, 0x00ffff, 0x8040c0)
```

**touch-pos.py**

```

from m5stack import lcd
from m5stack import touch

lcd.font(lcd.FONT_DejaVu24)

lcd.clear()
while not touch.status():
    pass

(x, y) = touch.read()
lcd.print("Tryk (%i, %i)" %(x, y), 10, 10, 0xff0000)

```

### cirkel-gyro.py

```

from m5stack import lcd
import imu
import time

imu0 = imu.IMU()
lcd.clear()

x = 160
y = 120

while not touch.status():
    (dx, dy, dz) = imu0.acceleration
    x = x - int(dx*10)
    y = y + int(dy*10)
    lcd.circle(x, y, 10, 0x00ffff, 0x8040c0)
    time.sleep(0.1)
    lcd.circle(x, y, 11, 0x0, 0x0)

```

## 13 Links

Skriv gerne links her, og et par ord om hvad man kan finde via linket.

Officiel M5Stack hjemmeside: <https://m5stack.com/>  
 Firmware download: <https://m5stack.com/pages/download>  
 Guide på dansk til M5StickC <https://m5guide.readthedocs.io/>

### Python M5Stack API

Her er en fin start guide til M5Stack Python API'et.

<https://m5guide.readthedocs.io/en/latest/index.html#>

Den dækker primært M5StickC, men de fleste funktioner vil virke på alle enheder.