

# M5STACK

Dokument version 1.00 (Maj 2021)

Dette dokument beskriver hvad en M5Stack enhed er og hvordan man kan bruge den.



## INDHOLDSFORTEGNELSE

[Indholdsfortegnelse](#)

[Hardware overblik](#)

[Installation af værktøjer på desktop](#)

[USB driver \(kun Windows\)](#)

[M5Burner \(i alle tilfælde\)](#)

[UIFlow desktop IDE \(til visuel programmering\)](#)

[Visual Studio code med udvidelse \(til Python programmering\)](#)

[M5-Udvidelsen](#)

[Arduino Desktop IDE \(til C/C++ programmering\)](#)

[Programmering af M5stack](#)

[Medfølgende firmware](#)

[Python programmering](#)

[Python M5Stack API](#)

[UIFlow blocks programmering](#)

[Tilføj hardware og sensorer](#)

[Diverse](#)

[Reset og power](#)

[M5StickC \(Python\)](#)

[Generelle tips](#)

[Ting som bør løses](#)

[Problemer](#)

[Funktions overview](#)

[Generisk include](#)

[Tekst & grafik](#)

[Knapper](#)

[Lyd](#)

[Gyros](#)

[Battery](#)

[Tip](#)

[Escape room game](#)

[Labyrint](#)

[Flyt kuglen](#)

[Lav labyrint](#)

[Selve spillet](#)

[Forløb 1](#)

[EScaperoom - Kuffert edition](#)

[Ideen er her at kombinere flere af M5stacks features til hver del af gåden som skal løses.](#)

[Eksempelvis indtastning af kode.](#)

[Nedtællingsur.](#)

[Optagelse af korrekt beat/rytme.](#)

[Forløb 1](#)

## HARDWARE OVERBLIK

M5Stack findes i flere varianter, som har mange ligheder, men også en del forskelle. De fleste er baseres 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.

Her er nogle eksempler:

De mindste hedder M5StickC og M5StickC+ og har en 0.96" skærm, 1 knap, gyro og en extension connector på siden.



Den næste i rækken er M5Stack core basic, som har en 2" skærm og 3 knapper, SD kort, højtalere og en extension connector bag på.



Den mest avanceret er M5Stack core 2, som har en 2" touch skærm, gyro, sd kort, mikrofon, højtalere og en extension connector bag på.



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 externe enheder som kan tilsluttes den fælles 4 pin connector, såsom temperatur sensorer, neopixels mv. Her er de fysiske ben desværre heller ikke de samme og det er ikke alle sensore 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.

## INSTALLATION AF VÆRKTØJER PÅ DESKTOP

For at arbejde med M5stack er det nemmest at arbejde med forskellige værktøjer på sin arbejdscomputer.

Der er et antal værktøjer som kan være nyttige til programmering af M5stack. Nedenfor er der beskrivelser af softwaren og hvordan den installeres og køres for Windows og Ubuntu Linux.

## USB driver (kun Windows)

For at kommunikere med M5stack er det muligvis nødvendigt at installere den rigtige USB driver, den kan findes her:

[http://www.ftdichip.com/Drivers/CDM/CDM21228\\_Setup.zip](http://www.ftdichip.com/Drivers/CDM/CDM21228_Setup.zip)

## M5Burner (i alle tilfælde)

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. Pas på alle zip-filer er “[Tarbombs](#)”.

### 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`.

## 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. Pas på alle zip-filer er “[Tarbombs](#)”.

### Windows

Udpak zip-filen et passende sted.

Herefter startes programmet som `UIFlow-Desktop-IDE.exe`

### Linux

Udpak filerne med

```
$ unzip -d UIFlow-Desktop-IDE_Linux{,.zip}
```

Herefter startes programmet som  
`UIFlow-Desktop-IDE_Linux/bin/uiflow-desktop-ide.`

## Visual Studio code med udvidelse (til Python programmering)

Til Visual studio code (VSCode) findes der en udvidelse der gør det let at udvikle og overføre Python programmer fra samme editor.

### Windows

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

### 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
```

### Mac

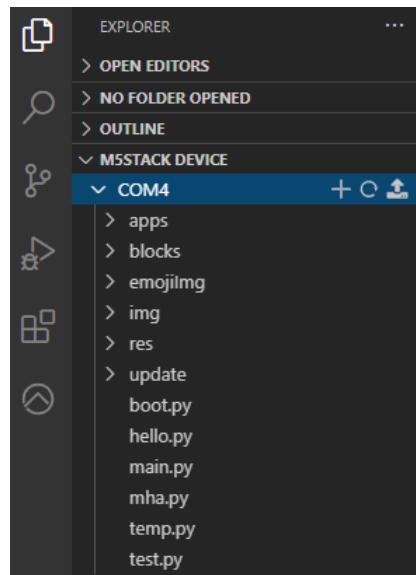
Følg instrukser for at installere Visual Studio Code på <https://code.visualstudio.com/Download>

## M5-Udvidelsen

Efter installation af VSCode skal man installere en M5 stack plugin som hedder

**vscode-m5stack-mpy**

Tryk på  ude til venstre og skriv **M5Stack** og installer plug-in'en. Efter installationen, så kommer der en knap frem i bunden af editoren: . Hvis ikke der kommer en sådan knap er det sikkert fordi du bruger en nyerer version af VSCode, se [Visual Studio code med udvidelse \(til Python programmering\)](#). Tryk på knappen  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, det skal vi bruge når der skal udvikles Python software, se [Python programmering](#)

## Arduino Desktop IDE (til C/C++ programmering)

Hvis man gerne vil programmere M5stack direkte så skal man bruge Arduino Desktop IDE.

### Windows

Selve Arduino installeres fra denne side:  
<https://www.arduino.cc/en/software>

### Linux

```
$ aptitude install arduino
```

Når man har installeret editoren, så skal man have hentet nogle forskellige biblioteker for at kunne oversætte koden.

Først skal man ind i File->Preferences, og opdatere Additional Boards manager URLs: til at indeholde:

```
https://dl.espressif.com/dl/package_esp32_index.json,  

http://arduino.esp8266.com/stable/package_esp8266com_index.json
```

Additional Boards Manager URLs:

Herefter skal man ind i Tools->Board->Boards Manager og installere esp32.

Derefter skal man ind i Tools->Manage Libraries. Søg på m5stack og installer M5Stack og afhængig af hvad man har af boards også M5Core 2, M5StickC etc.

Nu skulle alle filerne være installeret. Vælg det board du vil programmere:

Tools->Board->ESP32 Arduino->...

For at finde et eksempel, gå til File->Examples->M5Stack-> ... (eller hvad board man nu har).

Herefter er det 'normalt' Arduino flow, compile og download koden.

## PROGRAMMERING AF M5STACK

M5stack'en er baseret på en ESP32 microcontroller, og den forstår egentlig kun maskinkode. For at køre programmer skrevet i andre sprog er der brug for at disse sprog enten bliver oversat til ESP32 maskinkode af en oversætter (som er tilfældet for C), eller at der på M5stacken er installeret en fortolker af sproget (som er tilfældet for Python).

- Hvis man programmerer i C, så findes der værktøjer, heriblandt en oversætter, som oversætter C-programmet til ESP32 maskinkode. Den kode kan så brændes direkte på enheden med Arduino desktop.
- Hvis man vil køre (micro)Python programmer på M5stacken, så skal der på M5stacken installeres en Python fortolker, også kaldet firmware, for at kunne køre disse Python programmer på M5stacken.
- Hvis man vil køre programmer som man har programmeret/"tegnet" i UIFlow, som omsættes disse program-tegninger til Python som så kan køres som et Python program.

Maskinkode programmer til M5stacken kaldes også for firmware. Disse kan man som sagt selv udvikle i C/C++. Men med M5Stacken følger der også en masse eksisterende firmware. Disse kan hentes og installeres til M5Stacken via programmet M5Burner, se [M5Burner \(i alle tilfælde\)](#).

## Medfølgende firmware

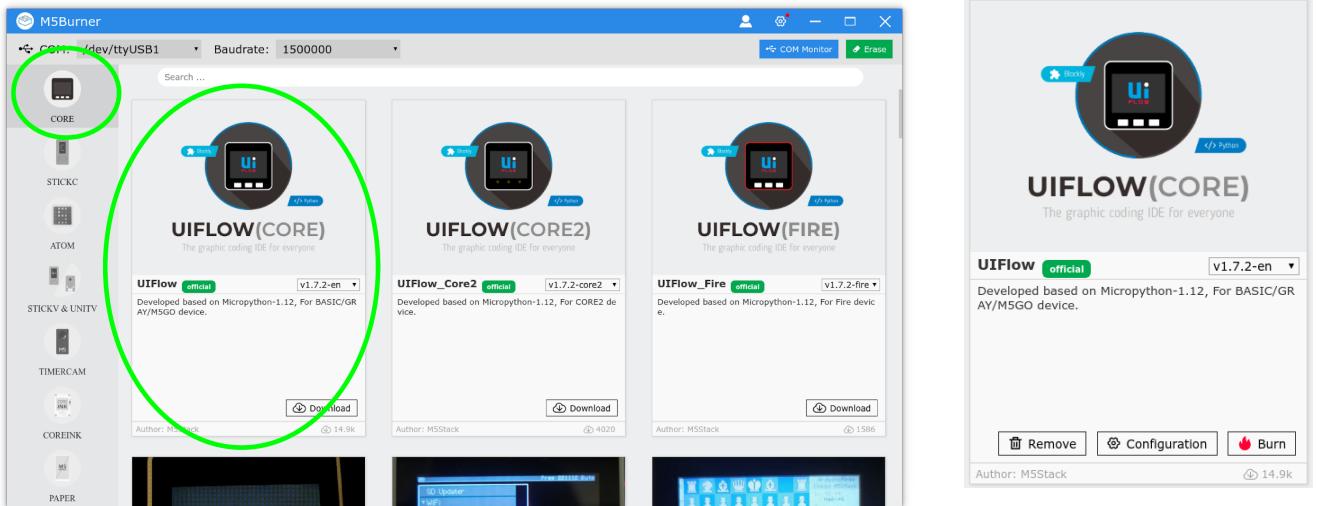
Med enheden sat i USB-porten og ved at starte programmet M5Burner (se [M5Burner \(i alle tilfælde\)](#)) kan man se en stribe muligheder for at installere allerede udviklede firmware.

Firmware er specifik for en bestemt enhed, så du skal være opmærksom på hvilken enhed du har. Her tager vi udgangspunkt i [ESP32 Basic Core IoT Development Kit](#) som CPDK har købt nogle eksemplarer hjem af.

Man vælger enhed i højre side, vi vælger Core.

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





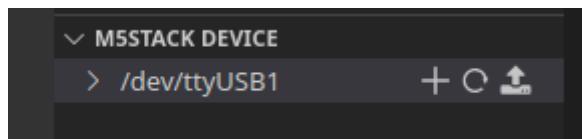
Efter man har valgt version (i billedets tilfælde 1.7.2-en, så vælges klikkes der "Download", hvorefter firmware hentes til din arbejdscomputer og der fremkommer 3 nye knapper. Så vælger man "Burn" og til Wifi-billedet som kommer frem klikker man blot "Start". Så går der noget tid (ca. 30 sekunder), hvorefter firmware er brændt på enheden. Nu er enheden klar til at fortolke Python-programmer, heriblandt de Python programmer der er et resultat af visuel programmering i UIFlow.

## Python programmering

Forudsætning:

- Der er installeret firmware med Python fortolker på M5-enheten, se [Medfølgende firmware](#)
- Der er installeret VSCode med M5-udvidelse på din arbejdscomputer, se [Visual Studio code med udvidelse \(til Python programmering\)](#)
- Enheden er i USB-mode. Enheden sættes i USB-Mode lidt forskelligt afhængig af hardware typen. Generelt gælder at man skal trykke på en tast inden for 1 sek efter reset og så navigere til USB-mode. Når først det er sat op, så husker den det også efter power-off.

Start VSCode og sørge for at der er forbindelse til M5-enheten.



Tryk på + (ud for M5-forbindelsen) for at tilføje en fil på enheden, kald den `my_demo.py`. Skriv følgenden i filen:

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

Gem programmet og 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.

## 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 flest funktioner vil virke på alle enheder.

## UIFlow blocks programmering

Forudsætning:

- Der er installeret firmware med Python fortolker på M5-enheden, se [Medfølgende firmware](#).
- Der er installeret UIFlow-desktop-IDE på din arbejdscomputer, se [UIFlow desktop IDE \(til visuel programmering\)](#).

Start UIFlow-desktop-IDE. Lav et Blocks-program (som vist på billede) og vælg derefter ➤ oppe i højre hjørne.



## Tilføj hardware og sensorer

Der er en række værktøjer man kan tilføje, hvis man har dele såsom RFID og sensorer af forskellig art. Når man har tilføjet det via det grafisk interface, så kommer der automatisk flere muligheder til at blok programmere med det specifikke stykke hardware.

## DIVERSE

### Reset og power

De forskellige enheder har i de fleste tilfælde en kombineret reset og power knap, som er placeret i nærheden af USB-C stikket, men den fungere ikke ens! Et enkelt tryk er på nogle reset, men power off er enten dobbelttryk eller tryk i 6 sekunder.

### M5StickC (Python)

For M5StickC har man også mulighed for at installere en ren Python version og bruge MU editoren i stedet. Det er alt sammen beskrevet meget fint her:

<https://m5guide.readthedocs.io/>

Her er også en god vejledning i, hvordan man skriver Python kode til enheden. Denne guide kan også bruges til de andre enheder, da selve M5Stack API'et er det samme.

## FORLØB (PYTHON)

---

### 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.

### TING SOM BØR LØSES

- Hvordan afspiller man en MP3 fil?
- Hvordan overfører man filer til og fra M5Stack (på en nem måde)?
- Hvordan installerer man flere Python biblioteker? (Kan man det?)

### Problemer

Her kan man beskrive de problemer som man endnu ikke har fået løst eller som måske ikke kan løses.

- Man kan ikke vise et billede direkte fra SD kortet, så resetter M5Stack!  
`lcd.image(20, 20, "/sd/image.jpg")`  
Man kan godt læse filen og kopiere den ind på flash'en og så vise den, men ikke vise den direkte fra SD kortet.
- Der er desværre forskel på hvad den enkelte Python firmware supporterer (selvom hardwaren har samme funktionalitet). Fx. **M5Stack core 2** supporterer WAV fil afspilning, men det gør **M5Stack basic** ikke.
- 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.

### FUNKTIONS OVERVIEW

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

## Generisk include

I de følgende eksempler, skal der bruges diverse imports. Det er ikke sikkert alle er nødvendige hver gang, men her er en kort liste 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)
```

## Knapper

Fysiske knapper:

```
btnA.wasPressed()
btnB.wasPressed()
btnC.wasPressed()
```

Touch (Core 2) 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 (Core 2) raw metode:

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

## Lyd

PWM lyd (ikke Core 2):

```
speaker.tone(freq=800, duration=300)
```

Wav lyd (Core 2):

```
speaker.playWAV('/sd/tadar.wav', rate=44100, dataf=speaker.F16B)
```

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.

## Battery

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

## Tip

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 bootet, 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.

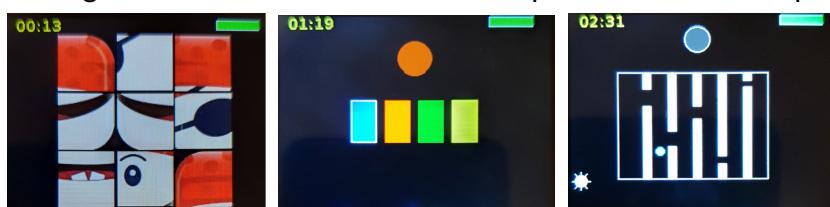
## ESCAPE ROOM GAME

Jeg har lavet et Escape room spil til M5Stack Core 2, som jeg har lagt her:

<https://github.com/mhfalken/m5stack/tree/main/escaperoom>, og her ligger en video demo:

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

Spillet er skrevet i Python og viser hvordan man bruger de forskellige kald til hardwaren. Spillet har tre baner som bruger touchskærmen, en farve sensor og en lys sensor til at løse et problem. Det tager 5-10 minutter at løse. Her er et par screenshots fra spillet.



Hver bane fylder ca. 80 linjer, så det burde være muligt for børnene selv at lave flere baner. Installation og hardware krav er beskrevet på GitHub linken.

## LABYRINT

Her er en vejledning i, hvordan man selv laver et simpelt kugle labyrinth spil på en M5Stack core 2.

Forudsætninger: Man skal have kendskab til Python og vide hvad en liste er. Derudover skal man bruge "Funktions overview" fra dette dokument.

## 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)

## Lav labyrinth

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

- Let at kode i Python
- Let at printe på skærmen
- Let at finde rundt i

```
labyrinth = [
    "XXXXXXXX",
    "X       X",
    "X XXXX X",
    "X XE     X",
    "X X XX X",
    "X       X",
    "XXXXXXXX"]
```

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: `labyrinth[y][x]`

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

(Hint: man kan læse listen på følgende måde:

```
for line in labyrinth:
    for c in line:
()
```

## Selv spillet

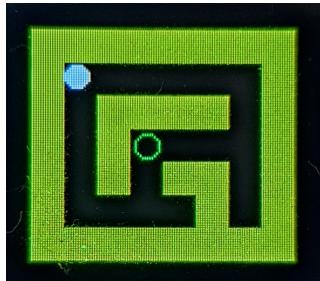
Sæt kuglen ind i labyrinthen.

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 labyrinthen er 14 pixels bred og kuglen har en radius på 5. Så man finder indekset i labyrinthen 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)

## FORLØB (UI FLOW)

---

Placeholder. Her er ikke noget indhold i øjeblikket.

### FORLØB 1

#### ESCAPEROOM - KUFFERT EDITION

Ideen er her at kombinere flere af M5stacks features til hver del af gåden som skal løses.

Eksempelvis indtastning af kode.

Nedtællingsur.

Optagelse af korrekt beat/rytme.

## FORLØB (C/C++)

---

Placeholder. Her er ikke noget indhold i øjeblikket.

## FORLØB 1

### 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/>