

Lego Spike Prime

Af: Michael Hansen, Coding Pirates Furesø, 2022-24, version 0.70

Dokumentet ligger her: <https://github.com/mhfalken/lego>



Her er lidt ideer til hvordan man kan bruge Lego Spike Prime sættet, hvor hoved fokus er på programmering i **Python**. I de skoleopgaver som 'følger med' sættet får man generelt udlevet al koden i scratch og primært skal man bare se hvad der sker. Det egner sig i mine øjne ikke særligt godt til Coding Pirates, hvor fokus er mere på programmering.

Note: Jeg har tegnet et 'kort/bane' på et A0-ark ala FLL, og det er det jeg refererer til i nogle af opgaverne. Nogle af disse opgaver kan ikke løses uden dette kort eller noget tilsvarende af køre efter. Der er et billede af mit kort i bunden af dette dokument. Jeg har også lavet 2 paprør, som afstandssensoren har lettere ved at se end nogle Legoklodser.

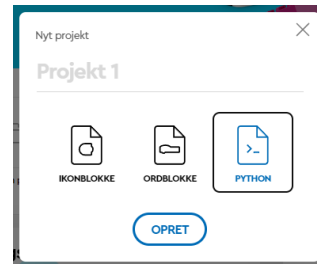
1	Hello world	3
2	Bil kontrol	3
2.1	Klods til cirkel.....	3
2.2	Firkant.....	4
2.3	Bane.....	4
2.4	Hent en klods.....	4
2.5	Hent flere klodser	5
3	Bil farvesensor	5
3.1	Kør til stregen	5
3.2	Tæl antal streger.....	6
3.3	Print farverne ud	6
3.4	Farverekation.....	6
4	Bil afstandssensor.....	7
4.1	Kør tæt på en klods.....	7
4.2	Ryd området	7
5	ToDo	8
5.1	Bil følg en streg.....	8
5.2	Reager på farverækkefølgen (avanceret).....	8
5.3	Skridt tæller	8
5.4	Plotter (CNC-maskine)	8
6	Løsninger	9
6.1	Bil kontrol	9
6.2	Bil farvesensor	10
6.3	Bil afstandssensor.....	11
7	Appendiks	11
7.1	Kort/bane.....	11
7.2	Links.....	12

1 Hello world

Man kan programmere Lego HUB'en direkte fra en browser (det kræver derfor ingen installation af software) ved at bruge følgende link: <https://spike.legoeducation.com/prime/lobby/>

Start med at trykke på **Settings** nede i venstre hjørne og skift sproget til Dansk.

- 1) Tryk på '**Nyt projekt**',
- 2) Tryk på **Python**
- 3) Tryk på **OPRET** knappen (se billede).



Nu kommer der et program frem på skærmen.

- 4) Tænd HUB'en (tryk på den store knap)
- 5) Tryk derefter i browseren oppe i venstre hjørne på **Tilslut** og følg vejledning for at forbinde HUB'en til computeren.



Vi skal nu downloade programmet til HUB'en.

- 6) Tryk på **Afspil** (gul pil) nede i højre hjørne.
- 7) Se at HUB'en skriver 'Hi!' på skærmen.



Hvis man vil køre programmet igen, skal man trykke på den store knap på HUB'en to gange efter hinanden. (Første gang stopper programmet, anden gang starter programmet igen.)

Opgave

- 1) Ret teksten til noget andet end 'Hi!'.
- 2) Download programmet og se at det virker.

Man slukker HUB'en ved at holde den store knap inde på HUB'en til den slukker.

2 Bil kontrol

2.1 Klods til cirkel

Bilen skal starte i startområdet og skubbe en klods ind i cirklen i midten og køre baglæns tilbage til start området.

- 1) Opret et nyt program (tryk på **Åbn hjem** ikonet oppe i venstre hjørne), og kald det *bil-klods* og tryk **OPRET**.
- 2) Slet al koden!
- 3) Byg en bil fx den som hedder *Grundmodel 1* under **Byggevejledninger**. Sæt motorerne i **port C og D**.

Bilen har to motorer, og dem vil vi gerne kunne styre som en enhed, dvs. i stedet for at styre de to motorer hver for sig, så styrer vi dem sammen, så vi angiver hvordan de indbyrdes skal køre. Det gør man ved at pare de to motorer sammen og så angive hvor meget bilen skal dreje med et tal mellem -100 og 100. 0 får bilen til at køre lige ud.

Opgave

1. Få bilen til at køre lige ud i 2 sekunder. Brug følgende linjer:

```

from hub import port
import runloop
import motor_pair

async def main():
    motor_pair.pair(motor_pair.PAIR_1, port.D, port.C)
    await motor_pair.move_for_time(motor_pair.PAIR_1, 2000, 0, velocity=500)

runloop.run(main())

```

2. Test at koden virker
3. Få bilen til at køre lidt hurtigere ved at bruge hjælpen ude til højre i browseren. Tryk på **API-moduler**, derefter på **Motorpar** og til sidst på **move_for_time**.
4. Ændre koden så bilen også kører baglæns.
5. Få bilen til at skubbe en legoklods ind i cirklen og køre tilbage igen.
6. Gem programmet

2.2 Firkant

Vi skal nu have bilen til at køre i en firkant.

- 1) Lav et nyt projekt og kald det *bil-firkant*.

Opgave

- 1) Få bilen til at køre 20 cm frem (det kan være en fordel at kopiere sin kode fra før som start)
- 2) Få derefter bilen til at dreje 90 grader til højre
- 3) Lav en loop rundt om disse linjer, så den laver det 4 gange og der med en firkant. Man laver en loop på følgende måde:

```
for i in range(4):
```

Bilen skal gerne ende samme sted som den startede.

- 4) Gem programmet

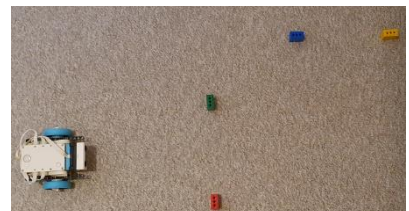
2.3 Bane

Byg en lille bane af Legoklodser og få bilen til at følge banen vha. simple motor kommandoer.

- 1) Lav et nyt projekt og kald det *bil-bane*.

Opgave

- 1) Byg en lille bane af Legoklodser
- 2) Få bilen til at køre gennem banen
- 3) Gem programmet



2.4 Hent en klods

Sæt en gribe-arm på bilen og få den til at hente en klods.

- 1) Lav et nyt program og kald det *bil-arm*.

Opgave

- 1) Byg en gribe-arm foran på bilen som er styret af en motor. Sørg for at gribearmen er monteret foran på bilen og stikket sat i **port E**.
- 2) Sæt bilen i start området og få den til at køre frem til klodsen og tilbage igen (brug koden fra tidligere opgave)
- 3) Få gribearmen til at åbne sig inden bilen kører frem. Her kan man bruge følgende linjer som inspiration

```

import motor
...
await motor.run_for_degrees(port.E, 90, 100)

```

- 4) Få gribearmen til at lukke sig når bilen når klodsen
- 5) Test at bilen nu kan hente klodsen

- 6) Gem programmet

2.5 Hent flere klodser

Få bilen til at hente flere klodser, ved at køre flere gange

- 1) Lav et nyt program og kald det *bil-klodser*.
- 2) Læg to klodser med ca. 15 cm imellem sig, så bilen kan hente en af gangen
- 3) Kopier koden fra før til at hente den første klods

Opgave

Når bilen kommer tilbage fra første tur, så skal den drejes lidt, så den peger mod den næste klods.

- 1) Tilføj en linje som drejer bilen lidt så den peger mod den næste klods
- 2) Lav den kode som henter klods to. (Det er det samme kode som henter klods et)
- 3) Test at bilen kan hente begge klodser
- 4) Gem programmet

3 Bil farvesensor

3.1 Kør til stregen

Få bilen til at køre frem og stoppe når den kører over en sort streg.

Lav et nyt program og kald det *bil-streg*.

- 1) Sæt farvesensoren fast på bilen så den peger nedad ca. 15 mm over gulvet. Sæt stikket i **port A**.
- 2) Placer bilen i startområdet så den peger mod en eller flere sorte linjer.

Farvesensoren aflæses på følgende måde:

```
import color_sensor
import color
...
farve = color_sensor.color(port.A)
```

Vi skal også have bilen til at køre 'af sig selv' dvs. at den kører indtil at den skal stoppe. Det gøres ved at bruge følgende kommando:

```
motor_pair.move(motor_pair.PAIR_1, 0, velocity=300)
...
motor_pair.stop()
```

Første linje starter motorerne og sidste linje stopper motorerne.

Opgave

- 1) Lav følgende program og se at bilen kører frem

```
from hub import port
from hub import light_matrix
import runloop
import motor_pair
import color_sensor
import color

async def main():
    motor_pair.pair(motor_pair.PAIR_1, port.D, port.C)
    motor_pair.move(motor_pair.PAIR_1, 0, velocity=300)
    for i in range(50):
        await runloop.sleep_ms(50)
    motor_pair.stop(motor_pair.PAIR_1)
runloop.run(main())
```

- 2) Aflæs farvecensoren og lav en *if* sætning som stopper bilen når den ser en sort streg. Den sorte farve er defineret sådan her: `color.BLACK`
- 3) Gem programmet

3.2 Tæl antal streger

Få bilen til at køre et stykke og tæl hvor mange linjer den kører over.

- 1) Lav et nyt program og kald det *bil-count*.

Opgave

- 1) Lav en variable `count = 0` i 'toppen' af `main()`
- 2) Ændre koden så i stedet for at stoppe motorerne, så tælles `count` én op.
- 3) Skriv tallet ud på displayet `light_matrix.write(str(count))`

Hvis man læser farverne for hurtigt, så kan man komme til at tælle den samme streg flere gange. Det kan løses ved at få bilen til at køre lidt hurtigere.

- 4) Gem programmet

3.3 Print farverne ud

Få bilen til at køre et lille stykke og lav en liste over de farver den ser.

- 1) Lav et nyt program og kald det *bil-farve-liste*.
- 2) Placer bilen i start området udfor den liste af farver som har 5 farver.
- 3) Kopier koden fra før

Lego mangler desværre en funktion til at printe farven ud med, så den laver vi selv. Indsæt følgende linje i koden lige efter alle `imports`

```
farveTxt = ["BLACK", "MAGENTA", "PURPLE", "BLUE", "AZURE",  
"TORQUISE", "GREEN", "YELLOW", "ORANGE", "RED", "WHITE"]
```

Opgave

- 1) Ændre koden så i stedet for at tælle antal sorte linjer, så at printe farven ud. Brug `print(farveTxt[farve])` til at printe farven ud på **Konsol** vinduet i bunden af browseren.

Problemet er at den printer alle farverne ud mange gange og at den også skriver WHITE. Det skal vi nu have ændret.

- 2) Tilføj følgende linjer de rigtige steder i koden, og se at den nu kun skriver farverne ud én gang.

```
prevFarve = color.WHITE  
...  
if farve != color.WHITE and farve != color.UNKNOWN and farve != prevFarve:  
    prevFarve = farve
```

- 3) Test at det virker med forskellige farvemønstre.
- 4) Gem programmet

3.4 Farverreaktion

Lav et program, som få bilen til at køre i forskellige mønstre afhængig af hvilken farve farvesensoren ser.

Fx *rød* får bilen til at køre 10 cm frem, *grøn* får bilen til at dreje 90 grader etc.

- 1) Lav et nyt program og kald det *bil-farve*.
- 2) Slet al koden
- 3) Sæt farvesensoren fast på bilen, så man kan holde en klods op foran den.

Opgave

- 5) Tilføj følgende linjer, og test det virker med en rød klods.

```

from hub import port
import runloop
import motor_pair
import color_sensor
import color

async def main():
    motor_pair.pair(motor_pair.PAIR_1, port.D, port.C)
    while True:
        farve = color_sensor.color(port.A)
        if farve == color.RED:
            await motor_pair.move_for_time(motor_pair.PAIR_1, 1000, 0, velocity=500)

runloop.run(main())

```

- 6) Tilføj support for flere farver, så bilen gør noget forskelligt afhængig af farven.
- 7) Lav en lille bane og styr bilen igennem ved hjælp af klodser i forskellige farver.
- 8) Gem programmet

4 Bil afstandssensor

4.1 Kør tæt på en klods

Få bilen til at køre så tæt på en 'væg' som muligt.

- 1) Lav et nyt program og kald det *bil-afstand*.
- 2) Monter afstandssensoren foran på bilen og sæt stikket i **port F**.
- 3) Sæt bilen ca. 30sm fra en væg eller anden forhindring.

Afstandssensoren aflæses på følgende måde og returnerer afstanden i mm:

```

import distance_sensor
...
afstand = distance_sensor.distance(port.F)

```

Vi skal også have bilen til at køre 'af sig selv' dvs. at den kører indtil at den skal stoppe. Det gøres på samme måde som i "Kør til strengen":

Opgave

- 1) Lav et program som få bilen til at køre lige frem i nogle sekunder som i "Kør til strengen".

Vi skal nu have afstandssensoren til at stoppe bilen, når den er tæt nok på.

- 2) Aflæs afstandssensoren og tilføj en **if** sætning som får bilen til at stoppe, når afstanden er mindre end 50 mm. Afstandssensoren kan desværre returnere -1, hvis den ikke kan 'se' noget og **if** sætningen skal derfor være 'lidt' avanceret:

```

if afstand > 0 and afstand < 50:

```

- 3) Test nu at bilen stopper foran forhindringen.
- 4) Afstandssensoren har 'øjne' som kan lyse. Kik i API referencen ude til højre og skriv noget kode som tænder lysdioderne.
- 5) Tilføj yderlige en linje som slukker for 'øjnene' når bilen stopper.
- 6) Gem programmet

4.2 Ryd området

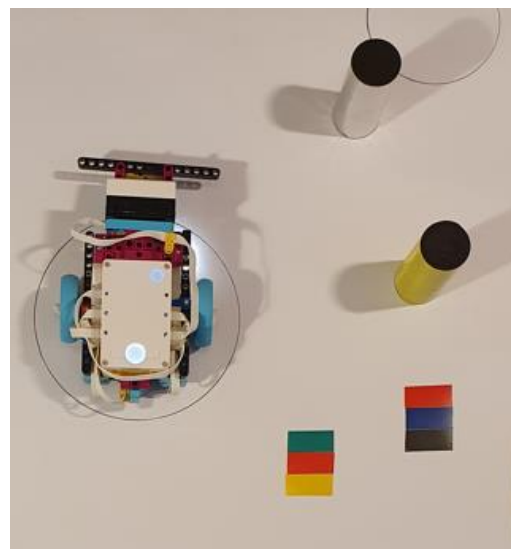
Placer bilen i cirklen i midten og de to paprør uden om (eller anden dims) – se billede. Bilen skal nu drejerundt og skubbe de to paprør ud til siderne.

Lav et nyt program og kald det *bil-rydop*.

- 1) Placer bilen i cirklen i midten. Man kan med fordel montere en pind på tværs på gribearmen for at gøre den lidt bredere – se billedet.
- 2) Placer de to paprør ca. 15 cm uden for cirklen
- 3) Kopier koden fra før ("Kør tæt på en klods")

Opgave

- 1) Få bilen til at dreje langsomt rundt om sig selv i små steps, dvs. drej lidt, vendt lidt, drej lidt osv.
- 2) Aflæs afstandssensoren og stop bilen hvis den ser noget tættere på end 20 cm
- 3) Kør bilen frem og tilbage for at skubbe tingen væk (koden skal stå inde i `if` sætningen)
- 4) Forsæt med at dreje bilen rundt og kik efter næste ting
- 5) Brug lysdioderne i afstandssensoren til noget, som fx at tænde dem når den ser noget.
- 6) Gem programmet



5 ToDo

5.1 Bil følg en streg

5.2 Reager på farverækkefølgen (avanceret)

5.3 Skridt tæller

5.4 Plotter (CNC-maskine)

Lego lektion: **Opfinderholdet->Defekt**

Lav koden i Python

6 Løsninger

6.1 Bil kontrol

6.1.1 Klods til cirkel

```
from hub import port
import runloop
import motor_pair

async def main():
    motor_pair.pair(motor_pair.PAIR_1, port.D, port.C)
    await motor_pair.move_for_time(motor_pair.PAIR_1, 6000, 0, velocity=500)
    await motor_pair.move_for_time(motor_pair.PAIR_1, 6000, 0, velocity=-500)

runloop.run(main())
```

6.1.2 Firkant

```
from hub import port
import runloop
import motor_pair

async def main():
    motor_pair.pair(motor_pair.PAIR_1, port.D, port.C)
    for i in range(4):
        await motor_pair.move_for_time(motor_pair.PAIR_1, 1000, 0, velocity=500)
        await motor_pair.move_for_time(motor_pair.PAIR_1, 810, 100, velocity=200)

runloop.run(main())
```

6.2 Bil farvesensor

6.2.1 Bil-count

```
from hub import port
from hub import light_matrix
import runloop
import motor_pair
import color_sensor
import color

async def main():
    count = 0
    motor_pair.pair(motor_pair.PAIR_1, port.D, port.C)
    motor_pair.move(motor_pair.PAIR_1, 0, velocity=400)
    for i in range(60):
        farve = color_sensor.color(port.A)
        if farve == color.BLACK:
            count += 1
        await runloop.sleep_ms(50)
        light_matrix.write(str(count))
    motor_pair.stop(motor_pair.PAIR_1)
runloop.run(main())
```

6.2.2 Bil-farve-liste

```
from hub import port
from hub import light_matrix
import runloop
import motor_pair
import color_sensor
import color

farveTxt = ["BLACK", "MAGENTA", "PURPLE", "BLUE", "AZURE",
"TORQUISE", "GREEN", "YELLOW", "ORANGE", "RED", "WHITE"]

async def main():
    prevFarve = color.WHITE
    motor_pair.pair(motor_pair.PAIR_1, port.D, port.C)
    motor_pair.move(motor_pair.PAIR_1, 0, velocity=300)
    for i in range(50):
        farve = color_sensor.color(port.A)
        if farve != color.WHITE and farve != color.UNKNOWN and farve
!= prevFarve:
            prevFarve = farve
            print(farveTxt[farve])
            await runloop.sleep_ms(50)

    motor_pair.stop(motor_pair.PAIR_1)
runloop.run(main())
```

6.3 Bil afstandssensor

6.3.1 Kør tæt på klods

```
from hub import port
import runloop
import motor_pair
import distance_sensor

async def main():
    pixels = [100] * 4
    distance_sensor.show(port.F, pixels)
    motor_pair.pair(motor_pair.PAIR_1, port.D, port.C)
    motor_pair.move(motor_pair.PAIR_1, 0, velocity=300)
    for i in range(50):
        afstand = distance_sensor.distance(port.F)
        if afstand < 50:
            motor_pair.stop(motor_pair.PAIR_1)
            distance_sensor.clear(port.F)
            break
        await runloop.sleep_ms(50)
    motor_pair.stop(motor_pair.PAIR_1)
runloop.run(main())
```

6.3.2 Ryd området

```
from hub import port
import runloop
import motor_pair
import distance_sensor

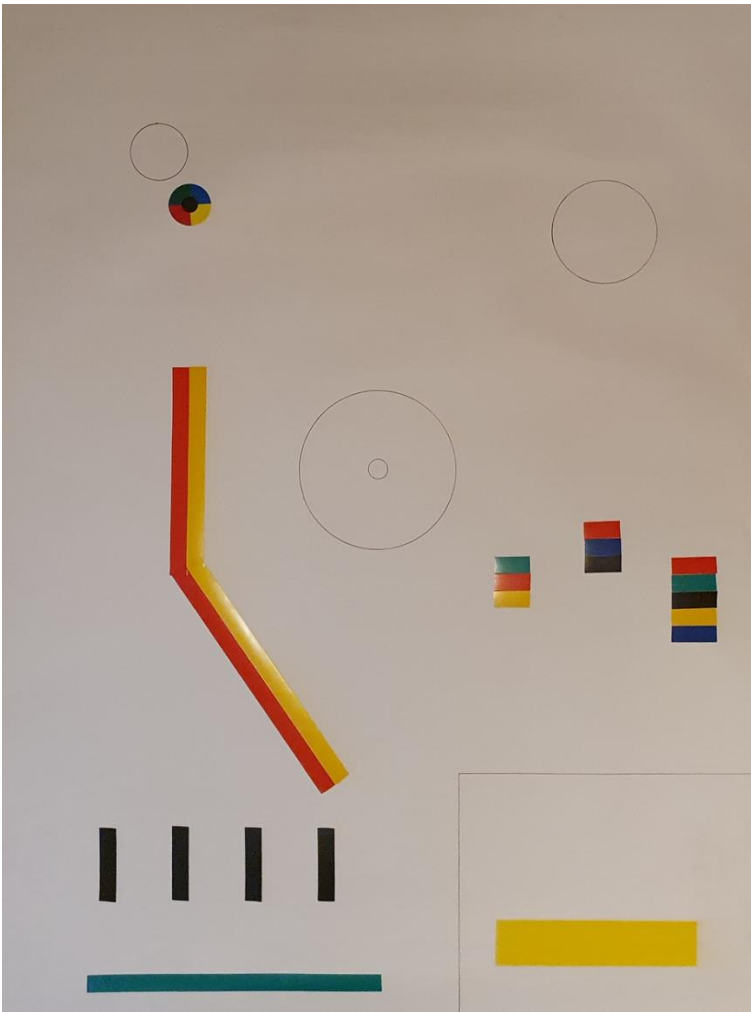
async def main():
    pixels = [100] * 4
    motor_pair.pair(motor_pair.PAIR_1, port.D, port.C)

    for i in range(20):
        await motor_pair.move_for_time(motor_pair.PAIR_1, 500, 100, velocity=50)
        await runloop.sleep_ms(50)
        afstand = distance_sensor.distance(port.F)
        if afstand > 0 and afstand < 200:
            motor_pair.stop(motor_pair.PAIR_1)
            distance_sensor.show(port.F, pixels)
            await motor_pair.move_for_time(motor_pair.PAIR_1, 1500, 0, velocity=500)
            await motor_pair.move_for_time(motor_pair.PAIR_1, 1500, 0, velocity=-500)
            distance_sensor.clear(port.F)
        motor_pair.stop(motor_pair.PAIR_1)
    runloop.run(main())
```

7 Appendiks

7.1 Kort/bane

Kortet er tegnet på et A0 papir (bagsiden af en plakat) og striberne er lavet med tape i forskellige farver.



7.2 Links