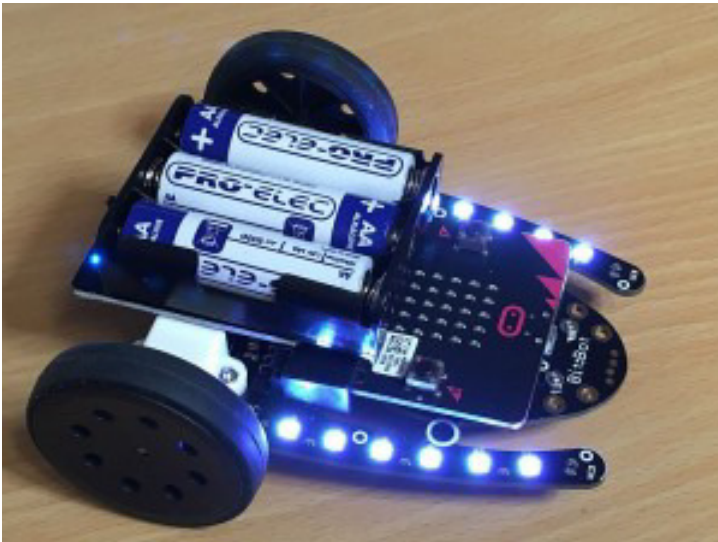


# Bit:Bot – Flere opgaver!



## INDLEDNING

Her er yderligere 10 opgaver, som jeg har samlet og oversat. Programmerne er hentet fra Firmaet 4tronix hjemmeside: <http://4tronix.co.uk/blog/?p=1490> og fra Mark Atkinsons hjemmeside "Multiwingspan:" <http://multiwingspan.co.uk/micro.php?page=bitbot>

I **Øvelse 2 & 3**, er det en fordel at programmere din Bit:Bot ved hjælp af [Microsoft PXT-Editoren](#).

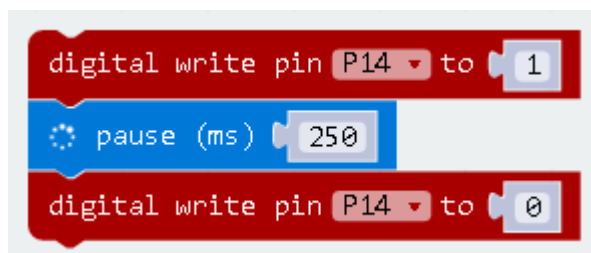
I **Øvelse 4 - 10** er det dog [MicroPython Editoren](#) du skal bruge!

*Venlig hilsen og god fornøjelse!*

*Søren Westerholm, Coding Pirates Sorø.*

## OPGAVE 2: BRUG HORNET!

Dette er et yderst simpelt program:



Programmet ser sådan ud i JavaScript:

```
pins.digitalWritePin(DigitalPin.P14, 1)
basic.pause(250)
pins.digitalWritePin(DigitalPin.P14, 0)
```

## UDFORDRING:

1. Indsæt programstumpen til Program 1, så din Bit:Bot dytter 1 gang når den kører forlæns og 2 gange når den kører baglæns!

2. Konstruer selv en udfordring!

### ØVELSE 3: 'NEOPIXELS'

er navnet for sammenkædede lysdioder (RGB LEDs), der kører på jævnstrøm. Med en enkelt **GPIO pin** kan du styre en anseelig bunke af disse!

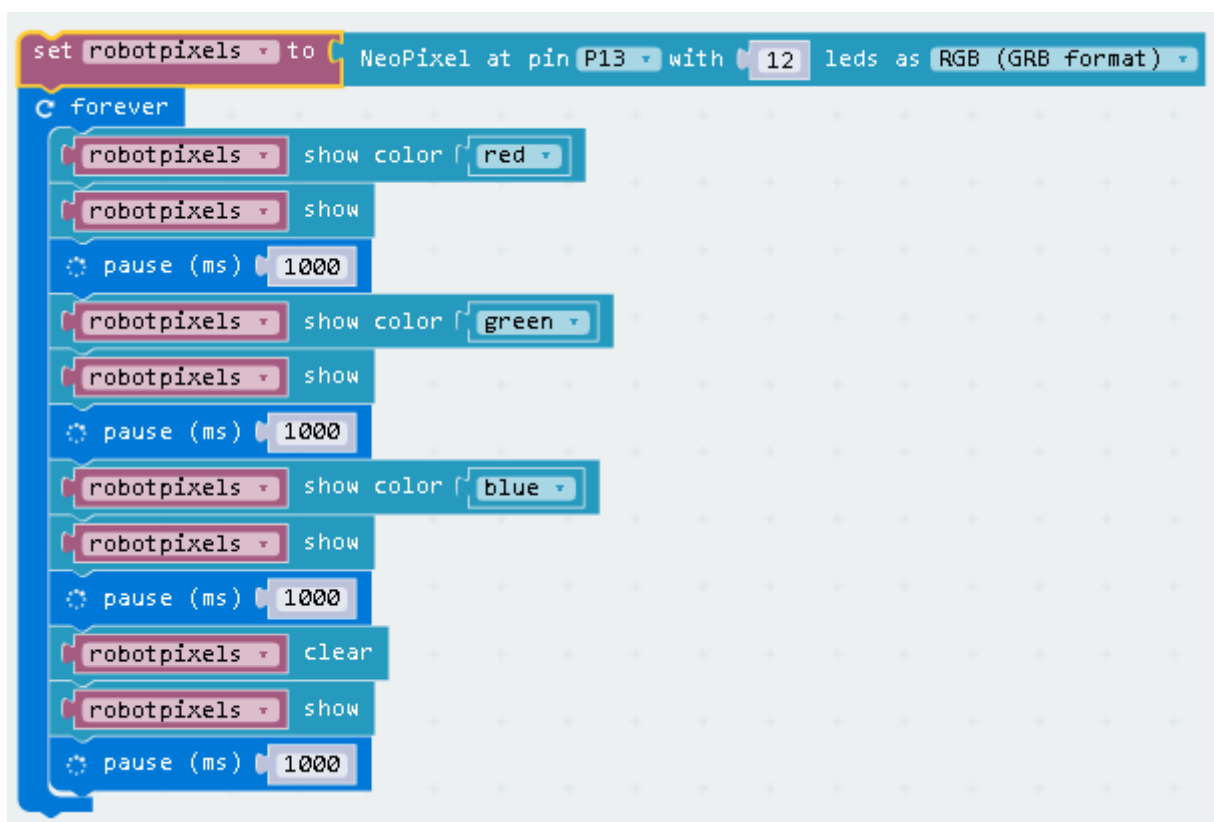
Der er 12 Neopixels på din Bit:Bot, 6 på hver side. De er praktisk nok nummereret fra 0 - 11 på PCB forbindelsen.

De 12 Neopixels er forbundet til pin 13.

#### INDLEDNING

For at kunne bruge Neopixel blokkene, skal du først importere biblioteket. Dette gør du ved at klikke på "**Settings link.**" Vælg **import package** for at importere biblioteket.

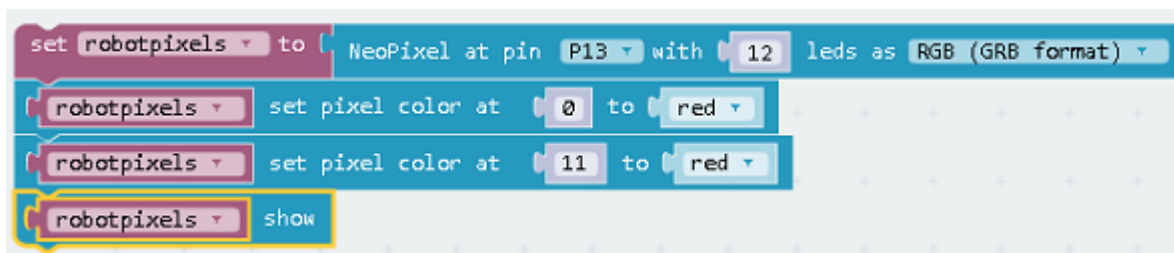
De nedenstående blokke bruges til at tænde dine Neopixels i 1 farve!



Som JavaScript ser programmet ud som herunder:

```
let robotpixels = neopixel.create(DigitalPin.P13, 12, NeoPixelMode.RGB)
basic.forever(() => {
  robotpixels.showColor(neopixel.colors(NeoPixelColors.Red))
  robotpixels.show()
  basic.pause(1000)
  robotpixels.showColor(neopixel.colors(NeoPixelColors.Green))
  robotpixels.show()
  basic.pause(1000)
  robotpixels.showColor(neopixel.colors(NeoPixelColors.Blue))
  robotpixels.show()
  basic.pause(1000)
  robotpixels.clear()
  robotpixels.show()
  basic.pause(1000)
})
```

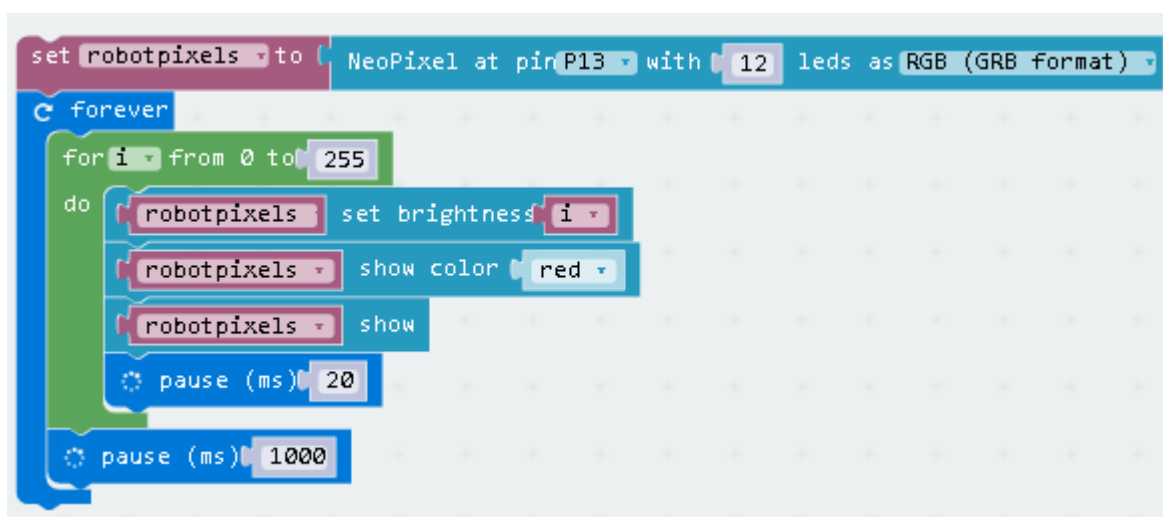
Du kan tænde dine Neopixels individuelt. Nedenstående program tænder den første og den sidste Pixel. De er nummereret fra 1 - 11.



Som JavaScript ser programmet ud som herunder:

```
let robotpixels = neopixel.create(DigitalPin.P13, 12, NeopixelMode.RGB)
robotpixels.setPixelColor(0, neopixel.colors(NeopixelColors.Red))
robotpixels.setPixelColor(11, neopixel.colors(NeopixelColors.Red))
robotpixels.show()
```

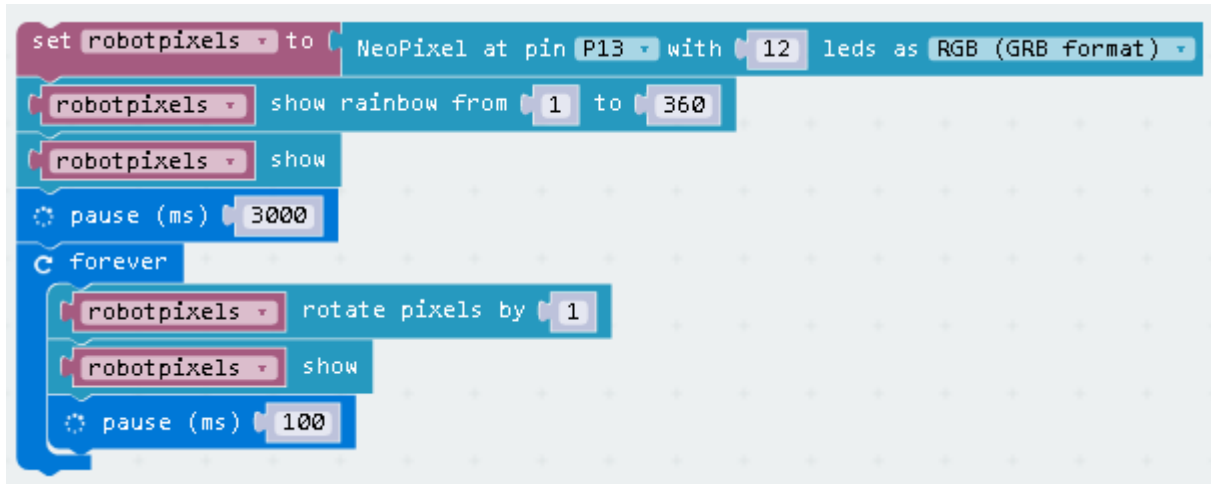
Du kan ændre på lysintensiteten af dine Neopixels med følgende programstump:



Som JavaScript ser programmet ud som herunder:

```
let robotpixels = neopixel.create(DigitalPin.P13, 12, NeopixelMode.RGB)
basic.forever(() => {
  for (let i = 0; i <= 255; i++) {
    robotpixels.setBrighthness(i)
    robotpixels.showColor(neopixel.colors(NeopixelColors.Red))
    robotpixels.show()
    basic.pause(20)
  }
  basic.pause(1000)
})
```

Der er indbyggede funktioner som du kan lege med. De giver nogle flotte effekter. Her er et par stykker som du kan prøve:



Som JavaScript ser programmet ud som herunder:

```
let robotpixels = neopixel.create(DigitalPin.P13, 12, NeopixelMode.RGB)
robotpixels.showRainbow(1, 360)
robotpixels.show()
basic.pause(3000)
basic.forever(() => {
  robotpixels.rotate(1)
  robotpixels.show()
  basic.pause(100)
})
```

## UDFORDRING:

1. Prøv i forbindelse med Øvelse 1 og Øvelse 2 at bruge lysstyrke og farver til f. eks at vise af til venstre / højre.
2. Ved at bruge forskellige grader af rødt kan du prøve at efterligne et hjertes pulsslag

## ØVELSE 4: FØLG LINIEN!

### Indledning

Din Bit:Bot har 2 linesensorer monteret på undersiden. De er forbundet til de samme PIN's som knapperne på din Micro:Bit. Du bruger en `read_digital()` kommando på disse pinde til at kontrollere om sensorerne ser en sort linje.

Ved at ændre indstillingerne på venstre og højre motors hastighed, kan du, ud fra sensorernes data få din Bit:Bot til f.eks. at følge en cirkel!

Hvis den venstre sensor detekterer en sort linje, skal du øge hastigheden på den højre motor og hvis den højre gør det, er det den venstre motors hastighed, der skal øges!

Her så programmet, skrevet i MicroPython. Det fungerer fint på en Lego Mindstorm bane, som jo har en ret tyk linje at følge!

Det kan sagtens være, at du kommer til at ændre en smule på værdierne for at få din Bit:Bot til at følge en anden linjetykkelse!

```
from microbit import *

def Drive(lft, rgt):
    pin8.write_digital(0)
    pin12.write_digital(0)
    if lft<0:
        pin8.write_digital(1)
        lft = 1023 + lft
    if rgt<0:
        rgt = 1023 + rgt
        pin12.write_digital(1)
    pin0.write_analog(lft)
    pin1.write_analog(rgt)

def FollowLine():
    lft = pin11.read_digital()
    rgt = pin5.read_digital()
    if lft==0 and rgt==1:
        #turn right
        Drive(400,100)
    elif lft==1 and rgt==0:
        # turn left
        Drive(100,400)
    elif rgt==0 and lft==0:
        # straight on
        Drive(400,400)
    sleep(20)

# a small delay before starting
sleep(1000)

# keep on following the line forever
while True:
    FollowLine()
```

## UDFORDRING:

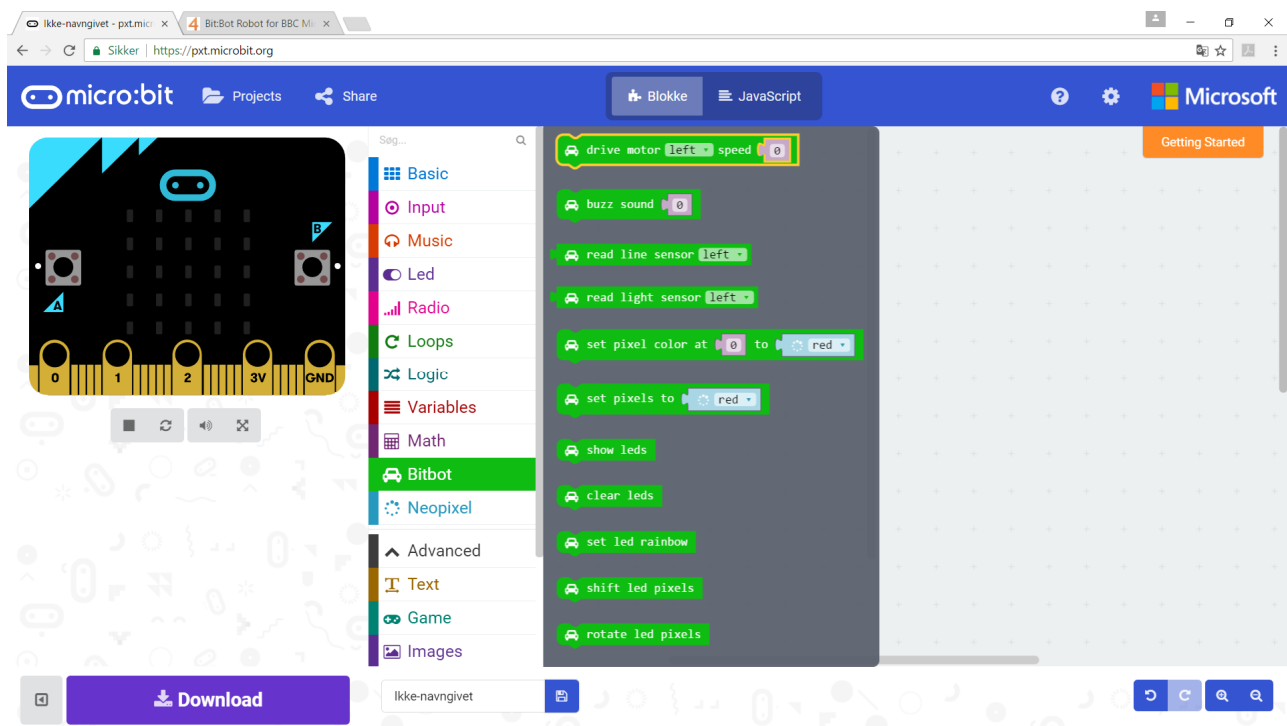
1. Brug sort elektrisk tape til at konstruere dine baner. Det er let at fjerne efter brug, men fungerer bedst hvis kontrasten er stor - altså brug hvidt papir som underlag!
2. Brug funktionerne LightLeft() and LightRight() til at bruge dine Neopixels!
3. Styr din Bit:Bot igennem en farlig labyrint!

## TILFØJELSESPAKKE TIL PXT-EDITOREN

[Microsoft PXT-Editoren](https://pxt.microbit.org). Er blevet udvidet med en "Bit:Bot" pakke.

- Du installerer pakken, når du har åbnet PXT-Editoren.
- Klik på "Advanced" folderen.
- Klik på "Add Package" og søg efter "Bit:Bot."

Din PXT-Editor har nu fået tilføjet nogle nye "Byggeklodser," som passer perfekt til programmeringen af din Bit:Bot!



## UDFORDRING:

3. Løs ØVELSE 4, men gør programmeringen nemmere og simpleere ved at bruge dine nye "Byggeklodser."

## **ØVELSE 5 - 10: FORSKELLIGE MICROPYTHON PROGRAMMER!**

Hvis du nu har fået blod på tanden med hensyn til Python programmering, er her en række småprogrammer som du sikkert kan få glæde af.

Programmerne klippe / klistres ind i MicroPython Editoren. Her kan du eventuelt tilpasse og ændre dem for til sidst at downloade dem til din Micro:Bit.

Jeg har hentet programmerne, så du kan klippe / klistre fra dem. Bemærk, at enkelte af programmerne fylder flere spalter!

Programmerne er hentet fra firmaet 4Tronics hjemmeside:

<http://4tronix.co.uk/blog/?p=1490>

## 5. MOTORTEST

```
# Add your Python code here. E.g.

from microbit import *

leftA = pin0
leftB = pin8
rightA = pin1
rightB = pin12

def forward (speedpct):
    speed = speedpct * 10.23
    leftA.write_analog (speed)
    leftB.write_digital (0)
    rightA.write_analog (speed)
    rightB.write_digital (0)

def reverse (speedpct):
    speed = speedpct * 10.23
    leftA.write_analog (1023-speed)
    leftB.write_digital (1)
    rightA.write_analog (1023-speed)
    rightB.write_digital (1)

def spinLeft (speedpct):
    speed = speedpct * 10.23
    leftA.write_analog (1023-speed)
    leftB.write_digital (1)
    rightA.write_analog (speed)
    rightB.write_digital (0)

def spinRight (speedpct):
    speed = speedpct * 10.23
    leftA.write_analog (speed)
    leftB.write_digital (0)
    rightA.write_analog (1023-speed)
    rightB.write_digital (1)

def stop():
    leftA.write_analog (0)
    leftB.write_digital (0)
    rightA.write_analog (0)
    rightB.write_digital (0)

while True:
    display.show(Image.ARROW_N)
    forward (50)
    sleep (3000)
    display.show(Image.ARROW_E)
    spinRight (50)
    sleep (3000)
    display.show(Image.ARROW_S)
    reverse (50)
    sleep (3000)
    display.show(Image.ARROW_W)
    spinLeft (50)
    sleep (3000)
    display.show(Image.HAPPY)
    stop()
    sleep (3000)
```

## 6. NEOPIXEL FARVE TEST!

```
from microbit import *
import neopixel
from random import randint

np = neopixel.NeoPixel(pin13, 12)
bright = 40

while True:
    for pixel_id in range(0, len(np)):
        np[pixel_id] = (bright,0,0)
    np.show()
    sleep(1000)
    for pixel_id in range(0, len(np)):
        np[pixel_id] = (0,bright,0)
    np.show()
    sleep(1000)
    for pixel_id in range(0, len(np)):
        np[pixel_id] = (0,0,bright)
    np.show()
    sleep(1000)
    for pixel_id in range(0, len(np)):
        np[pixel_id] = (bright,bright,bright)
    np.show()
    sleep(1000)
```

## 7. FØLG LYSET!

```
from microbit import *

leftA = pin0
leftB = pin8
rightA = pin1
rightB = pin12
lightSensor = pin2
lightSelect = pin16

while True:
    lightSelect.write_digital(0) # select left
    sensor
    rval = lightSensor.read_analog()
    leftA.write_analog(rval)
    leftB.write_digital(0)
    sleep(1)
    lightSelect.write_digital(1) # select
    right sensor
    rval = lightSensor.read_analog()
    rightA.write_analog(rval)
    rightB.write_digital(0)
    sleep(1)
```



## 8. SKANNER

```
from microbit import *
import neopixel
from random import randint

numpixels = 12
np = neopixel.NeoPixel(pin13, numpixels)
bright = 40

def setPixel(id, val):
    if (id>=0 and id<=5):
        np[id] = (val, 0, val)
        np[id+6] = (val, 0, val)

while True:
    for i in range (0, 6):
        setPixel(i-3, 0)
        setPixel(i-2, 2)
        setPixel(i-1, 10)
        setPixel(i, 40)
        setPixel(i+1, 10)
        setPixel(i+2, 2)
        setPixel(i+3, 0)
        np.show()
        sleep(100)
    for i in range (6, 0, -1):
        setPixel(i-3, 0)
        setPixel(i-2, 2)
        setPixel(i-1, 10)
        setPixel(i, 40)
        setPixel(i+1, 10)
        setPixel(i+2, 2)
        setPixel(i+3, 0)
        np.show()
        sleep(100)
```

## 9. FØLG LINIEN!

# Add your Python code here. E.g.

```
from microbit import *

leftA = pin0
leftB = pin8
rightA = pin1
rightB = pin12
leftLine = pin11
rightLine = pin5
speedFwd = 100
speedTurn = 40

def reverse (spd):
    spd = spd * 10.23
    leftA.write_analog(1023-spd)
    leftB.write_digital(1)
    rightA.write_analog(1023-spd)
    rightB.write_digital(1)

def forward (spd):
    spd = spd * 10.23
    leftA.write_analog(spd)
    leftB.write_digital(0)
    rightA.write_analog(spd)
    rightB.write_digital(0)

def spinLeft (spd):
    spd = spd * 10.23
    leftA.write_analog(1023-spd)
    leftB.write_digital(1)
    rightA.write_analog(spd)
    rightB.write_digital(0)

def softLeft (spd):
    spd = spd * 10.23
    leftA.write_analog(350)
    leftB.write_digital(1)
```

```
rightA.write_analog(spd)
rightB.write_digital(0)

def spinRight (spd):
    spd = spd * 10.23
    leftA.write_analog(spd)
    leftB.write_digital(0)
    rightA.write_analog(1023-spd)
    rightB.write_digital(1)

def softRight (spd):
    spd = spd * 10.23
    leftA.write_analog(spd)
    leftB.write_digital(0)
    rightA.write_analog(350)
    rightB.write_digital(1)

def stop():
    leftA.write_analog(0)
    leftB.write_digital(0)
    rightA.write_analog(0)
    rightB.write_digital(0)

lastfwd = 0
while True:
    lline = leftLine.read_digital()
    rline = rightLine.read_digital()
    if((lline == 1) and (rline == 0)):
        softLeft(speedTurn)
        #sleep(1)
        lastfwd = 0
        while ((leftLine.read_digital() == 1)
and (rightLine.read_digital() == 0)):
            pass
            #sleep(1)
    elif((rline == 1) and (lline == 0)):
        softRight(speedTurn)
        #sleep(1)
        lastfwd = 0
        while ((rightLine.read_digital() == 1)
and (leftLine.read_digital() == 0)):
            pass
            #sleep(1)
    else:
        if (lastfwd == 0):
            forward(speedFwd)
            lastfwd = 1
```

## 10. POWER ON SELFTEST!

**#Power On Self Test: Tester næsten alle  
Bit:Bottens funktioner!**

```
from microbit import *
import neopixel

leftA = pin0
leftB = pin8
rightA = pin1
rightB = pin12
leftLine = pin11
rightLine = pin5
buzzer = pin14
lightSensor = pin2
lightSelect = pin16
np = neopixel.NeoPixel(pin13, 12)

sleepDel = 800

def forward():
    leftA.write_digital(1)
    leftB.write_digital(0)
    rightA.write_digital(1)
    rightB.write_digital(0)

def reverse():
    leftA.write_digital(0)
    leftB.write_digital(1)
    rightA.write_digital(0)
    rightB.write_digital(1)

def stop():
    leftA.write_digital(0)
    leftB.write_digital(0)
    rightA.write_digital(0)
    rightB.write_digital(0)

def showAll(red, green, blue):
    for i in range(12):
        np[i] = (red, green, blue)
    np.show()

def showLeftLine():
    aset = leftLine.read_digital() * 9
    for i in range(4):
        display.set_pixel(0, i, aset)
    display.set_pixel(0, 4, 9)

def showRightLine():
    aset = rightLine.read_digital() * 9
    for i in range(4):
        display.set_pixel(4, i, aset)
    display.set_pixel(4, 4, 9)

def showLight():
    lightSelect.write_digital(0)
    leftLight = lightSensor.read_analog()
    lightSelect.write_digital(1)
    rightLight =
lightSensor.read_analog()
    for i in range(5):
        if leftLight > i*200:
            display.set_pixel(1, 4-i, 9)
        else:
            display.set_pixel(1, 4-i, 0)
        if rightLight > i*200:
            display.set_pixel(3, 4-i, 9)
        else:
            display.set_pixel(3, 4-i, 0)
```

```
while True:
    forward()
    showAll(50,0,0)
    showLeftLine()
    showRightLine()
    showLight()
    sleep(sleepDel)
    stop()
    showAll(0,50,0)
    showLeftLine()
    showRightLine()
    showLight()
    sleep(sleepDel)
    stop()
    reverse()
    showAll(0,0,50)
    showLeftLine()
    showRightLine()
    showLight()
    sleep(sleepDel)
    stop()
    showAll(50,50,50)
    buzzer.write_digital(1)
    showLeftLine()
    showRightLine()
    showLight()
    sleep(sleepDel)
    buzzer.write_digital(0)
    stop()
```