

## 1. SOS

- Programmeer de SOS-code als drie korte tonen gevolgd door drie lange tonen en nogmaals dezelfde drie korte tonen; allemaal in dezelfde toonhoogte. Zend de SOS-code 10-maal na mekaar uit.
- Combineer de audio SOS-code met het knipperen van een led: drie korte flitsen, drie lange en opnieuw drie korte.



## 2. Straatverlichting



Schrijf een programma dat een led aanzet indien het duister wordt, b.v. de ingebouwde kleur-led met rgb-waarde wit = (255,255,255), en uitzet indien er terug opnieuw voldoende licht is.

## 3. Licht-muziek

Codeer het volgende deuntje gebruik makend van letternamen en van frequenties.

C D E C  
*Broeder Jacob*  
C D E C  
*Broeder Jacob*  
E F G  
*Slaap jij nog*  
E F G  
*Slaap jij nog*  
G A G F E C  
*Alle klokken luiden*  
G A G F E C  
*Alle klokken luiden*  
C G C  
*Bim Bam Bom*  
C G C  
*Bim Bam bom*

Va- der Ja- cob, slaapt gij nog?  
Al- le klok- ken lui- den. Bim bam bom.



Pas de code aan zodat bij iedere noot de RGB-led in een bepaalde kleur oplicht.

Ga creatief aan de slag met muziek!

## 4. Rij/Wandel de grafiek

### a. Cirkelbeweging

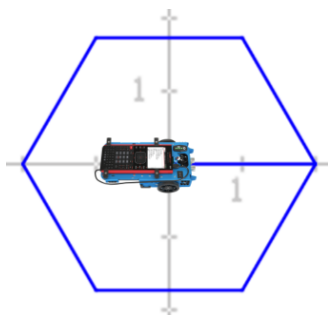
Voor de onderstaande code krijgen we de volgende resultaten.

#### TI-Innovator Rover

```
from math import *
import ti_rover as rv

points = 6
scale = 2*pi/points
p=[i*scale for i in range(points+1)]

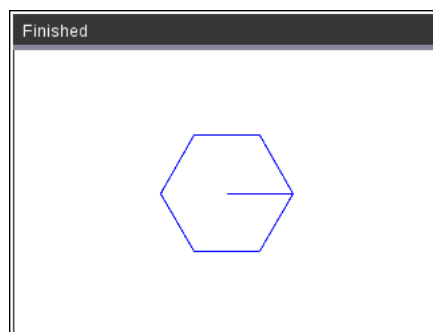
for t in parameter:
    ♦♦x=2*cos(t)
    ♦♦y=2*sin(t)
    ♦♦rv.to_xy(x,y)
```



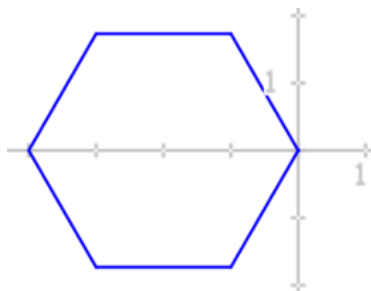
#### CX Turtle

```
from cx_turtle import *
from math import *

rv=turtle()
rv.color(0,0,255)
points=10
scale=2*pi/points
p=[i*scale for i in range(points+1)]
for t in p:
    ♦♦x=50*cos(t)
    ♦♦y=50*sin(t)
    ♦♦rv.goto(x,y)
```



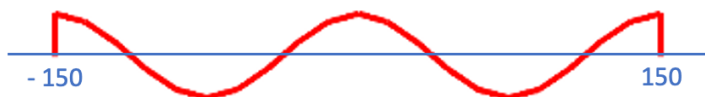
Pas de code voor de parameterkromme aan om de volgende output te krijgen.



b. SinRide

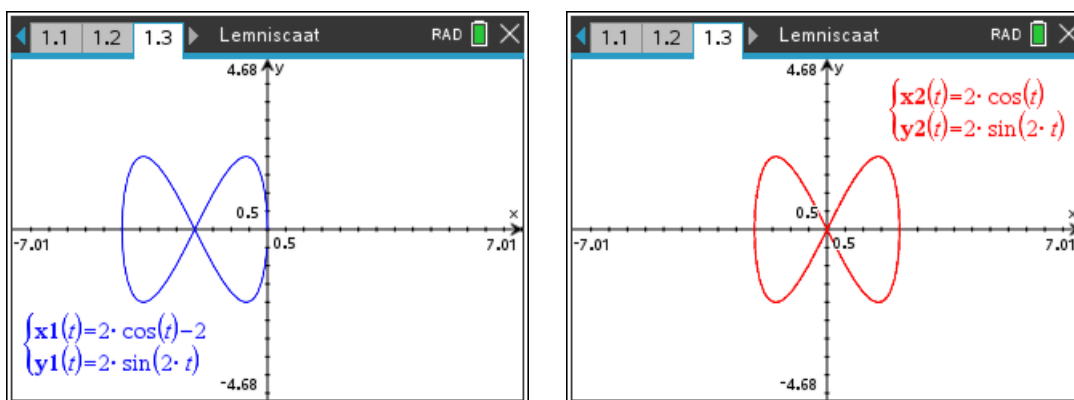
met fase-verschil van grootte  $\frac{\pi}{2}$

- o Laat de Rover een cosinus-curve rijden/tekenen startend in de oorsprong tot  $4\pi$ .
- o Teken de volgende cosinus curve met de CX Turtle met amplitude 20 pixels.



c. Oneindige lus

Laat de Rover een oneindige lus rijden/tekenen:



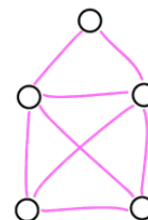
Voor de CX-Turtle geldt dat 1 eenheid gelijk is aan 1 pixel. Voor het tekenen van een oneindig lus met de Turtle best een schaalvergroting toevoegen, b.v. 30 in de x-richting en 20 in de y-richting.

d. Eenrichtingsverkeer

In grafentheorie noemt met het hiernaast afgebeelde huisje een Euler-pad. Dit wil zeggen dat deze figuur in één pennentrek kan getekend worden, van punt naar punt, zonder één lijnstuk dubbel te tekenen.

Schrijf een programma dat de Rover (of de CX-Turtle) zo'n huisje laat tekenen:

- o gebruikmakend van coördinaten,
- o zonder gebruik te maken van coördinaten.



e. Mandelbrot-verzameling

Het hart van de Mandelbrot-verzameling, waar oneindig veel bollen aangehecht zijn, heeft als rand de vorm van een cardioïde.

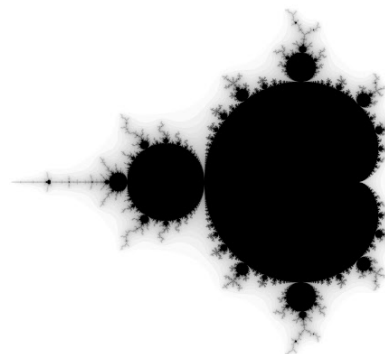
De vergelijking van een cardioïde,  $a \in \mathbb{R}$ :

- o Parametervoorstelling  

$$x(t) = 2a \cdot \cos(t) (1 - \cos(t))$$

$$y(t) = 2a \cdot \sin(t) (1 - \cos(t))$$
- o Polaire coördinaten  

$$r(\theta) = 2a(1 - \cos(\theta))$$

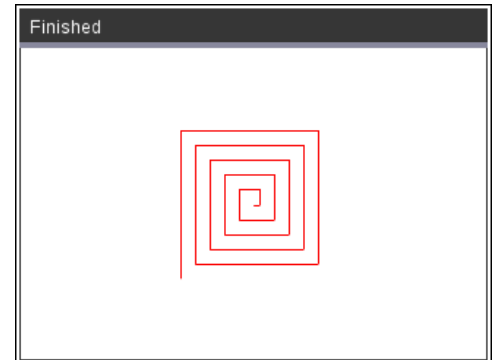


Laat de Rover/Turtle een cardioïde rijden. Merk op dat de TI Rover-module een functies heeft om naar een punt gedefinieerd door poolcoördinaten te rijden: `rv.to_polar(radius,theta)`.

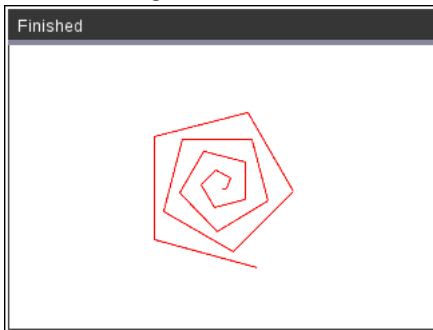
## 5. Spiraal-plezier

Gebruikmakend van de `cx_turtle`-module kan je met de volgende code creatief spiralen tekenen.

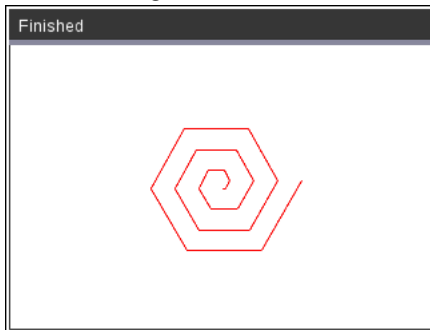
```
from cx_turtle import *
rv=turtle()
width=5
angle=90
segments=20
rv.color(250,0,0)
for x in range(1,segments+1):
    ♦♦rv.forward(x*width)
    ♦♦rv.left(angle)
```



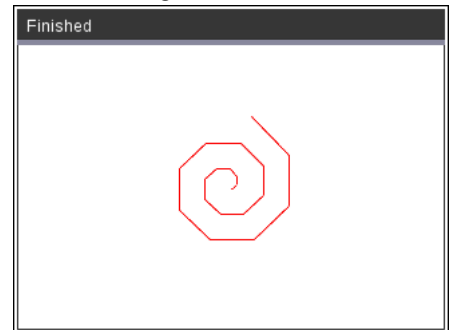
width=4 ; angle=75



width=3 ; angle=60

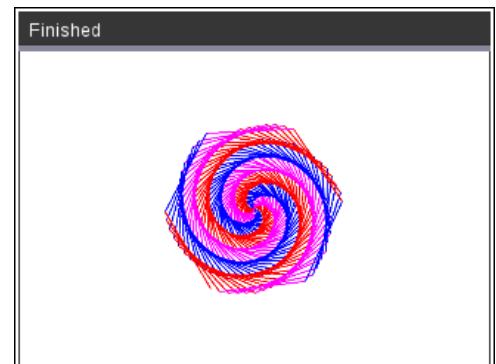


width=2 ; angle=45

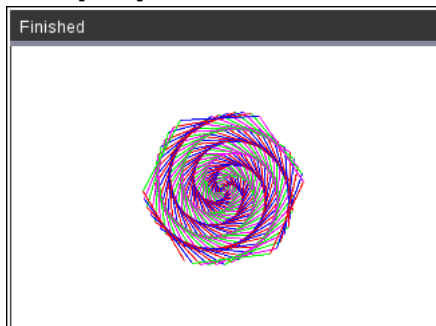


Experimenteren met kleur kan als volgt:

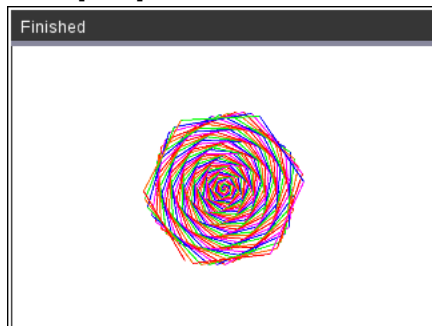
```
from cx_turtle import *
rv=turtle()
width=6
angle=59
segments=360
colors=['red','magenta','blue','green','orange','cyan']
for x in range(1,segments+1):
    ♦♦rv.color(colors[x%3])
    ♦♦rv.forward(x/width)
    ♦♦rv.left(angle)
```



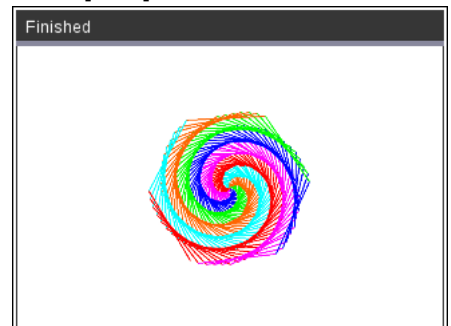
colors[x%4]



colors[x%5]

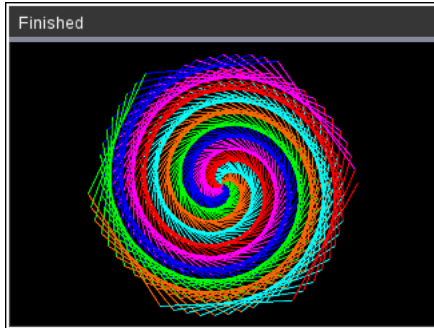


colors[x%6]

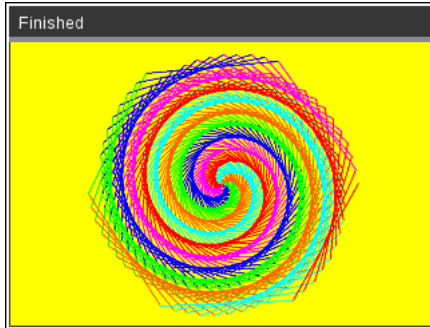


Een manier, om met deze relatief beperkte Turtle-module, de achtergrond te kleuren is gebruik te maken van de dot()-functie(methode) en zo b.v. een gevulde cirkel tekenen met middelpunt de oorsprong en straal 200:

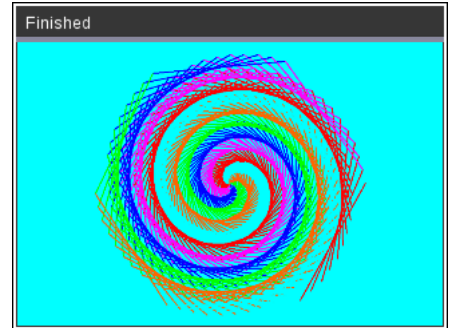
```
rv.home()  
rv.color('black')  
rv.dot(200)
```



```
rv.home()  
rv.color('yellow')  
rv.dot(200)
```



```
r.home()  
rv.color('cyan')  
rv.dot(200)
```



Veel plezier met het kunstig tekenen van spiralen!

