**OEFENINGEN Grafische user interface met tkinter**  
  
Oef 1

Stuur met een GUI een led verbonden met GPIO 25.

Plaats een knop ON en een knop OFF op je GUI

from tkinter import \*

import RPi.GPIO as GPIO

window = Tk()

window.geometry("250x50")

window.title ("Led on/off")

led=25

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(led, GPIO.OUT)

def led\_on():

GPIO.output(led, GPIO.HIGH)

def led\_off():

GPIO.output(led, GPIO.LOW)

btn\_on = Button(window, text="On", command=led\_on)

btn\_on.place(x=10, y=10)

btn\_off = Button(window, text="Off", command=led\_off)

btn\_off.place(x=100, y=10)

window.mainloop()

Oef 2

Stuur met een GUI een led verbonden met GPIO 25.

Plaats een knop TOGGLE op je GUI

from tkinter import \*

import RPi.GPIO as GPIO

window = Tk()

window.geometry("250x50")

window.title ("Toggle LED")

led=25

status\_led= 0

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(led, GPIO.OUT)

def led\_toggle():

global status\_led

status\_led= not status\_led

GPIO.output(led, status\_led)

print("LED nu:",status\_led)

btn\_toggle = Button(window, text="TOGGLE LED", command=led\_toggle)

btn\_toggle.place(x=10, y=10)

window.mainloop()

Oef 3

Herhaal oef 1 en 2 en toon in de GUI de status van een LED ( 0 of 1) in een label  
  
from tkinter import \*

import RPi.GPIO as GPIO

window = Tk()

window.geometry("250x50")

window.title ("Toggle LED")

led=25

status\_led= 0

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(led, GPIO.OUT)

def led\_toggle():

global status\_led

status\_led= not status\_led

GPIO.output(led, status\_led)

txt\_led\_status["text"] = status\_led

print("LED nu:",status\_led)

btn\_toggle = Button(window, text="TOGGLE LED", command=led\_toggle)

btn\_toggle.place(x=10, y=10)

txt\_led\_status = Label(window, text="0" , relief = "sunk")

txt\_led\_status.place(x=150 , y = 15)

window.mainloop()

Oef 4

Gebruik een slider om de knippersnelheid van je LED in te stellen tussen 1 en 10 keer per seconde

from tkinter import \*

import RPi.GPIO as GPIO

import time

window = Tk()

window.geometry("250x100")

window.title ("Toggle LED")

led=25

status\_led= 0

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(led, GPIO.OUT)

def led\_toggle():

global status\_led

status\_led= not status\_led

GPIO.output(led, status\_led)

print("LED nu:",status\_led)

def change\_slider(p):

global knippersnelheid

knippersnelheid=int(p)

print("p=",p)

slider = Scale(window, from\_=1,to=10, tickinterval =1 , length= 200, orient = HORIZONTAL , command = change\_slider)

slider.place(x=10, y=10)

time\_vorige=0

knippersnelheid=1

while True:

if time.time() - time\_vorige > (1/knippersnelheid)/2:

led\_toggle()

time\_vorige=time.time()

window.update()

Oef 5

Gebruik een slider om de tijdsduur van je LED in te stellen tussen 1 en 100 seconden.

Met een knop met opschrift “LICHT” start je de led met de ingestelde tijd.

Uitbreiding: Geef feedback over de status met kleuren of met afbeeldingen ( zelf opzoeken hoe dat moet)

from tkinter import \*

import RPi.GPIO as GPIO

import time

window = Tk()

window.geometry("500x200")

window.title ("Toggle LED")

led=25

tijd\_aan=1

status\_led =0

time\_start=0

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(led, GPIO.OUT)

def led\_aan():

global status\_led, time\_start

time\_start=time.time()

status\_led = 1

GPIO.output(led, 1)

print("LED nu aan!")

def change\_slider(p):

global tijd\_aan

tijd\_aan=int(p)

print("tijd\_aan=",tijd\_aan)

btn\_toggle = Button(window, text="LICHT AAN", command=led\_aan)

btn\_toggle.place(x=10, y=10)

slider = Scale(window, from\_=1,to=100, tickinterval =10 , length= 450, orient = HORIZONTAL , command = change\_slider)

slider.place(x=10, y=50)

time\_vorige=0

knippersnelheid=1

while True:

if (time.time() - time\_start >= tijd\_aan) and status\_led == 1:

GPIO.output(led, 0)

print("LED nu uit!")

status\_led = 0

print("loop still running")

window.update()

Oef 6-5b

from tkinter import \*

import RPi.GPIO as GPIO

import time

window = Tk()

window.geometry("700x200")

window.title ("Toggle LED")

led=25

tijd\_aan=0

status\_led =0

time\_start=0

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(led, GPIO.OUT)

gif\_led\_on =PhotoImage(file="on.gif")

gif\_led\_off =PhotoImage(file="off.gif")

def led\_aan():

global status\_led, time\_start

time\_start=time.time()

status\_led = 1

GPIO.output(led, 1)

print("LED nu aan!")

label\_knop\_status["image"] = gif\_led\_on

def change\_slider(p):

global tijd\_aan

tijd\_aan=int(p)

print("tijd\_aan=",tijd\_aan)

btn\_toggle = Button(window, text="LICHT AAN", width = 52 , height = 5, relief = RAISED,overrelief=SUNKEN,activebackground="lightblue", activeforeground= "white",command=led\_aan)

btn\_toggle.place(x=10, y=10)

label\_knop\_status=Label(window, image=gif\_led\_off , relief = SUNKEN)

label\_knop\_status.place(x=500 , y =10)

slider = Scale(window, from\_=0,to=100, tickinterval =10 , length= 450, orient = HORIZONTAL , command = change\_slider)

slider.place(x=10, y=120)

time\_vorige=0

knippersnelheid=1

while True:

if (time.time() - time\_start >= tijd\_aan) and status\_led == 1:

GPIO.output(led, 0)

print("LED nu uit!")

status\_led = 0

label\_knop\_status["image"] = gif\_led\_off

print("loop still running")

window.update()

Oef 6-6 schrijf de toestand 1 of 0 aan RP-pin 18 in een label!

from tkinter import \*

import RPi.GPIO as GPIO

import time

window = Tk()

window.geometry("100x200")

window.title ("Toggle LED")

s1=18

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(s1, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)

label\_knop\_status=Label(window, text=0 , font=('Times', '128'), relief = SUNKEN)

label\_knop\_status.place(x=5 , y =5)

while True:

toestand\_input = GPIO.input(s1)

if toestand\_input== 1:

label\_knop\_status["text"] = "1"

else:

label\_knop\_status["text"] = "0"

print("loop still running")

window.update()

Oef 7 Verander de kleur van een label van groen naar rood indien input van gesloten naar open verandert!

#Oef 6-7 Verander de kleur van een label van groen naar rood indien input van gesloten naar open verandert!

from tkinter import \*

import RPi.GPIO as GPIO

import time

window = Tk()

window.geometry("100x200")

window.title ("Toggle LED")

s1=18

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(s1, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)

label\_knop\_status=Label(window, text=0 , bg = 'lightgreen' , font=('Times', '128'), relief = SUNKEN)

label\_knop\_status.place(x=5 , y =5)

while True:

toestand\_input = GPIO.input(s1)

if toestand\_input== 1:

label\_knop\_status["text"] = "1"

label\_knop\_status["bg"] = "lightgreen"

else:

label\_knop\_status["text"] = "0"

label\_knop\_status["bg"] = "#ff5500" # rrggbb

print("loop still running")

window.update()

Oef 6-8 toon een open of gesloten schakelaar in je GUI die de toestand van je schakelaar aan een pin volgt.

from tkinter import \*

import RPi.GPIO as GPIO

import time

window = Tk()

window.geometry("350x75")

window.title ("Toggle LED")

s1=18

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(s1, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)

gif\_s\_open =PhotoImage(file="open.gif")

gif\_s\_closed =PhotoImage(file="closed.gif")

label\_knop\_status=Label(window, image= gif\_s\_closed, relief = SUNKEN)

label\_knop\_status.place(x=5 , y =5)

while True:

toestand\_input = GPIO.input(s1)

if toestand\_input== 1:

label\_knop\_status["image"] = gif\_s\_closed

else:

label\_knop\_status["image"] = gif\_s\_open

print("loop still running")

window.update()

Oef 9 Maak een simpel alarmsysteem dat je aan en uit schakelt met je GUI.

* Je kan de PIR sensor gebruiken om het alarmsysteem te triggeren.
* De buzzer kan het alarmsignaal genereren. ( 30 sec lang)  
  Je kan eventueel een in/uit-vertraging van 20 seconden toevoegen.
* In de GUI kan je de toestand vd pir tonen en ook de status van het alarm OFF/ARMED.
* Indien alarm ON en PIR getriggerd wordt zal het alarm 30 seconden lang de buzzer sturen.

Oef 10 Maak een programma dat je rotary encoder volgt en een getal tussen 0-255 op je GUI aanpast.

Draaien naar links getal vermindert.

Draaien naar rechts getal vermeerdert.

Bv met stapjes van +/-5

Probeer dit eerst zelf te vinden ( gebruik geen gevonden oplossing)

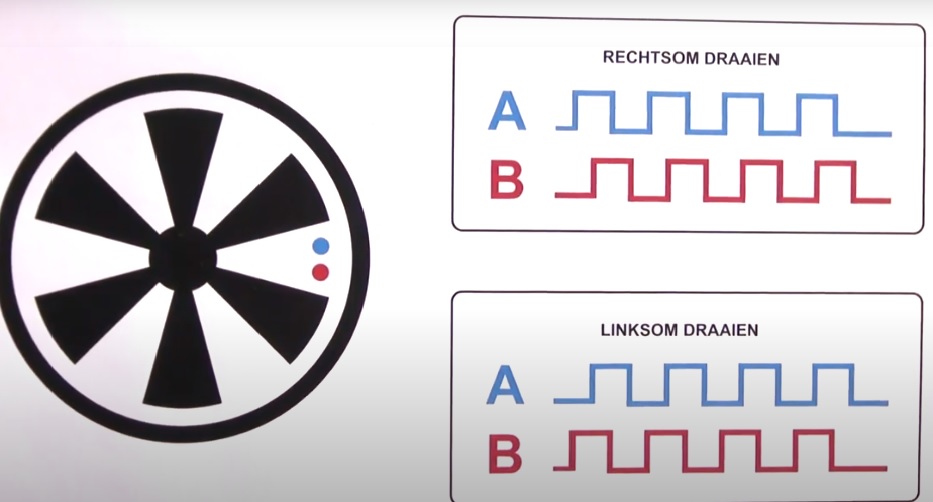
Oef 11 Gebruik je rotary encoder om een RGB waarde te genereren.

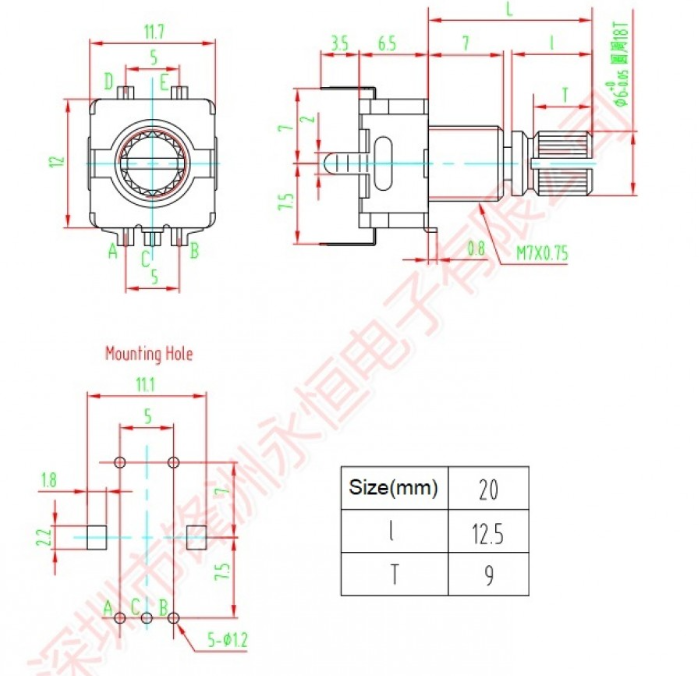
Draaien aan de rotary encoder verhoogt/verlaagt de waarde met +/- 5

Drukken op de rotary encoder verplaatst de keuze van R-G-B

Toon de waarde van RGB in 3 tekstvelden of labels in je GUI.

Toon het resultaat van de RGB selectie in je GUI door de kleur aan te passen en te tonen in een label.





A en B signalen

C common

D-E NO contact