

LISTING UITLEG

- > Start nieuwe regel
- regel met spatie na afbreking
- regel zonder spatie na afbreking

Meer informatie over de listing-
uitleg vind je in de inhoudsopgave.



JE VINDT DE LISTING
OOK OP LINUXMAG.NL

> Drie kleuren op een e-ink scherm lijkt niet zoveel nuttiger dan twee, zou je denken. Toch maakt het in de praktijk een heel verschil <

```
> while True:
>     time.sleep(0.01)

> if __name__ == '__main__':
>     main()
```

Voer de code uit en voilà! Als je op een knop drukt, toont het programma welke knop dat is.

IN- EN UITVOER COMBINEREN

En zo kunnen we tot slot in- en uitvoer in één programma combineren. Voor de eenvoud schrijven we een programma dat de vier knoppen als een soort menu gebruikt: achter elke knop schuilt andere inhoud die op het scherm wordt getoond. Deze code kun je eenvoudig aanpassen aan je eigen situatie, bijvoorbeeld als je op het scherm informatie van je domotica systeem wilt tonen.

```
> import epd2in7b
> import Image
> import ImageFont
> import ImageDraw
> import time
> import RPi.GPIO as GPIO
> import subprocess

> epd = epd2in7b.EPD()
> epd.init()

> COLORED = 1
> UNCOLORED = 0

> GPIO.setmode(GPIO.BCM)

> knop1 = 5
> knop2 = 6
> knop3 = 13
> knop4 = 19

> knoppen = [knop1, knop2, knop3, knop4]

> GPIO.setup(knoppen, GPIO.IN, pull_up_down=GPIO.PUD_UP)

> def knop1_ingedrukt(pin):
>     toonTekstZwart(subprocess.check_output(["date", "%H:%M:%S"]))
>     print('knop 1')

> def knop2_ingedrukt(pin):
>     toonTekstZwart(subprocess.check_output(["hostname", "-I"]))
>     print('knop 2')

> def knop3_ingedrukt(pin):
>     toonTekstZwart(subprocess.check_output(["curl", "ipinfo.io/ip"]))
>     print('knop 3')

> def knop4_ingedrukt(pin):
>     toonTekstZwart(subprocess.check_output(["uptime", "-p"]))
>     print('knop 4')

> GPIO.add_event_detect(knop1, GPIO.FALLING, callback=knop1_ingedrukt, bouncetime=200)
> GPIO.add_event_detect(knop2, GPIO.FALLING, callback=knop2_ingedrukt, bouncetime=200)
> GPIO.add_event_detect(knop3, GPIO.FALLING, callback=knop3_ingedrukt, bouncetime=200)
> GPIO.add_event_detect(knop4, GPIO.FALLING, callback=knop4_ingedrukt, bouncetime=200)

> def toonTekstZwart(tekst):
>     frame_black = [0] * (epd.width * epd.height / 8)
>     frame_red = [0] * (epd.width * epd.height / 8)

>     font = ImageFont.truetype('/usr/share/fonts/truetype/freefont/FreeMonoBold.ttf', 18)
>     epd.draw_string_at(frame_black, 4, 80, tekst, font, COLORED)
>     epd.display_frame(frame_black, frame_red)

> def toonTekstRood(tekst):
>     frame_black = [0] * (epd.width * epd.height / 8)
>     frame_red = [0] * (epd.width * epd.height / 8)

>     font = ImageFont.truetype('/usr/share/fonts/truetype/freefont/FreeMonoBold.ttf', 18)
>     epd.draw_string_at(frame_black, 4, 80, tekst, font, COLORED)
>     epd.display_frame(frame_black, frame_red)
```

```
> epd.draw_string_at(frame_red, 4, 80, tekst, font, COLORED)
> epd.display_frame(frame_black, frame_red)

> def main():
>     epd.set_rotate(epd2in7b.ROTATE_270)
>     toonTekstRood('Info')
>     while True:
>         time.sleep(0.01)

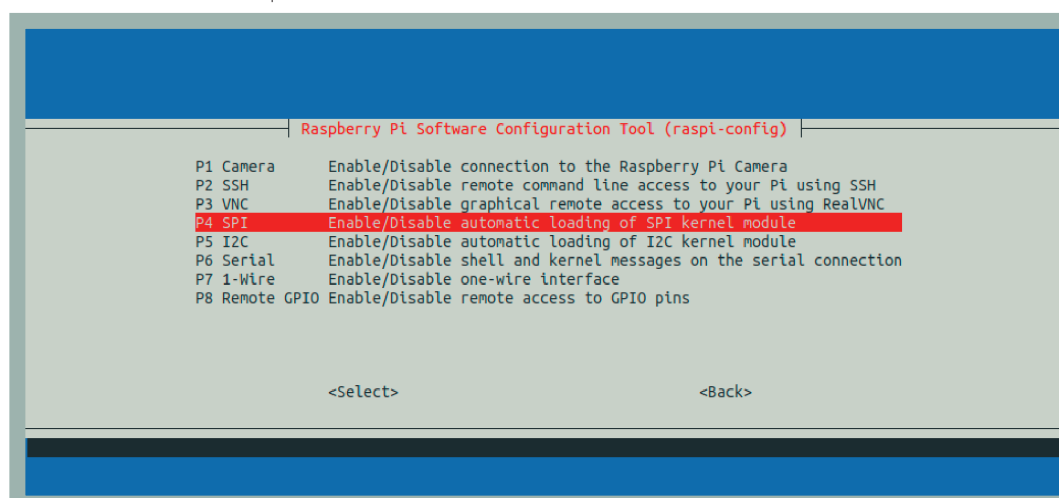
> if __name__ == '__main__':
>     main()
```

We leggen de code kort even uit. Na de benodigde bibliotheken te importeren, initialiseren we het scherm en stellen we de gpio's voor de knoppen in. Daarna definiëren we voor elke knop een callbackfunctie die wordt aangeroepen als we die knop indrukken. Elke functie toont iets anders op het scherm, in dit geval voor de eenvoud de uitvoer van een shellopdracht. Daarna voegen we voor elk van die gpio-pinnen een event toe dat de juiste callback-functie oproept.

We definiëren twee functies om tekst in het midden van het scherm te tonen: één voor zwarte tekst en één voor rode tekst. Heb je een monochroom scherm, dan verwijder je de functie voor de rode tekst. Vergeet dan ook niet een verschillend epd-object aan te maken in het begin van de code en laat frame_red weg als parameter bij epd.display_frame.

Tot slot hoeven we in de functie main alleen nog maar het scherm te roteren zodat het correct staat (in ons geval 270°, maar dat hangt ervan af hoe je de Raspberry Pi plaatst), een begintekst te tonen en een oneindige lus te starten. In een volwaardig programma kun je in de functie main gewoon allerlei taken uitvoeren; de callbackfuncties worden door RPi.GPIO in een andere thread uitgevoerd. <

✓ Schakel SPI in om met het e-Paper HAT van Waveshare te kunnen communiceren.



Neem de codes nu rechtstreeks over van linuxmag.nl. Je kan ze vinden bij Listings, rechts boven in het menu.