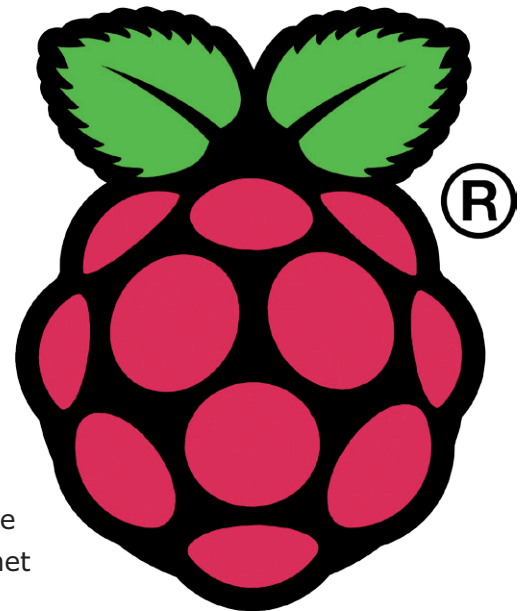


Raspberry Pi recepten deel 2

Wat extra keukenhulpjes komen altijd van pas



In deel 1 hebben we laten zien hoe Raspbian moet worden geïnstalleerd en hoe onze Raspberry Pi moet worden geconfigureerd. Zoals de vorige keer beloofd gaan we nu kijken naar de uitbreidingsconnector en hoe we de GPIO-pennen die we daar vinden kunnen programmeren. Wie het maartnummer van Elektor heeft gelezen, zal een aantal dingen herkennen omdat de uitbreidingsconnector ook aan de orde komt in het artikel over het *Raspberry Pi Prototyping Board* [1].

RPi uitbreidingsconnector: nog meer smaken

Voor de hardware-liefhebbers moet de uitbreidingsconnector wel het meest fascinerende aspect van de Pi zijn, naast zijn belachelijk lage prijs natuurlijk.

De uitbreidingsconnector zit in de hoek naast de composiet-video-aansluiting. Het is een simpele dubbele pinheader met 26 pennen en een steek van 0,1" (2,54 mm).

Op de uitbreidingsconnector vinden we drie soorten signalen:

- Voeding: +5 V DC, 3,3 V DC en massa (0 V)
(Let op: de 3,3 V kan niet meer stroom leveren dan 50 mA)
- Input/Output: universele input/output (GPIO) signalen
- Communicatie-interfaces: Seriële UART, SPI en I²C

Er zijn 17 universele input/output (GPIO) signalen op de uitbreidingsconnector beschikbaar. De meeste hiervan hebben een alternatieve functie waarmee de UART-, SPI- en I²C-interfaces worden gerealiseerd. Zie **tabel 1**.

Elke GPIO-aansluiting kan tussen de 2 en 16 mA leveren, afhankelijk van de *drive strength* die in een configuratieregister wordt ingesteld. Na een reset is deze standaard ingesteld op 8 mA.

Met versie 2 van de the Raspberry Pi werd een tweede kleinere uitbreidingsconnector (P5) geïntroduceerd (zie **tabel 2**). Deze voegt nog eens vier GPIO-signalen toe, maar het belangrijkste is (voor de audio-hobbyisten) dat deze toegang geeft tot de PCM audio-interface van de Broadcom 2835 chip.

Ook werd bij versie 2 van de Pi een kleine wijziging in de signalen op uitbreidingsconnector P1 aangebracht waarbij de I2C0-interface werd vervangen door de I2C1-interface. Belangrijk om te onthouden als je I²C-apparaten op je Pi wilt aansluiten.

Installatie van Python's GPIO-bibliotheek

We gaan onze voorbeelden in Python programmeren. Gelukkig is Python een standaard onderdeel van de Raspbian-distributie, maar om de GPIO van de Pi te kunnen gebruiken moet er een geschikte hard-

Tony Dixon
(Verenigd Koninkrijk)

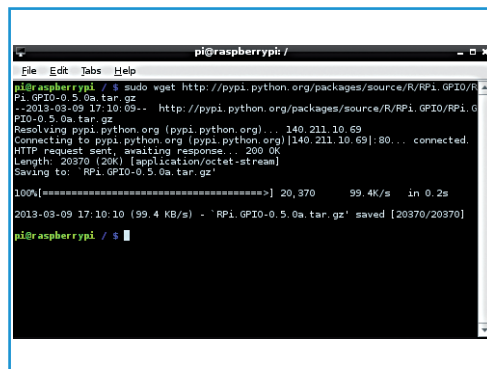
Tabel 1 - Aansluitgegevens uitbreidingsconnector P1

Naam	Functie	Alternatief	RPi.GPIO	Board Versie 1		Board Versie 2	
				Functie	Alternatief	Functie	Alternatief
P1-02	5,0 V	-	-	P1-01	3,3 V	-	3,3 V
P1-04	5,0 V	-	-	P1-03	GPIO0	I2C0_SDA	GPIO2
P1-06	GND	-	-	P1-05	GPIO1	I2C0_SCL	GPIO3
P1-08	GPIO14	UART0_TXD	RPi.GPIO8	P1-07	GPIO4	GPCLK0	GPIO4
P1-10	GPIO15	UART0_RXD	RPi.GPIO10	P1-09	GND	-	GND
P1-12	GPIO18	PWM0	RPi.GPIO12	P1-11	GPIO17	RTS0	GPIO17
P1-14	GND	-	-	P1-13	GPIO21	-	GPIO27
P1-16	GPIO23	-	RPi.GPIO16	P1-15	GPIO22	-	GPIO22
P1-18	GPIO24	-	RPi.GPIO18	P1-17	3,3 V	-	3,3 V
P1-20	GND	-	-	P1-19	GPIO10	SPI0_MOSI	GPIO10
P1-22	GPIO25	-	RPi.GPIO22	P1-21	GPIO9	SPI0_MISO	GPIO9
P1-24	GPIO8	SPI0_CE0_N	RPi.GPIO24	P1-23	GPIO11	SPI0_SCLK	GPIO11
P1-26	GPIO7	SPI0_CE1_N	RPi.GPIO26	P1-25	GND	-	GND

Noot: I2C0_SDA en I2C0_SCL (GPIO0 & GPIO1), en I2C1_SDA en I2C1_SCL (GPIO2 & GPIO3) hebben 1K8 pullup-weerstanden naar 3V3.

Tabel 2. Aansluitgegevens connector P5

Naam	Functie	Alternatief
P5-01	5,0 V	
P5-02	3,3 V	
P5-03	GPIO28	PCM_CLK
P5-04	GPIO29	PCM_FS
P5-05	GPIO30	PCM_DIN
P5-06	GPIO31	PCM_DOUT
P5-07	GND	
P5-08	GND	



Figuur 1. LX terminal.

ware-bibliotheek worden geïnstalleerd. We gebruiken de Python RPi.GPIO-bibliotheek om toegang te krijgen tot de GPIO-pennen.

Als u de Python-ontwikkel-tools of de Python GPIO-bibliotheek nog niet hebt gedownload, beginnen we met het downloaden van de Python-ontwikkeltools met het volgende commando in een LX-terminal (zie **figuur 1**) op de Pi:

```
sudo apt-get install python-dev
```

Vervolgens installeren we het Python GPIO-pakket [1] met:

```
wget http://pypi.python.org/packages/source/R/RPi.GPIO/RPi.
```

GPIO-0.5.0a.tar.gz

Na het downloaden moeten de bestanden worden uitgepakt. Type:

```
tar -zxf RPi.GPIO-0.5.0a.tar.gz
```

Er wordt een nieuwe map aangemaakt met de Python-bestanden. Type vervolgens:

```
cd RPi.GPIO-0.5.0a
```

Installeer het pakket met:

```
sudo python setup.py install
```

Hiermee is de Python RPi.GPIO-bibliotheek geïnstalleerd.

Voorbeeldprogramma - blinky.py

Nu RPi.GPIO is geïnstalleerd gaan we een testprogramma schrijven om een LED te laten knipperen. We hebben hiervoor een breadboard (MiniPiio ProtoBoard [2]) op onze Pi gemonteerd en vervolgens een LED en een weerstand van 680 Ω aangesloten tussen GPIO17 en 0 V (zie **figuur 2**).

Nadat de onderdelen zijn geplaatst dubbelklikken we op het Pi-bureaublad op IDLE om de Python shell en IDE te starten. (**figuur 3**).

Selecteer File in het menu en open een nieuw programma. Hiermee wordt de IDE-editor gestart.

Type het programma van **listing 1** in de IDLE-editor (**figuur 4**).

Vergeet niet om het programma na het invoeren op te slaan, schakel over naar een LX-terminalscherf en type het volgende commando om het programma executable te maken:

```
chmod +x blinky.py
```

Nu kan het programma worden uitgevoerd met het volgende commando:

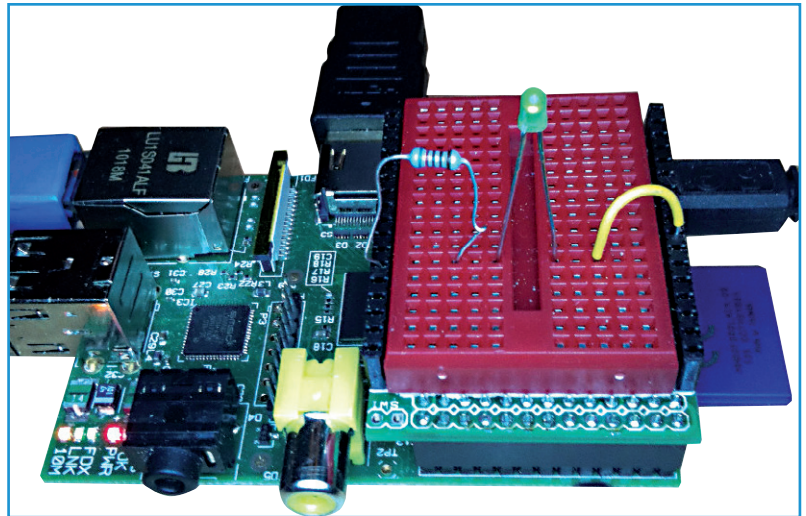
```
sudo ./blinky.py
```

TIP: als u IDLE vanuit een LX-terminalsessie start met sudo voor de programma naam (sudo idle), hebt u de juiste permissies om uw RPi.GPIO-programma vanuit IDLE te starten.

RPi.GPIO pennummering

RPi.GPIO kent twee manieren om de GPIO-sigtaalpennen op de Raspberry Pi te nummeren. In het RPi.GPIO programma moet dus altijd worden aangegeven welk GPIO-nummersysteem (GPIO.setmode(GPIO.BOARD) of GPIO.setmode(GPIO.BCM)) wordt gebruikt.

Het eerste nummersysteem GPIO.setmode(GPIO.BOARD) verwijst naar de pennummers van uitbreidingsconnector P1 van het Raspberry Pi board. Dit heeft



Figuur 2. Pi met breadboard.

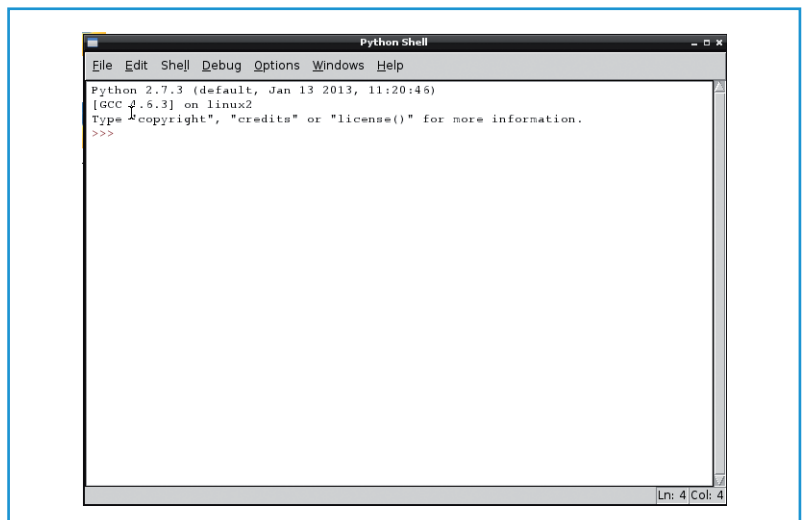
Listing: blinky.py

```
#!/usr/bin/python
import time
import RPi.GPIO as GPIO

# Configure Pi's GPIO pins
GPIO.setmode (BCM)
GPIO.setup (17,GPIO.OUT)

# Program loop
while True :
    GPIO.output (17,True)
    time.sleep (1)
    GPIO.output (17,False)
    time.sleep (1)
```

Figuur 3. I DLE Python Shell.



als voordeel dat de code niet hoeft te worden aangepast bij toekomstige wijzigingen in de signalen van de Raspberry Pi uitbreidingsconnector.

Het tweede nummersysteem GPIO.setmode(GPIO.BCM) verwijst direct naar de signaalnamen van de Broadcom 2835.

Omdat dit soms een beetje verwarrend kan zijn, laten we in **tabel 3** de relatie zien tussen P1-pennummers, GPIO-signaalnamen en GPIO.setmode.

(130110)

```

#!/usr/bin/python
import time
import RPi.GPIO as GPIO

print "Setting up GPIO"
GPIO.setmode(GPIO.BOARD)
GPIO.setup(7, GPIO.OUT)

print "go, LED's"
while True:
    GPIO.output(7, True)
    time.sleep(1)
    GPIO.output(7, False)
    time.sleep(1)
    
```

Figuur 4. IDLE Editor.

Weblinks

- [1] www.elektor-magazine.nl/120483
- [2] <https://pypi.python.org/pypi/RPi.GPIO>
- [3] www.dtronixs.com

Tabel 3. GPIO.setmode(GPIO.BCM) en GPIO.setmode(GPIO.BOARD)

Naam	Functie	GPIO.setmode	
		GPIO.BCM	GPIO.BOARD
P1-01	3,3 V	-	-
P1-02	5,0 V	-	-
P1-03	GPIO0/2*	0/2	3
P1-04	5.0V	-	-
P1-05	GPIO1/3*	1/3	5
P1-06	GND	-	-
P1-07	GPIO4	4	7
P1-08	GPIO14	14	8
P1-09	GND	-	-
P1-10	GPIO15	15	10
P1-11	GPIO17	17	11
P1-12	GPIO18	18	12
P1-13	GPIO21/27*	21	13
P1-14	GND	-	-
P1-15	GPIO22	22	15
P1-16	GPIO23	23	16
P1-17	3,3 V	-	-
P1-18	GPIO24	24	18
P1-19	GPIO10	10	19
P1-20	GND	-	-
P1-21	GPIO9	9	21
P1-22	GPIO25	25	22
P1-23	GPIO11	11	23
P1-24	GPIO8	8	24
P1-25	0 V	-	-
P1-26	GPIO7	7	26

Noot: * Wijzigingen in versie 2