

Raspberry Pi Recepten (Deel 6)

Als De Chef komt...

Tony Dixon (Verenigd Koninkrijk)

Tot nu toe hebben we in onze Elektor.POST-projecten voornamelijk gekeken naar de digitale signalen zoals GPIO, Serial UART, SPI en I²C op de uitbreidingsconnector van de Raspberry Pi. In deze aflevering gaan we wat analoge functionaliteit aan onze Raspberry Pi toevoegen met behulp van Analooq-naar-Digitaal (ADC)-poorten via de SPI-bus.



Een tekort aan (analoog) kookgerei ☹️

De uitbreidingsconnector van de Raspberry Pi heeft geen analoge interfaces, niet één. Dat is jammer, andere platforms zoals de Arduino en de BeagleBone Black hebben er zelfs meerdere.

Maar geen nood, we kunnen ook de seriële SPI- en/of I²C-interface gebruiken om een seriële ADC aan te sluiten. Dus laten we beginnen met de SPI-interface. We gaan een 4-kanaals MCP3004 ADC-chip met een resolutie van 10 bits aansluiten.

SPI-Interface

We hebben al eerder naar de Serial Peripheral Interface (SPI)-interface gekeken in Elektor.POST project nummer 9, maar we zullen nog

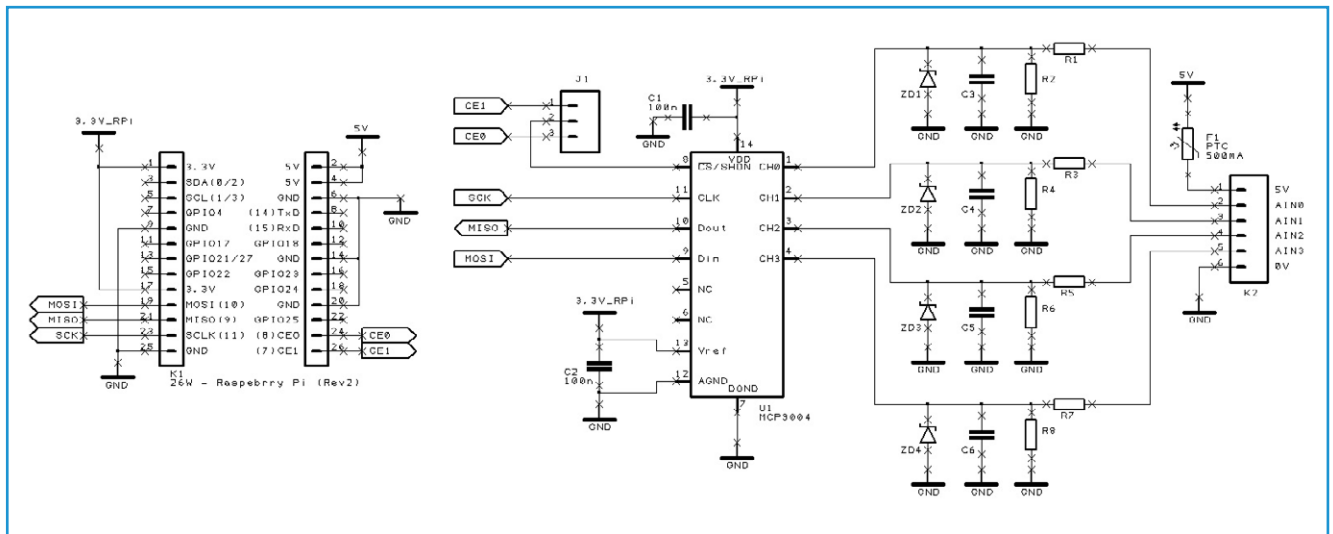
even snel het geheugen opfrissen.

In tabel 1 zien we een overzicht van de signalen op de uitbreidingsconnector. De SPI-interface is te vinden op pen 19 (MOSI), pen 21 (MISO) en pen 23 (SCK). Verder zijn er twee CE-lijnen voor de SPI-interface: pen 24 (CE0) en pen 26 (CE1).

Zie het tekstkader 'Installeren van de Python-SPI-Library' voor details over het installeren van de SPI-bibliotheek voor Python en het configureren van de Pi voor het gebruik van de SPI-interfaces.

ADC-Hardware

In dit eerste analoge project met onze Raspberry Pi gebruiken we een MCP3004 van Microchip [1]. De MCP3004 is een 4-kanaals, 10-bits ADC.



In figuur 1 zien we een eenvoudig schema ter illustratie van het gebruik van de MCP3004. De chip is verbonden met de SPI-interface van de RPi. Met een jumper kunnen we één van de twee CE-lijnen van de SPI kiezen (CE0 of CE1). Deze schakeling is gemakkelijk op te bouwen op een breadboard. De MCP3004 wordt gevoed met 3,3 volt, en we moeten opletten wat voor spanningen we er op aansluiten. Het ingangscircuit voor elk van de ADC-kanalen is hetzelfde. Als we kijken naar kanaal 1 van de ADC, zien we een spanningsdeler opgebouwd uit de weerstanden R1 en R2. Met de formule voor een spanningsdeler, kunnen we de juiste waarden uitrekenen:

$$V_{out} = \frac{R2}{R1 + R2} \times V_{in}$$

- Voor meten van 0 V tot 5 V, kiezen we R1 = 10 k en R2 = 10 k. Dat geeft een ADC-bereik van 0 V - 2,5 V.
- Voor meten van 0 V tot 10 V, kiezen we R1 = 10 k en R2 = 22 k. Dat geeft een ADC-bereik van 0 V tot 3,125 V.
- Voor meten van 0 V tot 3,3 V, kiezen we R1 = 1 k en R2 wordt niet geplaatst. Dat geeft een ADC-bereik van 0 V tot 3,3 V.

We kunnen de weerstanden voor de andere ADC kanalen #2, #3 en #4 op dezelfde manier kiezen.

Als we last hebben van storingen in het analoge signaal, kunnen we wat filtering toevoegen door condensatoren C3-C6 te plaatsen. Geschikte waarden zijn ongeveer 1 tot 10 nF.

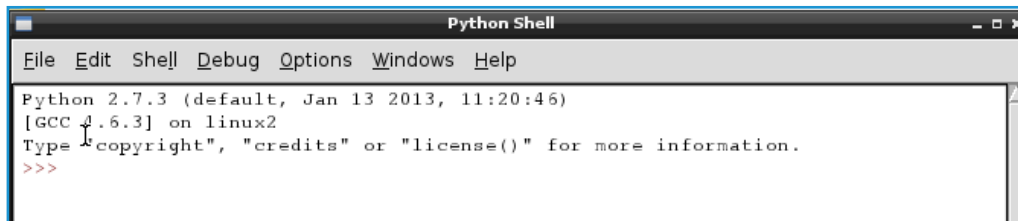
Figuur 1. Schema van de MCP3004 ADC-interface voor de Raspberry Pi.

Tabel 1. Pinbezetting van de uitbreidingsconnector

Naam	Funcctie	Alternatief	RPi.GPIO
P1-02	5,0V	-	-
P1-04	5,0V	-	-
P1-06	GND	-	-
P1-08	GPIO14	UART0_TXD	RPi.GPIO8
P1-10	GPIO15	UART0_RXD	RPi.GPIO10
P1-12	GPIO18	PWM0	RPi.GPIO12
P1-14	GND	-	-
P1-16	GPIO23		RPi.GPIO16
P1-18	GPIO24		RPi.GPIO18
P1-20	GND	-	-
P1-22	GPIO25		RPi.GPIO22
P1-24	GPIO8	SPI0_CE0_N	RPi.GPIO24
P1-26	GPIO7	SPI0_CE1_N	RPi.GPIO26

Naam	Board Revision 1		Board Revision 2	
	Funcctie	Alternatief	Funcctie	Alternatief
P1-01	3,3V	-	3,3V	-
P1-03	GPIO0	I2C0_SDA	GPIO2	I2C1_SDA
P1-05	GPIO1	I2C0_SCL	GPIO3	I2C1_SCL
P1-07	GPIO4	GPCLK0	GPIO4	GPCLK0
P1-09	GND	-	GND	-
P1-11	GPIO17	RTS0	GPIO17	RTS0
P1-13	GPIO21		GPIO27	
P1-15	GPIO22		GPIO22	
P1-17	3,3V	-	3,3V	-
P1-19	GPIO10	SPI0_MOSI	GPIO10	SPI0_MOSI
P1-21	GPIO9	SPI0_MISO	GPIO9	SPI0_MISO
P1-23	GPIO11	SPI0_SCLK	GPIO11	SPI0_SCLK
P1-25	GND	-	GND	-

Opmerking: I2C0_SDA, I2C0_SCL (GPIO0 & GPIO1), I2C1_SDA en I2C1_SCL (GPIO2 & GPIO3) hebben pullup-weerstanden van 1k8 naar 3,3 V.



Figuur 2.
IDLE Python-Shell.

Als we onze schakeling willen voorzien van een bescherming tegen overspanning, kunnen we ook nog de zenerdiodes ZD1 t/m ZD4 plaatsen. Deze hebben een zenerspanning van 3,3 V.

Voorbeeldprogramma: `mcp3004.py`

Als de schakeling is opgebouwd en spidev is geïnstalleerd (zie tekstvenster 'Installeren

van de SPI-Library voor Python'), kunnen we een klein testprogramma schrijven om een spanning op ADC-kanaal 1 te meten en weer te geven.

Dubbelklik het pictogram IDLE op het bureaublad van de Pi om de Python Shell en IDE te starten (zie figuur 2).

Kies nu de optie File in het menu en maak een nieuw programma. Dit start de IDE-editor.

Installeren van de SPI-library voor Python

We hebben de SPI-bibliotheek voor Python al geïnstalleerd in Elektor.POST project nummer 9, maar voor alle zekerheid geven we hier de noodzakelijke handelingen opnieuw. Start een LXTerminal-sessie (zie **figuur 4**) en geef de volgende commando's:

```
sudo apt-get install git-core
cd ~
git clone git://github.com/doceme/py-spidev
cd py-spidev/
sudo python setup.py install
```

Of gebruik 'Pip', de Python Package Installer

```
sudo apt-get install git-core python-dev
sudo apt-get install python-pip
sudo pip install spidev
```

Standaard is de hardware-SPI uitgeschakeld, dus we moeten dat veranderen door de blacklist-file aan te passen:

```
sudo nano /etc/modprobe.d/raspi-blacklist.conf
```

Zoek de regel met **blacklist spi-bcm2708** en plaats een # (hekje) aan het begin van de regel. Daarmee verandert de regel in een commentaarregel. Sla daarna het bestand op. Na het opslaan, moeten we het systeem opnieuw booten met:

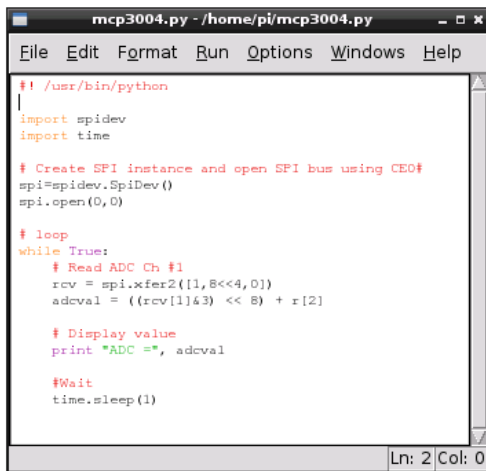
```
sudo reboot
```

Start na de reboot opnieuw een LXTerminal-sessie en type...

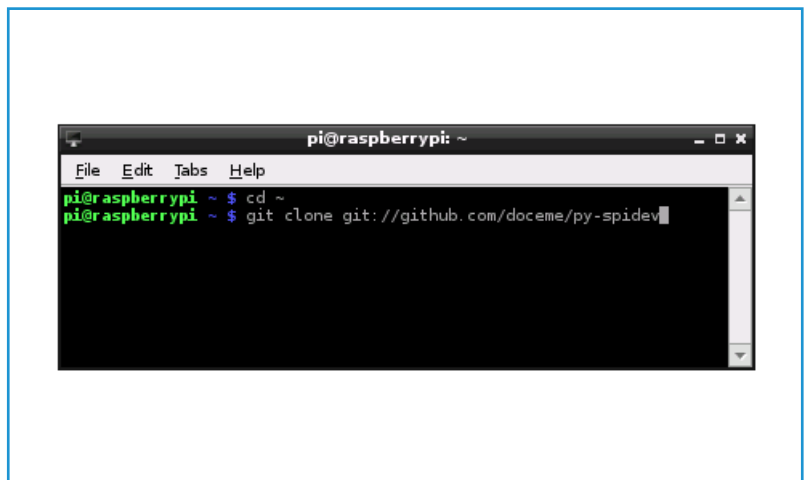
```
ls /dev/spi*
```

...om te controleren of we nu twee SPI apparaten hebben (één voor elk SPI Chip Select-sigitaal). Dat moet er als volgt uit zien:

```
/dev/spidev0.0
/dev/spidev0.1
```



Figuur 3. IDLE-Editor.



Figuur 4. LXTerminal.

Voer nu met de IDLE-editor (zie figuur 3) het programma van listing 1 in.

Sla het programma na het intypen op en schakel over naar een LXTerminal. Geef dan het volgende commando om het programma uitvoerbaar te maken:

```
chmod +x mcp3004.py
```

Nu kan het programma worden uitgevoerd met het commando:

```
sudo ./mcp3004.py (130260)
```

Weblinks

- [1] <http://goo.gl/eMSOQ>
- [2] <https://github.com/doceme/py-spidev>

```
Listing 1
#!/usr/bin/python

import spidev
import time

# Create SPI instance and open SPI bus using CE0
spi = spidev.SpiDev()
spi.open(0,0)

# Loop
while True:

    # Read ADC Ch #1
    rcv = spi.xfer2 ([1,8<<4, 0])
    adcval = ((rcv [1]&3) << 8) + rcv[2]

    # Display value
    Print "ADC = ", adcval

    # Wait
    time.sleep(1)
```

Korte samenvatting van de commando's van spidev:	
spi.open (0,0)	Opent SPI-bus 0 met CE0.
spi.open (0,1)	Opent SPI-bus 0 met CE1.
spi.close ()	Verbreekt de verbinding met de interface.
spi.writebytes ([array van bytes])	Schrijft een array van bytes naar het SPI-device.
spi.readbytes (len)	Leest len bytes van het SPI-device.
spi.xfer2 ([array van bytes])	Stuurt een array van bytes, waarbij CEx continu actief gehouden wordt.
spi.xfer ([array van bytes])	Stuurt een array van bytes, waarbij CEx telkens tussen twee bytes gedeactiveerd wordt.