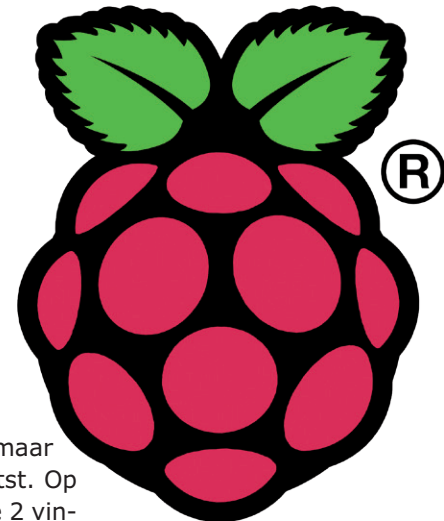


Raspberry Pi recepten

(Deel 5)

I²C: Inter-IC-Cake

Tony Dixon
(Verenigd Koninkrijk)



In de vorige twee delen hebben we gekeken naar de UART- en SPI-interfaces op de uitbreidingsconnector van de Raspberry Pi. We beginnen al behoorlijk ervaren *elektro-bakkers* te worden. Hoog tijd om te gaan kijken naar de laatste seriële interface van de Raspberry Pi: de I²C-bus.

I²C-interface

De Inter-IC- of I²C-bus is de laatste van de drie seriële interfaces op de uitbreidingsconnector van de Raspberry Pi. De andere twee zijn de UART-interface (zie deel 3) en de SPI-interface (zie deel 4).

In **tabel 1** zien we de signaallijnen van de uitbreidingsconnector. De I²C-interface is te vinden op pen 3 (SDA) en pen 5 (SCL).

Net als SPI is de I²C-interface ontworpen om met zo min mogelijk signalen te interfacen met andere apparaten. I²C maakt slechts gebruik van twee bidirectionele open-drain-lijnen: Serial Data Line (SDA) en Serial Clock (SCL). Meestal zijn deze voorzien van pullup-weerstanden naar 3,3 V. In de Raspberry Pi zijn dit twee weerstanden van 1k8.

I²C is niet zo snel als bijvoorbeeld de SPI-bus, maar haalt toch snelheden van 100 kbit/s in Standard Mode en 400 kbit/s in Fast Mode.

De Broadcom SoC-chip op de Raspberry Pi heeft twee I²C-interfaces. De originele versie van de Raspberry Pi had maar één I²C interface. Dat is de eerste I²C-interface (I2C_SDA0 en I2C_SCL0) op de uitbreidingsconnector.

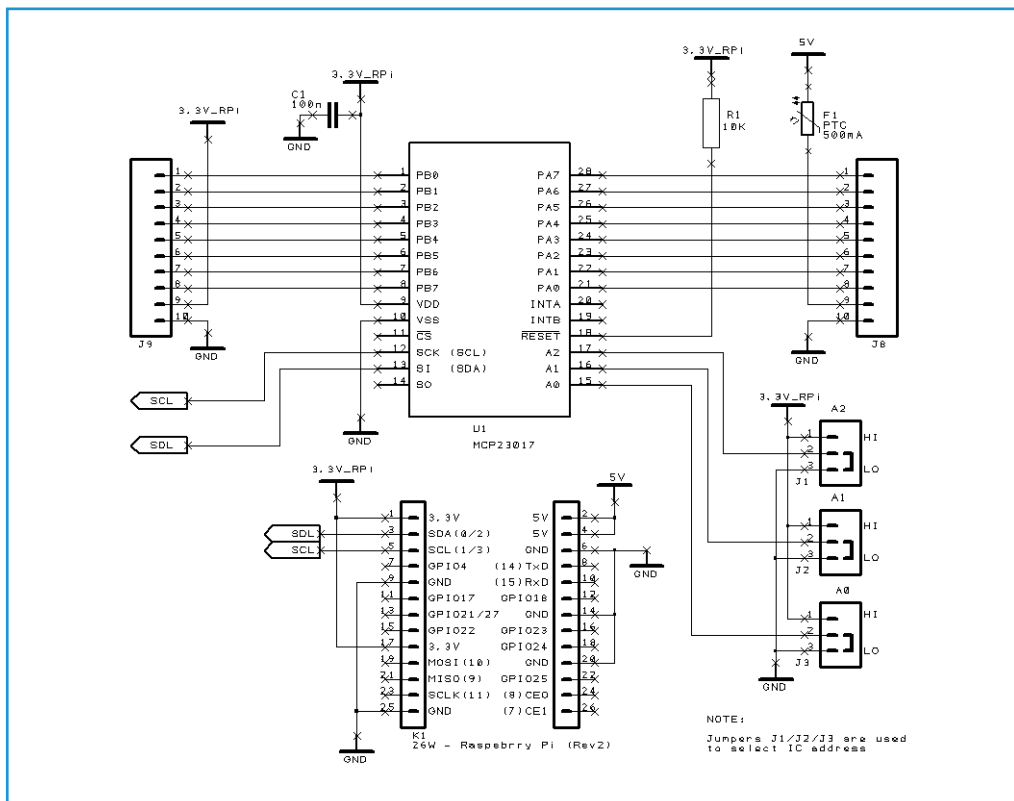
Bij de tweede versie van de Raspberry Pi werd een kleine extra uitbreidingsconnector toegevoegd. Daarmee kwam ruimte voor de

tweede I²C-interface beschikbaar, maar er werden ook wat dingen verplaatst. Op de uitbreidingsconnector van versie 2 vinden we nu de tweede I²C-interface (I2C_SDA1 en I2C_SCL1), terwijl de eerste I²C-

Tabel 1. Penbezetting 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	-



Figuur 1.
Schema voor de MCP23017
Raspberry Pi port-expander.

interface (I2C_SDA0 en I2C_SCL0) is verplaatst naar de nieuwe, kleinere connector. Dat is een klein maar belangrijk detail om in de gaten te houden als we de I²C op onze Raspberry Pi gaan gebruiken.

Port-expander hardware (opnieuw)

Voor ons I²C-project breiden we het aantal GPIO-lijnen van de Raspberry Pi opnieuw uit met behulp van nog een port-expander. Net als eerder gebruiken we een port-expander van Microchip, maar dit keer een 16-kanaals MCP23017 [1]. Dat is het I²C-broertje van de MCP23S17 die we bij het SPI-project hebben gebruikt.

In **figuur 1** zien we een vereenvoudigd schema van het gebruik van de MCP23017. De chip is verbonden met de I²C-interface van de RPi. In tegenstelling tot de SPI-versie van de schakeling zijn er geen chip-enable-signalen nodig, dus het schema is heel eenvoudig. Jumpers J1, J2 en J3 zijn voor optionele adreslijnen voor de port-expander, zodat meer dan één expander op de I²C-bus kan worden aangesloten.

Figuur 2 geeft onze hardware weer. We gebruiken weer een kleine extra kaart voor

de interface van onze MCP23017. Oplettende lezers hebben al gezien dat dit dezelfde kaart [2] is die we eerder gebruikt hebben. Dat klopt, want deze kaart is zowel geschikt voor de MCP23017 voor I²C als voor de MCP23S17 voor SPI. Met een jumper kiezen we voor SPI of voor I²C.

Installatie van de I²C-tools

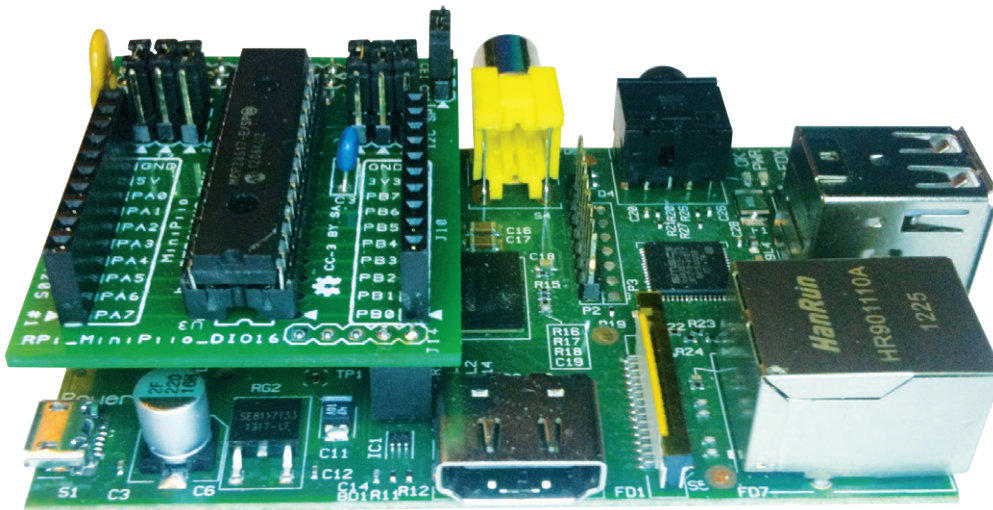
Voordat we de I²C-tools kunnen installeren, moeten we wat organisatorische werkzaamheden uitvoeren:

Raspbian moet weten dat we de hardware-I²C-interface willen gebruiken. Standaard is de hardware-I²C 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 i2c-bcm2708** en plaats een # (hekje) aan het begin van de regel. Daarmee verandert de regel in een commentaarregel. Sla daarna het bestand op.

Daarna moeten we het module-bestand aanpassen met:



Figuur 2. Pi met een MCP23017-opsteekprint.

`sudo nano /etc/modules`

Voeg de tekst ***i2c-dev*** toe op een nieuwe regel en sla het bestand op.

Installeer dan het pakket `i2c-tools` met het commando:

`sudo apt-get update`
`sudo apt-get install i2c-tools`

Als dit is geïnstalleerd, moeten we een nieuwe gebruiker toevoegen aan de `i2c`-groep. Type:

`sudo adduser pi i2c`

Doe daarna een snelle reboot met:

`sudo reboot`

Na de reboot kunnen we controleren of de I²C-interfaces beschikbaar zijn. Start een nieuwe LXTerminal-sessie en type...

`ls /dev/i2c*`

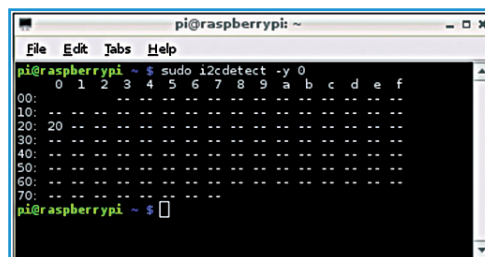
...om te controleren of er twee I²C-apparaten worden weergegeven (één voor elke I²C-interface). Als het goed is, zien we:

`/dev/i2c-0`
`/dev/i2c-1`

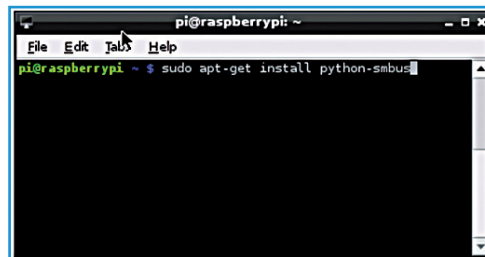
We kunnen ze ook testen. Op een Rev 1 Pi gaat dat met:

`sudo i2cdetect -y 0`

en op een Rev 2 Pi met:



Figuur 3. Uitvoer van "i2cdetect".



Figuur 4. LXTerminal.

`sudo i2cdetect -y 1`

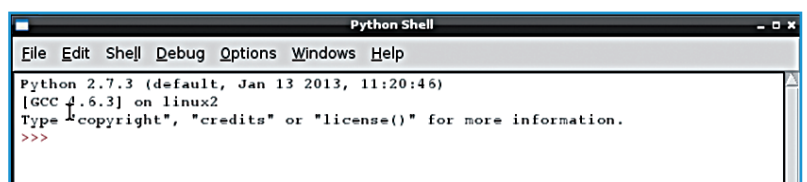
Het resultaat moet er ongeveer uit zien als in **figuur 3**.

Installeren van de I²C-library voor Python

Voor de voorbeelden in dit project gebruiken we Python 2. Zoals we al weten uit de vorige delen, is Python standaard al geïnstalleerd in de Raspbian-distributie.

Maar er is geen voorziening voor de I²C-interface. We kunnen dat verhelpen door

Figuur 5. IDLE Python-shell.



de I²C Python wrapper/library te installeren. Start dus weer een LXTerminal-sessie, zoals in **figuur 4** en geef de volgende commando's:

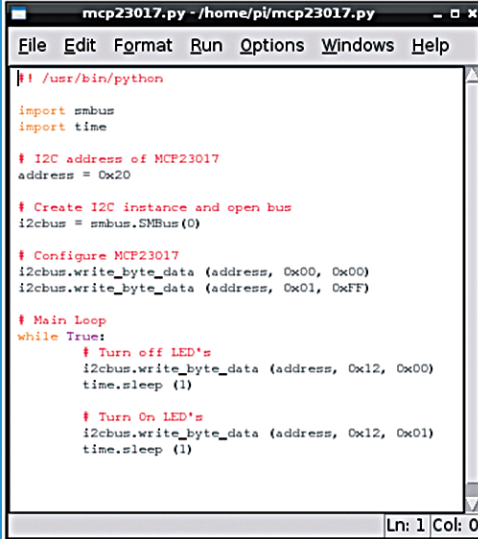
```
sudo apt-get install python-smbus
```

Als dit is geïnstalleerd, zijn we klaar om de I²C met Python te gebruiken.

Voorbeeldprogramma: mcp23017.py

Nu smbus is geïnstalleerd kunnen we een klein testprogramma schrijven om LED's aan te sturen via de Port-Expander GPIO.

Dubbelklik het pictogram IDLE op het bureaublad van de Pi om de Python Shell



```
mcp23017.py - /home/pi/mcp23017.py
File Edit Format Run Options Windows Help
#!/usr/bin/python
import smbus
import time

# I2C address of MCP23017
address = 0x20

# Create I2C instance and open bus
i2cbus = smbus.SMBus(0)

# Configure MCP23017
i2cbus.write_byte_data (address, 0x00, 0x00)
i2cbus.write_byte_data (address, 0x01, 0xFF)

# Main Loop
while True:
    # Turn off LED's
    i2cbus.write_byte_data (address, 0x12, 0x00)
    time.sleep (1)

    # Turn On LED's
    i2cbus.write_byte_data (address, 0x12, 0x01)
    time.sleep (1)
Ln: 1 Col: 0
```

Figuur 6.
IDLE-editor

Listing

```
#!/usr/bin/python

import smbus
import time

# I2C address of MCP23017
address = 0x20

# Create I2C instance and open bus
i2cbus = smbus.SMBus(0)

# Configure MCP23017
i2cbus.write_byte_data(address,0x00,0x00) # Set Bank A to outputs
i2cbus.write_byte_data(address,0x01,0xFF) # Set Bank B to inputs

# Main loop
while True:
    # Turn off LEDs
    i2cbus.write_byte_data (address,0x12,0x00)
    time.sleep(1)

    # Turn on PortA.0
    i2cbus.write_byte_data (address,0x12,0x01)
    time.sleep(1)
```

Opmerking: Verander voor Rev 2 Pi-kaarten de regel:

```
i2cbus = smbus.SMBus(0)      naar      i2cbus = smbus.SMBus(1)
```

en IDE te starten (zie **figuur 5**).

(130178)

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

Weblinks

[1] ww1.microchip.com/downloads/en/devicedoc/21952b.pdf

[2] www.dtronixs.com

Voer nu met de IDLE-editor het programma in de **listing** in (zie **figuur 6**).

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

```
sudo chmod +x mcp23017.py
```

Nu kan het programma worden uitgevoerd met het commando:

```
sudo ./mcp23017.py
```

Tabel 2 geeft een kort overzicht van de besturingsregisters van de MCP23x17.

Tabel 2. Registeradressen van de MCP23x17

Adres IOCON.BANK = 1	Adres IOCON.BANK = 0	Register	Beschrijving
0x00 / 0 dec	0x00 / 0 dec	IODIRA	I/O Direction Register voor Poort A
0x10 / 16 dec	0x01 / 1 dec	IODIRB	I/O Direction Register voor Poort B
0x01 / 1 dec	0x02 / 2 dec	IPOLA	Input Polarity Poortregister voor Poort A
0x11 / 17 dec	0x03 / 3 dec	IPOLB	Input Polarity Poortregister voor Poort B
0x02 / 2 dec	0x04 / 4 dec	GPINTENA	Interrupt-n-Change Control Register Poort A
0x12 / 18 dec	0x05 / 5 dec	GPINTENB	Interrupt-n-Change Control Register Poort B
0x03 / 3 dec	0x06 / 6 dec	DEFVALA	Default Compare Register voor GPINTENA
0x13 / 19 dec	0x07 / 7 dec	DEFVALB	Default Compare Register voor GPINTENB
0x04 / 4 dec	0x08 / 8 dec	INTCONA	Interrupt Control Register voor Poort A
0x14 / 20 dec	0x09 / 9 dec	INTCONB	Interrupt Control Register voor Poort B
0x05 / 5 dec	0x0A / 10 dec	IOCON	I/O-Expander Configuration Register
0x15 / 21 dec	0x0B / 11 dec	IOCON	I/O-Expander Configuration Register
0x06 / 6 dec	0x0C / 12 dec	GPPUA	Pull-Up Resistor Configuration Register Poort A
0x16 / 22 dec	0x0D / 13 dec	GPPUB	Pull-Up Resistor Configuration Register Poort B
0x07 / 7 dec	0x0E / 14 dec	INTFA	Interrupt Flag Register voor Poort A
0x17 / 23 dec	0x0F / 15 dec	INTFB	Interrupt Flag Register voor Poort B
0x08 / 8 dec	0x10 / 16 dec	INTCAPA	Interrupt Capture Register voor Poort A
0x18 / 24 dec	0x11 / 17 dec	INTCAPB	Interrupt Capture Register voor Poort B
0x09 / 9 dec	0x12 / 18 dec	GPIOA	Poortregister voor Poort A
0x19 / 25 dec	0x13 / 19 dec	GPIOB	Poortregister voor Poort B
0x0A / 10 dec	0x14 / 20 dec	OLATA	Output Latch Register voor Poort A
0x1A / 26 dec	0x15 / 21 dec	OLATB	Output Latch Register voor Poort B