Raspberry-Pi-voltmeter

Met kleurendisplay

Hermann Nieder
(Duitsland)

Er zijn maar weinig externe onderdelen nodig om met de Raspberry Pi gelijkspanningen tot 5 V te meten en de resultaten in duidelijke kleuren weer te geven op een monitor. We kunnen daarbij het hele scherm gebruiken, dus deze applicatie is goed te gebruiken voor demonstraties, bijvoorbeeld in een klaslokaal.

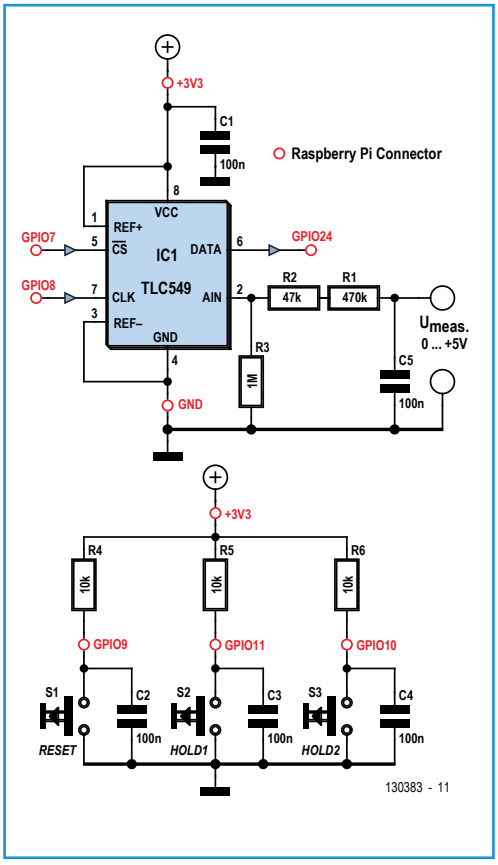
De auteur raakte onder andere geïnspireerd door drie artikelen in Elektor over de overstap van Basic naar Python [1]; dus de software voor de Raspberry Pi (Rev. 2) moest vanzelfsprekend in Python geschreven worden. Er zijn uiteindelijk twee versies van het programma ontstaan. In beide programma's communiceert de Raspberry Pi via enkele GPIO-pennen met een A/D-converter-chip en geeft de meetwaarden weer op het aangesloten beeldscherm.

Uitbreidingsschakeling

De uitbreidingsschakeling voor de Raspberry Pi (**figuur 1**) bestaat uit een A/D-converter met een resolutie van 8 bits (TLC549 [2] van Texas Instruments). Deze converter heeft een seriële interface en is inmiddels meer dan 30 jaar (!) oud, gemakkelijk verkrijgbaar (natuurlijk als DIP) en al even gemakkelijk te gebruiken. Er is in deze schakeling een spanningsdeler (R1...R3) toegevoegd aan de analoge ingang. In de testfase heeft de auteur daar een 10k-potentiometer aangesloten en een digitale multimeter gebruikt als apparaat om de spanning mee te vergelijken.

De weergegeven waarden kunnen tijdens het meten worden opgeslagen met de druktoetsen 'HOLD1' en 'HOLD2'. De opgeslagen waarden worden voortdurend weergegeven door markeringen boven de schaal. De opgeslagen waarden kunnen weer gewist worden met de druktoets 'RES', zodat we twee nieuwe waarden kunnen opslaan en weergeven.

De uitbreidingsschakeling kan gemakkelijk op een voor de Raspberry Pi verkrijgbaar breadboard worden opgebouwd. Dit wordt op de uitbreidingsconnector van de RPi-print gestoken. De voedingsspanning van 3,3 V wordt verzorgd door de RPi. Het meetbereik kan worden vergroot door extra weerstanden in serie te schakelen voor R1. Dan moeten de beide Python-programma's wel op enkele plaatsen



Figuur 1. Uitbreidingsschakeling voor de Raspberry Pi met een 'antieke' A/D-converter.

worden aangepast. Dankzij het commentaar in de listings zal dat weinig problemen opleveren.

Versies van het programma

In **figuur 2** zien we een screenshot van de eerste versie van het programma. Er worden drie meetwaarden weergegeven. De huidige waarde staat in het midden en wordt groter weergegeven. Verder kunnen we met de twee drukknoppen twee meetwaarden als digitale waarde opslaan. Naast de digitale weergave is er een quasi-analoge weergave in de vorm van een *bargraph* (staafdiagram). De door het indrukken van een toets opgeslagen waarden worden als markeringsstrepen boven de *bargraph* weergegeven.

Als de vast te leggen meetwaarden niet te dicht bij elkaar liggen, kunnen de markeringen met een kleine verandering in de eerste versie van het programma worden weergegeven als pijlpunten zoals in **figuur 3**.

In de tweede versie van het programma (**figuur 4**) wordt een voltmeter met wijzer gesimuleerd. De meetwaarde wordt ook weergegeven in cijfers. Ook bij dit programma kunnen we twee meetwaarden opslaan door op 'HOLD1' of 'HOLD2' op de uitbreidingsprint met de TLC549 te drukken. Deze meetwaarden worden ook weergegeven als markeringsstrepen boven de schaalverdeling. Net als bij het eerste programma kunnen we de beide opgeslagen meetwaarden en de markeringen ook weer wissen. in **figuur 5** is een aangepaste versie van het tweede programma te zien: een voltmeter met een kader om het 'meetwerk'.

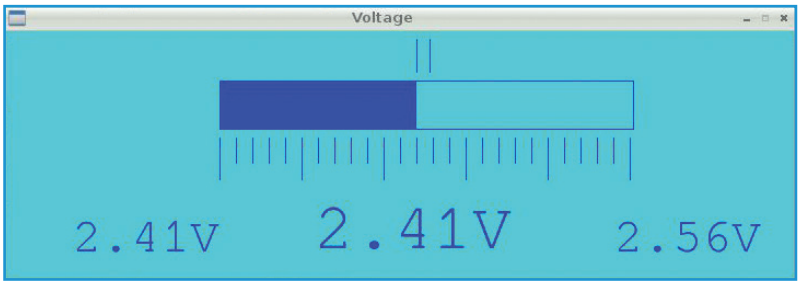
In alle versies van het programma kunnen we bij het opstarten kiezen uit een kleurenpalet voor de voor- en achtergrond en een factor voor de grootte van de weergave. We kunnen die weergaven ook een naam geven.

Uitproberen

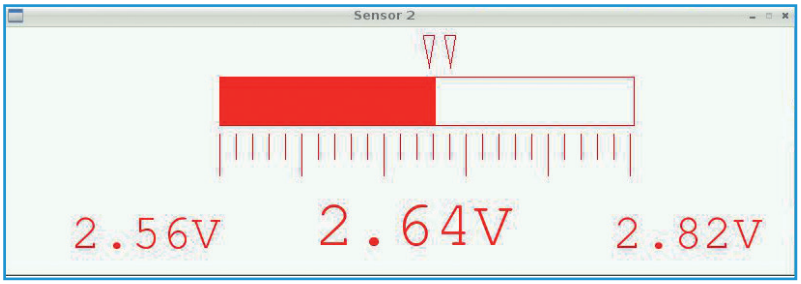
De Python-programma's van de auteur zijn te downloaden van de Elektor-website [3]. Het is het beste om dat te doen met een Raspberry Pi met internetverbinding.

Maak een subdirectory in /home/pi, sla de zip-bestanden daar op en pak ze uit.

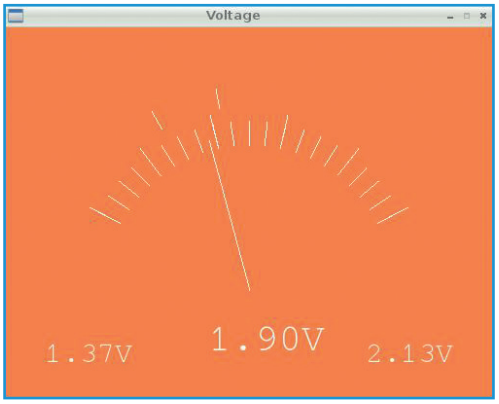
Installeer eerst de Pygame-bibliotheek. Die wordt in alle programmavoorbeelden van de auteur gebruikt. In de Debian-versie van het besturingssysteem gaat dat met het volgende commando:



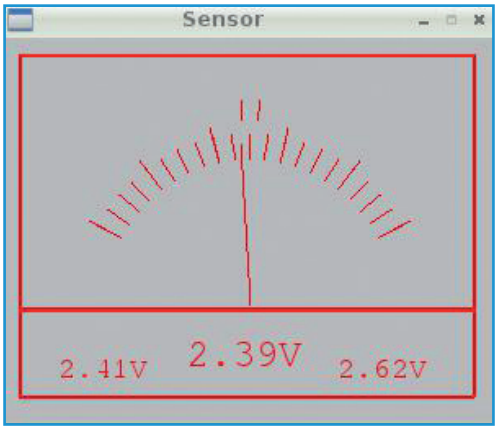
Figuur 2. Digitale voltmeter met quasi-analoge weergave.



Figuur 3. De variant met andere kleuren en twee kleine pijlen voor het weergeven van de opgeslagen waarden.



Figuur 4.
Voltmeter met gesimuleerd wijzerinstrument en digitale weergave.



Figuur 5.
De 'luxe-uitvoering' met een kader om het meetwerk.

```
sudo apt-get install python-game
```

Nu kunnen we de programma's van de auteur uitproberen.
Ga in bestandsbeheer naar de directory waar de Python-bestanden staan en start het programma met:

```
sudo python ADW_PTR_E.py
```

(Dit is de Engelse versie; er is ook een Duitstalige versie ADW_PTR_D.py)

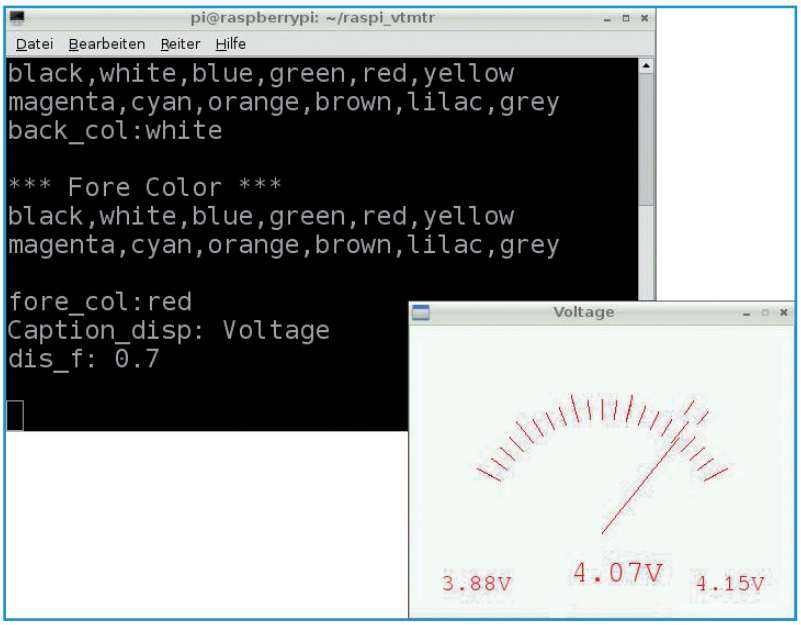
Voer in de terminal één van de beschikbare achtergrondkleuren in en bevestig deze met ENTER. Geef dan een voorgrondkleur, een naam voor de weergave en een factor voor de grootte van de presentatie op het beeldscherm. Daarna opent meteen het weergavevenster.
Blijkt de weergave bij de gekozen factor te groot of te klein, dan kan het programma in de terminal met Ctrl+C worden gestopt en opnieuw gestart om op de juiste plaats een andere factor voor de grootte te kiezen. De bediening van de andere versies van het programma gaat op ongeveer dezelfde manier.

In het voorbeeld in **figuur 6** is 'white' gekozen als achtergrondkleur en 'red' als voorgrondkleur voor de wijzer, de schaal en de getalweergave. Als factor voor de grootte is voor figuur 6 de waarde 0.7 ingevoerd.
In **figuur 7** zien we de invoer die nodig is om de Raspberry Pi een meetspanning aan de ingang van de TLC549 blauw te laten weergeven op een gele achtergrond.

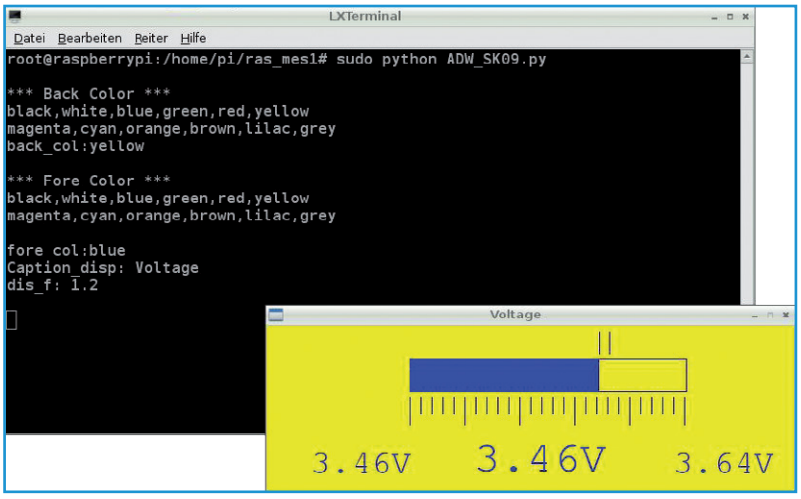
Listing in Python

De A/D-converter TLC549 wordt aangestuurd via de GPIO-pennen 7, 8 en 24 van de Raspberry Pi. GPIO7 is in de Python-software gedefinieerd als AD_CS en GPIO8 als AD_Clk om de code begrijpelijk te maken. In de datasheet van de A/D-converter [3] is te vinden hoe de pennen moeten worden aangestuurd om een meetwaarde in te lezen. De acht bits van het resultaat verschijnen achter elkaar op de uitgang DATA voor verdere verwerking. De uitgang is verbonden met GPIO24, die in de software is gedefinieerd als AD_Dat.

In beide versies van het programma wordt de TLC549 aangestuurd met de regels in het programma weergegeven in **Listing 1**.



Figuur 6. Simulatie van een meetwerk met wijzer en de daarvoor benodigde invoer in de LXTerminal van de Raspberry Pi.



Figuur 7. De invoer in de LXTerminal voor de afgebeelde simulatie.

In de eerste versie van het programma moet voor de weergave van een actuele meetwaarde als *bargraph* of als numerieke waarde de vorige weergegeven waarde eerst gewist worden. Dat gebeurt door om te beginnen een 'gevulde' rechthoek met de achtergrondkleur te tekenen. Pas daarna wordt de *bargraph* of de numerieke weergave van de actuele waarde getekend.

De tweede versie van het programma werkt op een vergelijkbare manier. Maar in dit geval

is het een stuk lastiger dan bij de eerste versie. De begin- en eindpunten van de schaalverdelingsstrepen worden berekend met sinus- en cosinusfuncties. De uitslag van de wijzer van de gesimuleerde voltmeter komt altijd overeen met de huidige meetwaarde, dus worden ook daarvoor berekeningen met sinus en cosinus uitgevoerd. De wijzer steekt, net als bij een echt meetinstrument, uit tot in de schaalverdeling. En ook voor het bepalen van de positie van de beide markeringen die de opgeslagen waarden weergeven wordt zo'n berekening gedaan.

Voor het afbeelden van de wijzerpositie wordt, in tegenstelling tot het eerste programma, altijd de hele meterweergave gewist. Verdere details zijn te vinden in het commentaar in de programma-listings.

(130383)

Weblinks

[1] "Van Basic naar Python", Elektor mei, juni en juli/augustus 2013, www.elektor-magazine.nl

[2] www.ti.com/lit/ds/symlink/tlc549.pdf

[3] www.elektor-magazine.nl/130383

[4] http://lxde.org

Listing 1.

```
def ADin():
    GPIO.output(AD_Clk,GPIO.LOW)# set AD_Clk to 0
    AD_res=0
    for n in range(10):
        time.sleep(0.05)
        GPIO.output(AD_CS,GPIO.HIGH)# set AD_CS to 1
        MSB=128
        time.sleep(0.001)
        GPIO.output(AD_CS,GPIO.LOW)# set AD_CS to 0
        time.sleep(0.0005)
        AD_value=0
        for z in range(8):
            if (GPIO.input(AD_Dat)):
                AD_value=AD_value+MSB
            GPIO.output(AD_Clk,GPIO.HIGH)# set AD_Clk to 1
            time.sleep(0.0005)
            GPIO.output(AD_Clk,GPIO.LOW)# set AD_Clk to 0
            MSB=MSB>>1
            time.sleep(0.0005)
            result=AD_value
        GPIO.output(AD_CS,GPIO.HIGH)# set AD_CS to 1
        AD_res=AD_res+AD_value
    AD_res=AD_res/10
    result=AD_res
    return result
```

Voor de digitale weergave en de grafische presentatie moeten de volgende omrekeningen uitgevoerd worden:

```
def conversion(value0):
    value=value0
    value=value*1960 # for 5V
    value=value/1000
    pic_value=value/2 # for bargraph display
    one=value/100
    rest_z=value % 100
    tenth=rest_z/10
    hundredth=rest_z%10
    return one, tenth, hundredth, pic_value,value
...
```