

Hands-on met de Raspberry Pi

26 september 2013

Inhoudsopgave

1	Inleiding	1
1.1	Benodigdheden	2
2	De Raspberry Pi opstarten	2
2.1	De Raspberry Pi connecteren	2
2.2	Initiële configuratie	3
2.3	Inloggen	4
2.4	WiFi configureren	4
3	Opdrachten	5
3.1	De Raspberry Pi Camera module	5
3.1.1	De camera module connecteren	5
3.1.2	Foto's nemen met de camera module	6
3.1.3	Video's maken met de camera module	6
3.2	De afstandssensor	7
3.2.1	Werking van de GPIO pinnen	7
3.2.2	Werking van de afstandssensor	8
3.2.3	Connecteren van de afstandssensor	9
3.2.4	Sensor waarden uitlezen met Python	9
3.2.5	Sensor waarden uitlezen met Java	11
4	Alles netjes achterlaten	15

1 Inleiding

In dit document wordt eerst beschreven hoe je de Raspberry Pi moet verbinden met de verschillende randapparatuur, hoe je de Raspberry Pi aanzet, en hoe je de Raspberry Pi de eerste keer configureert. Daarna wordt beschreven hoe je met de camera en de afstandssensor aan de slag kan.

Lees ook Sectie 4, zelfs als je niet klaar bent met de opdrachten. Daar staat immers beschreven hoe je de Raspberry Pi klaar maakt voor de volgende groep indien je in de eerste groep zit, en hoe je de software ontwerp labo's netjes achterlaat.

1.1 Benodigdheden

De volgende items zouden zeker in je kit moeten zitten:

1. Raspberry Pi Model A
2. HDMI naar DVI kabel
3. SD kaart
4. Muis & toetsenbord
5. USB Hub
6. USB WiFi dongle
7. Raspberry Pi camera
8. Micro USB adapter
9. HC SR04 Afstandssensor
10. 4 female to female jumper kabeltjes

2 De Raspberry Pi opstarten

Dit hoofdstuk beschrijft hoe je de Raspberry Pi moet aansluiten op alle randapparatuur. Daarna zal de Raspberry Pi opstarten. Indien het de eerste keer is dat je Raspbian opstart moeten er een aantal zaken geconfigureerd worden.

2.1 De Raspberry Pi connecteren

Voor je de Raspberry Pi voor het eerst opstart moet je het toestel connecteren met alle randapparatuur. De Raspberry Pi heeft geen aan/uit knop. De Pi zal opstarten zodra er stroom is.

SD kaart Het slot voor de SD kaart zit onderaan de Raspberry Pi. Schuif de SD kaart met de contactpunten naar boven in het slot.

Monitor Kies bij voorkeur een monitor met een vrije DVI ingang. Op die manier hoef je geen kabels uit te trekken. Kies de DVI ingang als *source* voor de monitor.

USB toestellen Sluit de USB Hub aan op de USB poort van de Raspberry Pi. Je gebruikt hiervoor de stekker waaruit twee draden vertrekken. De Wifi dongle, het toetsenbord en de muis sluit je aan op de USB Hub.

Micro USB adapter Connecteer de Micro USB adapter met de Raspberry Pi. De Raspberry Pi zal nu opstarten. Je ziet een gekleurd vierkant op het scherm verschijnen gevolgd door de Linux boot sequentie.

2.2 Initiële configuratie

Wanneer je Raspbian voor het eerst opstart kom je na de start-up in de `raspi-config` configuratie tool. Met deze tool kan je een aantal Raspberry Pi specifieke configuratie instellingen wijzigen.

Als Raspbian reeds is opgestart van op deze SD kaart zal `raspi-config` niet geladen worden en zal je in de plaats daarvan het login scherm zien. Normaal gezien zijn alle instellingen reeds in orde. Je kan inloggen zoals beschreven in Sectie 2.3. Moest je later toch nog instellingen willen wijzigen kan dat door `raspi-config` manueel te starten zoals hieronder beschreven.

Als je de `raspi-config` tool per ongeluk verlaat kan je deze terug starten door het volgende commando in de console te typen:

```
pi@raspberrypi ~ $ sudo raspi-config
```

Filesystem uitbreiden Wanneer de Raspbian software op de SD kaart wordt gezet neemt deze niet de ganse SD kaart in beslag. Om meer plaats te hebben in je filesystem kan je het uitbreiden.

1. Kies optie 1 `Expand Filesystem`.
2. Even later verschijnt normaal de bevestiging dat de root partitie groter is gemaakt.
3. Druk op `enter`.

Correcte toetsenbord layout kiezen De standaard instellingen voor de toetsenbord indeling komen niet overeen met het toetsenbord dat jullie gekregen hebben. Je moet het juiste toetsenbord selecteren door je instelling steeds verder te verfijnen.

1. Kies optie 4 `Internationalisation Options`.
2. Kies vervolgens I3 `Change Keyboard Layout`.
3. Selecteer `Generic 105-key (Intl) PC` in de lijst met modellen.
4. Selecteer `Other` in de lijst met keyboard layouts.
5. Selecteer `English (US)`.
6. Selecteer `English (US) (helemaal bovenaan)`.
7. Selecteer `The default for the keyboard layout`.
8. Selecteer `No compose key`.
9. Selecteer `<Yes>` om via `Control+Alt+Backspace` de X server te kunnen stoppen.

De nieuwe instellingen zijn onmiddellijk van kracht.

Camera activeren Om later de camera te kunnen gebruiken moet deze eerst geconfigureerd worden. Deze wijziging zal pas actief zijn na een reboot.

1. Kies optie 5 `Enable Camera`.

2. In de volgende dialoog kies je voor <Enable>.

SSH activeren Op termijn is het eenvoudiger om op de Raspberry Pi in te loggen via SSH. Op die manier kan je van op je eigen computer werken en kunnen meerdere gebruikers tegelijk de Raspberry Pi gebruiken.

1. Kies 8 Advance Options.
2. Kies A4 SSH.
3. Kies <Enable>.
4. Je krijgt de bevestiging dat de SSH server vanaf nu geactiveerd is.

Rebooten Nadat je de nodige configuratie instellingen hebt aangepast kan je best je Raspberry Pi rebooten. Selecteer <Finish> om `raspi-config` te beëindigen. Kies nadien voor <Yes> om te rebooten.

2.3 Inloggen

De Raspbian distributie komt met een standaard gebruiker: **pi** en paswoord **raspberry**. Na het inloggen kom je in de standaard Linux console. Je kan de grafische gebruikers interface starten met:

```
pi@raspberrypi ~ $ startx
```

2.4 WiFi configureren

Om je WiFi connectie te configureren kan je een grafische configuratie tool gebruiken (`wpa_gui`). De snelkoppeling op je bureaublad werkt echter niet. De tool heeft `root` rechten nodig.

Open *LXTerminal* via de snelkoppeling op je bureaublad. Je krijgt een console te zien. Je kan de configuratie tool als `root` starten met het volgende commando:

```
pi@raspberrypi ~ $ sudo wpa_gui
```

In het venster dat opent klik je op de *Scan* knop onderaan. Er opent zich een nieuw venster *Scan results*. Hier moet je weer op de knop *Scan* drukken om te scannen naar beschikbare WiFi netwerken.

Kies het *eduroam* netwerk in de lijst door er op te dubbelklikken. Dit opent een configuratie venster voor het *eduroam* network.

Op <https://admin.kuleuven.be/icts/english/wifi/eduroam-ubuntu> vind je hoe je het eduroam netwerk kan instellen op een Ubuntu Linux. Het merendeel van de instellingen is identiek op Raspbian. Tabel 1 geeft een overzicht van de configuratie nodig voor eduroam.

SSID	eduroam
Authentication	WPA2-Enterprise (EAP)
Encryption	CCMP
EAP method	PEAP
Identity	xxxxxxxxx@kuleuven.be
Password	your-password
CA certificaat	<i>blanco laten</i>

Tabel 1: Configuratie voor het eduroam netwerk in `wpa_gui`. Het *CA certificaat* veld mag blanco blijven.

3 Opdrachten

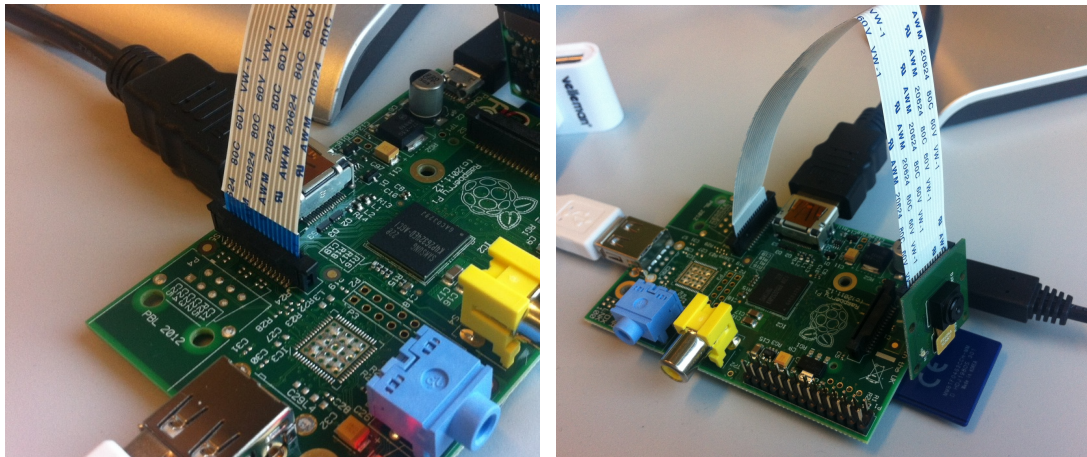
Dit hoofdstuk beschrijft een aantal opdrachten die je helpen vertrouwd te worden met de camera en de afstandssensor.

3.1 De Raspberry Pi Camera module

Je kan foto's en video's nemen met de Raspberry Pi camera module met behulp van respectievelijk het `raspistill` en `raspivid` commando.

3.1.1 De camera module connecteren

De kabel komende van de camera module moet verbonden worden met de connector vlak naast de HDMI aansluiting. De kant met het blauwe kleeflint moet *weg* wijzen van HDMI connector. De kant met de metalen connectors moet *naar* de HDMI connector wijzen (Figuur 1).



Figuur 1: De kant met het blauwe kleeflint moet *weg* wijzen van HDMI connector. De kant met de metalen connectors moet *naar* de HDMI connector wijzen.

Nadat je de camera hebt aangesloten moet je de Raspberry Pi herstarten. Dit kan je doen door in een console het volgende commando te geven:

```
pi@raspberrypi ~ $ sudo reboot
```

Dit roept het reboot commando aan als root gebruiker.

Wanneer de Raspberry Pi opnieuw is opgestart moet je opnieuw inloggen en de grafische omgeving starten met startx.

3.1.2 Foto's nemen met de camera module

Met het commandline programma raspistill kan je foto's nemen.

```
pi@raspberrypi ~ $ raspistill -o image.jpg
```

Na het geven van dit commando komt er op het scherm een preview van de foto. Na 5 seconden preview wordt er een foto genomen en bewaard in de image.jpg file in je huidige directory.

Je kan de foto nu bekijken met het volgende commando:

```
pi@raspberrypi ~ $ gpicview image.jpg
```

of door in een file browser naar het foto bestand te navigeren en het zo te openen.

Het raspistill commando heeft erg veel opties. Een overzicht van alle opties krijg je door het volgende commando uit te voeren.

```
pi@raspberrypi ~ $ raspistill --help
```

3.1.3 Video's maken met de camera module

Met het commandline programma raspivid kan je video's opnemen. Het volgende commando maakt een video van 10 seconden en schrijft deze video weg in het video.h264 bestand.

```
pi@raspberrypi ~ $ raspivid -o video.h264 -t 10000
```

Je kan de video bekijken met de omxplayer player:

```
pi@raspberrypi ~ $ omxplayer video.h264
```

Net als raspistill heeft raspivid veel configuratie opties. Je krijg een overzicht van al deze opties met het commando:

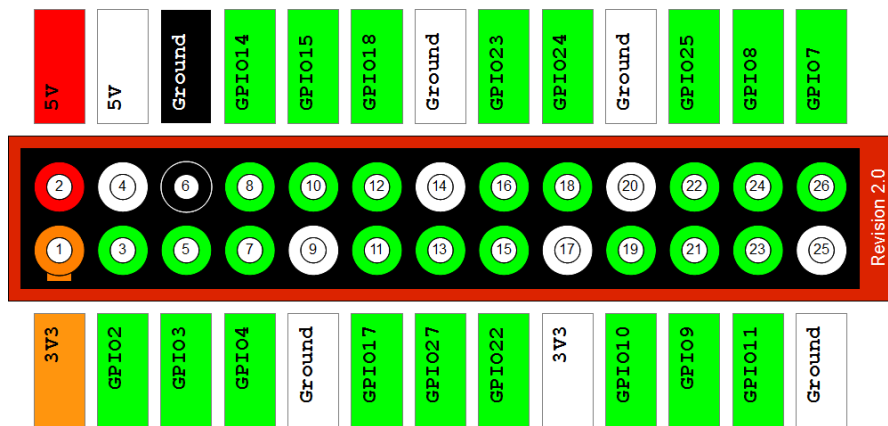
```
pi@raspberrypi ~ $ raspivid --help
```

3.2 De afstandssensor

In dit deel wordt beschreven hoe je waarden van de afstandssensor uitleest met behulp van Python of Java. Om te werken met de afstandssensor moet je gebruik maken van de GPIO pinnen.

3.2.1 Werking van de GPIO pinnen

De Raspberry Pi heeft in totaal 26 GPIO pinnen. Hiervan kunnen er 17 gebruikt worden voor algemene in- en uitvoer. Figuur 2 toont de verschillende pinnen.



Figuur 2: De verschillende GPIO pinnen. De pinnen worden op twee verschillende manieren benoemd: op basis van hun fysieke locatie (nummer in de cirkels) of op basis van hun connectie met de CPU (de GPIOXX labels in het groen).

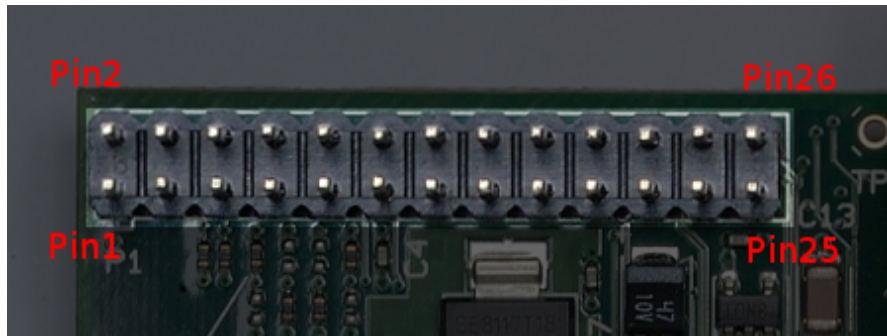
Pin 1, links onderaan Figuur 2, heeft onderaan een rechthoekig symbool. Pin 1 op de Raspberry Pi heeft hetzelfde symbool.

De pinnen kunnen op twee manieren benoemd worden. Op basis van de fysieke locatie van de pin en op basis van de connectie met de CPU. De fysieke locatie wordt als volgt bepaald: De pin met symbool, onderaan links is pin 1. De pin daarboven is pin 2. De pin rechts van pin 1 is pin 3. De pin boven pin 3 is pin 4. En zo verder. De fysieke pin nummers worden getoond in Figuur 3.

De alternatieve benaming voor de pinnen is gebaseerd op hun connectie met de CPU. Zo is de pin op plaats 3 verbonden met ingang 2 van de CPU, en de pin op plaats 5 is verbonden met ingang 3 van de CPU. Deze pinnen krijgen dus respectievelijk de labels GPIO2 en GPIO3.

Figuur 2 toont de mapping tussen beide benamingen. Enkel de pinnen met een groen label zijn verbonden met de CPU en hebben een GPIO label. Enkel deze pinnen kunnen gebruikt worden voor in en uitvoer.

De GPIO pinnen kunnen werken in twee verschillende modi. Ze werken als invoer of als uitvoer. Van een pin in invoer modus kan je in je software lezen of er spanning ten opzichte van de aarding op staat. Op een pin in uitvoer modus kan je een spanning ten opzichte van de aarding plaatsen.



Figuur 3: De fysieke pin nummering

De GPIO pinnen, de groene pinnen in Figuur 2, mogen niet gebruikt worden om stroom te leveren. Ze dienen enkel maar om een signaal te geven. Omwille van hun rechtstreekse verbinding met de CPU kan een te grote stroom doorheen deze pinnen leiden tot schade aan de CPU.

De overige pinnen, de 3V3, 5V en Ground pinnen mogen wel gebruikt worden om stroom aan sensoren te leveren. Ze kunnen echter enkel de door de Raspberry Pi ongebruikte stroom leveren. Als de Micro USB adapter 500 mA levert en de Raspberry Pi 300 mA verbruikt kunnen de pinnen samen maar 200 mA leveren. Voor sensoren zoals de afstandssensor is dit voldoende, voor het aansturen van een motor is dit wellicht niet voldoende.

Er bestaan voor verschillende programmeertalen libraries die het gebruik van de GPIO pinnen mogelijk maken. Voor Python is er WiringPi¹ en RPi.GPIO². Voor Java is er Pi4J³. Naast bibliotheken om de GPIO pinnen te gebruiken kan je ook het `sysfs` bestandssysteem gebruiken.

3.2.2 Werking van de afstandssensor

De HC SR04 afstandssensor heeft 4 pinnen, gelabeled *Vcc*, *Trig*, *Echo* en *Gnd*. De buitenste twee, *Vcc* en *Gnd*, dienen om de sensor van stroom te voorzien. De binnenste twee, *Trig* en *Echo*, dienen respectievelijk om de sensor te activeren en uit te lezen.

Wanneer via een GPIO pin in uitvoer modus een signaal gegeven wordt aan de *Trig* pin zal de sensor een korte ultrasone puls verzenden. Nadien zal de sensor een spanning plaatsen op de *Echo* pin. Deze spanning kan gedetecteerd worden met een GPIO pin in invoer modus. De duur van de spanning komt overeen met de tijd die het signaal nodig had om het dichtstbijzijnde object te bereiken, te weerkaatsen en terug te keren naar sensor⁴. Door de lengte van het signaal te meten (Δt) kan je dus de afstand tot het object berekenen (d) met de formule:

$$d = \frac{\Delta t * v_{snd}}{2} \quad (1)$$

De snelheid van het geluid, v_{snd} in de formule, is 340 m/s. Om de afstand in centimeters te

¹<https://projects.drogon.net/raspberry-pi/wiringpi/>

²<https://pypi.python.org/pypi/RPi.GPIO>

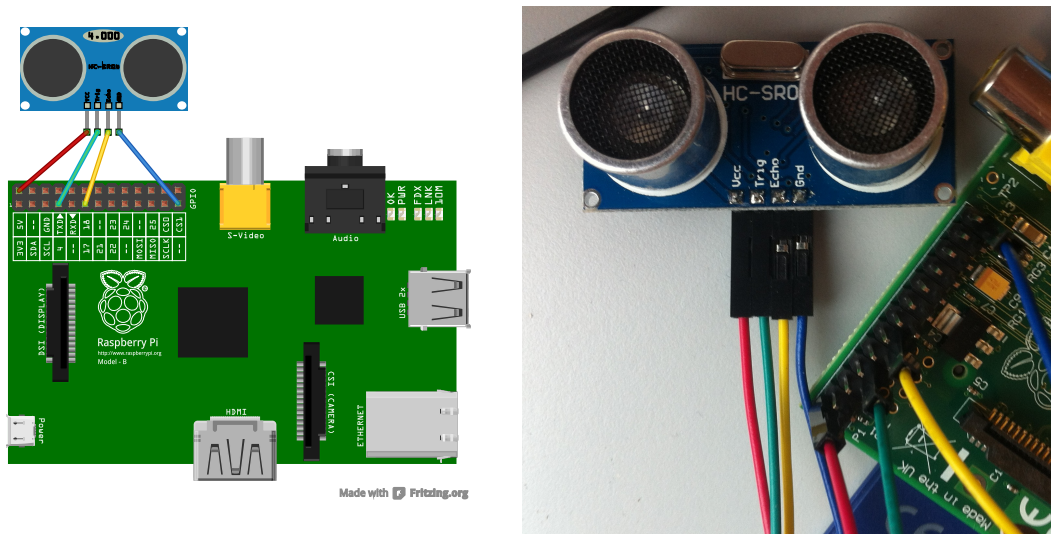
³<http://pi4j.com/>

⁴[url: http://elecxfreaks.com/store/download/HC-SR04.pdf](http://elecxfreaks.com/store/download/HC-SR04.pdf)

bereken op basis van een tijdsduur in microseconden (μs) volstaat het dus Δt te vermenigvuldigen met 0.017.

3.2.3 Connecteren van de afstandssensor

Connecteer de *Vcc* pin van de sensor met pin 2 (5V) van de Raspberry Pi. Connecteer de *Gnd* pin van de sensor met pin 25 (Ground) van de Raspberry Pi. Connecteer de *Trig* pin van de sensor met pin 7 (GPIO4) van de Raspberry Pi. En connecteer tot slot de *Echo* pin van de sensor met pin 11 (GPIO17). De uiteindelijke connecties moeten overeenkomen met Figuur 4.



Figuur 4: De verbinding tussen de afstandssensor en de Raspberry Pi

3.2.4 Sensor waarden uitlezen met Python

Listing 1 toont de code nodig om de afstandssensor uit te lezen op de Pi. Je kan deze code ook op het internet bekijken op <https://gist.github.com/rutgerclaes/6701244>. Om de code snel op de Pi te krijgen kan je volgend command gebruiken:

```
pi@raspberrypi ~ $ wget -O distance.py http://goo.gl/X9ot8j
```

```
1  #!/usr/bin/env python
2
3  # Import time module to keep track of time
4  import time
5
6  # Import RPi.GPIO module to access GPIO pins
7  import RPi.GPIO as GPIO
8
9  # Set the pins we use
10 # -----
11 # Here we use the BCM notation. This is the notation corresponding
```

```

12 # to the GPIO labels. 17 means GPIO17, 4 means GPIO4.
13 echo_gpio = 17
14 trig_gpio = 4
15
16 # Initiate the GPIO pins
17 # -----
18 # First we set the addressing mode to BCM, corresponding to how we
19 # have setup the echo_gpio and trig_gpio variables.
20 #
21 # Next we set the echo gpio in input mode and the trig_gpio in
22 # output mode. Finally, we set the output signal on trig_gpio to
23 # False, meaning no signal.
24 #
25 # For more information about the RPi.GPIO module check out:
26 # http://pypi.python.org/pypi/RPi.GPIO
27 GPIO.setmode( GPIO.BCM )
28 GPIO.setup( echo_gpio, GPIO.IN, pull_up_down=GPIO.PUD_DOWN )
29 GPIO.setup( trig_gpio, GPIO.OUT )
30 GPIO.output( trig_gpio, False )
31
32 # Setup some other variables
33 # -----
34 loop = True # Infinite loop
35 trig_duration = 0.0001 # Trigger duration
36 inttimeout = 2100 # Timeout on echo signal
37 v_snd = 340.29 # Speed of sound in m/s
38
39 # Wait for 2 seconds for the ultrasonics to settle
40 # -----
41 # Probably not needed...
42 time.sleep( 2 )
43
44 while loop:
45     # Trigger trig_gpio for trig_duration
46     GPIO.output( trig_gpio, True )
47     time.sleep( trig_duration )
48     GPIO.output( trig_gpio, False )
49
50     # Wait for the echo signal (or timeout)
51     countdown_high = inttimeout
52     while ( GPIO.input( echo_gpio ) == 0 and countdown_high > 0 ):
53         countdown_high -= 1
54
55     # If we've gotten a signal
56     if countdown_high > 0:
57         echo_start = time.time()
58
59         countdown_low = inttimeout
60         while( GPIO.input( echo_gpio ) == 1 and countdown_low > 0 ):
61             countdown_low -= 1
62         echo_end = time.time()
63
64         echo_duration = echo_end - echo_start
65
66     # Display the distance, unless there was a timeout on
67     # the echo signal
68     if countdown_high > 0 and countdown_low > 0:
69         # echo_duration is in seconds, so multiply by speed
70         # of sound. Divide by 2 because of roundtrip and
71         # multiply by 100 to get cm instead of m.
72         distance = echo_duration * v_snd * 100 / 2
73         print "Distance = " + str( distance ) + "cm"

```

```

74         else:
75             print "Timeout"
76
77     # Wait before retriggering
78     time.sleep( 1 )

```

Listing 1: distance.py

Je kan het `distance.py` programma aanpassen met een editor zoals `leafpad`. Nadien moet je het script uitvoerbaar maken en uitvoeren als `root`.

```

pi@raspberrypi ~ $ chmod +x distance.py
pi@raspberrypi ~ $ sudo ./distance.py

```

Van zodra het programma gestart is zie je de afstandswaarden voorbij komen. Je kan het programma afsluiten met `Ctrl+c`. Als je het programma voor de tweede keer opstart zal je een foutboodschap te zien krijgen: `RuntimeWarning: This channel is already in use . . .`. De GPIO pinnen zijn reeds geïntialiseerd en de `RPi.GPIO` geeft daarom een waarschuwing. Je mag deze foutboodschap negeren.

3.2.5 Sensor waarden uitlezen met Java

De Java JDK is standaard niet geïnstalleerd op Raspbian. Indien je de eerste gebruiker bent van deze Raspbian installatie zal je die zelf moeten installeren. Dit kan je doen met het volgende commando:

```

pi@raspberrypi ~ $ sudo apt-get update
pi@raspberrypi ~ $ sudo apt-get install openjdk-7-jdk
pi@raspberrypi ~ $ java -version

```

Het installeren van de nodige software zal een tijdje duren. Nadien zou het `java` commando moeten werken.

Ook vanuit Java kan je de GPIO pinnen gebruiken. De library die dit mogelijk maakt is `Pi4J`. Om de library te gebruiken moet je code downloaden met het volgende commando:

```

pi@raspberrypi ~ $ wget pi4j.googlecode.com/files/pi4j-0.0.5.zip
pi@raspberrypi ~ $ unzip pi4j-0.0.5.zip

```














Listing 2 toont voorbeeld code die gebruik maakt van `Pi4J` om de afstandssensor uit te lezen⁵. `PI4J` gebruikt *nog* een andere manier om de pinnen te adresseren. Figuur 5 toont de mapping in `Pi4J`. Zowel de naam als de nummer in Figuur 5 zijn `Pi4J` specifiek. Je kan de fysieke pin nummer gebruiken om de mapping te maken zoals in Tabel 2.

```

1  /**
2   * Class to monitor distance measured by an HC-SR04 distance sensor on a
3   * Raspberry Pi.
4   *
5   * The main method assumes the trig pin is connected to the pin # 7 and the echo
6   * pin is connected to pin # 11. Output of the program are comma separated lines

```

⁵Je kan deze code ook vinden op <http://goo.gl/dpOsbx>

Raspberry Pi P1 Header					
PIN #	NAME			NAME	PIN #
	3.3 VDC Power	1		5.0 VDC Power	2
8	SDA0 (I2C)	3		DNC	4
9	SCL0 (I2C)	5		0V (Ground)	6
7	GPIO 7	7		TxD	15
	DNC	9		RxD	16
0	GPIO 0	11		GPIO1	1
2	GPIO2	13		DNC	
3	GPIO3	15		GPIO4	4
	DNC	17		GPIO5	5
12	MOSI	19		DNC	
13	MISO	21		GPIO6	6
14	SCLK	23		CE0	10
	DNC	25		CE1	11
http://www.pi4j.com					

Figuur 5: De pin nummering voor PI4J. Onder Pin # staat de nummer die PI4J gebruikt. Onder Name de naam die PI4J gebruikt. Naast de pinnen staat ten slotte nog de fysieke pin nummer.

Fysieke pin nr.	RPi.GPIO label	Pi4J label
7	GPIO04	RaspiPin.GPIO07
11	GPIO17	RaspiPin.GPIO00

Tabel 2: Mapping tussen fysieke pin, RPi.GPIO en Pi4J labels.

```

7  * where the first value is the number of milliseconds since unix epoch, and the
8  * second value is the measured distance in centimeters.
9  */
10
11 import com.pi4j.io.gpio.GpioController;
12 import com.pi4j.io.gpio.GpioFactory;
13 import com.pi4j.io.gpio.GpioPinDigitalInput;
14 import com.pi4j.io.gpio.GpioPinDigitalOutput;
15 import com.pi4j.io.gpio.Pin;
16 import com.pi4j.io.gpio.RaspiPin;
17
18 /**
19  * DistanceMonitor class to monitor distance measured by sensor
20  *
21  * @author Rutger Claes <rutger.claes@cs.kuleuven.be>
22  */
23 public class DistanceMonitor {
24
25     private final static float SOUND_SPEED = 340.29f; // speed of sound in m/s
26

```

```

27 private final static int TRIG_DURATION_IN_MICROS = 10; // trigger duration of 10 micro s
28 private final static int WAIT_DURATION_IN_MILLIS = 60; // wait 60 milli s
29
30 private final static int TIMEOUT = 2100;
31
32 private final static GpioController gpio = GpioFactory.getInstance();
33
34 private final GpioPinDigitalInput echoPin;
35 private final GpioPinDigitalOutput trigPin;
36
37 private DistanceMonitor( Pin echoPin, Pin trigPin ) {
38     this.echoPin = gpio.provisionDigitalInputPin( echoPin );
39     this.trigPin = gpio.provisionDigitalOutputPin( trigPin );
40     this.trigPin.low();
41 }
42
43 /*
44  * This method returns the distance measured by the sensor in cm
45  *
46  * @throws TimeoutException if a timeout occurs
47  */
48 public float measureDistance() throws TimeoutException {
49     this.triggerSensor();
50     this.waitForSignal();
51     long duration = this.measureSignal();
52
53     return duration * SOUND_SPEED / ( 2 * 10000 );
54 }
55
56 /**
57  * Put a high on the trig pin for TRIG_DURATION_IN_MICROS
58  */
59 private void triggerSensor() {
60     try {
61         this.trigPin.high();
62         Thread.sleep( 0, TRIG_DURATION_IN_MICROS * 1000 );
63         this.trigPin.low();
64     } catch (InterruptedException ex) {
65         System.err.println( "Interrupt during trigger" );
66     }
67 }
68
69 /**
70  * Wait for a high on the echo pin
71  *
72  * @throws DistanceMonitor.TimeoutException if no high appears in time
73  */
74 private void waitForSignal() throws TimeoutException {
75     int countdown = TIMEOUT;
76
77     while( this.echoPin.isLow() && countdown > 0 ) {
78         countdown--;
79     }
80
81     if( countdown <= 0 ) {
82         throw new TimeoutException( "Timeout waiting for signal start" );
83     }
84 }
85
86 /**
87  * @return the duration of the signal in micro seconds
88  * @throws DistanceMonitor.TimeoutException if no low appears in time

```

```

89     */
90     private long measureSignal() throws TimeoutException {
91         int countdown = TIMEOUT;
92         long start = System.nanoTime();
93         while( this.echoPin.isHigh() && countdown > 0 ) {
94             countdown--;
95         }
96         long end = System.nanoTime();
97
98         if( countdown <= 0 ) {
99             throw new TimeoutException( "Timeout waiting for signal end" );
100        }
101
102        return (long)Math.ceil( ( end - start ) / 1000.0 ); // Return micro seconds
103    }
104
105    public static void main( String[] args ) {
106        Pin echoPin = RaspiPin.GPIO_00; // PI4J custom numbering (pin 11)
107        Pin trigPin = RaspiPin.GPIO_07; // PI4J custom numbering (pin 7)
108        DistanceMonitor monitor = new DistanceMonitor( echoPin, trigPin );
109
110        while( true ) {
111            try {
112                System.out.printf( "%1$d,%2$.3f%n", System.currentTimeMillis(), monitor.measureDistance() );
113            }
114            catch( TimeoutException e ) {
115                System.err.println( e );
116            }
117
118            try {
119                Thread.sleep( WAIT_DURATION_IN_MILLIS );
120            } catch (InterruptedException ex) {
121                System.err.println( "Interrupt during trigger" );
122            }
123        }
124    }
125
126    /**
127     * Exception thrown when timeout occurs
128     */
129    private static class TimeoutException extends Exception {
130
131        private final String reason;
132
133        public TimeoutException( String reason ) {
134            this.reason = reason;
135        }
136
137        @Override
138        public String toString() {
139            return this.reason;
140        }
141    }
142
143 }

```

Listing 2: DistanceMonitor.java

Je kan bovenstaande code compileren en uitvoeren met de volgende twee commandos. Het eerste zorgt voor de compilatie. Het tweede voert het programma uit en schrijft de waarden naar een `distance.csv` file.

```
pi@raspberrypi ~ $ javac -cp pi4j-0.0.5/lib/pi4j-core.jar DistanceMonitor.java
pi@raspberrypi ~ $ sudo java -cp pi4j-0.0.5/lib/pi4j-core.jar:. DistanceMonitor > distance.csv
```

Visualiseren van de sensor waarden. Om de data te visualiseren kan je R gebruiken. Je kan R installeren op de Raspberry Pi met het volgende commando. R is een erg uitgebreid pakket, de installatie kan even duren.

```
pi@raspberrypi ~ $ sudo apt-get install r-base
```

Eens R is geïnstalleerd kan je een sessie starten door in je console het volgende te typen:

```
pi@raspberrypi ~ $ R
```

In de R console geef je de volgende commando's. De eerste twee installeren nuttige R bibliotheken. Het downloaden en installeren kan lang duren. Bij het installeren van de bibliotheken kan het zijn dat R vraagt waar hij ze moet downloaden (CRAN Mirror). Hier kan je kiezen voor *Belgium*.

```
> install.packages( 'ggplot2' )
> install.packages( 'zoo' )
> distance.data = read.csv( 'distance.csv' )
> colnames( distance.data ) = c( 'time', 'distance' )
> distance.data$time = ( distance.data$time - min( distance.data$time ) ) / 1000
> distance.data$smooth = rollmedian( distance.data$distance, 9, na.pad = T )
> plot = ggplot( data = distance.data, aes( x = time ) )
> plot = plot + geom_point( aes( y = distance ) )
> plot = plot + geom_line( aes( y = smooth ) )
> print( plot )
```

4 Alles netjes achterlaten

Het eduroam paswoord dat je gebruikte om de WiFi te connecteren wordt in plaintext opgeslagen. Als je niet wil dat de studenten van de volgende groep het kunnen lezen verwijder je best de configuratie files.

```
pi@raspberrypi ~ $ sudo rm /etc/wpa_supplicant/wpa_supplicant.conf
```

Daarnaast kan je ook de code die je hebt geschreven of gedownload verwijderen. Zeker als je in de eerste groep studenten zit.

Ten slotte zet je de Raspberry Pi door de Micro USB connectie te verbreken. Verwijder ook alle randapparatuur van de Raspberry Pi.

Zet de invoer van de schermen terug op displayport en kijk na of ze een Ubuntu login scherm tonen.