

## ROBOBITS<sup>-75</sup>

### VAN DE BESTUURSTAFEL

Beste lezer,

Het is weer winter, volgens sommige mensen donkere dagen...  
Maar dat is toch niet erg?  
Lekker in de hobbykamer (als je die hebt natuurlijk!) knutselen aan de robots.

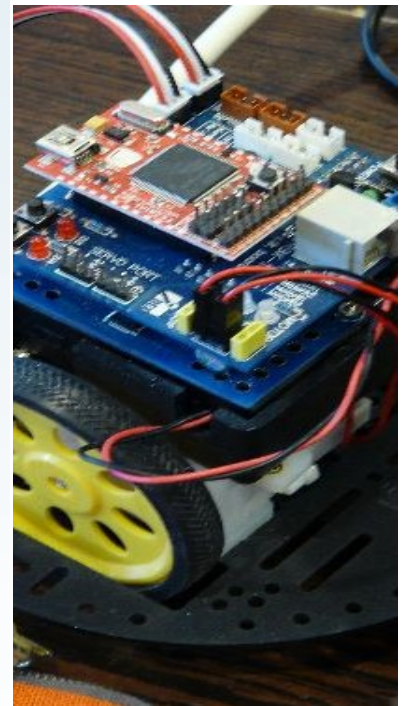
We hebben een maandje geleden onze 10e Roborama race gehad, en het was een groot succes. Veel deelnemers en belangstellenden.  
Op de website is een link te vinden in het hoofdmenu bovenaan, naar de film die is gemaakt die dag door ons HCC Videoteam. Hij is leuk geworden, dus ga snel eens een kijkje nemen als je dat nog niet gedaan hebt.

Ik ben benieuwd na de actie van Joep, om banen te laten printen en die aan de leden te verkopen. Hoe snel gaan we meerdere robots krijgen die goed en snel over de baan zullen gaan?  
Hopelijk hebben we binnenkort ook een tijdswaarneming die voor iedereen te zien is tijdens de wedstrijden.

Ik wil het hier even bij laten, want mijn robots roepen mij. En ik wil daar gehoor aan geven.

Namens het bestuur wens ik alle leden een gelukkig en gezond 2017. En ik hoop weer op een schitterend jaar waarin we onze robotdromen kunnen verwezenlijken.

Met vriendelijke groet,  
Bert Berrevoets.



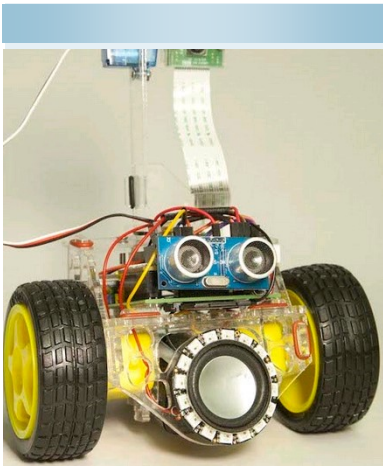
### IN DIT NUMMER

Van de bestuurstafel.....	1
PIBOT .....	2
Roborama 2016 .....	5
Let's e-mail en twitter met onze robots.....	6
Workshop Arduino en I2C.....	12

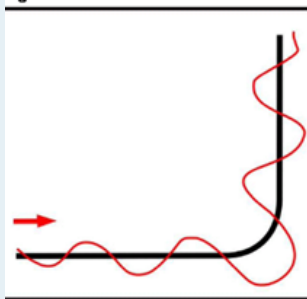
### Star wars BB8

?? [Wie bouwt hem na ??](#)





PiBot is an exciting new robot kit for the Raspberry Pi, designed for digital makers wanting to learn and have fun with technology.



A line following robot has the ability to move forward, change direction and detect a line on the ground. This is a classic example of autonomous behaviour.



Raspberry Pi

## PIBOT

Artikel uit: The MagPi25, ISSUE 25 – JUL. 2014, A Magazine for Raspberry Pi Users

### Introduction

The robot revolution is coming. Robots are no longer just machines from science fiction. From self-driving cars, to flying drones, robots are on the march. Over the next few years robots are going to be seen all over the place and will be increasingly used in agriculture, manufacture, medicine and education, as well as in our own homes. The amazing thing is that now almost anyone can become a robotist and if you have a Raspberry Pi you are already half way there.

As an example we will cover the building of the simplest robot possible for the Raspberry Pi.

Before we get into the what and how-to of any robotics, the first question may be why would you want to build a robot with a Raspberry Pi in the first place? Also, what kind of things will a Raspberry Pi robot be able to do?

I first got interested in Raspberry Pi robots when I realised how good they can be for learning about technology and also the teaching of technology to others. The great thing

about robots is that they are physical and immersive. Instead of providing output to pixels on a screen, a robot is in your personal space and its movements, lights and sounds go way beyond the limits of a screen. With the Raspberry Pi and some low cost hardware, not only can you make a robot move, you can make a robot that can speak, dance and a whole lot more besides! Why wouldn't you want to turn a Raspberry Pi into a robot?

What makes the Raspberry Pi so good for robotics (as well as so many other projects) is its special GPIO port (General Purpose Input Output). This allows the Raspberry Pi to connect to all kinds of electronics and hardware.

The fundamental requirements for the most interesting robots are the ability to both sense and interact with their environment, and it is the GPIO port of the Raspberry Pi that makes this possible.

Before going into further details for building a simple robot, let's first consider some robot fundamentals.

### What makes a robot robotic?

Wikipedia defines a robot as "a machine which can be electronically programmed to carry out a variety of physical tasks or actions". In addition to this requirement to perform physical actions, a proper robot should also have an autonomous ability. Autonomy allows a robot to act independently in the world and this makes them different from things like radio controlled vehicles that are not able to function by themselves. The Raspberry Pi's processing ability (both CPU and GPU), along with its GPIO port, gives it lots of potential for developing autonomous behaviour. As a simple example let's take a look at the classic line following robot.

A line following robot has the ability to move forward, change direction and detect a line on the ground. This is a classic example of autonomous behaviour. No matter where the line leads, the robot is programmed to follow the line without the need for any external control.

For our simple robot we are going to add an infrared,(IR) sensor that detects a change in reflectivity. The reflectivity of the ground changes when the robot moves off the line. When this happens it changes direction to get back on the line.

While autonomous behavior can be very useful in a robot it can also be a lot of fun to control it directly. Some of the most interesting applications happen when a robot combines the two. Imagine a flying robot that can fly around at your command but can also be programmed never to crash into walls and other obstacles.

### Anatomy of a Raspberry Pi robot

Let us go through all the things you need to build your own robot using a Raspberry Pi. I will detail a minimal possible Raspberry Pi robot step by step. These basic steps are:

- Remote access to the Raspberry Pi
- Powering a Raspberry Pi without cables
- Building a robot chassis
- Making the robot move
- Adding sensors

The first hurdle to turning a Raspberry Pi into a robot is to untether it from all wires and cables.

# Vervolg PIBOT

## Remote access

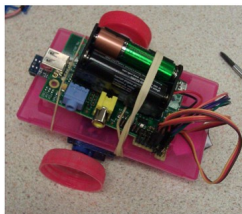
Most people will be familiar with interfacing a Raspberry Pi to a monitor/TV, keyboard and mouse, but this is just not going to work for a robot. The best thing for connecting remotely is a wireless dongle connected to the Raspberry Pi USB port. I won't go into all the details for doing this here. Instead, visit <http://pihw.wordpress.com/guides/guide-to-remoteconnections/>.

Another interesting thing to note is that you can either connect wirelessly across an existing network or you could turn your Raspberry Pi into a wireless hotspot and connect to it via its own network. See Issue 1 1 of The MagPi, page 1 0 ([http://www.themagpi.com/issue/issue1\\_1/article/turn-your-raspberry-pi-into-a-wireless-access-point](http://www.themagpi.com/issue/issue1_1/article/turn-your-raspberry-pi-into-a-wireless-access-point)) for details.

Rather than using the Raspbian Desktop on the robot, a terminal interface is more suitable. When communicating across the wireless network, a remote SSH terminal session is the best way to send commands. Details are in the previously mentioned guide to remote connections.

## Portable robot power

The next essential for the robot is to get the Raspberry Pi running on portable electrical power. The simplest way to do this is to use a backup USB mobile phone charger. Personally I opted for a basic 4 x AA battery pack with a 5V voltage regulator. If you are using a Model B then a 8 x AA battery pack may be preferable to get longer running time.



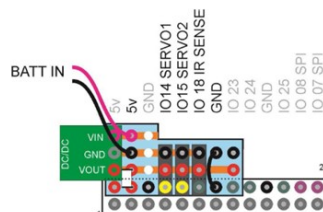
## Building a robot chassis

While it is possible to use some kind of existing chassis to build your robot, it can also be fun to make your own. Even some stiff cardboard can be used for this. The photo overleaf shows a basic design.

## Making the robot move

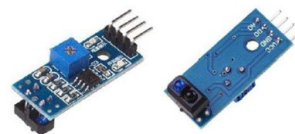
Now that you have successfully untethered your Raspberry Pi and can connect to it remotely, we can give it some robot powers - motion! The simplest and probably cheapest way to give your robot motion is by connecting two continuous rotation servos. Normally servos are designed to turn just 90 or 180 degrees though modified versions exist that provide continuous rotation to be able to drive wheels. Adafruit show how to modify a low cost servo for continuous rotation (<http://learn.adafruit.com/modifying-servos-for-continuous-rotation>). If you want to be really inventive you can also make your own wheels. In the photo above, two large bottle tops are being used for wheels. With two wheels to drive the robot a third wheel is usually required for support. It is positioned in the middle towards the back of the chassis. Ball bearing castors or furniture wheels can work well but to keep things really simple a smooth sliding support can be made using a paperclip.

Now that the robot has wheels the next important step is to connect the servos to the GPIO port. As a hack I made up a simple circuit. This connects the data pins of the servos and the IR sensor to the GPIO and also connects the positive and negative power pins of the servo to a 5V voltage regulator. The circuit is shown on the right.



## Making your robot sensational

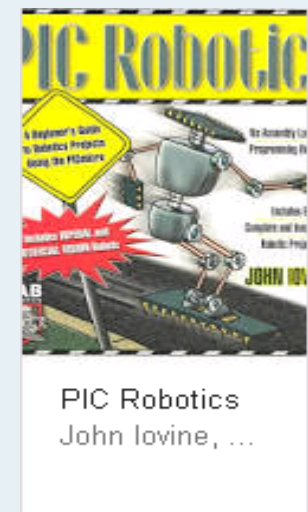
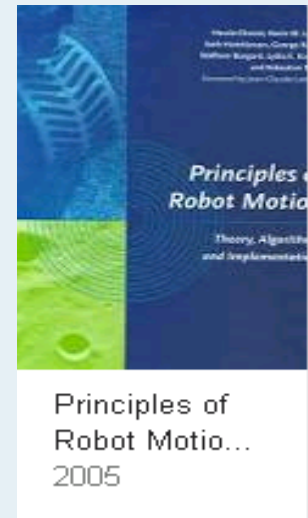
To keep things simple we will only add an IR sensor to our robot. The way this works is that it has one LED that is an IR emitter and one that is an IR receiver. The receiver measures the light reflected from the ground and the amount received will depend of the reflectivity of the surface. A line that has a different reflectivity to the ground around it can be measured by the sensor so it knows whether it is on or off the line. A suitable sensor can be obtained from <http://www.banggood.com> (search for LM393 IR).



## Coding autonomous behavior

Now that we have the hardware of the robot sorted, it is time for the code. The Python script shown on the next page is all that is required to add both remote control and autonomous line-following behaviour to our robot. A detailed explanation of the code is available at <http://www.pibot.org/tiddlybot/code>.

## BOEKEN ROBOTICA



## Kobi: een handige robot "tuinman" voor alle seizoenen

Kobi is een autonome robot die je een handje helpt in ieder seizoen. Kobi is meer dan een standaard grasmaaier robot, zo gaat hij ook blaadjes oprakelen en versnipperen en sneeuw wegblazen wanneer de winter in het land is.

Dankzij een internetverbinding maakt Kobi connectie met sites die het weer voorspellen en kan hij tijdig de sneeuw wegscheppen zodat het niet te dik bedekt in je tuin zal liggen. Verder beschikt Kobi ook over een



ingebouwde GPS technologie en diverse sensoren waardoor hij in no time de grootte van je tuin leert kennen en obstakels ontwijkt. En is er een anti-diefstalsysteem ingebouwd.

De verwachting is dat de Kobi robot rond de 3999 dollar gaat kosten.

Deelnemers en  
toeschouwers  
Roborama NL  
2016 >>>>>

### Beyond the basics

Next month we will look at some more capable robots with features like speech, voice recognition and environment mapping. We will also discuss adding more hardware via an Arduino interface.

In this first part I hope you've become more familiar with the basics of building a robot and hope you agree that building robots with the Raspberry Pi is affordable and fun.

To help make it easier for Raspberry Pi users to get into building robots I have now developed the "TiddlyBot". This brings the ideas described in this article to an easier "build-your-own" robot kit. For "TiddlyBot" I have also added some exciting software as well as things like line drawing capabilities.

To find out more and to support the "TiddlyBot" project, please see our Kickstarter ([http://kck.st/1\\_pOgU1J](http://kck.st/1_pOgU1J)) until 26 July 2014.

With the Raspberry Pi anyone can now be a roboticist and I hope this helps people to both learn and have fun!

```
import RPi.GPIO as GPIO
from RPi.GPIO import PWM
import time
import curses
import sys
left_motor = 14
right_motor = 15
line_sensor = 18
GPIO.setmode(GPIO.BCM)
GPIO.setup(line_sensor, GPIO.IN)
servo = PWM.Servo()
def display_input(stdscr, kboutput):
    stdscr.clear()
    stdscr.addstr(kboutput)
    stdscr.refresh()
    stdscr.move(0, 0)
def turn_movement_servos(left_speed,
right_speed):
    servo.set_servo(left_motor, 1520 + (-1 * left_speed))
    servo.set_servo(right_motor, 1520 + (1 * right_speed))
def check_on_line():
    if GPIO.input(line_sensor) == GPIO.LOW:
        on_line = True
    else:
        on_line = False
    return on_line
def kb_input(stdscr):
    k = 0
    in_line_follow = False
```

```
try:
    stdscr.nodelay(1)
    while True:
        c = stdscr.getch()
        ## Remote control button actions
        if (c != -1) and (in_line_follow is
False):
            if (c == 261) and (c != k): # right
                k = c
                turn_movement_servos(0, 80)
                display_input(stdscr, "Right")
            elif (c == 260) and (c != k): # left
                k = c
                turn_movement_servos(80, 0)
                display_input(stdscr, "Left")
            elif (c == 259) and (c != k): # up
                k = c
                turn_movement_servos(100, 100)
                display_input(stdscr, "Forwards")
            elif (c == 258) and (c != k): # down
                k = c
                turn_movement_servos(-100, -100)
                display_input(stdscr, "Backwards")
            elif (c == 10) and (c != k): # enter
                k = c
                turn_movement_servos(0, 0)
                display_input(stdscr, "Stopped")
            elif (c == 113) and (c != k): # quit
                display_input(stdscr, "QUITTING!")
                time.sleep(0.5)
                RPi.GPIO.cleanup()
                sys.exit()
            elif (c == 32) and (c != k): # space
                k = c
                in_line_follow = True
                line_following()
                display_input(stdscr, "Follow Line")
            else:
                if k != c:
                    d = "ASCII "+str(c)+" not valid!"
                    display_input(stdscr, d)
                ## Line following button actions
                elif c == 32:
                    in_line_follow = False
                    stop() # stop wheels turning
                    display_input(stdscr, "Stopped")
                elif c == 113:
                    display_input(stdscr, "QUITTING!")
                    time.sleep(0.5)
                    RPi.GPIO.cleanup()
                    sys.exit()
            else:
                if in_line_follow:
                    line_following()
                    display_input(stdscr, "Follow Line")
                except curses.error:
                    curses.wrapper(kb_input)
                def line_following():
                    if check_on_line():
                        turn_movement_servos(100, 40)
                    else:
                        turn_movement_servos(40, 100)
                    time.sleep(0.1)
                ## Start program
                curses.wrapper(kb_input)
```



## Roborama NL 2016

Op 5 november 2016 is de Nederlandse versie van de Roborama wedstrijd weer gehouden in Hooglanderveen. Met een record aantal deelnemende robots. Dit jaar hebben we ervoor gekozen om de mini\_Sumo als demonstratie project te doen. Wel hebben we de uitslag meegenomen in de wedstrijdresultaten. Daarnaast was er ook een presentatie doolhof, Maze solving.

Beginners		2	eindstand
Rien van Harmelen	Slalom	83	1
Rien van Harmelen	Wall	68	2
Jack Ruben	Krispa	65	3
Raoul Xavier Boerlage	Jabo-1	58	4
Ben Hooijenga	Madeleine	9	5
Edith van Putten	Red Dwarf	DNF	6

Verder een nieuw onderdeel, Lijnvolgen op een aparte rondlopende baan. Gemeten wordt in feite een rondetijd. En tot slot over de hele dag verspreid de onderdelen van de Roborama wedstrijden.

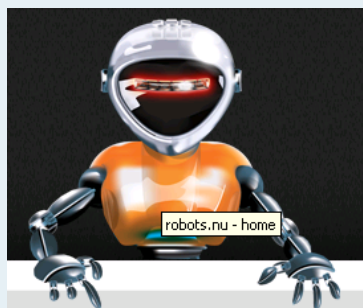
deelnemer	robot	eindscore	eindstand	Kolom1
Jan Heynen	Discovery 1	396	1	
Jan Heynen	Olimex 3	389	2	
Coen Roos	Octopus	281	3	
Patrick de Wachter	PlankTrick	252	4	
Jan Heynen	Discovery 2	242	5	
Gerard van Uden	Bos	225	6	ex aequo
Niek van Uden	geenidee	225	6	ex aequo
Herman Tiggelaven	DFRobot	132	8	
Abraham Vreugenhil	Intel Inside	131	9	
Joep Sujs	PrintBot II	89	10	
Gerard van Uden	1R3	88	11	
Ewoud Huttner	Roborammar	87	12	
Jan Heynen	F103	78	13	
Leo Fortuyn	Black	77	14	
Abraham Vreugenhil	PLV	76	15	
Aloys Verstraeten	Strandbeest	75	16	
Leo Fortuyn	Yellow	68	17	
Edith van Putten	Kryten	67	18	
Jack Ruben	Mocar	67	19	
Fred Eisen	SimpelBot	66	20	
Wim de Boer	lelifo	65	21	
Fred Eisen	Tribot	64	22	
Fred Eisen	Sparky2	64	23	
Rien van Harmelen	Line	DNF	24	ex aequo
Fred Eisen	RedBot	DNF	24	ex aequo
Jan Blok	MOOF	DNS	26	

Van tevoren hebben we omschrijvingen opgesteld voor gevorderden (de allrounders genoemd) en de beginners. Die klassemten hebben we ook berekend uit de resultaten. Ze zijn op de website terug te vinden. Overall winnaar werd Jan Heijnen met de robot Discovery 1. In het beginners klassemten zien we Rien van Harmelen (beginner?) de score aanvoeren met zijn robot Slalom. Bijzonder daarbij is dat door slechts deel te nemen aan 1 onderdeel je al winnaar kunt zijn in dit klassemten. We hebben de trofee uitgereikt aan de in onze ogen beste beginner, Herman Tiggelaven met de DFRobot. Hoewel volgens de van te voren vastgestelde definitie van beginner en allrounder niet de formele winnaar, vonden we de beginnersprijs toch goed bij Herman passen.

Uiteraard is er ook een prijs voor de best-scorende in het beginnersklassemten, Rien. De prijs zal op de volgende Robotica bijeenkomst door de voorzitter uitgereikt worden.

Alle prijswinnaars proficiat.

Voor meer informatie over Roborama NL 2016 kun je naar de website van [HCC Robotica](http://HCC.Robotica).



**ROBOTS.NU**

Robot Discounter

## Vervolg Roborama NL 2016

Ook het eindklassement van het Line-racen is opgemaakt. De winnaar daar is geworden: PrintBot II van Joep Suijs. Het parcours is nu nog eenvoudig, recht toe recht aan, waardoor vrijwel iedereen mee kan doen. De snelste (die niet uit de bocht vliegt) wint. Het ligt in de lijn der verwachting dat er handicaps in het parcours ingebouwd gaan worden. Welke, daar heeft de scheidsrechter nu nog geen idee van. Dat zal de toekomst leren. Leuk wordt en blijft het in ieder geval.

Line-Race	Kolom1	16/11/05	eindklassering	Kolom2
PrintBot II	Joep Suijs	11.93		1
Race	Rien van Harmelen	18.03		3
	nieuwe robot Niek van Uden	21.06		5
Kryten	Edith van Putten	52.24		7
PlankTrick	Patrick de Wachter	77.33		9
PLV	Abraham Vreugdenhil	135.14		11
Ielifo	Wim de Boer	DNF		
Black	Leo Fortuyn	DNF		

Bijzondere vermeldingen zijn er ook nog. De robot van Abraham draait niet op een microcontroller, maar op een antieke microprocessor (De Intel 4004) en heeft een programma van ruim minder dan 100 bytes. Let wel, gewone bytes. Alhoewel het ook nog nibbles zouden kunnen zijn, want volgens mij is de 4004 een 4-bitter...

Daarnaast wandelde Strandbeest van Aloys over het parcours. Niet de snelste, maar veruit de meest bijzondere. Tot slot eigenlijk een eervolle vermelding voor alle robots. Dankzij iedere deelnemende robot hebben we een kijkje kunnen nemen in de mogelijkheden en moeilijkheden die we tegenkomen bij het optimaliseren van robots voor de roborama. Daardoor is het een evenement geworden dat zowel in de breedte als in de diepte veel mooie dingen heeft gebracht. Iedereen dank daarvoor.

© Kampen, Bert Ruben

## Let's e-mail en twitter met onze robots

De huidige informatie- en communicatietechnologie (ICT) biedt de computerhobbyist een heel scala aan mogelijkheden betreffende het meten, regelen en besturen van apparatuur met een computer of microcontroller. Het stelt nauwelijks grenzen aan de inventiviteit en creativiteit van onze robothobby.

Tegenwoordig is het een koud kunstje om een robot te laten communiceren met een netwerk en gegevens te laten uitwisselen. Opslag van data in een database voor registratie en in line analyse is eveneens makkelijk te realiseren. De verwachting is zelfs dat bijvoorbeeld het internet overspoeld gaat worden door IoT (internet of things); alledaagse voorwerpen met embedded elektronica zullen het netwerk overspoelen met data en gegevens oftewel onze fysieke (robot)wereld komt in contact met de virtuele (cloud)wereld.

Om als hobbyist hieraan ook een steentje te kunnen bijdragen wordt in dit project voorgesteld om een computer, microcontroller of autonome robot, zoals wij die binnen de HCC toepassen, informatie te laten uitwisselen via het internet met de gebruiker.

Eerst laat ik zien hoe je vanuit een windows omgeving met een zelf geschreven programma of script, eenvoudig kunt e-mailen.

Het tweede voorbeeld toont een raspberry pi (robot), die kan e-mailen.

Een laatste voorbeeld laat zien dat een raspberry pi ook kan twitteren. Ook kan de robot reageren op commando's, die door een twitteraar naar de robot zijn verzonden.

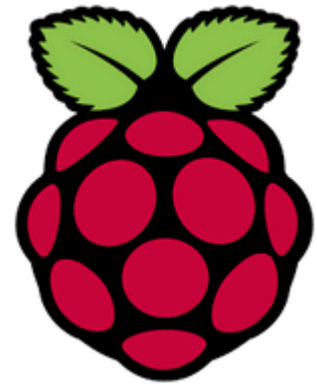
### Raspberry Pi

Een raspberry pi is een zeer populaire micro computer, met afmetingen van een credit card. Naast de vele mogelijkheden om apparatuur aan te sluiten via de Input/Output lijnen (IO) en randapparatuur erop aan te sluiten kan er direct op worden geprogrammeerd met bijvoorbeeld een zeer toegankelijke programmeertaal Python.

Voor communicatie met de buitenwereld kan direct gebruik worden gemaakt van Wifi, Bluetooth, webserver mogelijkheden en dataopslag (bijvoorbeeld in een MySQL database). Dit alles onder het zeer stabiele operating systeem van Linux, met daarbovenop een complete gebruikersvriendelijke grafische user interface.

De gehele microcomputer kan via een VNC verbinding of RDP verbinding volledig op een andere desktop computer of laptop worden bediend zodat aan de Pi geen toetsenbord, muis of beeldscherm hoeft te worden aangesloten. Ideaal dus voor het besturen van (autonome) robots.

Een voorbeeld om een robot te bouwen met een raspberry pi is in een ander artikel weergegeven (zie PIROBOT).



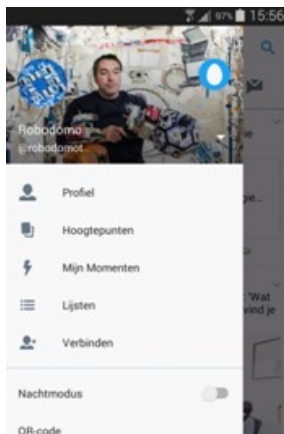
---

*‘Het grote voordeel van twitter ten opzichte van andere social media programma’s is het feit dat twitterberichtjes kort zijn en ‘to the point’ informatie leveren. Ideaal dus voor berichtjes van en voor onze robots.’*

---

### E-mail

Om e-mail te kunnen verzenden is een mailserver of ‘smtp’ (Simple Mail Transfer Protocol) server nodig. Deze server kun je zelf hosten maar het merendeel van ons zal gebruik maken van de mailserver van de provider waarvan de internettoegang wordt verkregen. Het nadeel kan echter zijn dat niet elke provider toestaat dat er mail wordt verstuurd van apparatuur die niet gekoppeld is aan het domein van de provider, zoals bijvoorbeeld straks onze robot. Gelukkig zijn er ook een aantal smtp servers die publiekelijk kunnen worden toegepast, mits je een account hebt. Bijvoorbeeld bij Google gmail. Heb je nog geen gmail account dan is het noodzakelijk een gmail account aan te maken, misschien alleen al voor jouw robot (<https://support.google.com/mail/answer/56256?hl=nl>).



### Twitter

Het grote voordeel van twitter ten opzichte van andere social media programma’s is het feit dat twitterberichtjes kort zijn en ‘to the point’ informatie leveren. Ideaal dus voor berichtjes van en voor onze robots. Bovendien kan zo’n twitterbericht makkelijk worden teruggestuurd naar de robot, die daar dan op kan reageren.

Om te kunnen twitteren heeft de robot dus een twitteraccount nodig. Deze kun je eenvoudig aanmaken via <http://www.twitterinfo.nl/twitteraccountaanmaken.htm>. Mijn robots twitteren via Robodomo (#Robodomot).

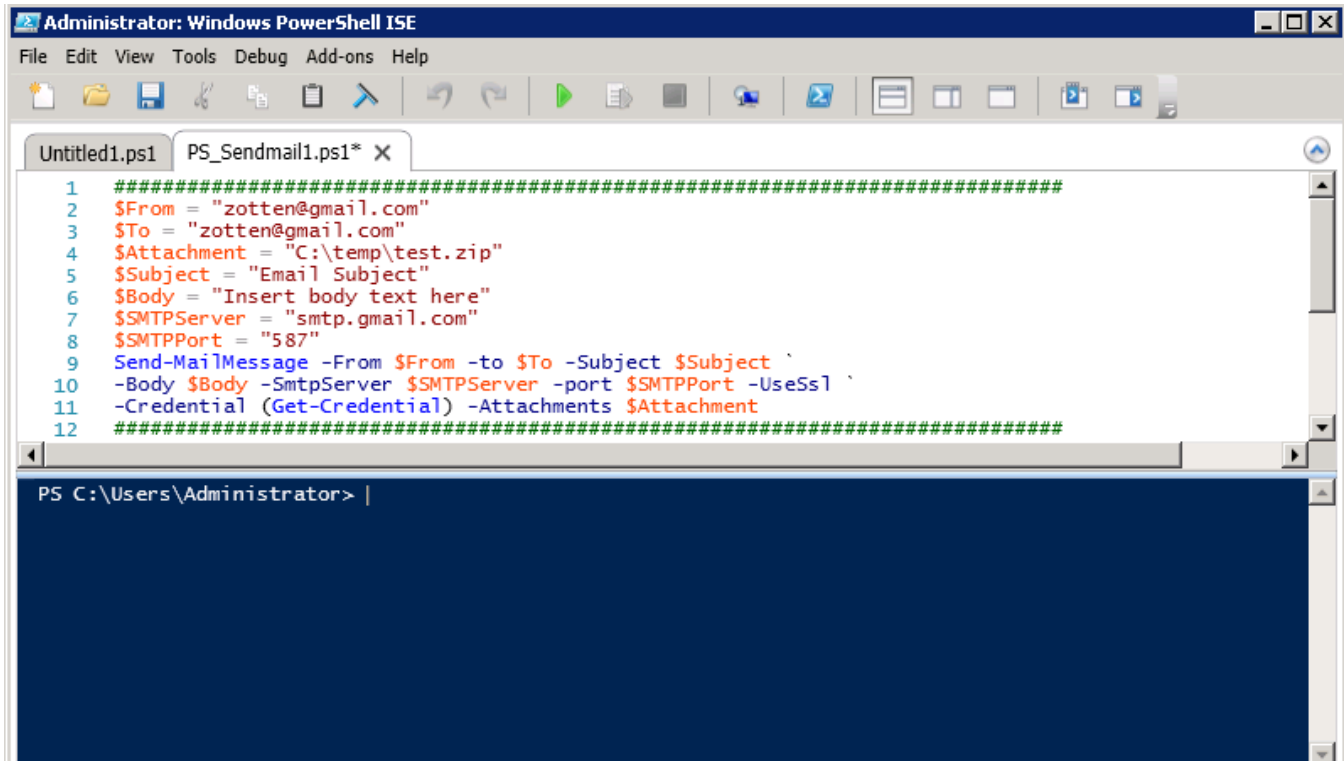
## Vervolg Let's e-mail en twitter met onze robots

Ik ga nu 4 voorbeelden geven waarbij een (mini)computer wordt gebruikt om te e-mailen en te twitteren.

### Voorbeeld 1. E-mail versturen met Windows met behulp van powershell scripting

Windows 10 is standaard uitgerust met Powershell; een Microsoft script taal waarmee de gehele Windows omgeving te beheren is op basis van commando's. Een groot voordeel van script bestanden is het feit dat ze in gewone, leesbare, tekstbestanden worden opgeslagen.

Met het onderstaand script kun je met jouw nieuw aangemaakt gmail account e-mailen zonder e-mail programma:



```
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Untitled1.ps1 PS_Sendmail1.ps1* x
1 #####
2 $From = "zotten@gmail.com"
3 $To = "zotten@gmail.com"
4 $Attachment = "C:\temp\test.zip"
5 $Subject = "Email Subject"
6 $Body = "Insert body text here"
7 $SMTPServer = "smtp.gmail.com"
8 $SMTPPort = "587"
9 Send-MailMessage -From $From -to $To -Subject $Subject `
10 -Body $Body -SmtServer $SMTPServer -port $SMTPPort -UseSsl `
11 -Credential (Get-Credential) -Attachments $Attachment
12 #####
PS C:\Users\Administrator> |
```

In dit voorbeeld is het getoonde script opgeslagen in het bestand *PS-Sendmail1.ps1*.

Wanneer je *PS-Sendmail1.ps1* runt dan krijg je binnen 5 seconden een e-mail.

Als voorbeeld staat er [zotten@gmail.com](mailto:zotten@gmail.com) wat natuurlijk moet worden gewijzigd in *jouw e-mailadres@gmail.com*.

De tag 'Get-Credential' vraagt om de inlognaam en wachtwoord van het gmail account.

Verder kun je een attachment meegeven en natuurlijk de mailtekst (Body) voorzien van zinvolle informatie.

Dit script kun je integreren met jouw windows programmeer omgeving, zoals bijvoorbeeld Visual Basic.

### Voorbeeld 2. Raspberry pi en e-mail

Er zijn meerdere types raspberry Pi's. Een belangrijk verschil tussen de laatste *+3B versie* met eerdere versies is het feit dat de *+3B versie* geïntegreerd Wifi en Bluetooth aan boord heeft. Dus zonder een extra (usb) dongle kun je een wifi verbinding opzetten in jouw (thuis) omgeving. De programmeertaal Python wordt op elke versie ondersteund en de volgende voorbeelden werken dus op alle type Raspberry's mits ze zijn aangesloten op een Wifi netwerk.

De Python code op de volgende bladzijde beschrijft het uitvoeren van een e-mail met de mailserver smtp.gmail.com en met het e-mail account [zotten@gmail.com](mailto:zotten@gmail.com). Vervang dit adres dus weer door jouw gmail adres en wachtwoord! Wanneer je de code in een bestandje *mailme.py* kopieert dan kun je het bestand op een linux command prompt aanroepen. Je kunt de code natuurlijk ook onderdeel laten uitmaken van jouw Python programma die een robot bestuurt of metingen verricht.

```

# -*- coding: utf-8 -*-
#=====
# mailme.py
# algemeen mailprogramma voor platte tekst
# tbv. onderhoud raspberry pi's
# Z.E.H. Otten

import time
import smtplib

from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.image import MIMEImage
from email.mime.application import MIMEApplication

msg = MIMEMultipart('Bericht van mijn robot')
# Definieer e-mail adressen.
adres_naar = 'zotten@gmail.com'          # maar naar een adres!
Tolist = ["zotten@gmail.com"]
adres_van = "mailme@ede"                 #maakt niet uit wat er staat
smtp_server = 'smtp.gmail.com'           #Definieer SMTP e-mail server gegevens.
smtp_gebruiker = 'zotten@gmail.com'
smtp_wachtwoord = '*****'
smtp_poort = '465'

def SendE-mail(tekst):
#http://domoticx.com/raspberry-pi-mailen-via-een-python-script/
#denk aan beveiligingsnivo van google account omlaag brengen!!
#mail opbouwen:
    msg = MIMEMultipart('Bericht van de robot')
    msg['To'] = adres_naar
    msg['From'] = adres_van
    msg['Subject'] = " MailNotification bericht van robot"
    deel0 = MIMEText(tekst, 'plain')
    msg.attach(deel0)

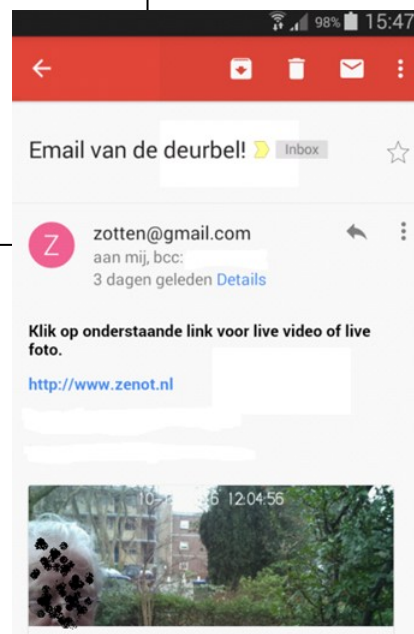
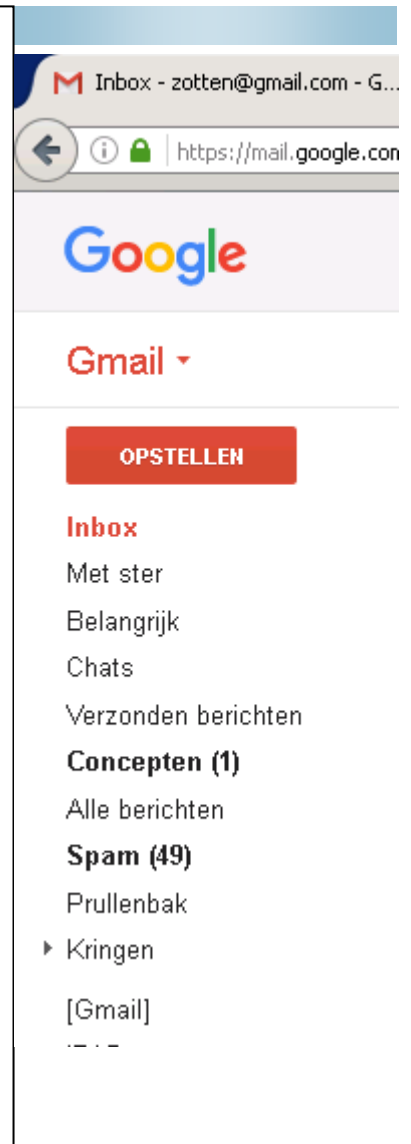
    # Stuur het bericht via de SMTP server.
    s = smtplib.SMTP(smtp_server)
    s.ehlo()
    s.starttls()
    s.ehlo()

    s.login(smtp_gebruiker, smtp_wachtwoord)
    s.sendmail(adres_van, tolist, msg.as_string())
    print "Mail gestuurd"
    s.quit()

#=====
#=====      Main loop      =====
#=====
#print ("mailme.py")
#print time.strftime('%d-%m-%Y %H:%M:%S', time.localtime())
SendE-mail("Tekst die je wilt mailen Raspberry Pi")
#print "Quit"

```

Ik gebruik deze manier van e-mail op een van mijn raspberry pi's (ik heb er meerdere in huis) om mij te berichten wanneer er iemand aan mijn deurbel belt. Ook wordt er een foto meegezonden, hier onherkenbaar gemaakt. De ontvangen mail is hiernaast te zien, bekeken op mijn mobiele telefoon >>





### Voorbeeld 3. Raspberry pi en twitter

Uitgangspunt is dat er een twitter account is aangemaakt. In dit voorbeeld # **robodomot** . Omdat we twitter gaan gebruiken vanuit een applicatie moet die applicatie toestemming krijgen van twitter.com om dit te mogen doen. Hiervoor moet de applicatie worden geregistreerd. Via <http://apps.twitter.com> kun je een nieuwe applicatie registreren door de knop *Create new app* te klikken. Je krijgt vervolgens een key en access token die je in de applicatie moet inlezen. De zogenaamde *consumer key* , *consumer secret access token* and *access token secret* . Het verkrijgen van de keys is beschreven in het <https://www.raspberrypi.org/learning/getting-started-with-the-twitter-api/worksheet/>

De verkregen keys zien er ongeveer zo uit:

```
consumer_key = 'cGo9oepokdp,m,amasq5YDIszxnrs5'  
consumer_secret = 'X87847854-H6sjX66TdjkkOPIO8UfkfdqGmjnNg'  
access_token_secret = 'XwKuVVltsplgJjdjksksnucluyms7EOBZjhRC'
```

Deze keys moeten worden ingelezen in de applicatie.

Vervolgens wordt de twython software bibliotheek voor het twitteren ingelezen. Dit is eenvoudig uit te voeren door de volgende commando's te gebruiken op de raspberry pi:

```
Sudo apt-get update  
Sudo apt-get upgrade  
Sudo apt-get install python-setuptools  
Sudo easy_install pip  
Sudo pip install twython
```

Nu kunnen we gaan twitteren. De meeste simpele tweet die je nu kunt versturen vanuit python ziet er dan als volgt uit:

```
from twython import Twython  
from auth import (  
    consumer_key,  
    consumer_secret,  
    access_token,  
    access_token_secret  
)  
twitter = Twython(  
    consumer_key,  
    consumer_secret,  
    access_token,  
    access_token_secret  
)  
  
message = "Hello world! from PI "  
twitter.update_status (status=message)  
print ("Tweeted: %s" % message)
```

Thuis heb ik een weerstation, een deurbel en een energiemonitor waarbij een raspberry Pi centraal staat. Het weerstation twittert mij de buitentemperatuur, de deurbel mailt mij indien er aangebeld wordt en mijn energiemonitor waarschuwt mij op twitter indien mijn energieverbruik een drempelwaarde van 3.5 kWh bereikt. Deze twitterberichtjes ontvang ik op mijn telefoon. Hiernaast een paar voorbeelden (mijn raspberry's hebben allemaal een naam) .

### Voorbeeld 4: Robot opdrachten geven via twitter

Een twitter bericht kan ook worden gebruikt om de raspberry pi (robot) opdrachten te geven. Door een filter te programmeren in het Python programma kan de Pi selectief reageren op twitter berichten die worden ontvangen. Wanneer (een deel van) de inhoud van het bericht overeenkomt met een gekozen keyword of regel, dan zal de software reageren door het afhandelen van een actie.

In volgend voorbeeld reageert de Pi op een door #Robodomo getwitterde woord #onxin:

```

import time

from twython import TwythonStreamer
from twython import Twython
from auth import (
    consumer_key,
    consumer_secret,
    access_token,
    access_token_secret
)
twitter = Twython(
    consumer_key,
    consumer_secret,
    access_token,
    access_token_secret
)

# Search terms
keyword = '#onxin'

# Setup callbacks from Twython Streamer
class BlinkyStreamer(TwythonStreamer):
    def on_success(self, data):
        if 'text' in data:
            print data['text'].encode('utf-8')
            print
            print 'aan'
            time.sleep(0.5)
            print 'uit'

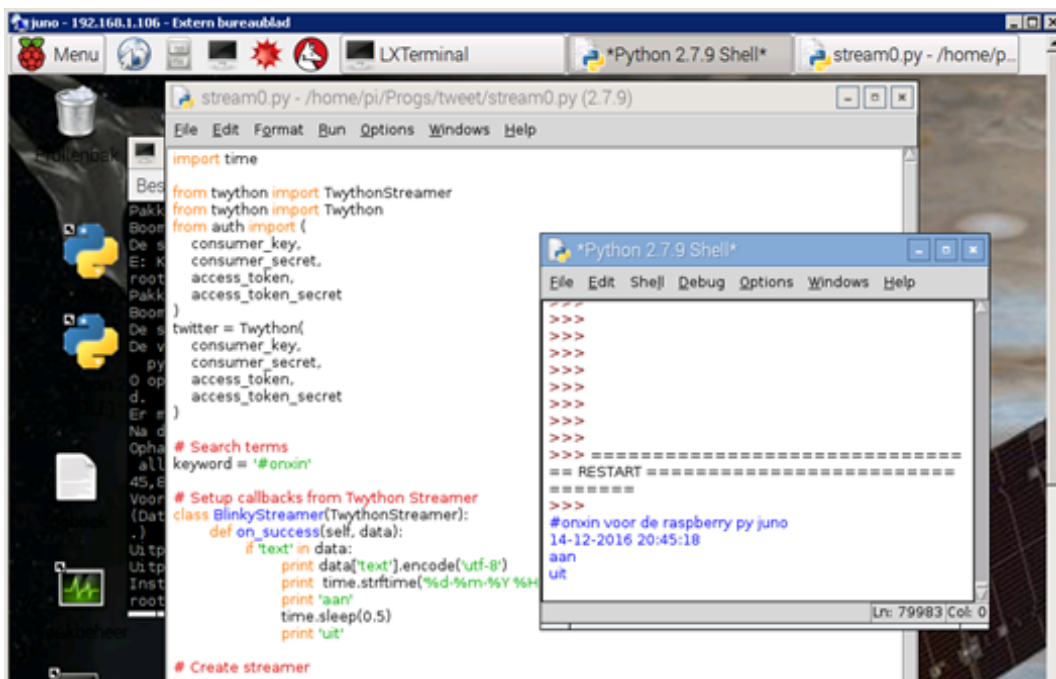
# Create streamer
try:
    stream = BlinkyStreamer(consumer_key,consumer_secret, access_token, access_token_secret)
    stream.statuses.filter(track=keyword)
except KeyboardInterrupt:

```



De reactie van het programma is het 'print aan' en 'print uit' commando op het scherm afdrucken. (zie ook <https://learn.sparkfun.com/tutorials/raspberry-pi-twitter-monitor> ).

Zie hier een schermafdruck van de raspberry Pi waarop het python programma runt met de uitvoer van het programma:



Succes met e-mailen en twitteren!  
Z. Otten

## HCC!Robotica ig

HCC-Robotica is een interessegroep die zich bezig houdt met het ontwikkelen, ontwerpen, programmeren en bouwen van elektronica en mechatronica, toegepast op robots. Deze meer of minder intelligente en autonome robots en machines met verschillende sensoren, actuatoren, processoren en bewegende onderdelen worden onder andere ingezet bij de jaarlijkse georganiseerde Roborama wedstrijden. Wij komen elke eerste zaterdag van de maand bijeen in dorps huis de Dissel te Hooglanderveen. Kennis delen, kennis vergaren, presentaties en workshops bijwonen zijn terugkerende activiteiten tijdens deze bijeenkomsten.

U bent van harte welkom!

## HCC!ROBOTICA: workshop Arduino en I2C

### Aankonding workshop Arduino en I2C

Twee jaar na de succesvolle workshop C, Arduino en Robots geven Karel en Joep opnieuw een workshop.

Dit keer is het onderwerp I2C, een handige interface die toegang geeft tot allerlei sensoren en andere randapparatuur.

Met de bekende mix van theorie en praktijk gebruik je verschillende type slaves en maak je zelfs je eigen slave op basis van een Arduino.

De workshop start in januari 2017 en zal tijdens vier achtereenvolgende bijeenkomsten (januari t/m april) gehouden worden.

### Locatie:

De Dissel waar ook de maandelijkse bijeenkomsten plaatsvinden. Hooglanderveen nabij Hoevelaken.

**Tijd:** 13:00 - 15:00



# hcc!robotica

### Dagelijks bestuur:

Voorzitter : Bert Berrevoets

Secretaris : Edith van Putten

Penningmeester : Joep Suijs

### Het Kernledenbestand ziet er als volgt uit en zal het dagelijks bestuur ondersteunen:

Redactie : Zeno Otten

Website : Pim v. d. Bos

Techniek : Tim Woldring

Roborama : Bert Ruben

Public Relations : Rien van Harmelen

Externe Contacten : Ed Buzzi

Website: <http://www.hccrobotica.nl>