

HCC ROBOTICA

December 2020
Jaargang 23
Nummer 4

ROBOBITS₋₉₀

VAN DE BESTUURSTAFEL

Beste clubgenoten,

Helaas kunnen we weinig organiseren in deze tijd en zoals het er nu uit ziet zal het nog even duren voor we weer life bij elkaar kunnen komen in Hooglanderveen.

Gelukkig hebben we ons digitale café en de maandelijkse virtuele bijeenkomst.

De virtuele bijeenkomst biedt ook mogelijkheden voor leden die normaal niet kunnen komen.

Maak er gebruik van en doe mee met de HCC Robotica!

De link https://groups.google.com/g/hcc_robotmc wordt veel gebruikt; als je nog geen lid hiervan bent meld je dan aan. Op dit platform komen ook de aanmeldingen voor de virtuele bijeenkomsten voorbij en kun je altijd vragen stellen over je hobby.

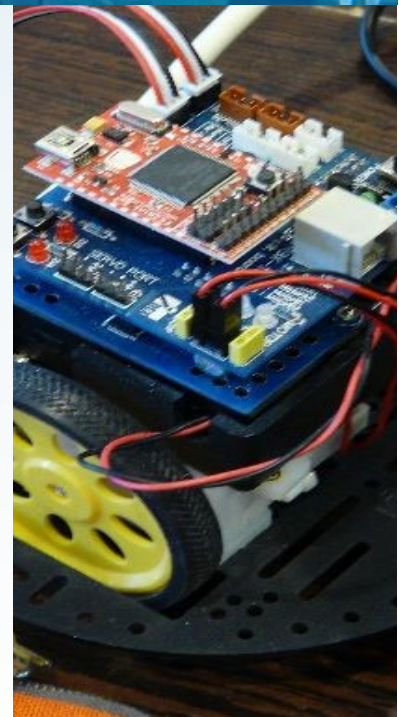
Namens mij en de HCC:

2020: een jaar om nooit te vergeten, bedankt voor je inzet

2021: een jaar van verbinding en interactie, online en fysiek

Voor nu: Een gezond en voorspoedig nieuwjaar met hopelijk weer meer bijeenkomsten in Hooglanderveen.

Wim de B.

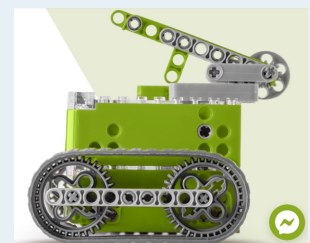


IN DIT NUMMER

Van de bestuurstafel	1
Webscraping met python.....	2
Python voor robots?.....	4
Het raadsel van de Kerise micro- muizen	9
HCC Agenda.....	10

ROBOCAMP

Op deze leuke site leer je werken met robots. Een aantal lessen om tientallen robot projecten heen, interactieve bouw- en programmeer-instructies en er is veel kennis op te doen en veel te ontdekken. Allemaal beschikbaar voor de hobbyist en vooral ook jongeren.



Webscraping met Python

Deze Robobits gaat over de programmeeromgeving Python en over robotica. Ook is de Robobits een clubblad met nieuws uit de dagelijkse praktijk van de hobbyist. Daarom past het volgende verhaal over de techniek van het 'webscrapen' zeker in robobits. Het voorbeeld dient als inspiratie om onze robots 'iets intelligenter' te maken met behulp van het internet.

In Robobits89 gaf ik een boeksuggestie: 'Webscraping met Python'. Webscraping is de techniek om geautomatiseerd gegevens van het internet te vergaren. Zonder browser! Stel je voor dat een grasmaairobot weet wat de weercondities zijn en dat het voorlopig droog genoeg blijft om het gras te gaan maaien? De informatie die nodig is om deze beslissing te nemen is op allerlei (weer) websites te vinden. De robot hoeft alleen maar de informatie te webscrapen.

... De robot hoeft alleen maar de informatie te webscrapen.....

Probleem

In mijn woonplaats heeft elk huishouden minstens een viertal afvalcontainers; een voor plastic afval, een voor GFT (composteerbaar) afval, een voor papier en een voor 'rest' afval. Elke container wordt minstens een keer per maand geleegd. Soms op maandag, soms op dinsdag en ook op donderdag, mits er geen feest - of vrije dagen zijn. Dan wordt het weer anders.

Je begrijpt het al. Ik raak in de war van de 'regelmaat' en het gebeurt dus dat ik te laat ben met het aanbieden van mijn afval. De hulp van een jaarlijkse 'afvalkalender' is weliswaar een oplossing maar hoe leuk is het om dit probleem te laten oplossen door een 'webscraping' programma, geschreven in Python?

Doel van het scrape programma

Ik wil een dag van te voren een berichtje ontvangen op mijn telefoon (mail, tweet of ander berichtje) waarin wordt vermeld welke afvalcontainer ik moet klaarzetten. In dit geval kies ik voor een e-mail naar robodomo@gmail.com omdat ik de procedure om mail te sturen al vaker in Python heb toegepast.

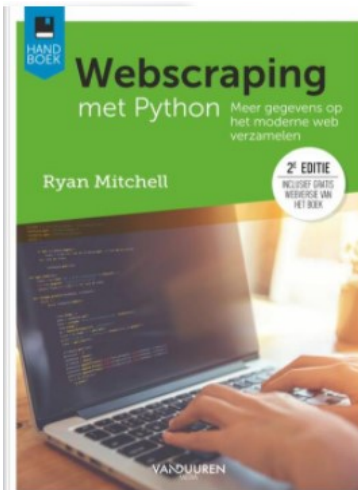
Werkwijze

Ik gebruik een Raspberry Pi Zero (circa 10 euro) om mijn Python programma op te draaien. Tevens heeft de Zero Wifi aan boord en daardoor internet toegang. Een volledige Python3 omgeving is standaard aanwezig op de Linux distributie Raspbian. Om het programma te schrijven kan ik kiezen uit meerdere editors, zoals Geanny, Idle of Thonny.

Het programma zal via de Linux taakmanager 'crontab' één keer per dag worden aangeroepen om te checken of er een mail moet worden gestuurd.

Ik gebruik de website van de lokale afval verwerkingsdienst die de afhaaldata daar publiceert. Ik maak gebruik van een paar extra, maar zeer vaak toegepaste python libraries zoals:

Smtplib voor het sturen van e-mail, urllib voor het bezoeken van de website en voor het scrapen van data en BeautifulSoup. Deze laatste is geen standaard Python library en moet apart worden geïnstalleerd. BeautifulSoup helpt bij het formatteren en organiseren van html opgemaakte internet pagina's.



[Webscraping met Python, Ryan Mitchell](#) | 9789463561006 | Boek - bruna.nl



Op de website `url = 'https://afvalkalender.dar.nl/adres/postcode:nummer'` is de gewenste data te vinden.

`html = urlopen(url)` opent de site en de regel `bs = BeautifulSoup(html.read(), 'html.parser')` leest de website in de variabele `bs`.

De regel `s = bs.find_all(class_='date')` zoekt in `bs` alle `html <date>` tags en plaatst die in `s`.

De procedure `analyseer` kijkt in de gevonden `s` om welke ophaaldata het gaat. Indien het de volgende dag een ophaaldag is dan stuurt de procedure 'sendmail' mij een email.

Dit webscraping voorbeeld gaat ervan uit dat bekend is waar de gewenste data te vinden is op de website, namelijk achter de `<date>` tags.

De `google chrome browser` beschikt over een tool om `html pagina's` te bekijken (via de menu optie 'hulpprogramma's voor ontwikkelaars'). Die kun je gebruiken om gewenste data te vinden.

Het gehele scrape programma ziet er als volgt uit:

```
#
# webscraping programma voorbeeld met beautiful soup !
# Webscraping met Python, Ryan Mitchell
#
# (c) Z.E.H. Otten
# november 2020
#

import datetime
import smtplib
from locale import setlocale, LC_ALL
from datetime import date, datetime, timedelta
from email.mime.text import MIMEText
from urllib.request import urlopen
from bs4 import BeautifulSoup

debugPrint = False
setlocale(LC_ALL, "nl_NL.utf-8")
url = "https://afvalkalender.dar.nl/adres/postcode:nummer"
html = urlopen(url)
bs = BeautifulSoup(html.read(), 'html.parser')
afval = ["Plastic", "Rest-afval", "GFT-afval", "Papier"]

def sendMail(subject, body):
    msg = MIMEText(body)
    msg['Subject'] = subject
    msg['From'] = "mailadres"
    msg['To'] = "mailadres"

    s = smtplib.SMTP('smtp.gmail.com')
    s.ehlo()
    s.starttls()
    s.ehlo()
    s.login("mailadres", "password")
    s.send_message(msg)
    s.quit()
    return
```

```
def analyseer():
    s[0]= str(s[0])[16:25]
    s[1]= str(s[1])[16:25]
    s[2]= str(s[2])[16:25]
    s[3]= str(s[3])[16:25]

    #een dag van te voeren waarschuwen
    nextDay = datetime.now() + timedelta(days=1)
    if debugPrint: print (nextDay.strftime('%a %-d %b'))

    message = ""
    for d in range(4):
        if debugPrint: print (d, s[d], afval[d])
        if nextDay.strftime('%a %-d %b') == s[d]:
            message = message + "Morgen" + " (" + s[d] +)" " + "ophaaldag " + afval[d]
            message = message + "\n"
        else:
            if debugPrint: print ('morgen geen ophaaldag')

    if message != "":
        sendMail("Kalenderbericht:",message)
        print (message)
        if debugPrint: print ('Mail gestuurd')
    return

# main loop
print ("Running scrape program...")
s = bs.find_all(class_='date')

if len(s) == 4: # vier afvalstromen
    analyseer()
else:
    print ("error..")
```

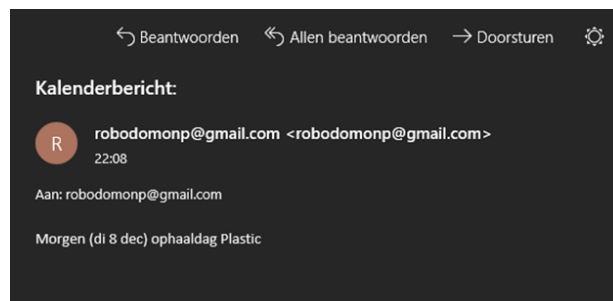
Het resultaat is hiernaast weergegeven: een e-mail aan `robodomnp@gmail.com` met de boodschap dat morgen de container met plastic afval wordt opgehaald.

Zeno O.

Beautiful soup, so rich and green,
Waiting in a hot tureen!
Who for such dainties would not stoop?
Soup of te evening, beautiful Soup!

(gedicht van Lewis Carroll)

Wordt er veel gebruikt gemaakt van Java(script) op een website of moet er eerst worden ingelogd op een website dan blijft het mogelijk om de site te scrapen. Het boek biedt daarvoor meerdere oplossingen met voorbeelden.



Python voor robots?

Python is een populaire programmeertaal die breed wordt gebruikt, onder meer voor educatie. Het is bijvoorbeeld één van de populairste talen voor de Raspberry Pi.

Wikipedia schrijft:

“Python is ontwikkeld met het oog op leesbare code. Hieruit vloeit haar "zuivere" stijl voort. Met weinig woorden kan men veel zeggen. Dit uit zich op verschillende manieren. Structuur wordt aangebracht door indentatie (regelinspringing) in plaats van bijvoorbeeld de accolades uit C-achtige talen. Statements (vergelijkbaar met zinnen uit gewone taal) worden simpelweg beëindigd door het eind van de regel.”

En dat klinkt veelbelovend, want de accolades en puntkomma's van C/C++ vinden de meeste mensen niet fijn. En er zijn nog veel meer voordelen, maar dat is wat voor een andere keer. Wat ik met jullie wil delen zijn mijn eerste ervaringen met Python voor de besturing van een robotje.

Python of MicroPython?

Python is oorspronkelijk bedoeld voor 'grote' computer met bijvoorbeeld Windows. Of Linux, zoals de Raspberry Pi. MicroPython is een Python implementatie die speciaal ontwikkeld is voor (de wat stevigere) microcontroller, zonder operating system. Dit zou wel eens het beste van twee werelden kunnen zijn: de relatieve eenvoud van een microcontroller en de kracht en toegankelijkheid van Python. Voldoende reden om dit uit te proberen.

Hardware.

Enig onderzoek leert dat encoders uitlezen met MicroPython wellicht lastig wordt. Om dit te vermijden – en meteen eindelijk eens een test te doen met een SPI slave – heb ik gekozen voor de volgende configuratie.

Een Arduino Nano leest de encoders van de motoren uit en slaat deze op in registers, die via SPI opgevraagd kunnen worden door de ESP32. De ESP32 geeft via SPI aan de Arduino Nano door wat de gewenste PWM-waarden zijn voor de motoren.

Voor de test heb ik gekozen voor de Wemos LOLIN D32 Pro V2. De kern van dit bord is de ESP32, een module met een dual core 240 MHz processor en 16MB flash. De wifi van de ESP32 maakt het mogelijk om draadloos te communiceren. Het bordje is voorzien van 8MB extra RAM, RAM die nodig is als je programma groter wordt.

Configuratie.

Op het bordje is door de fabrikant al MicroPython geïnstalleerd, dus je kunt hier direct mee aan de slag. Sluit het bordje aan via USB en maak verbinding met de favoriete terminal-programma op 115200 baud. Als je op reset drukt, zie de opstart van de module die eindigt op:

Een compleet overzicht van de functies in de module 'os', waaronder het wijzigen van directories, vind je hier: <https://docs.micropython.org/en/latest/library/uos.html>

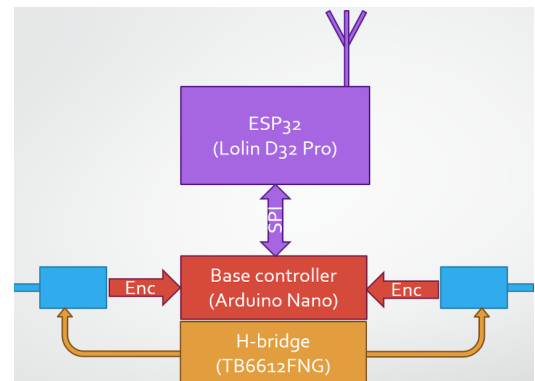
```
I (0) cpu_start: Starting scheduler on APP CPU.  
MicroPython v1.13 on 2020-09-02; ESP32 module (spiram) with ESP32  
Type "help()" for more information.  
>>>
```

Je kunt nu Python commando's ingeven. Bijvoorbeeld:

```
import os  
os.listdir()
```

De import op de eerste regel hoeft (na reset) slechts eenmaal uitgevoerd te worden en laadt de module 'os'. De functie 'listdir' uit de module 'os' geeft een lijst terug met files in de huidige directory. In mijn geval staat hier alleen de file boot.py:

```
>>> import os  
>>> os.listdir()  
['boot.py']  
>>>
```



Nu het bord werkt, gaan we eerst REPL activeren. REPL – Read-Evaluate-Print-Loop – is een tool om interactief en zonder vertraging software te ontwikkelen en testen. De ESP32 versie van MicroPython heeft een script om REPL te activeren voor toegang vanaf het netwerk. Start dit script en beantwoord de vragen:

```
>>>import webrepl_setup
WebREPL daemon auto-start status: enabled
Would you like to (E)nable or (D)isable it running on boot?
(Empty line to quit)
> e
Would you like to change WebREPL password? (y/n) y
New password (4-9 chars): 1234
Confirm password: 1234
Changes will be activated after reboot
Would you like to reboot now? (y/n) y
```

Met de volgende 4 commando's maken we verbinding met ons wifi-netwerk. De regels die beginnen met '#' zijn commentaar en kun je weglaten.

```
# Laden van de module 'network'
import network
# Maak een 'WLAN' object van het type STA(tion) aan en bewaar deze
# in de variabele mySta. Dit kinkt ingewikkeld, maar mySta is
# eigenlijk gewoon de wifi-adapter waarmee je verbinding kunt maken
# met een acces-point.
mySta = network.WLAN(network.STA_IF)
# Schakel mySta in.
mySta.active(True)
# Laat mySta verbinding maken met het wifi netwerk 'ewave' en
# gebruik daarbij het wachtwoord 'geheim!'. Uiteraard vul je hier
# de gegevens van je eigen wifi-netwerk in.
mySta.connect('ewave', 'geheim!')
```

Bij het uitvoeren van de commando's komt logging voorbij, inclusief een aantal warnings. En als het goed is, eindigt de logging met:

```
(1336835) network: CONNECTED
I (1336885) wifi:AP's beacon interval = 102400 us, DTIM period = 3
I (1338985) event: sta ip: 192.168.178.124, mask: 255.255.255.0, gw: 192.168.178.1
I (1338985) network: GOT_IP
```

Onthoud het IP-adres (in dit voorbeeld 192.168.178.124), dit gaan we later gebruiken.

Overzicht wifi netwerken

Wil je een overzicht van de wifi netwerk die je bordje ziet? Dit kan met het commando

```
mySta.scan()
```

Dit is de (ingekorte) output bij mij:

```
[(b'ewave', b'd\xa5\xc3qX\xf6', 5, -54, 4, False), (b'ziggo100', b'\xecU\xf9b\x8d\xe0', 1, -78, 3, False), (b'Ziggo',
b'\xeeU\xf9b\x8d\xe0', 1, -78, 5, False), (b'Coblo', b'\xb0N&6\x97\xa9', 1, -83, 4, False), (b'H368N0F5327', b't\xda\x88\xb7
\x9e\xe1c', 8, -84, 4, False), (b'Ziggo23260', b'\x84\x16\xf9\xd4\xc8\xfd', 10, -94, 3, False)]
```

De output is niet erg overzichtelijk, maar waarom dat zo is, is een verhaal voor later. Voor nu is het relevant om te weten dat b'ewave' een byte-string is met de inhoud ewave. En de eerste byte-string na een openingshaak '(' is de naam (SSID) van het wifi netwerk. Voor meer informatie over de output van deze scan en andere functies van het WLAN object zie: <https://docs.micropython.org/en/latest/library/network.WLAN.html>

vervolg: Python voor robots?

Toegang vanuit de browser

We hebben nu, met ons favoriete terminal-programma, het bord beschikbaar gesteld in ons lokale netwerk. En daar kunnen we gebruik van maken met behulp van WebREPL. Download dit vanaf <https://github.com/micropython/webrepl>. Het eenvoudigst gaat dat waarschijnlijk door op de knop 'Code' te drukken en te kiezen voor 'download zip'. Pak die zipfile vervolgens uit in de gewenste folder en start `webrepl.html` door hierop te klikken. De browser opent dan met het volgende scherm:



Vul het IP-adres van het bordje in en druk op `Connect`. Na het invullen van het eerder opgegeven WebREPL wachtwoord (1234 in ons voorbeeld) kunnen we inloggen en krijgen we de bekende `>>>` prompt.

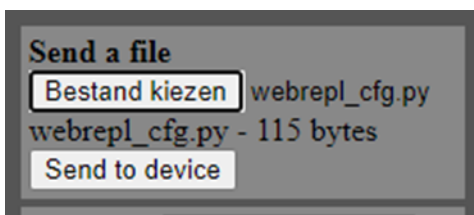


Maak configuratie permanent

We hebben nu vanuit een browser draadloos toegang tot het bordje. Totdat het bordje opnieuw start, want dan moet opnieuw verbinding worden gemaakt met het wifi-netwerk. WebREPL wordt wel meegenomen bij het opstarten en het wachtwoord wat we hebben opgegeven is daarvoor vastgelegd in `webrepl_cfg.py`. Dit script kunnen we mooi gebruiken om ook verbinding te maken met het wifi-netwerk. Maak hiervoor, met de favoriete tekst-editor, op je PC het bestand '`webrepl_cfg.py`' aan met de volgende inhoud:

```
PASS = '1234'  
import network  
sta = network.WLAN(network.STA_IF)  
sta.active(True)  
sta.connect('ewave', 'geheim')
```

Kies vervolgens het bestand met de browser en druk op 'Send to device'.



Daarna het bordje resetten en opnieuw verbinding maken vanuit de browser om te controleren of alles goed is gegaan. Werkt het niet? Open dan je terminal-programma en kijk via de usb-poort waar het mis gaat.

Ervaringen en conclusie

Ik heb een test-robotje gemaakt met 2 processoren, een Arduino die o.a. de encoders uitleest en een ESP32-bordje met extra geheugen, waarop MicroPython draait voor de besturing van de robot. Deze configuratie werkt goed en via wifi heb je ook draadloos toegang, wat handig is met robotjes.

Python is een krachtige taal, waardoor programmeren sneller gaat. Daar staat tegenover dat de uitvoeringssnelheid van een Python programma aanzienlijk lager dan die van een vergelijkbaar C/C++ programma. Dit wordt deels opgevangen door moderne, snellere processoren en door het (tijd kritische) uitlezen van de encoders uit te besteden aan een tweede processor. Of het bordje ook snel genoeg is om grotere hoeveelheden (sensor)data te bewerken zal moeten blijken.

Beschikbaar geheugen is wel een punt van aandacht. Eerste experimenten op een ESP32 zonder het extra geheugen zijn gestrand omdat het geheugen vol was zodra het programma iets groter werd of iets meer bewerkingen uitgevoerd werden.

Python is een taal waarin je snel dingen kunt proberen en interactief commando's kunt ingeven en gegevens opvragen, bijvoorbeeld om diagnose te stellen. Over Python is veel informatie te vinden en er zijn ook veel nuttige bibliotheken beschikbaar. Een veelbelovende bibliotheek voor robotica is uasyncio, waarmee taken op de achtergrond uitgevoerd kunnen worden.

Kortom, dit experiment met MicroPython voor besturing van een robot verdient een vervolg!

Joep S.

uasyncio — asynchronous I/O scheduler

Core functions

Additional functions

class Task

class Event

class Lock

TCP stream connections

Event Loop

ubinascii — binary/ASCII conversions

ucollections — collection and container types

uerrno — system error codes

uhashlib — hashing algorithms

uheapq — heap queue algorithm

uio — input/output streams

Maqueen Plus voor Micro:bit

Dit MicroPython-op-ESP32 project is gestart in het tweede kwartaal van 2020. In dezelfde periode werd in Shanghai de laatste hand gelegd een robot met een vergelijkbare architectuur: de Maqueen Plus voor Micro:bit. Net als mijn robot heeft deze een basis-board met een eigen controller en micromotors met encoders. Met de in oktober 2020 geannonceerde Micro:bit V2 kan de Maqueen Plus worden uitgerust met controller die voldoende rekenkracht en geheugen heeft om wat uitgebreidere MicroPython programma's te draaien. Enige minpuntje is dat je met de Micro:bit vooralsnog niet draadloos interactief met Python kunt werken. Maar daar staan vele pluspunten tegenover van de Maqueen tegenover, zoals:

Een mooie vormgeving en nette afwerking.

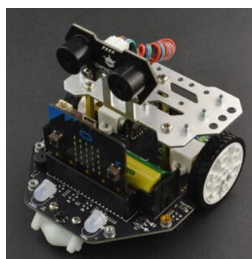
Laad-aansluiting en houder voor een LiPo accu.

Lijnsensoren, geschikt zijn lijnvolgen én maze-solving.

3 Servo-aansluiting.

Een header om sensoren aan te sluiten op de Micro:bit.

Kortom, een bijzonder interessant robotje!



<Maqueen Micro:bit

vervolg: Python voor robots?

MicroPython upgrade.

Op het bordje is standaard MicroPython geïnstalleerd. Bij mij was dit v1.9.4 van 2019-01-07. Een oude versie, onder meer zonder de library `uasyncio`. De eerste stap is daarom het installeren van een recente versie.

- Alle tools voor MicroPython zijn... in Python. Dus als je aan de slag gaat met MicroPython, begin je met installeren van Python 3 op je desktop computer als dit er nog niet op staat. Python kun je downloaden van: <https://www.python.org/downloads/>
- De actuele versie van MicroPython download je vanaf <https://micropython.org/download/esp32/>. Kies voor de meest recente stabiele versie met 'spiram' en 'idf3' (nu `esp32spiram-idf3-20200902-v1.13.bin`).
- Installeer 'esptool' met (vanaf de cmd-prompt)
`python -m pip install esptool`
kijk (bijvoorbeeld in de device-manager / apparatenbeheer) op welke com-poort de ESP32 is aangesloten.

Nu zijn we klaar voor de upgrade:

```
esptool.py --port com22 erase_flash  
esptool.py --chip esp32 --port com22 --baud 460800 write_flash -z 0x1000 esp32spiram-idf3-20200902-v1.13.bin
```

That's all! We hebben nu bordje met een actuele versie van MicroPython. We kunnen dit controleren door een terminal aan te sluiten op de com-port en daarna de reset-knop van het bordje in te drukken:

```
(249) cpu_start: Starting scheduler on PRO CPU.  
I (0) cpu_start: Starting scheduler on APP CPU.  
MicroPython v1.13 on 2020-09-02; ESP32 module (spiram) with ESP32  
Type "help()" for more information.  
>>>
```

Het raadsel van de Kerise micromuizen

Begin december 2020 verscheen Hackaday.com met een opmerkelijk artikel dat een onverwachte ontwikkeling in de micromouse wereld aankondigt. Goed om te weten dat het bij micromouse gaat om het oplossen van een doolhof.

Overzicht plaatje 1, overgenomen uit dat artikel, toont de succesvolle creaties Kerise 1 tm 5 van de bouwer en hierin zijn een paar inmiddels wereldwijd verspreide verbeteringen te zien. Gaande tegen de klok in van rechtsboven naar rechtsonder is te zien dat de muizen kleiner worden met als gevolg lichter en dat helpt het verhogen van wendbaarheid. Wie aan lijnvolgen doet weet dat bij het overschrijden van een kritische snelheid instabiliteit optreedt met als gevolg een slingerende gang. Groot motorkoppel gecombineerd met een lage massa en een strakke regeling zijn de ingrediënten voor verbetering.

Vanaf het begin ergens in 1970 waren tweewielige differentieel aangedreven muizen het standaard concept tot er in 2009 zo'n veertig jaar later !! een prestatie sprong werd gemaakt met Kato-san's Tetra. Het briljante idee was dat een hinderlijk effect kon worden geëlimineerd door aan weerszijde niet een maar twee wielen dicht bij elkaar te plaatsen.

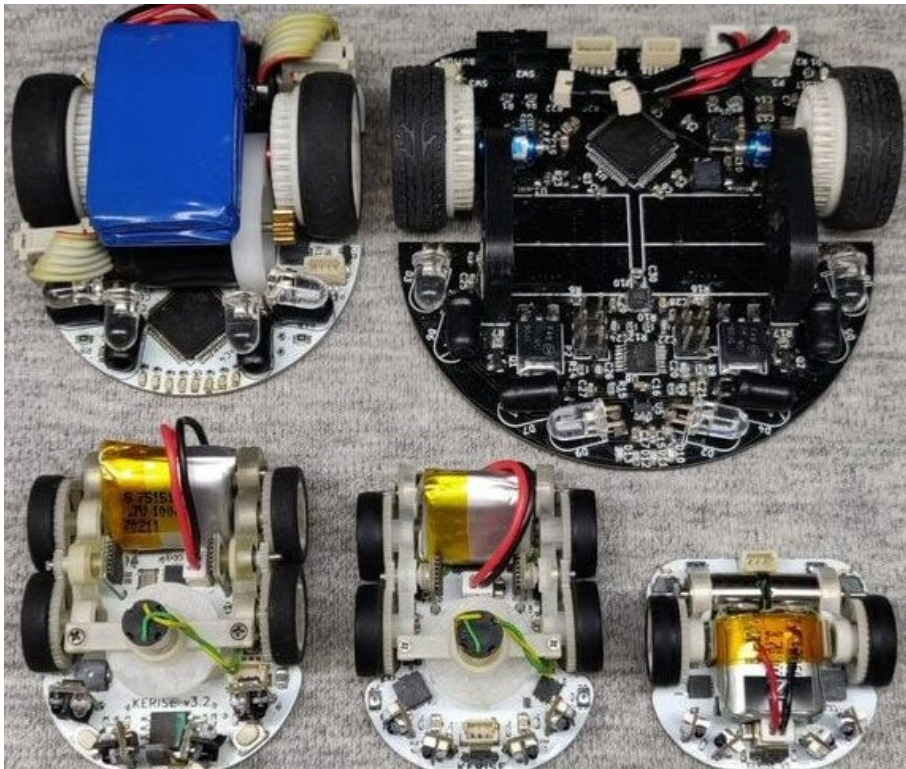
Voor die tijd zijn voor en achter op de structuur glijdopjes (skids) geplaatst met een ruimte t.o.v. de ondergrond van een millimeter om voor de horizontale stand te zorgen. Door een reactiekracht van de motoren wordt bij versnellen druk op de achterste schuif uitgeoefend wat ten koste gaat van de druk van de wielen op de vloer. Verminderde druk van de wielen leidt sneller tot wielslip wat noodzakelijk gecorrigeerd wordt met een meer hellend trapeziumprofiel. Dit drukverlies wordt met dubbele wielen in feite overwonnen.

Bij versnellen verplaatst de druk zich nu gedeeltelijk naar de achterste wielen maar de som van de neerwaartse kracht blijft hoe dan ook op wielen werken. Veel beter slijpdrag wat leidt tot steiler trapezium.

Op een screenshot (plaatje 2) van een slow motion filmpje waar een muis met 15 m/sec/sec versnelt is te zien hoe de voorste wielen loskomen van de grond. Als gevolg van dit dubbele wielen concept draait de muis in hoeken wel als een tank en dat gaat nogal slordig om de verticale as.

In filmpje "Tetra toegelicht" laat David Otten zijn licht schijnen over deze meesterlijk vondst en is in de leader het revolutionaire exemplaar van Katosan te zien.

De prestaties schoten omhoog maar daar bleef het niet bij. Ik schat dat een jaar of vijf geleden opnieuw een doorbraak werd bereikt met kunstmatige zwaartekrachtvergroting. Een vacuüm centrifugaalpomp zorgt in dat geval voor onderdruk onder de wielen met als gevolg dat de muis horizontaal blijft, verlaagt de slijdrempel en laat dus daarbij een nog hogere versnelling/vertraging toe.

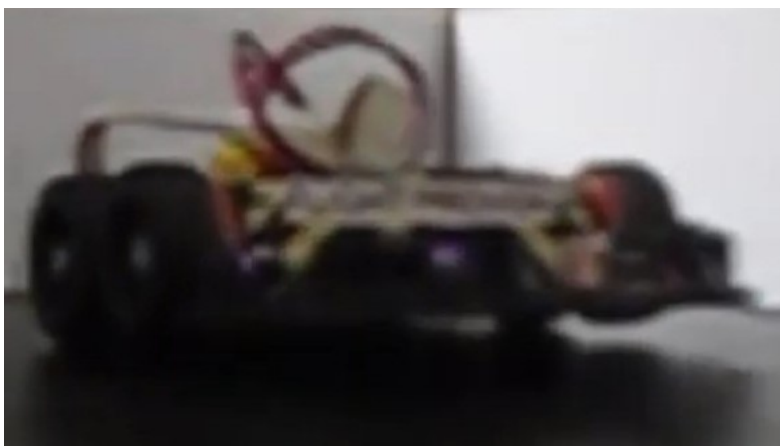


Terugkerend naar de iteraties van de Kerise muizen is het dus de vraag wat er is te verwachten van de prestaties van Kerise 5 waar de twee innovaties die wel zijn toegepast bij de nummers 3 en 4 worden weggelaten dit om winst in gewicht en afmeting te boeken. In het Hackaday artikel wordt overigens ook niet gesteld dat het doel is de voorgangers te overtreffen maar blijft de lezer dus achter met de vraag wat dan wel.

Tip: de twee baanbrekende technieken zijn ook op lijnvolgers toe te passen !!

Coen R.

Plaatje 1: Micro kerise



Plaatje 2: Decimus 5alpha: zo snel..

hackaday Kerise micromouse

<https://hackaday.com/2020/12/06/these-micro-mice-have-macro-control/>

Tetra toegelicht David Otten

<https://www.youtube.com/watch?v=Iw0FBkdj1n0>

Decimus 5A versnelt met 15m/sec/sec in slow motion

https://www.youtube.com/watch?v=CTFK3tTGdFY&feature=emb_logo

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!



Agenda

Bijeenkomsten - Virtueel en fysiek.

Door de corona maatregelen zijn wij genoodzaakt onze normale bijeenkomsten aan te passen, en moet men zich registreren als men naar deze bijeenkomsten wilt komen.

De tijden zijn:

Fysieke bijeenkomsten van 10:00 - 13:00.

Virtueel Inloop Café van 10:00 - 12:00

Voor de datum's zie de agenda op de web site.

Het virtuele inloop Café is iedere zaterdag geopend behalve de 1e zaterdag van de maand.

Om de 2 weken wordt er indien nodig rond 11:00 een korte sessie ingelast met mededelingen van het HCC/Robotica bestuur, werkgroepen etc.

Deze evenementen staan open voor alle HCC leden, en om hieraan deel te nemen, graag registreren.

Eenmalige registratie t.b.v. de Virtuele bijeenkomsten [hier](#)

Fysieke registratie per bijeenkomst [hier](#).

Discussiegroepen

HCCROBOTICA:

http://groups.google.nl/group/hcc_robotmc

Blogs

<http://zotten.wordpress.com/>

[Website van Coen](#)

<https://avretro.wordpress.com/>

<http://www.robotblog.nl/>

Facebook

[HCC!Robotica ook op Facebook.](#)

Gewoon om te laten weten, dat wij ook op [Facebook](#) actief zijn.

Raspberry Pi Compute Module 4



De Compute Module 4 is een uitgeklee Raspberry Pi-gebaseerde module.

Het stelt je o.a. in staat om op maat gemaakte elektronica nauw te integreren rond de Raspberry Pi SOC (bijvoorbeeld voor industrieel gebruik of narrowcasting).

HCC!Robotica ig

Dagelijks bestuur:

Voorzitter : Wim de Boer

Secretaris : Edith van Putten

Penningmeester : Ed Buzzi

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

Redactie : Zeno Otten

Website : Bert Berrevoets

Techniek : Tim Woldring

Roborama : Bert Ruben

Public Relations : Rien van Harmelen

Externe Contacten : Ed Buzzi

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