Inhoud

Gebruikte programma's 1

MPD 1

MPC 1

Installatie op de RPI 2

Configuratie van de muziekspeler op de RPI 2

Test met oude Packard Bell NAS 3

Configuratie NAS 3

Koppeling aan RPI 3

Afspelen via MPC 3

Aansturing van de GPIO Pinnen 4

Configuratie en scripting 4

Gebruikte programma's

http://www.musicpd.org/

MPD

Music player deamon. De aansturing van de muziek. Draait op een server en heeft geen interface. Dit programma ontvangt enkel de commando's via zijn netwerkprotocol. BV: Wanneer het commando "play" ontvangen wordt zal MPD het ingeladen liedje, stream… proberen afspelen.

MPC

Music player client is de eenvoudigste manier om de MPD server aan te sturen. Het is 1 van de vele clients die kan gebruikt worden in combinatie met de MPD Server. Bij onze radio staan dus beide programma's op 1 toestel geïnstalleerd en is de opsplitsing van Client / Server verwaarloosbaar. MPC zal dus enkel de aansturing verzorgen via simpele linux commando's.

Om een ander voorbeeld aan te halen waar er wel een opsplitsing van toepassing is:

Je kan bv ook een smartphone app installeren (dit is dan een andere client) welke een verbinding maakt met de MPD Server op de raspberry PI. En zo zou je dan via je smartphone liedjes kunnen afspelen.

Via MPC gebruiken wij volgende commando's:

mpc clear	Leegmaken van de playlist
mpc load	Toevoegen van stream aan de playlist: bv: mpc load http://www.listenlive.eu/vrtstubru-high.m3u
mpc play	Starten van spelen van playlist
mpc stop	Stoppen van de muziekspeler

Installatie op de RPI

Apt-get install mpd mpc

Configuratie van de muziekspeler op de RPI

De server bindt automatisch aan de localhost (de raspberry zelf) en de client kan direct met de lokale server communiceren. Hier is dus geen extra configuratie nodig.

/var/log/mpd/mpd.log	logbestand met informatie m.b.t. de muziekspeler
/etc/mpd.conf	Configuratiebestand van de muziekspeler
/var/lib/mpd/music	Locatie van de muziekbestanden (niet gebruikt voor ons project)

Test met oude Packard Bell NAS

De laatste directory zou je bv. kunnen laten verwijzen naar een NAS systeem waar al jouw favoriete muziek opgeslagen staat in mp3 formaat. Dit doe je dan via een symbolic link Indien je dit zou doen kan je via MPC deze liedjes dan automatisch inladen en afspelen.

Configuratie NAS

Packard bell NAS en CIFS server geconfigureerd op IP 192.168.0.5

Koppeling aan RPI

```
vim /etc/rc.local

## activate old SMB support
/sbin/modprobe cifs
echo 0 > /proc/fs/cifs/LinuxExtensionsEnabled
```

Bovenstaande hack is een workaround op de hopeloos verouderde Packard Bell NAS toch aan de praat te krijgen op onze RPI.

```
vim /etc/fstab

//192.168.0.5/public  /mnt/netstore  cifs
_netdev,rw,cache=loose,username=guest,password=,uid=1000,forceuid,gid=0,noforcegid,addr=192.
168.0.5,file_mode=0755,dir_mode=0755,nounix,serverino,rsize=16384,wsize=32768  0  0

mkdir /mnt/netstore

mount /mnt/netstore
```

Bovenstaande aanpassing koppelt de NAS aan onze RPI

Afspelen via MPC

```
ln -s /mnt/netstore/Muziek /var/lib/mpd/music/nas
mpc playlist
mpc clear
mpc playlist
mpc add nas/Kalimba.mp3
mpc update
mpc play
mpc status
mpc stop
```

Hierboven een voorbeeldje om een MP3 bestand af te spelen via de NAS en MPC

Aansturing van de GPIO Pinnen

Via python scripting kan je de RPI bibliotheken inladen die de pinnen op de RPI kunnen uitlezen. De tools en plugins heb ik op volgende manier geïnstalleerd:

sudo apt-get update && sudo apt-get upgrade
updaten van de RPI

sudo apt-get install python-smbus ipython bluetooth bluez-utils python-cwiidpython-scipy python-numpy python-pygame python-setuptools libsdl-dev
Installatie van python

sudo python
Testen van python

wget http://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/robot/resources/RPi.GPIO-0.3.1a.zip
unzip RPi.GPIO-0.3.1a.zip
cd RPi.GPIO-0.3.1a
sudo python setup.py install
Installatie van de RPI bibliotheken

Configuratie en scripting

loop.py

Dit script welke een oneindige loop start controleert om de 50ms indien er op onze retroradio van zender gewisseld wordt.

```
root@Kristof:~/python# cat loop.py
import os
import time
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)
GPIO.setup(18,GPIO.IN)
GPIO.setup(17,GPIO.IN)
GPIO.setup(27,GPIO.IN)
GPIO.setup(22,GPIO.IN)
GPIO.setup(23,GPIO.IN)
GPIO.setup(24,GPIO.IN)
GPIO.setup(25,GPIO.IN)

prev_input1 = 0
prev_input2 = 0
prev_input3 = 0
prev_input4 = 0
prev_input5 = 0
prev_input6 = 0
prev_input7 = 0
```

```
while True:
 #take a reading
 input = GPIO.input(18)
 #if the last reading was low and this one high, run btnscript
 if ((not prev_input1) and input):
  os.system("/root/knop1.sh")
 #update previous input
 prev_input1 = input

 #take a reading
 input = GPIO.input(17)
 #if the last reading was low and this one high, run btnscript
 if ((not prev_input2) and input):
  os.system("/root/knop2.sh")
 #update previous input
 prev_input2 = input

 #take a reading
 input = GPIO.input(27)
 #if the last reading was low and this one high, run btnscript
 if ((not prev_input3) and input):
  os.system("/root/knop3.sh")
 #update previous input
 prev_input3 = input

 #take a reading
 input = GPIO.input(22)
 #if the last reading was low and this one high, run btnscript
 if ((not prev_input4) and input):
  os.system("/root/knop4.sh")
 #update previous input
 prev_input4 = input

 #take a reading
 input = GPIO.input(23)
 #if the last reading was low and this one high, run btnscript
 if ((not prev_input5) and input):
  os.system("/root/knop5.sh")
 #update previous input
 prev_input5 = input

 #take a reading
 input = GPIO.input(24)
 #if the last reading was low and this one high, run btnscript
 if ((not prev_input6) and input):
  os.system("/root/knop6.sh")
 #update previous input
 prev_input6 = input

 #take a reading
 input = GPIO.input(25)
```

```
#if the last reading was low and this one high, run btnscript
if ((not prev_input7) and input):
  os.system("/root/shutdown.sh")
#update previous input
prev_input7 = input


#slight pause to debounce
time.sleep(0.05)
```

Indien er een andere GPIO Pin van laag naar hoog (3.3v) gaat d.m.v. het draaien aan de zenderkeuze knop wordt er ten laatste na 50ms opnieuw gecontroleerd welke GPIO pin hoog geworden is. Hierna wordt de overeenkomstige code verder uitgevoerd. (knop1, knop2….)

Voorbeeld van code onder knop1.sh

```
root@Kristof:~# cat knop1.sh
mpc clear
mpc load http://www.listenlive.eu/vrtradio1-high.m3u
mpc play
```

na uitvoering van bovenstaande code begint er een nieuwe livestream te spelen.

Voorbeeld code van het shutdown script

```
root@Kristof:~# cat shutdown.sh
shutdown -h now
```

vim /etc/rc.local

De toevoeging in dit bestand zorgt ervoor dat ons scriptje begint te lopen zodra de RPI opgestart is.

```
python /root/python/loop.py
exit 0
```