

Raspberry einrichten

Dienstag, 26. Juni 2018 20:16

Voraussetzungen

Raspberry PI
Netzteil für PI
microSD Karte (8GB oder mehr)
Netzwerkabel
Kartenleser für PC

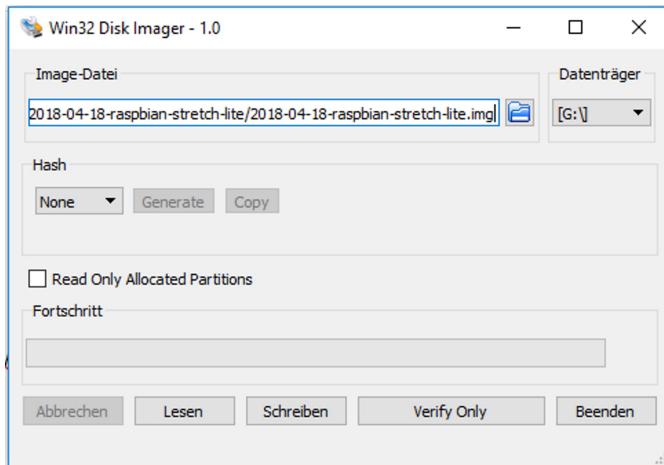
Software
SSH Client (Putty)
SFTP Client zum Datenaustausch (WinSCP)
Win32DiskImager

SD Karte vorbereiten

Raspberry Stretch Lite bzw. letzte Lite Version von raspberrypi.org herunterladen

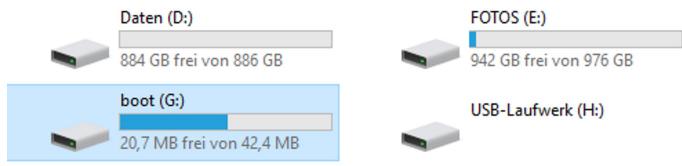


Nach dem herunterladen das Image entpacken und mittels Win32DiskImager auf die Karte schreiben. Dabei werden alle Daten auf der SD Karte überschrieben. Wird die Karte nicht als Datenträger erkannt, muss sie ggf. vorher formatiert werden.



Standardmäßig ist SSH an den Raspberry Images deaktiviert. Den Dienst benötigen wir aber um auf den PI zugreifen zu können. Da wir weder Bildschirm noch Tastatur anschließen wollen oder müssen, kann dies über einen simplen Trick erreicht werden.

Auf der SD Karte wurde eine Partition mit dem Name "boot" erstellt. Diese kann in Windows geöffnet werden. Um SSH zu aktivieren genügt es, eine leere Datei mit dem Namen ssh (ohne Dateiendung) hier zu erstellen

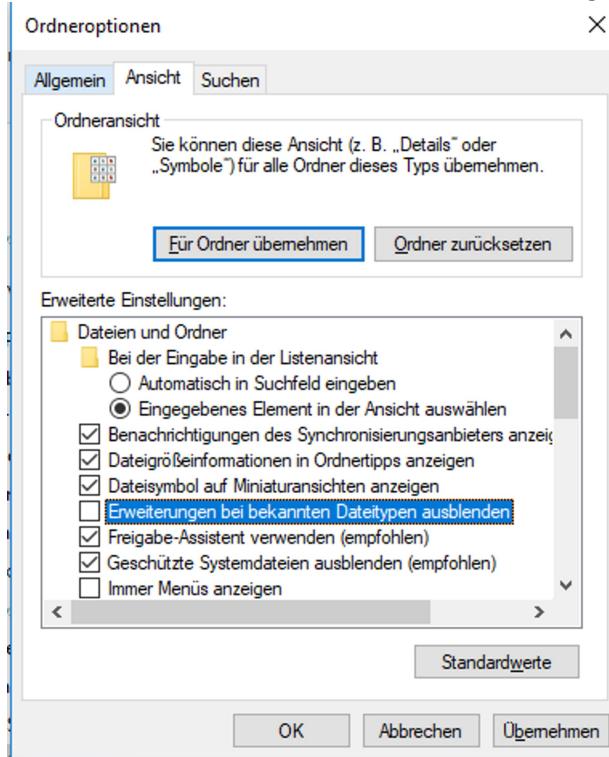


Windows wird meist eine Textdatei erstellen. Wir haben also eine Datei mit ssh.txt erstellt. Es muss aber eine Datei mit dem Namen ssh, ohne Dateiendung sein

start_db.elf	17.04.2018 11:50	ELF-Datei
start_x.elf	17.04.2018 11:50	ELF-Datei
ssh	26.06.2018 20:41	Textdokument

Dazu muss man in Windows einstellen, dass bekannte Dateinamen nicht ausgeblendet werden sollen. Dazu im Windows Explorer die Ordner- und Suchoptionen öffnen. Unter Windows 10 befinden sich diese im Reiter "Ansicht" unter "Optionen"

Hier deaktivieren wir den Haken bei "Erweiterungen bekannter Dateitypen ausblenden"



Die Datei wird nun als ssh.txt angezeigt und lässt sich umbenennen.

kernel7.img
LICENCE.broadcom
LICENSE.oracle
ssh.txt
start.elf

Beim Umbenennen kann die Dateierweiterung ".txt" gelöscht werden.

LICENSE.oracle	18.04.2018 01:08	ORACLE-Datei	19 KB
ssh	26.06.2018 20:41	Datei	0 KB
start.elf	17.04.2018 11:50	ELF-Datei	2.759 KB

Jetzt die Karte auswerfen und in den Raspberry einsetzen. Diesen nun mit Strom versorgen und das Netzwerkkabel einstecken.

Für die weitere Einrichtung wird nun eine SSH Verbindung zum Raspberry benötigt. Dies geschieht mit einem SSH Client, etwa Putty. Damit man sich mit dem PI verbinden kann, muss dieser zunächst

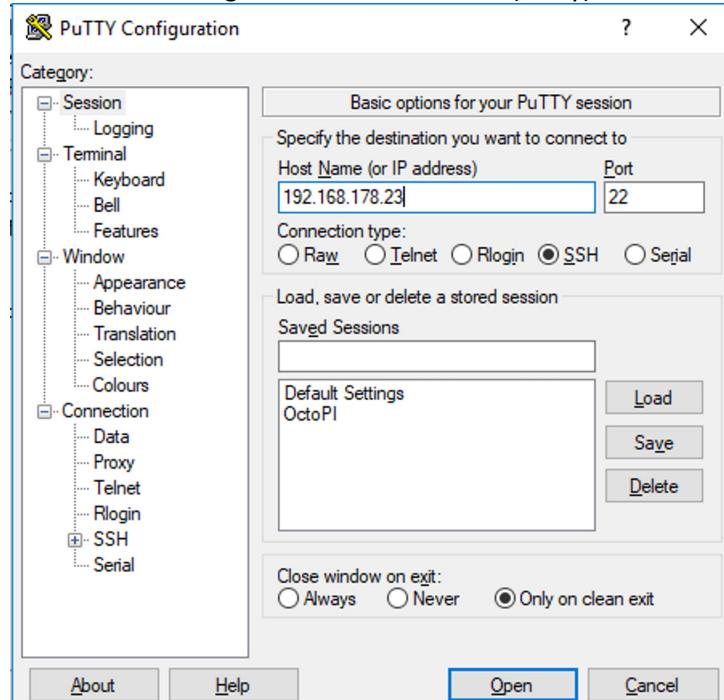
ins Netzwerk eingebunden werden. Da noch keine WLAN Verbindung besteh, muss dies zunächst über ein Netzwerkkabel erfolgen.

Der PI sollte nun vom Heimnetz eine IP Adresse erhalten. Meist geschieht dies durch den DHCP-Server, der auf dem DSL Router ausgeführt wird. Die vergebene IP Adresse kann am Router ausgelesen werden.

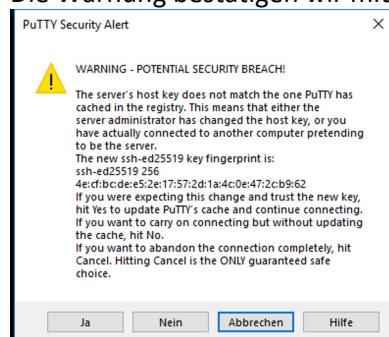
Es empfiehlt sich, im Router bzw. SHCP Server einzustellen, dass der Raspberry immer die gleiche IP Adresse erhält.

Systemkonfiguration

Für die Verbindung zum PI den SSH Client (Putty) öffnen und die IP Adresse eingeben



Die Warnung bestätigen wir mit "Ja"



Der PI erwartet nun von uns eine Anmeldung. Die initiale Anmeldung erfolgt mit dem Benutzer pi und dem Kennwort raspberry

A screenshot of a terminal window titled '192.168.178.23 - PuTTY'. The window shows a login prompt: 'login as: pi' followed by 'pi@192.168.178.23's password:'. A small green square icon is visible to the right of the password field.

Glückwunsch, die erste Hürde ist damit genommen

```
pi@raspberrypi: ~
login as: pi
pi@192.168.178.23's password:
Linux raspberrypi 4.14.34-v7+ #1110 SMP Mon Apr 16 15:18:51 BST 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi: ~ $
```

Als erstes sollte man das Kennwort ändern. Dies und weitere Einstellungen kann man komfortabel mit dem Programm raspi-config erledigen. In der Konsole rufen wir das Programm mit dem Kommando

```
sudo raspi-config
```

Auf Durch den Zusatz sudo teilen wir dem PI mit, dass wir als root (Administrator) aufrufen möchten. GGf. fragt uns der PI nun nach dem root Kennwort. Dies ist wieder raspberry (solange wir es nicht geändert haben)

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

Mit dem ersten Menüpunkt ändern wir das Kennowrt

```
Raspberry Pi 3 Model B Rev 1.2

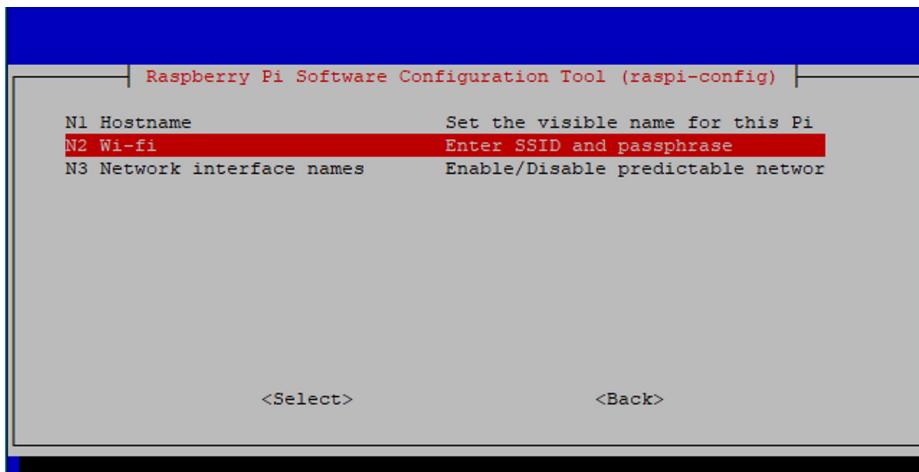
Raspberry Pi Software Configuration Tool (raspi-config)

1 Change User Password      Change password for the current u
2 Network Options           Configure network settings
3 Boot Options               Configure options for start-up
4 Localisation Options     Set up language and regional sett
5 Interfacing Options       Configure connections to peripher
6 Overclock                 Configure overclocking for your P
7 Advanced Options          Configure advanced settings
8 Update                     Update this tool to the latest ve
9 About raspi-config         Information about this configurat

<Select>                  <Finish>
```

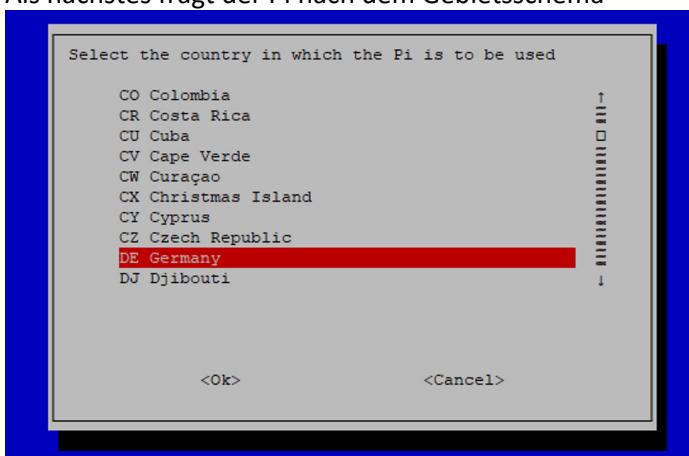
WLAN einrichten

Wo wir schon mal hier sind, können wir auch gleich das WLAN einrichten. Dies geschieht mit dem zweiten Menüpunkt "Network Options"

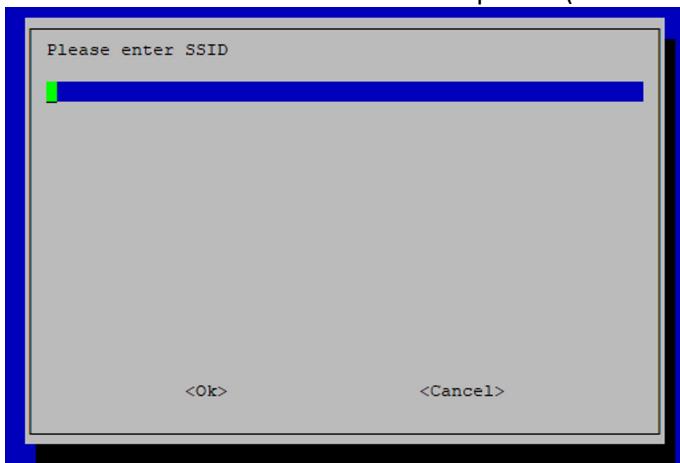


Und da wählen wir Wi-fi

Als nächstes fragt der PI nach dem Gebietsschema

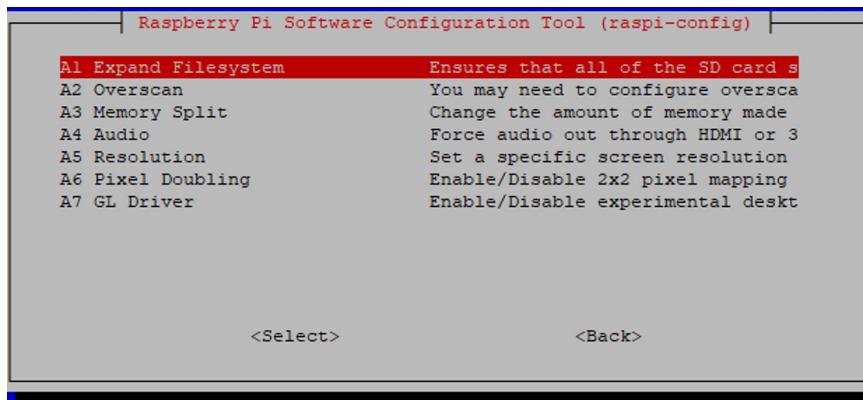


Und danach nach der SSID und der Passphrase (Kennwort) des WLANs

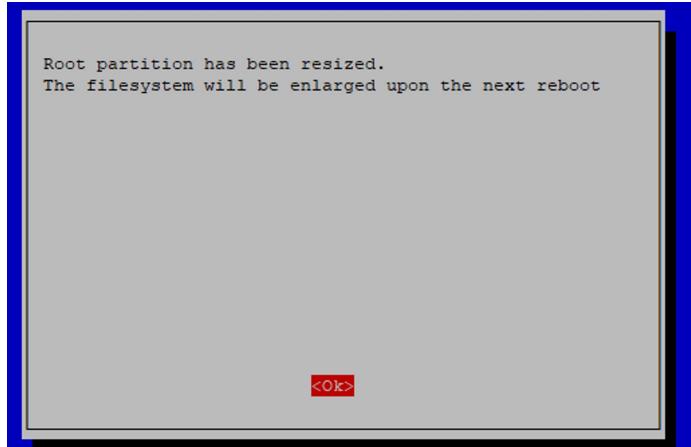


Nachdem die Einstellungen vorgenommen wurden, landet man wieder im Hauptmenü.

Ein letzter Schritt, bevor wir das Programm verlassen. Die Partition die auf der SD Karte erstellt wurde ist deutlich kleiner als die SD Karte. Somit ist wertvoller Speicherplatz verschenkt und unerreichbar. Dies können wir beheben, indem wir das Dateisystem auf den brachliegenden Teil erweitern. Dazu im Menü "Advanced Options" aufrufen und "A1 Expand filesystem" auswählen

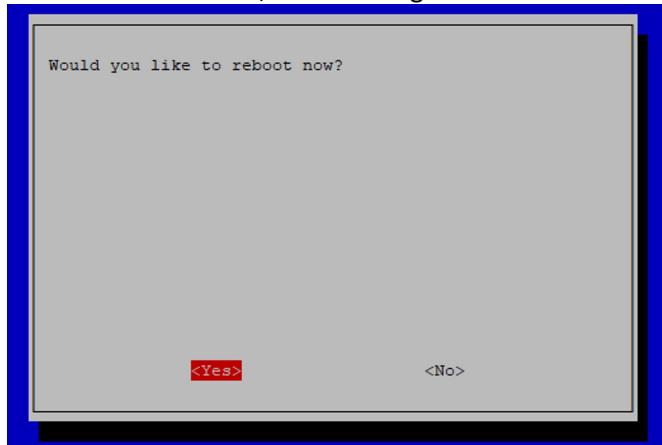


Nach erfolgreicher Erweiterung kommt diese Meldung, es muss also neugestartet werden



Wir landen wieder auf dem Hauptmenü und verlassen raspi-config mit <Finish>.

Zeit für einen Reboot, also bestätigen wir mit YES

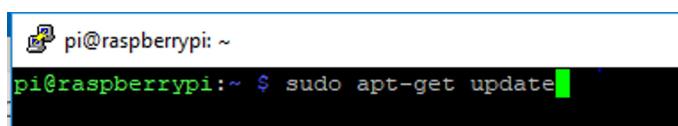


Nach dem Neustarten sollte auf dem WLAN Router nun eine neue WLAN Verbindung erscheinen. Auch hier empfiehlt es sich, die IP Adresse festzulegen, damit nicht beim nächsten anmelden ggf. eine andere IP Adresse vergeben wird.

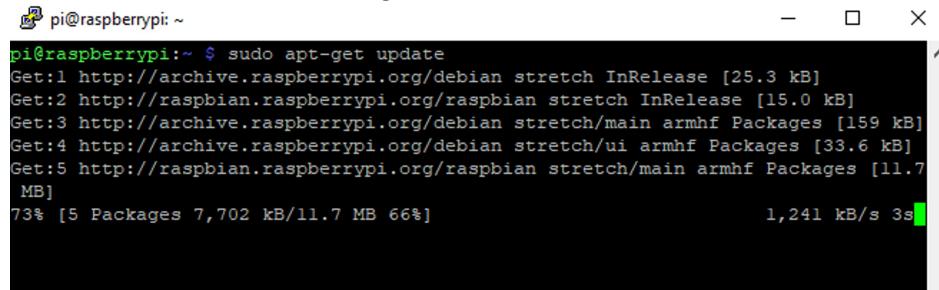
Systemupdate

Nachdem der PI wieder gestartet ist und wir neu angemeldet sind, wird es Zeit, einige Updates zu laden. Dazu müssen wir den Paketbaum neu aufbauen mit dem Kommando

```
sudo apt-get update
```



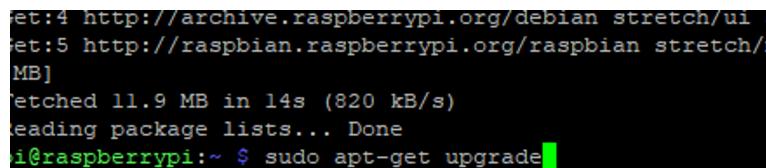
Nun warten wir, bis alles durchgelaufen ist



```
pi@raspberrypi:~ $ sudo apt-get update
Get:1 http://archive.raspberrypi.org/debian stretch InRelease [25.3 kB]
Get:2 http://raspbian.raspberrypi.org/raspbian stretch InRelease [15.0 kB]
Get:3 http://archive.raspberrypi.org/debian stretch/main armhf Packages [159 kB]
Get:4 http://archive.raspberrypi.org/debian stretch/ui armhf Packages [33.6 kB]
Get:5 http://raspbian.raspberrypi.org/raspbian stretch/main armhf Packages [11.7
 MB]
73% [5 Packages 7,702 kB/11.7 MB 66%] 1,241 kB/s 3s
```

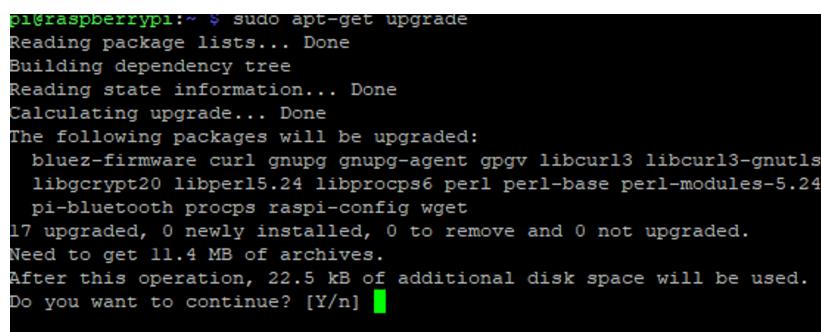
Jetzt kommen die eigentlichen Updates. Dies geschieht mit dem Kommando

```
sudo apt-get upgrade
```



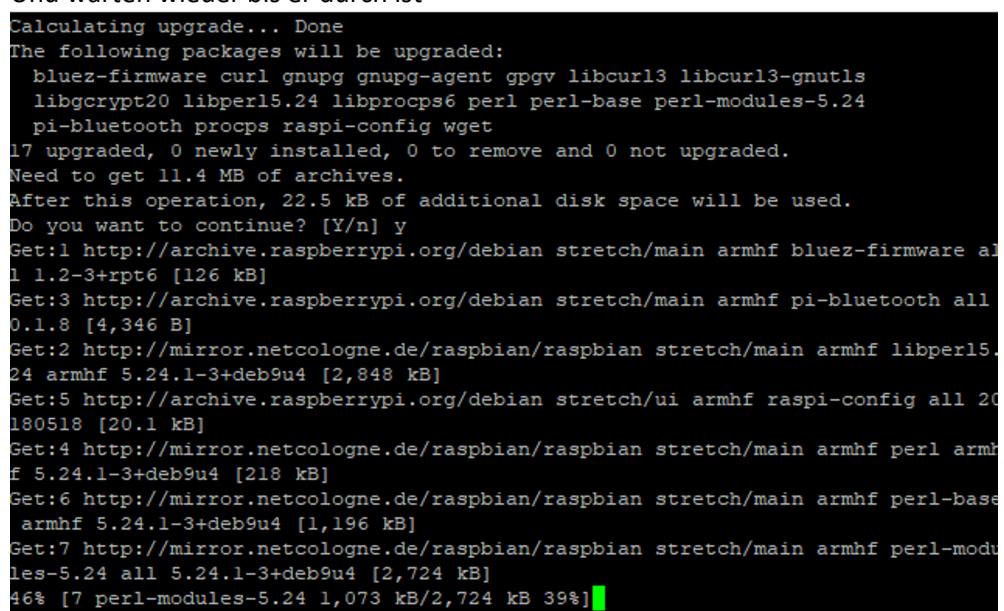
```
pi@raspberrypi:~ $ sudo apt-get upgrade
Get:4 http://archive.raspberrypi.org/debian stretch/ui
Get:5 http://raspbian.raspberrypi.org/raspbian stretch/
MB]
Fetched 11.9 MB in 14s (820 kB/s)
Reading package lists... Done
pi@raspberrypi:~ $ sudo apt-get upgrade
```

Wir haben ein frisches Image, somit sind kaum Updates vorhanden. Wir bestätigen die Installation mit "Y"



```
pi@raspberrypi:~ $ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  bluez-firmware curl gnupg gnupg-agent gpgv libcurl3 libcurl3-gnutls
  libgcrypt20 libperl5.24 libprocps6 perl perl-base perl-modules-5.24
  pi-bluetooth procps raspi-config wget
17 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 11.4 MB of archives.
After this operation, 22.5 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Und warten wieder bis er durch ist



```
Calculating upgrade... Done
The following packages will be upgraded:
  bluez-firmware curl gnupg gnupg-agent gpgv libcurl3 libcurl3-gnutls
  libgcrypt20 libperl5.24 libprocps6 perl perl-base perl-modules-5.24
  pi-bluetooth procps raspi-config wget
17 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 11.4 MB of archives.
After this operation, 22.5 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.raspberrypi.org/debian stretch/main armhf bluez-firmware all
1 1.2-3+rpt6 [126 kB]
Get:3 http://archive.raspberrypi.org/debian stretch/main armhf pi-bluetooth all
0.1.8 [4,346 B]
Get:2 http://mirror.netcologne.de/raspbian/raspbian stretch/main armhf libperl5.
24 armhf 5.24.1-3+deb9u4 [2,848 kB]
Get:5 http://archive.raspberrypi.org/debian stretch/ui armhf raspi-config all 20
180518 [20.1 kB]
Get:4 http://mirror.netcologne.de/raspbian/raspbian stretch/main armhf perl armhf
5.24.1-3+deb9u4 [218 kB]
Get:6 http://mirror.netcologne.de/raspbian/raspbian stretch/main armhf perl-base
armhf 5.24.1-3+deb9u4 [1,196 kB]
Get:7 http://mirror.netcologne.de/raspbian/raspbian stretch/main armhf perl-modu
les-5.24 all 5.24.1-3+deb9u4 [2,724 kB]
46% [7 perl-modules-5.24 1,073 kB/2,724 kB 39%]
```

Node-RED installieren

Node-RED installieren

Zum Installieren von Node-RED kann dieses Kommando verwendet werden

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/update-nodejs-and-nodered)
```

Aus <<https://nodered.org/docs/hardware/raspberrypi>>

```
[root login] pi@raspberrypi:~ $ bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/update-nodejs-and-nodered)
```

```
permitted by applicable law.  
Last login: Tue Jun 26 19:04:06 2018 from 192.168.178.30  
pi@raspberrypi:~ $ bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/update-nodejs-and-nodered)  
  
This script will remove versions of Node.js prior to version 6.x, and Node-RED and  
if necessary replace them with Node.js 8.x LTS (carbon) and the latest Node-RED  
from Npm.  
  
It also moves any Node-RED nodes that are globally installed into your user  
~/.node-red/node_modules directory, and adds them to your package.json, so that  
you can manage them with the palette manager.  
  
It also tries to run 'npm rebuild' to refresh any extra nodes you have installed  
that may have a native binary component. While this normally works ok, you need  
to check that it succeeds for your combination of installed nodes.  
  
To do all this it runs commands as root - please satisfy yourself that this will  
not damage your Pi, or otherwise compromise your configuration.  
If in doubt please backup your SD card first.  
  
Are you really sure you want to do this ? [y/N] ?
```

```
Running Node-RED install for user pi at /home/pi  
  
This can take 20-30 minutes on the slower Pi versions - please wait.  
  
Stop Node-RED ✓  
Remove old version of Node-RED ✓  
Remove old version of Node.js  
Install Node.js  
Clean npm cache  
Install Node-RED core  
Move global nodes to local  
Install extra Pi nodes  
Npm rebuild existing nodes  
Add menu shortcut  
Update systemd script  
Update update script  
  
Any errors will be logged to /var/log/nodered-install.log
```

Node-RED als Service einrichten

Damit Node-RED beim booten als Dienst automatisch gestartet wird, muss der Dienst eingerichtet werden

```
sudo systemctl enable nodered.service
```

Aus <<https://nodered.org/docs/hardware/raspberrypi>>

Motion installieren

Motion ist ein Programm, mit dem Bewegungen auf einem Kamerabild beobachtet werden. Eigentlich dient es dazu, bei einer Bewegung auf dem Kamerabild eine Aktion auszuführen, etwa ein Bild aufzunehmen und zu versenden. Motion bietet aber auch eine einfache Möglichkeit, das Kamerabild als Livestream zur Verfügung zu stellen.

Die Installation erfolgt mit

```
sudo apt-get install motion
```

Motion konfigurieren

Nun müssen wir einige Einstellungen in Motion durchführen. Diese sind z.B. den Stream im Netzwerk zur Verfügung zu stellen, die Web-Konfiguration im Netzwerk zur Verfügung zu stellen oder die Framerate des Streams einzustellen. Außerdem deaktivieren wir die Bewegungserkennung und das Abspeichern von Bildern. Gerade letzteres würde die SD Karte mit der Zeit vollschreiben.

Zur Konfiguration öffnen wir den Editor nano mot diesem Kommando

```
sudo nano /etc/motion/motion.conf
```

```
pi@raspberrypi:~ $ sudo nano /etc/motion/motion.conf
```

Als erstes Richten wir Motion als Dienst ein, dazu zu Beginn der Datei den daemon aktivieren (on)

```
# Daemon
#####
# Start in daemon (background)
daemon on
#
# File to store the process ID
process_id_file /var/run/motion/motion.pid
```

Zudem erhöhen wir die Anzahl der Frames, die pro Sekunde aufgenommen werden.

```
# Maximum number of frames to be captured per second.
# Valid range: 2-100. Default: 100 (almost no limit).
framerate 20
```

Bild sichern deaktivieren, indem output_pictures auf off gesetzt wird

```
#####
# Image File Output
#####

# Output 'normal' pictures when motion is detected (disabled by default)
# Valid values: on, off, first, best, center
# When set to 'first', only the first picture of an event is saved
# Picture with most motion of an event is saved when set to 'best'
# Picture with motion nearest center of picture is saved when set to 'center'
# Can be used as preview shot for the corresponding motion detection
output_pictures off

# Output pictures with only the pixels moving object (disabled by default)
output_debug_pictures off
```

Stream Framerate einstellen. Dazu den Wert stream_maxrate auf z.B. 10 erhöhen (10 Bilder pro Sekunde). Bei diesem Wert muss man etwas experimentieren, wie es am besten läuft. Hängt im wesentlichen von der Leistungsfähigkeit des PI und der Netzwerkverbindung ab. Hier aktivieren wir

auch gleich den Stream im Netzwerk, indem wir stream_localhost auf off setzen. Bitte beachtet, wenn die framerate niedriger eingestellt ist als die stream_maxrate, läuft das Video nicht flüssig

```
# rate given by stream_maxrate when motion is detected (def
stream_motion off

# Maximum framerate for stream streams (default: 1)
stream_maxrate 10

# Restrict stream connections to localhost only (default: on)
stream_localhost off
```

Die Einstellungen von Motion lassen sich auch über ein Webinterface einstellen. Dies ist besonders zum Testen praktisch. Die Weboberfläche ist normalerweise nicht über das Netzwerk aufrufbar. Da unser PI aber ohne Bildschirm läuft, muss die Weboberfläche im Netzwerk freigegeben werden, wenn man diese nutzen möchte. Dazu den Wert webcontrol_localhost auf off setzen

```
# Restrict control connections to localhost only
webcontrol_localhost off

# Output for http server, select off to choose ra
```

Die Einstellungen sichern wir mittels STRG+x und Y

```
# Authentication for the stream. Syntax username:password
# Default: not defined (Disabled)
; stream_authentication username:password
Save modified buffer? (Answering "No" will DISCARD changes.
Y Yes
N No          ^C Cancel
```

Das Kamerabild ist nach dem Start von Motion (entweder über sudo motion start oder nach Neustart) über den Port 8081 erreichbar. Die Weboberfläche zur Konfiguration über Port 8080. Angenommen, der Raspberry PI hat vom Router die IP Adresse 192.168.178.10 erhalten, kann man mit Eingabe der Adresse 192.168.178.10:8081 im Browser das Kamerabild prüfen

Damit Motion auch wirklich beim booten startet, muss eine weitere Datei angepasst werden

```
sudo nano /etc/default/motion
```

```
GNU nano 2.7.4          File: /etc/default/motion          Modified

# set to 'yes' to enable the motion daemon
start_motion_daemon=yes
```

Motion wird als Benutzer "motion" ausgeführt. Das Logfile wird unter /var/log/motion/motion.log abgelegt. Der Benutzer motion hat aber keine Berechtigung, darauf zuzugreifen. Ein Start als Dienst ist damit nicht möglich.

Wir können einfach das Verzeichnis löschen mit

```
sudo rm -rf /var/log/motion
```

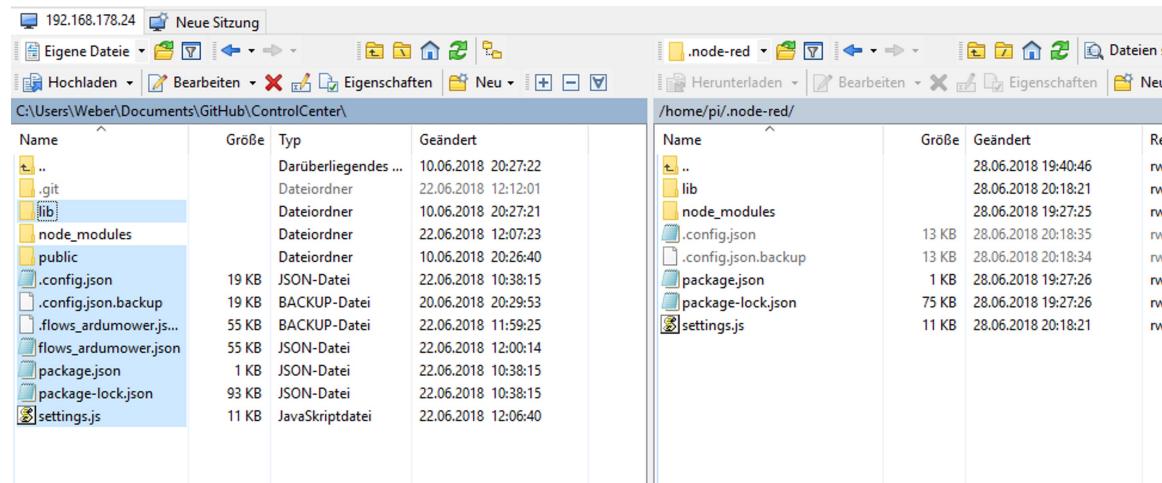
Beim Starten des Dienstes wird das Verzeichnis mit den korrekten Rechten erstellt. Man kann

natürlich auch den Owner des Verzeichnisses anpassen oder die Gruppenrechte setzen. Ich finde es so aber einfacher

Control Center einrichten

Nun ist es fast geschafft. Jetzt ist es Zeit, das Ardumower Control Center einzurichten. Dazu nutzen wir am besten eine SFTP Client wie WinSCP. Zuvor müssen wir aber die Dateien von Github heruntergeladen habe.

Anschließend kopieren wir alle Verzeichnisse und Dateien, mit Ausnahme von dem Verzeichnis node_modules auf unser .node-red Verzeichnis unter /home/pi

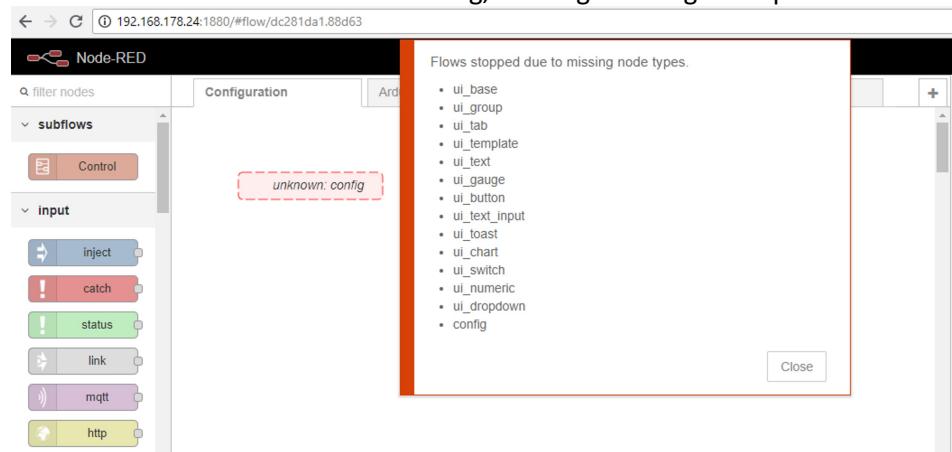


Dabei überschreiben wir alle alten Dateien

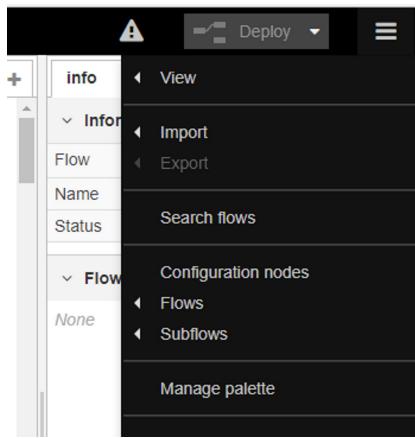
Jetzt starten wir den PI einmal durch mittels `sudo reboot` oder `sudo shutdown -r now`.

Nach dem Neustart wird es Zeit, die Node-RED Entwicklungsumgebung zum ersten mal aufzurufen. Dies geschieht mit dem Browser durch den Aufruf der Adresse <ip des PI>:1880, also etwa 192.168.178.10:1880

Wir erhalten direkt eine Fehlermeldung, da einige wichtige Komponenten nicht installiert sind



Dies holen wir sogleich nach, indem wir im Menü auf "Manage Pallette" gehen



Und unter dem Reiter "Install" die Komponenten installieren

User Settings

View	Nodes	Install
Keyboard		sort: a-z recent
search modules		
Palette	1487 modules available	

Node-red-dashboard

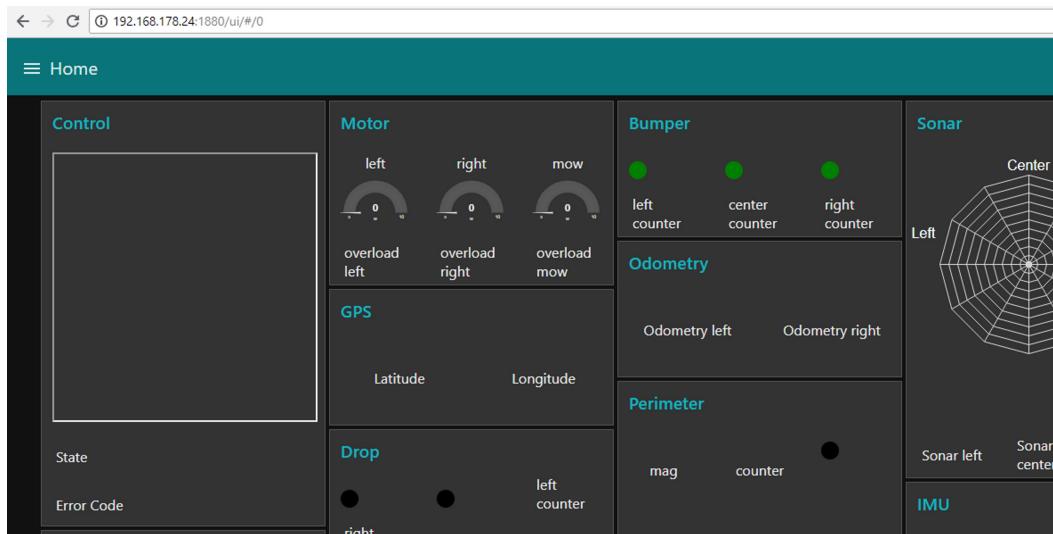
View	Nodes	Install
Keyboard		sort: a-z recent
search dashboard		
Palette	6 / 1487	
<ul style="list-style-type: none"> node-red-contrib-netatmo-dashboard Get JSON payload for NetAtmo dashboard. See http://netatmo.com, https://dev.netatmo.com/doc 0.3.2 3 months ago install node-red-contrib-polymer Polymer based dashboard UI for Node-RED 0.0.21 7 months ago install node-red-contrib-uibuilder Easily create web UI's for Node-RED using any (or no) front-end library. JQuery and normalize.css included by default but change as desired. 1.0.6 6 months ago install node-red-dashboard A set of dashboard nodes for Node-RED 2.9.4 1 month ago install 		

Sowie node-red-contrib-config

View	Nodes	Install
Keyboard		sort: a-z recent
search config		
Palette	2 / 1487	
<ul style="list-style-type: none"> node-red-configurable-ping A Node-RED node which takes input and pings a remote server. 1.0.1 2 years, 1 month ago install node-red-contrib-config A node-red node to set flow and global context values at start up. 1.1.2 6 months ago install 		

Am besten starten wir danach den Raspberry einmal neu

Nun ist das Control Center über die Adresse <ip des PI>:1880/ui erreichbar, also etwa 192.168.178.10:1880/ui



Glückwunsch, das Control Center ist eingerichtet.

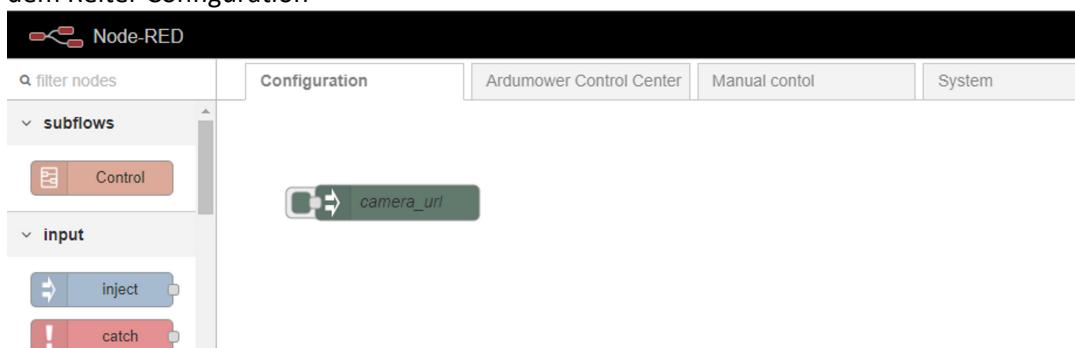
Raspberry mit Arduino verbinden

Den Raspberry PI verbinden wir mit einem einfachen USB Kabel mit dem Arduino. Wenn ein Arduino DUE zum Einsatz kommt, verbinden wir den PI mit dem Programming Port. Die Firmware des Arduinower kann auch über den Native USB Port geflasht werden, so dass man keine USB Kabel umstecken muss.

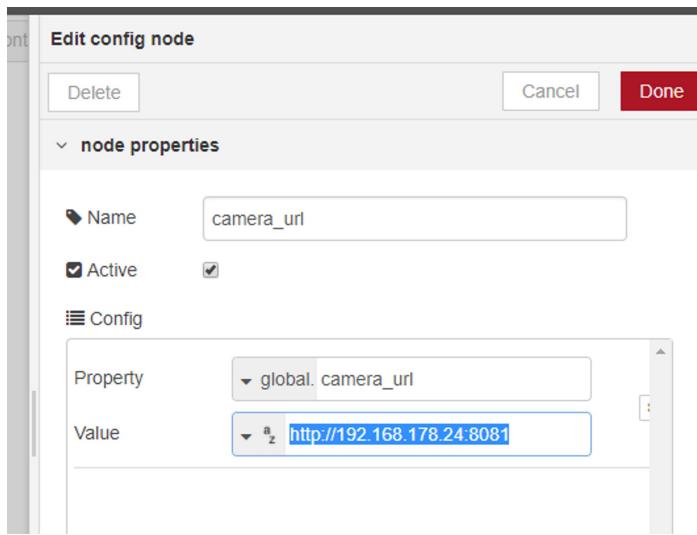
Control Center konfigurieren

Kamera URL

Die Adresse der Kamera stellen wir im Konfiguration-Node camera_url ein. Diesen finden wir auf dem Reiter Configuration

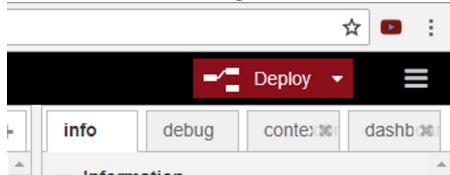


Auf camera_url einen Doppelklick machen. Hier kann die Adresse hinterlegt werden.



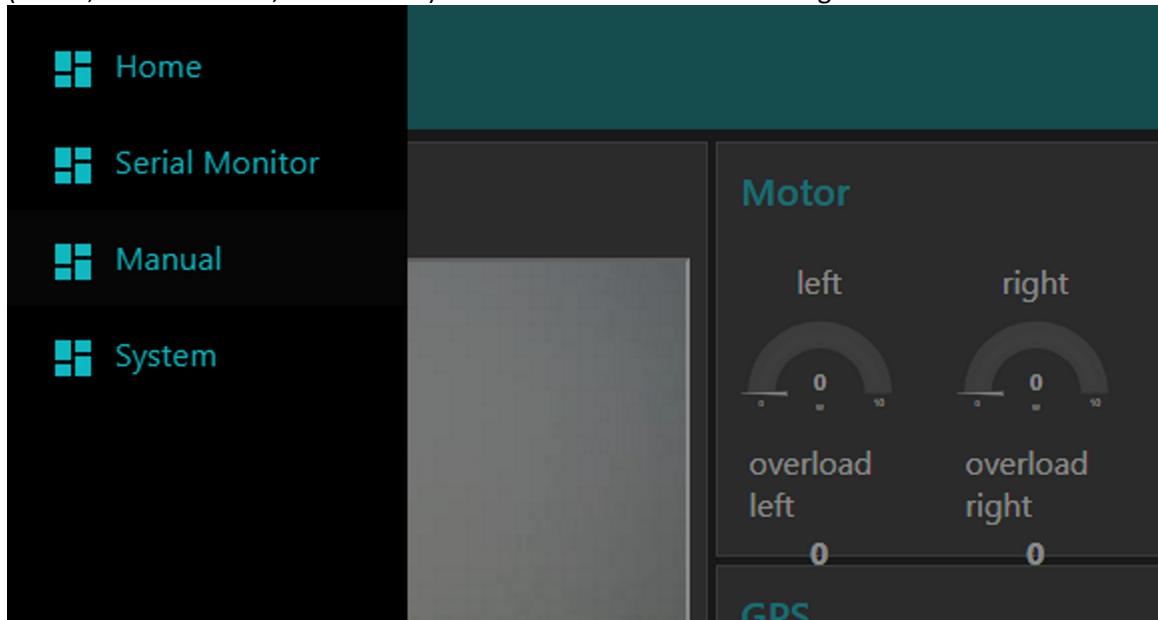
Anschließend auf Done klicken.

Damit die Änderung wirksam wird, müssen wir einmal auf den Button "Deploy" klicken



Das wars. Das Control Center ist jetzt einsatzbereit.

Abschließend noch ein paar Kommentare zum Control Center. Dieses besteht aus mehreren Seiten (Home, Serial Monitor, Manual etc.) welche im Menü oben links ausgewählt werden können.

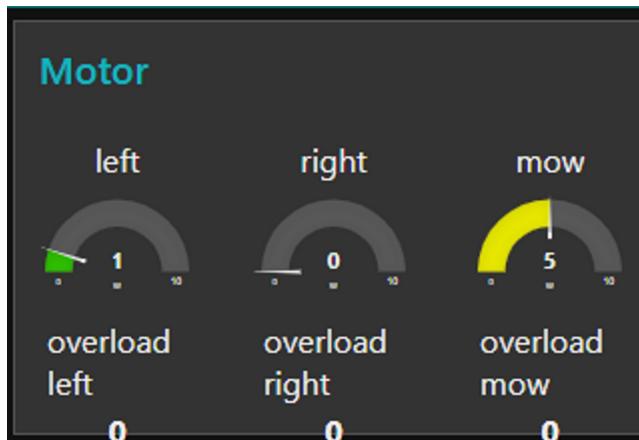


Auf Touch Geräten wie Smartphones geht es auch mit wischen. Die Verbindung zum Mower kann im Serial Monitor eingesehen werden. Hier müssten ständig neue Daten einlaufen. Bitte beachtet, dass das Control Center ein eigenes Nachrichtenformat verwendet (RMCS). Dieses muss in der ArduMower Firmware aktiviert werden. Implementiert ist das Protokoll in dem ArduMower Fork von mir (PaddyCube), ist aber hoffentlich bald auch im Master verfügbar.

Einstellen der Steuerelemente

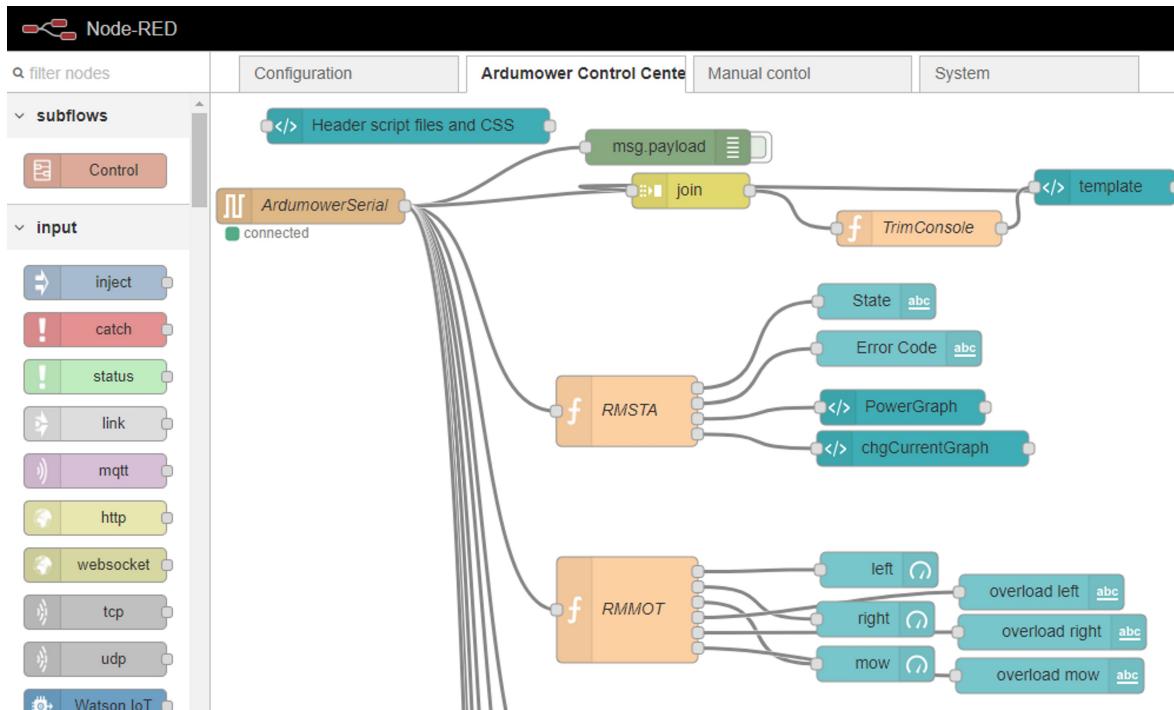
Im Control Center werden diverse Steuerelemente, etwa Zeiger oder Charts angezeigt. Diese haben fest hinterlegte Grenzwerte, die bei eurem Mower vermutlich nicht passen. Am Beispiel der

Motorleistung zeige ich noch schnell, wie diese angepasst werden können.



Meine Mähmotoren benötigen kaum mehr als 10W. Das liegt daran, da der Mower zwei Stück hat und jeder Motor eine kleine Messerscheibe. Manchmal steigt die Leistungsaufnahme aber auf 12W, so dass die Anzeige schon am Anschlag ist, bevor das Maximum erreicht wird. Man sieht dann die Anzeige ganz rechts stehen auf 10W, der Wert in der Mitte der Anzeige zeigt aber 12W an.

Um das zu ändern rufen wir wieder die Entwicklungsumgebung von Node-RED auf (Port 1880) und wechseln in den Reiter "ArduMower Control Center". Hier werden alle eingehenden Nachrichten verarbeitet. Wir suchen nun den Abschnitt, bei dem die Nachricht RMMOT verarbeitet wird.



Folgt man den Linien an der rechten Seite von RMMOT (das ist der Ausgang des Nodes), finden wir die drei Grafiken (left, right, mow). Durch Doppelklick auf mow können wir die Anzeige verändern. Unter Range tragen wir den gewünschten neuen Maximalwert ein

Edit gauge node

Delete Cancel Done

node properties

Group: Motor [Home]

Size: 2 x 2

Type: Gauge

Label: mow

Value format: {{value}}

Units: W

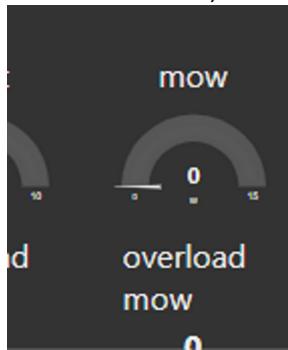
Range: min 0 max 10

Colour gradient: 

Sectors: 0 ... optional ... optional ... 10

node settings

Und klicken dann auf Done. Damit die Änderung wirksam wird, müssen wir einmal auf "Deploy" klicken. Das wars, die Anzeige geht jetzt von 0 bis 15.



Warnung manueller Modus

Das Control Center bietet auch die Möglichkeit, den Mower manuell zu steuern. Dazu gibt es eine Art Software Joystick (das graue Feld auf der Seite "Manual"). Damit der Mower manuell bewegt werden kann, muss zuerst der Ring MANUAL auf Grün gesetzt werden (anklicken).

Der Manuelle Modus ist experimentell. Benutzt ihn nur, wenn ihr direkt beim Mower steht. Die Steuerung ist sehr ungenau, verzögert und reagiert oftmals nicht. Bitte verletzt euch nicht und sonst niemanden. Passt auf, dass der Mower nicht abhaut, auf die Straße läuft, in den Teich fällt oder was auch immer. Besonders, verwendet den Modus niemals über das Internet.

Der Mower hält nicht an, wenn der Bumper auslöst oder der Perimeter verlassen wird. Bricht die Verbindung zwischen PC uns Raspberry ab, fährt er weiter bis der Akku leer ist. Ich möchte in den Verkehrsnachrichten keine Warnung vor Mährobotern auf der Fahrbahn hören.

Ich meine das wirklich ernst, passt auf damit. Ich werde hier nachlegen und es etwas sicherer gestalten müssen Aber auch danach bleibt es gefährlich und sollte niemals über das Internet von z.B.: dem Arbeitsplatz aus genutzt werden.

Raspberry PI Kommandos

Den Raspberry PI bitte nicht einfach vom Strom trennen. Er möchte heruntergefahren werden. Dies

geschieht ebenfalls über die SSH Konsole mit dem Befehl

```
sudo shutdown -h now
```

-h steht für "halt" also ausschalten, now für sofort. Ohne now plant der PI einen Job, der nach einiger Verzögerung herunterfährt.

Neustarten geht mittels

```
sudo shutdown -r now
```

Wobei -r für reboot steht.

Das Control Center verfügt auch über Buttons um den Raspberry herunterzufahren und um den Node-RED Dienst neu zu starten. Alles bedient sich unter der Seite "System". Hier kann auch eingestellt werden, welche Events der Mower senden soll und in welchen Intervallen aktuelle Daten gesendet werden sollen (Sonar, Motor, Perimeter usw.)

