



Software gsv-6toWAMP

Quellcode: <https://github.com/me-systeme/gsv-6ToWAMP>

Stand der Dokumentation: 01.09.2016

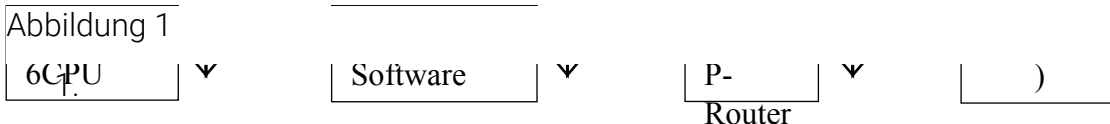
Zugangsdaten der Demo-Installation

Wlan Name	ME_AP
Passwort des Zugangspunktes	MeTestAP
IP Adresse (im Browser des Clients eingeben)	192.168.9.1:8000
SSH Zugang	user: pi passwort: metest12

Aufbau der Software

Die Software ist modular aufgebaut, somit können Komponenten einfacher ausgetauscht werden.

Die Software kommuniziert einerseits mit dem GSV-6CPU-Modul über die Serielle Schnittstelle und andererseits über das WAMP-protokoll via WebSockets mit der Crossbar-Instanz (WAMP-Router) (siehe Abbildung 1). Die Crossbar-Instanz kommuniziert mit den verbundenen Clients. Ein Clienten kann ein Browser sein, der die GUI darstellt oder ein beliebiges Programm welches den WAMP-Standard als Client unterstützt.



Der konkrete Aufbau soll nun beschrieben werden. Das Hauptprogramm Instanziert mit Hilfe der Twisted-Bibliothek einen Webserver der für das Ausliefern der Webseite (GUI), der CSV-Datei und der Log-Dateien verantwortlich ist. Der Port des Webserver kann über die Konfigurations-Datei gesetzt werden. Danach wird mit der Komponente ApplicationRunner, die von dem Autobahn.ws-Framework bereitgestellt wird und einen WAMP-Clients darstellt, die WAMP_Component instanziiert und blockierend gestartet. Die ApplicationRunner-Komponente baut auch die WAMP-Verbindung zu dem WAMP-Server auf. Ist dies erfolgreich, wird die FrameRouter-Komponente instanziiert, außerdem versucht das Programm die konfigurierte Serielle-Schnittstelle zu öffnen. Damit die Serielle-Schnittstelle geöffnet werden kann, muss der SerialPort-Komponente eine Protocol-Klasse übergeben werden. Diese Protocol-Klasse kann selbst implementiert werden und verwaltet die serielle Kommunikation. Ist die Serielle-Verbindung erfolgreich aufgebaut, wird der FrameRouter gestartet. Die FrameRouter-Komponente verteilt die über die Serielle-Schnittstelle

eingehenden Daten auf weitere Komponenten. Messwert-Frames werden die MessFrameHandler-Komponente und Antwort-Frames werden an die AntwortFrameHandler-Komponente weitergeleitet. Des Weiteren instanziiert der FrameRouter die EventHandler-Komponente, diese Komponente verarbeitet die eingehenden Events vom WAMP-Router. Dies wird in Abbildung 2 dargestellt.

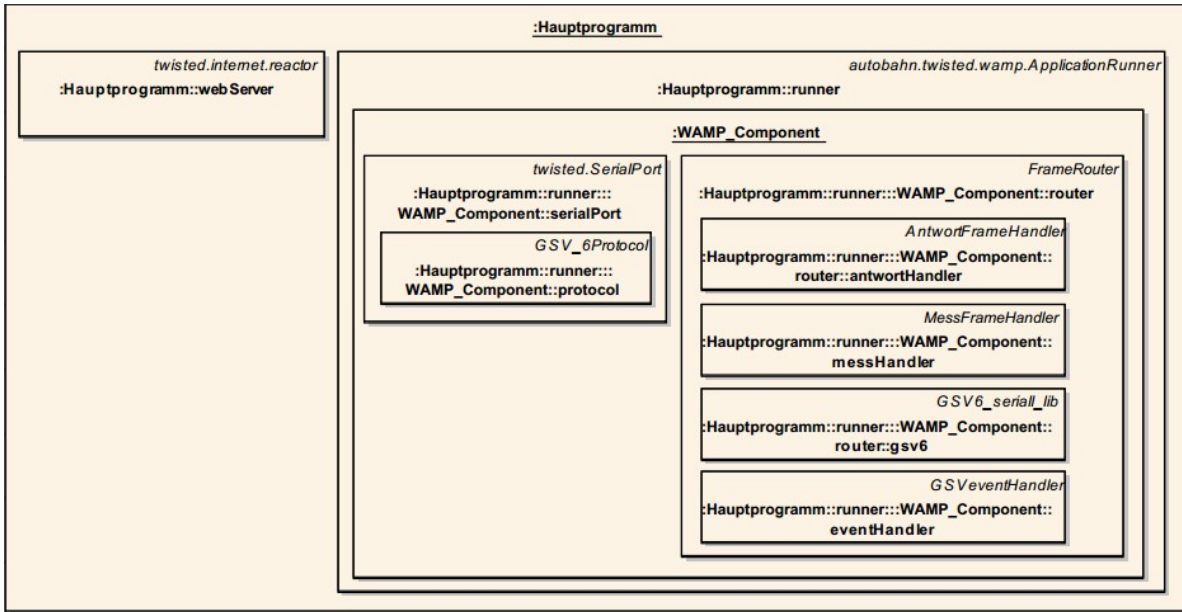


Abbildung 2 - Programmaufbau

Beschreibung

Die Software kommuniziert mit dem GSV-6CPU-Modul über die Serielle Schnittstelle und versteht dementsprechend das Protokoll des Moduls. Somit können Konfigurationseigenschaften ausgelesen und gesetzt werden und Messungen gestartet und gestoppt werden. Damit diese Daten bzw. die Konfiguration einfach zugänglich sind, werden sie über eine Webseite bereitgestellt.

Über „Messungen“ (index.html) kann eine Messung gestartet oder gestoppt werden, außerdem kann der „setZero“-Befehl auf alle Kanäle angewendet werden. Die Live-Messergebnisse werden auf dieser Seite dargestellt

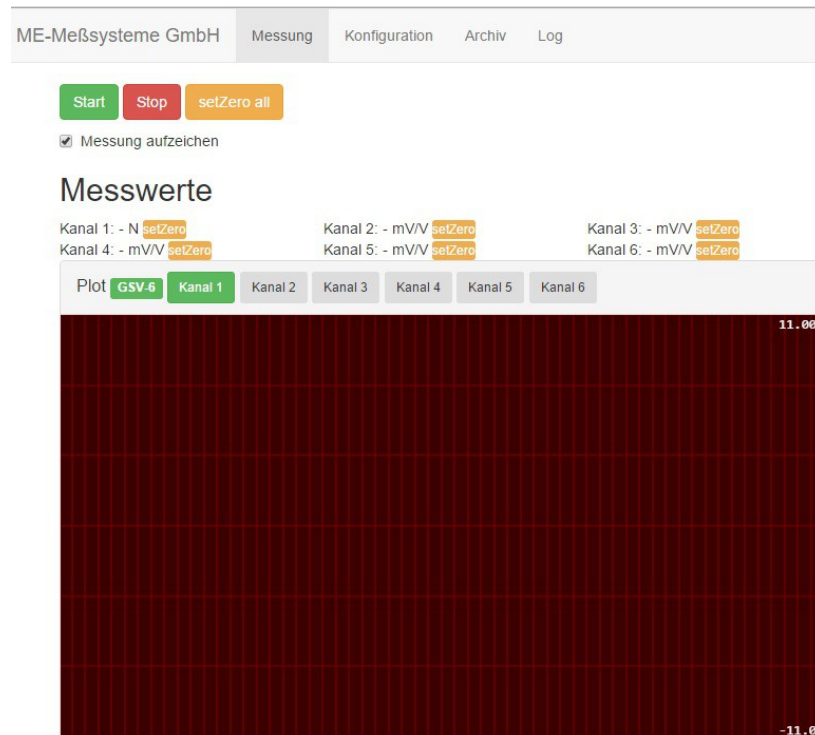


Abbildung 3 - Messung (index.html)

Die Konfiguration des GSV-6CPU erfolgt über die Konfigurations-Seite (config.html). Hier können diverse Informationen über die Konfiguration eingesehen und geändert werden. Geändert werden können:



- Datenrate (hierfür ist aktuell ein Power-Cycle des GSV-6CPU-Moduls nötig)
- UserOffset
- UserScale
- Die jeweilige Einheit separat je Kanal
- Eingangstype (für alle Kanäle gleich)
- UnitText Slot 0 und 1 (wird mit aktueller Firmware nicht unterstützt)
- Browser Uhrzeit und Datum auf dem RPi als System-Zeit setzen
- System Neustart (RPi-Reboot)

ME-Meßsysteme GmbH Messung **Konfiguration** Archiv Log

Modul Info

Eigenschaft	Wert	Eigenschaft	Wert	Eigenschaft	Wert
genereller Schreibschutz	deaktiviert	output-Datentype	float32	Betriebstunden	ERR_CMD_NOTIMPL
Schnittstellen spzif. Schreibs.	deaktiviert	aktive Kanäle	6	Firmware Version	Major: 3 Minor: 5
Modell	GSV-6	SerialNo	4294967295	Datenrate	10,00 Hz ✓

Modul Settings

Offset, Scaling, Einheiten, Eingangstype

UserOffset		UserScale		Einheiten		Eingangstype	
Kanal	Wert	Kanal	Wert	Kanal	Einheit	Kanal	Eingangstype
1	0,0000 ✓	1	10,0000 ✓	1	N ✓	Alle	4 mV/V ✓
2	0,0000 ✓	2	1,0000 ✓	2	mV/V ✓		

Abbildung 4 - Konfiguration Teil 1

Offset, Scaling, Einheiten, Eingangstype

UserOffset		UserScale		Einheiten		Eingangstype	
Kanal	Wert	Kanal	Wert	Kanal	Einheit	Kanal	Eingangstype
1	0,0000 ✓	1	10,0000 ✓	1	N ✓	Alle	4 mV/V ✓
2	0,0000 ✓	2	1,0000 ✓	2	mV/V ✓		
3	0,0000 ✓	3	1,0000 ✓	3	mV/V ✓		
4	0,0000 ✓	4	1,0000 ✓	4	mV/V ✓		
5	0,0000 ✓	5	1,0000 ✓	5	mV/V ✓		
6	0,0000 ✓	6	1,0000 ✓	6	mV/V ✓		

Danger-Zone





SaveAll
Browser Zeit setzten
reboot System

User-UnitText

set UnitText slot 0
set UnitText slot 1

Abbildung 5 - Konfiguration Teil 2

Aufgezeichnete Messungen können unter Archiv (archiv.html) wiedergefunden werden. Hier werden die CSV-Dateien aufgelistet und es können folgende Aktionen mit den Dateien ausgeführt werden:

-  Löschen
-  Download
-  Reduzieren
-  Ansehen

Löschen

Die CSV-Datei wird gelöscht (Achtung, es wird keine Kopie vorgehalten).

Download

Hierdurch wird eine Kopie heruntergeladen.

Reduzieren

Reduzieren bedeutet, dass die Anzahl der Einträge verringert wird. Außerdem werden die zusammengefassten Werte in einer neuen CSV-Datei abgespeichert, die den Präfix „red_“ tragen. Das Reduzieren geschieht, indem zeitlich zusammenliegende Werte zusammengefasst und nur noch deren Minimal- und Maximalwert abgelegt werden. Die Reduzierung kann nur auf CSV-Dateien angewendet werden, die mehr als 4000 Einträge haben.

Ansehen

Mit dem Klick auf Ansehen, erreicht der User die Detailansicht (messung_detail.html) der gewählten CSV-Datei. Hier werden die Daten der CSV-Datei dargestellt.

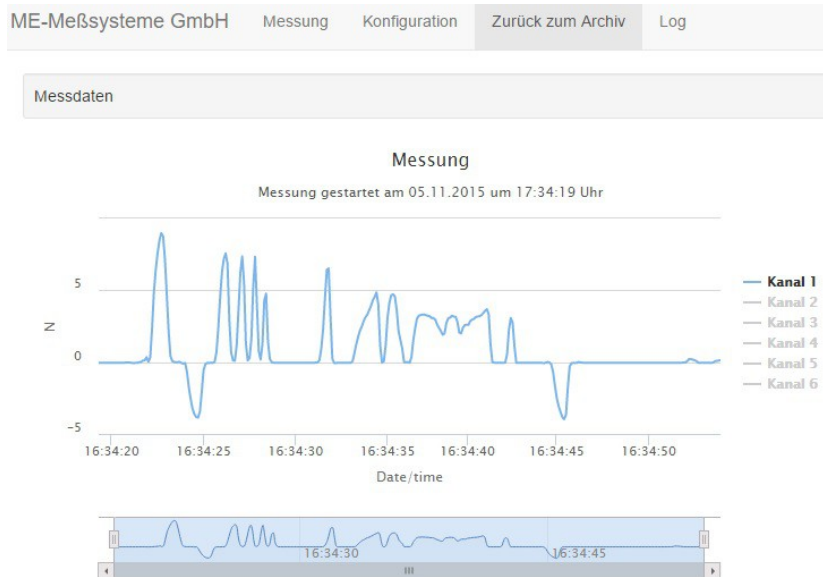


Abbildung 6 - Messung Detailansicht

Achtung: Wenn CSV-Dateien zu groß werden, kann es sein, dass Browser mobiler Geräte die Webseite nicht darstellen. An dieser Stelle sollte eine reduzierte Version der CSV-Datei erstellt werden, um die Darstellung zu ermöglichen. Die reduzierten Versionen der CSV-Dateien sind immer so groß, dass es zu keinen Darstellungsproblemen auf jeglichen Geräten kommen sollte.

CSV-Format

Jede vom Programm erstellte CSV-Datei hat einen Datei-Kopf (Header) und ein bis n Einträge. Der Datei-Kopf hat folgendes Format:

```
timestamp,channel10 [mV/V], channel11 [mV/V], channel12 [mV/V], channel13 [mV/V], channel14 [mV/V], channel15 [mV/V]
```

Abbildung 7 - CSV-Header

Der Datei-Kopf sagt in diesem Fall aus, dass die erste Spalte einen Zeitstempel enthält. Die Folgenden Spalten enthalten die Zahlenwerte der einzelnen Kanäle des GSV-6CPU. Zuzüglich enthalten die Kanäle noch die eingestellten Einheiten des jeweiligen Kanals in eckigen Klammern.

Der Zeitstempel ist intern ein Unix-Zeitstempel und wird in folgendem Format in der CSV-Datei abgelegt: „<Jahr>-<Monat>-<Tag> <Stunden>:<Minuten>:<Sekunden>.<Millisekunden>“

Die Messwerte werden in Float (Textuell) abgelegt. Ein Beispiel für eine komplette Zeile:

```
2015-10-13 18:32:43.278,1.0499999523162842,3.2044434192357585e-05,0.00012817773676943034,-6.408886838471517e-05,0.0,0.0001602221745997667
```



Abbildung 8 - CSV-Zeile

Eingesetzte Dritt-Software/Bibliotheken

An dieser Stelle wird ein Überblick über die eingesetzte Software gegeben.

PyPy

PyPy ist ein Interpreter für Python-Code, der selbst in Python geschrieben ist. Durch den integrierten JIT-Compiler können bestimmte Teile eines Python-Codes um ein Vielfaches schneller ausgeführt werden.

Twisted (Lib)

Twisted ist ein Event getriebenes Netzwerk Software Framework, welches durch PyPy beschleunigt werden kann. Für die GSV-6-Applikation stellt das Framework den Webserver bereit und stellt zusammen mit „Autobahn.ws“, die Websocket- und Serielle-Verbindungen bereit.

Autobahn.ws (Lib)

Autobahn.ws ist ein Software Framework, welches das Websocket-Standard und zusätzlich ein Subprotokoll, welches auf den Websocket-Standard aufbaut, mit dem Namen WAMP implementiert und bereitstellt. WAMP stellt Remote Procedure Calls und Publish- & Subscribe-Service via Websockets bereit.

Crossbar.io

Highcharts (Highstock)

Installation

Die Installation wird beispielhaft an einem Raspberry PI 2 mit vorinstalliertem Raspbian¹ (Version 2015-05-05) beschrieben. Es wird davon ausgegangen, dass der EDIMAX EW-7811Un oder Wi-Pi WLAN-USB-Adapter bereits vorhanden und eingesteckt ist. Außerdem wird vorausgesetzt, dass der User „pi“ genutzt wird. Es ist vorgesehen, dieser Installationsanleitung linear zu folgen.

¹ <https://www.raspbian.org/>



Installationsvorgehensweise

Als erstes wird PyPy installiert, eine Variante, die Python-Code in Teilen beschleunigt und der Paketmanager pip². Anschließend werden folgende Pakete installiert, die von dem Programm benötigt werden:

- Twisted
- AutobahnPython
- PySerial
- numpy

Damit USB-Speicher automatisch im Linux-Betriebssystem „gemounted“ werden, wird das Programm „usbmount“ benötigt.

Anschließend wird crossbar.io installiert, start-up-scripte für crossbar sowie das Hauptprogramm serial2ws installiert und der WLAN-Adapter so konfiguriert, dass er als Accesspoint fungiert.

Wenn eine Konfigurationsdatei mit nano geöffnet wird, kann sie mit der Tastenkombination strg+o gespeichert werden und mit der Tastenkombination strg+x geschlossen werden.

Außerdem werden alle Befehle in der Konsole eingegeben.

Für die Installation werden zwei Ordner (downloads und install) benötigt, die im Vorfeld angelegt werden müssen:

```
mkdir ~/downloads
mkdir ~/install
```

PyPy

PyPy ist vorinstalliert in Raspian (2015-05-05), allerdings in einer älteren Version (2.2.1), deswegen sollte eine neuere Version (4.0.0) installiert werden. Dies geschieht mit folgenden Befehlen in der Konsole

```
cd ~/downloads
wget https://bitbucket.org/pypy/pypy/downloads/pypy-4.0.0-linux-armhf-raspbian.tar.bz2
cd ~/install
tar xvjf ../downloads/pypy-4.0.0-linux-armhf-raspbian.tar.bz2
echo "export PATH=\${HOME}/install/pypy-4.0.0-linux-armhf-raspbian/bin:\${PATH}"
>> ~/.profile
source ~/.profile
```

Achtung: Zeile 6 und 7, sowie Zeile 8 und 9 sind jeweils eine Zeile.

² <https://github.com/pypa/pip>



Pip

```
cd ~/downloads
wget https://bootstrap.pypa.io/get-pip.py
pypy get-pip.py
```

Twisted

Twisted wird nicht mit Hilfe von pip installiert, da für die Installationen eine Zeile in der Setup.py auskommentiert werden muss. Kommentiert wird in Python mit der #.

```
cd ~/downloads
wget https://pypi.python.org/packages/source/T/Twisted/Twisted-15.4.0.tar.bz2
cd ~/install
tar xvjf ../downloads/Twisted-15.4.0.tar.bz2
```

Achtung Zeile 2 und 3 sind eine Zeile

```
cd Twisted-15.4.0
nano setup.py
line 63 auskommentieren mit # -> #conditionalExtensions=getExtensions(),
speichern und schließen (strg+o -> strg+x)
pypy setup.py install
```

PySerial

```
pip install pyserial
```

Autobahn Framework

```
pip install autobahn
```

numpy

Da in diesem Projekt PyPy zum Einsatz kommt, muss eine angepasste Variante von numpy mit folgenden Befehl installiert werden (Achtung: dies dauert eine Weile >15 Min.):

```
cd ~/downloads
pypy -m pip install git+https://bitbucket.org/pypy/numpy.git
```

Wenn ein Standard Python installiert wäre, würde folgender Befehl ausreichen:

```
pip install numpy
```

usbmount

```
sudo apt-get install usbmount
```

Anschließend muss die Konfiguration angepasst werden. Dazu wird die die Zeile FS_MOUNTOPTIONS in der usb,mount.conf angepasst:

```
sudo nano /etc/usbmount/usbmount.conf
```



```
FS_MOUNTOPTIONS="-fstype=vfat,gid=users,dmask=0007,fmask=0117"  
speichern und schließen und neustarten  
sudo reboot
```

crossbar.io

Für die Installation von crossbar.io werden noch ein paar Abhängigkeiten im Linux-System benötigt. Diese werden mit dem apt-get-Tool installiert und anschließend wird crossbar via pip installiert:

```
sudo apt-get install build-essential libssl-dev libffi-dev python-dev  
pip install crossbar
```

Setzen der Zeitzone

Damit serial2ws.py die korrekte Zeit nutzt, muss die Zeitzone noch konfiguriert werden.

```
cd ~/  
echo "TZ='Europe/Berlin';" >> ~/.profile  
echo "export TZ" >> ~/.profile  
source ~/.profile
```

Projektdateien von github.com holen

```
cd ~/  
git clone https://github.com/me-systeme/gsv-6ToWAMP.git
```

Konfigurationsdatei

Die defaults.conf ist die Konfigurationsdatei, in der folgenden Optionen gesetzt werden können:

Option	Beschreibung
router	Legt die Webadresse für die Corssbar-Instanz fest (z.B. ws://127.0.0.1:8001/ws/)
web	Setzt den Port, über den die Konfigurationswebseite erreichbar ist (z.B. 8000)
boot_wait	Setzt eine Zeit x, die nach dem Programmstart gewartet wird (z.B. 10 == 10 Sek.)
port	Setzt den Seriellen-Port (z.B. /dev/ttyAMA0 oder 4 (für Comport 5 unter Windows))
baudrate	Setzt die Baudrate des Seriellen-Ports
csvpath	Setzt den Speicherort, wo die CSV-Dateien abgelegt werden
log	Setzt das Log-Level (Verfügbar: CRITICAL, ERROR, WARNING, INFO, DEBUG, TRACE)

Tipp: Der csvpath muss vorhanden sein, sonst stoppt das Programm mit einer Fehlermeldung.



Die Optionen können bei einem Start aus der Kommandozeile heraus auch als Parameter übergeben werden. Beispiel:

```
python serial2ws.py --baud=115200 --port /dev/ttyAMA0
```

Start-up-Skript

Start-up-Skripte aus dem scripts Ordner nach /etc/init.d kopieren.

```
cd ~/gsv-6ToWAMP/scripts
sudo cp crossbar /etc/init.d/
sudo cp serial2ws /etc/init.d/
```

Skripte als Ausführbar markieren:

```
sudo chmod +x /etc/init.d/crossbar
sudo chmod +x /etc/init.d/serial2ws
```

Damit crossbar und serial2ws während des Boot-Vorgangs gestartet werden, könnte dies über init.d geschehen lassen. Allerdings beschreibt diese Anleitung, wie beide Programme mit Hilfe der rc.local-Datei gestartet werden. Dazu die /etc/rc.local-Datei mit nano öffnen:

```
sudo nano /etc/rc.local
```

Und die Datei folgendermaßen anpassen:

```
service networking restart

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi

/etc/init.d/crossbar start
/etc/init.d/serial2ws start
```

Nun werden crossbar und serial2ws während des Hochfahrens des Raspberry Pi gestartet. Außerdem können crossbar.io und serial2ws als daemon gestartet und gestoppt werden, mit folgenden Befehlen:

```
sudo /etc/init.d/crossbar start
sudo /etc/init.d/crossbar stop

sudo /etc/init.d/serial2ws start
sudo /etc/init.d/serial2ws stop
```



WLAN-Accesspoint einrichten

Damit der Raspberry Pi einen WLAN-Accesspoint bereit stellt, wird ein Programm benötigt, das diese Funktion bereit stellt. Außerdem wird ein DHCP-Server benötigt, der den verbunden Klienten eine IP zuweist. Dieser Abschnitt der Beschreibung ist aus zwei Quellen entstanden (Quelle1³ und Quelle2⁴).

In dieser Anleitung werden hostapd für den Accesspoint und dnsmasq als DHCP-Server genutzt. Allerdings muss darauf geachtet werden, dass der WLAN-Adapter mit dem hostapd-Daemon kompatibel⁵ ist.

Diese Anleitung beschreibt die Installation für einen kompatiblen WLAN-Adapter (z.B. [Wi-Pi](#) mit dem Ralink RT5370 Chipset) und für einen WLAN-Adapter mit dem Realtek RTL8188CUS Chip.

Für WLAN-Adapter mit Realtek RTL8188 Chip, kommt eine gepatchte Version von hostapd zum Einsatz, die in zwei Varianten bezogen werden kann: Erstens Es wird die vor-kompilierte Variante aus dem git-Repo genutzt oder zweitens es wird die gepachtete-Version auf dem Raspberry Pi gebaut (kompiliert). In beiden Fällen sollte jedoch hostap und dnsmasq über apt-get installiert werden. Anschließend wird die installierte hostapd-Version durch die gepatchte Version ersetzt.

Installation

```
sudo apt-get install firmware-ralink hostapd wireless-tools dnsmasq iw
```

Die nachfolgenden Schritte sind nur nötig, falls ein WLAN-Adapter mit dem Realtek RTL8188 Chip zum Einsatz kommt. Wenn der Adapter kompatibel ist, springen Sie bitte zur DHCP-Server Konfiguration.

Option 1, vor-kompilierte Version nutzen

Hierzu muss die vor-kompilierte binary in den /usr/sbin-Ordner kopiert werden. Damit bei Probleme, die originale Version erhalten bleibt, wird sie umbenannt.

```
cd /usr/sbin
sudo mv /usr/sbin/hostapd /usr/sbin/hostapd.bak
sudo cp gsv-6ToWAMP/binary/hostapd hostapd
sudo chown root:root hostapd
sudo chmod 755 hostapd
```

³ https://wiki.debianforum.de/WLAN-Access-Point_mit_hostapd_und_USB-Stick

⁴ <http://elinux.org/RPI-Wireless-Hotspot>

⁵ <https://w1.fi/hostapd/>



Option 2, gepachtete hostapd kompilieren

Dafür müssen als erstes alle Abhängigkeiten (Lib's) installiert werden:

```
sudo apt-get install libnl-3-dev libnl-genl-3-dev
```

Anschließend wird das git-Repo ausgechecked

```
cd ~/downloads  
git clone https://github.com/lostincynicism/hostapd-rtl8188
```

Im nächsten Schritt wird die gepachtete Variante von hostapd erstellt:

```
cd ~/downloads/hostapd-rtl8188/hostapd  
make
```

Abschließend muss das binary wie in Option1 in den /usr/sbin Ordner kopiert werden:

```
cd /usr/sbin  
sudo mv /usr/sbin/hostapd /usr/sbin/hostapd.bak  
cp ~/downloads/hostapd-rtl8188/hostapd/hostapd hostapd  
sudo chown root:root hostapd  
sudo chmod 755 hostapd
```

Konfiguration

DHCP-Server

Da jetzt die benötigten Tools installiert wurden, wird nun dnsmasq und anschließend hostapd konfiguriert.

Im Vorfeld sollte die mitgelieferte dnsmasq Konfigurations-Datei mit folgenden Kommando gesichert werden:

```
mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
```

Danach die neue dnsmasq.conf mit nano öffnen:

```
sudo nano /etc/dnsmasq.conf
```

Folgender Text wird eingefügt:

```
interface=wlan0  
no-dhcp-interface=eth0  
dhcp-range=interface:wlan0,192.168.9.2,192.168.9.30,infinite
```

Die Datei speichern und schließen.



WLAN-Adapter

Somit ist die Konfiguration für den DHCP –Server abgeschlossen. Es folgt die Konfiguration des WLAN-Adapters. Während des Boot-Vorgangs muss dem WLAN-Adapter, eine feste IP-Adresse zugewiesen werden. Die IP-Adresse hängt von dem IP-Adressenbereich des zu vor konfigurierten DHCP-Servers ab. In diesem Fall wird die IP-Adresse 192.168.9.1 gewählt, wofür wird die /etc/network/interfaces-Datei geöffnet und bearbeitet wird:

```
sudo nano /etc/network/interfaces
```

Anschließend die Zeile „iface wlan0 inet dhcp“ folgendermaßen abändern:

```
iface wlan0 inet static
    address 192.168.9.1
    netmask 255.255.255.0
    broadcast 192.168.9.255
```

Außerdem die folgenden Zeilen mit dem #-Zeichen auskommentieren:

```
#allow-hotplug wlan0
#wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
#iface wlan0 inet manual
```

Nun sollten folgende Zeilen ans Ende der interfaces-Datei angehängt werden:

```
# restart hostapd and dnsmasq
    up service hostapd restart
    up service dnsmasq restart
```

Abschließend speichern und die Datei schließen.

Nun folgt die Konfiguration des hostap-daemons. Dieses Programm richtet mit Hilfe des WLAN-Adapters einen Accesspoint ein, mit dem sich Geräte wie mit einem Standard-WLAN-Router verbinden können. Dafür muss die Konfigurationsdatei folgendermaßen angepasst werden.

Konfigurations-Datei öffnen:

```
sudo nano /etc/hostapd/hostapd.conf
```

Anschließend muss die Konfiguration folgendermaßen abgeändert werden:

Hierbei ist zu beachten, dass Sie ein selbst gewähltes Passwort benutzen, indem Sie <accesspoint_passwort> durch Ihr Passwort ersetzen. Außerdem muss die Zeile „driver“ mit dem Wert „rtl1871xdrv“ angepasst werden, falls Sie einen WLAN-Adapter mit Realtek Chip nutzen

```
interface=wlan0
driver=nl80211
ssid=ME_AP
hw_mode=g
```



```
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=<accesspoint_passwort>
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Anschließend muss dem hostap-daemon mitgeteilt werden, welche Konfiguration wären des Boot-Vorgangs geladen werden soll. Dazu wird die Datei /etc/default/hostapd geöffnet und bearbeitet:

```
sudo nano /etc/default/hostapd
```

Sie sollte folgendermaßen abgeändert werden:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Es muss kein automatischer Start der Programm hostapd und dnsmasq eingerichtet werden, da diese mit dem Netzwerk zusammen gestartet werden (siehe Anpassung der /etc/network/interfaces-Datei).

Abschließen das System neustarten:

```
sudo reboot
```

Jetzt dem Raspberry Pi genug Zeit geben zum Hochfahren und es sollte der Accesspoint mit dem Namen „ME_AP“ sichtbar sein. Wenn Sie sich mit einem Gerät verbunden haben, sollten Sie die GSV-6-Webseite über <http://192.168.9.1:8000> erreichen.

Teilweise gab es das Problem, das während des Boot-Vorganges dem WLAN-Adapter(Interfaces) keine IP zugewiesen wurde. Dies kann mit der Anpassung der /etc/ifplugd Konfiguration umgangen werden, in dem die Datei folgendermaßen geändert wird⁶:

Die Datei mit nano öffnen:

```
sudo nano /etc/defaults/ifplugd
```

und folgende Zeilen anpassen:

```
INTERFACES="eth0"
HOTPLUG_INTERFACES="eth0"
ARGS="-q -f -u0 -d10 -w -I"
SUSPEND_ACTION="stop"
```

⁶ <http://rpi.vypni.net/wifi-ap-rt5370-on-raspberry-pi/>