# Software gsv-6toWAMP

source code: https://github.com/me-systeme/gsv-6ToWAMP/tree/gsv-6ToWAMP-for-RASPI3

Updated: 2018-03-20

## Access data for the demo installation

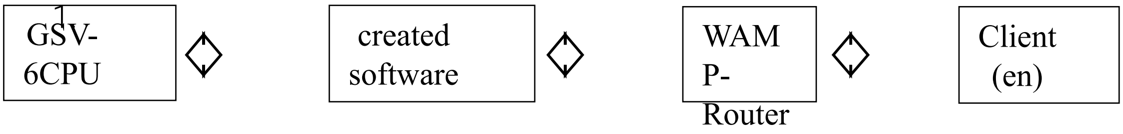
| | |
|---|---|
| Wlan Name | ME_AP |
| Password of the access point | MeTestAP |
| IP adress (im Browser des Clients eingeben) | 192.168.9.1:8000 |
| SSH Access | user: pi<br>passwort: metest12 |

## Structure of the software

The software is modular, which means that components can be exchanged more easily.

The software communicates, on the one hand, with the GSV-6CPU module via the serial interface and, on the other hand, via the WAMP protocol via WebSockets with the crossbar instance (WAMP router) (see Figure 1). The crossbar instance communicates with the connected clients. A client can be a browser that represents the GUI, or any program that supports the WAMP standard as a client.

Figure 1



The concrete structure should now be described. The main program uses the twisted library to instantiate a web server responsible for delivering the web page (GUI), the CSV file, and the log files. The port of the web server can be set via the configuration file. The WAMP_Component is then instantiated and blocked using the ApplicationRunner component, which is provided by the Autobahn.ws framework and represents a WAMP client. The ApplicationRunner component also builds the WAMP connection to the WAMP server. If this is successful, the FrameRouter component is instantiated and the program also tries to open the configured serial interface. In order to open the serial interface, the SerialPort component must be passed a Protocol class. This protocol class can be implemented itself and manages the serial communication. If the serial connection is successful, the FrameRouter is started. The FrameRouter component distributes the data received via the serial interface to other components. Measured-value frames will be passed to the FrayHandler component and response frames to the ReplyFrameHandler component. Des Weiteren instanziiert der FrameRouter die EventHandler-Komponente, diese

Komponente verarbeitet die eingehenden Events vom WAMP-Router. Furthermore, the FrameRouter instantiates the EventHandler component, which processes the incoming events from the WAMP router. This is shown in Figure 2.

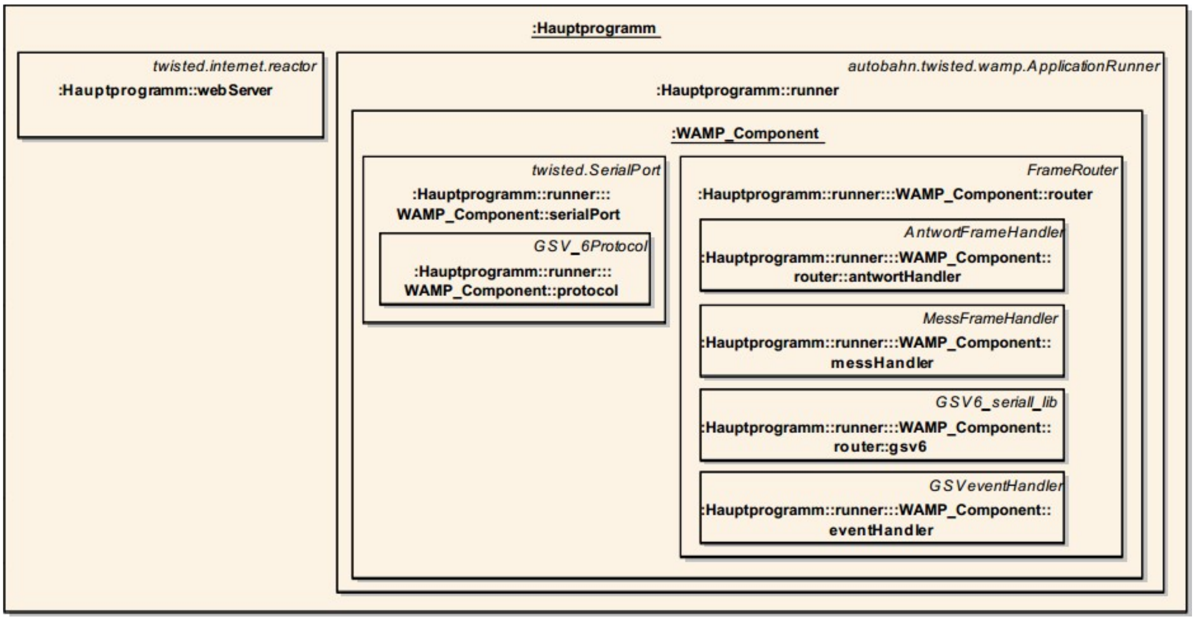

:Hauptprogramm
twisted.internet.reactor
:Hauptprogramm::webServer
autobahn.twisted.wamp.ApplicationRunner
:Hauptprogramm::runner
:WAMP_Component
twisted.SerialPort
:Hauptprogramm::runner:::WAMP_Component::serialPort
GSV_6Protocol
:Hauptprogramm::runner:::WAMP_Component::protocol
FrameRouter
:Hauptprogramm::runner:::WAMP_Component::router
AntwortFrameHandler
:Hauptprogramm::runner:::WAMP_Component::router::antwortHandler
MessFrameHandler
:Hauptprogramm::runner:::WAMP_Component::messHandler
GSV6_seriall_lib
:Hauptprogramm::runner:::WAMP_Component::router::gsv6
GSVeventHandler
:Hauptprogramm::runner:::WAMP_Component::eventHandler

Figure 2 - Program structure

## Description

The software communicates with the GSV-6CPU module via the serial interface and understands the protocol of the module accordingly. Thus configuration properties can be read out and set and measurements can be started and stopped. In order to make this data or the configuration easily accessible, they are provided via a web page.

A measurement can be started or stopped via "Measurements" (index.html), and the "setZero" command can be applied to all channels. The live measurement results are displayed on this page.

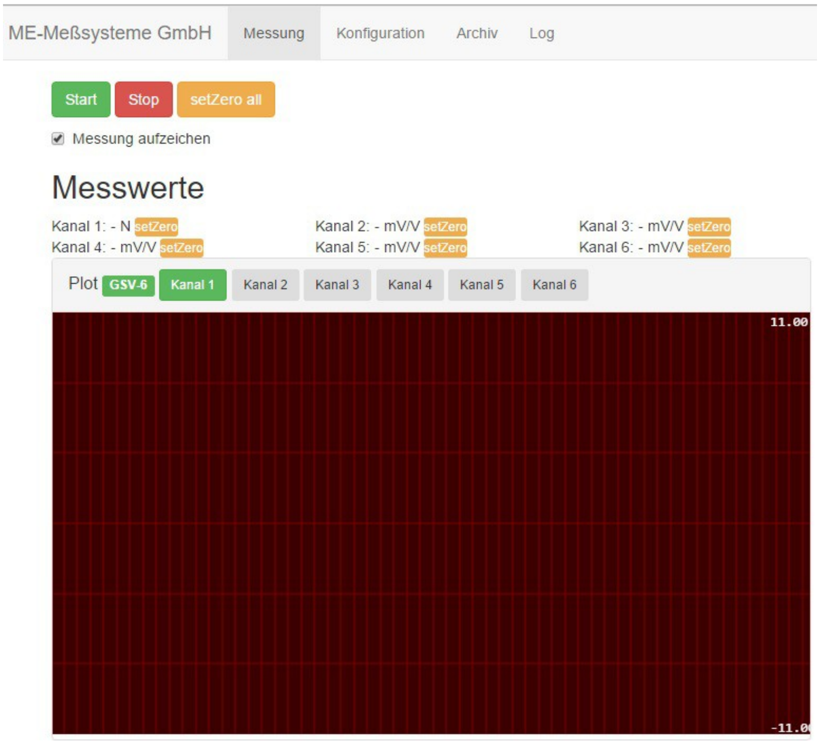
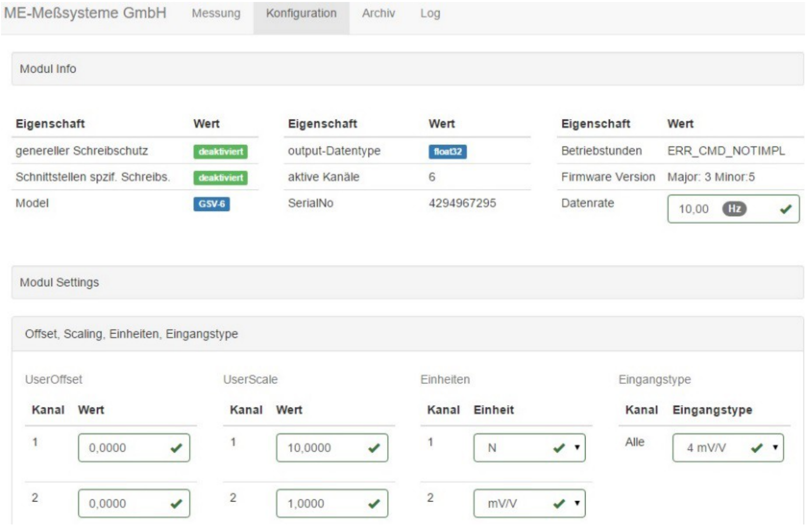


Figure 3 - Measurement (index.html)

The GSV-6CPU is configured via the configuration page (config.html). Various information about the configuration can be viewed and changed here. The following can be changed:

- Data rate (a power cycle of the GSV-6CPU module is currently required)
- UserOffset
- UserScale
- The respective unit separately per channel
- Input type (same for all channels)
- UnitText Slot 0 and 1 (Is not supported with current firmware)
- set the browser time and date on the RPi as the system time
- restart the system (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 |
| Model | GSV-6 | SerialNo | 4294967295 | Datenrate | 10,00 Hz ✓ |

Modul Settings

Offset, Scaling, Einheiten, Eingangstype

UserOffset

| Kanal | Wert |
|---|---|
| 1 | 0,0000 ✓ |
| 2 | 0,0000 ✓ |

UserScale

| Kanal | Wert |
|---|---|
| 1 | 10,0000 ✓ |
| 2 | 1,0000 ✓ |

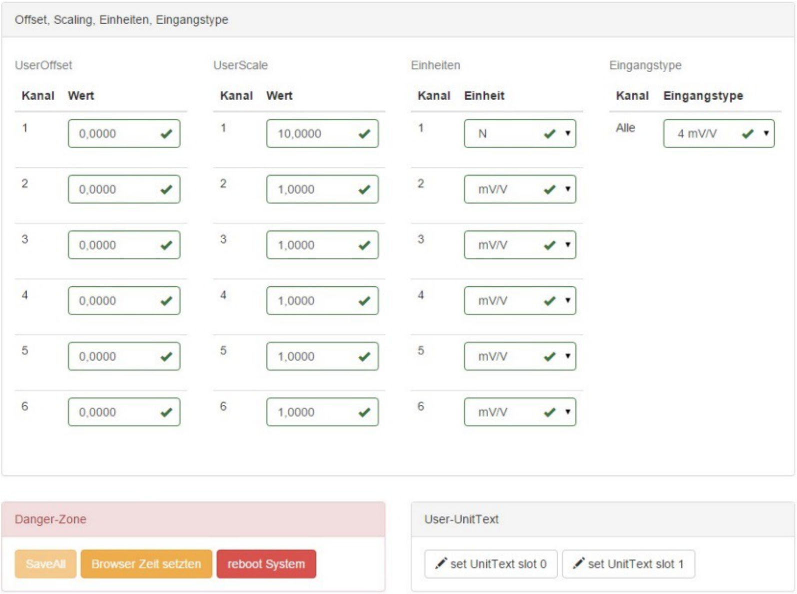
Einheiten

| Kanal | Einheit |
|---|---|
| 1 | N ✓ ▾ |
| 2 | mV/V ✓ ▾ |

Eingangstype

| Kanal | Eingangstype |
|---|---|
| Alle | 4 mV/V ✓ ▾ |

Figure 4 - configuration part 1



Offset, Scaling, Einheiten, Eingangstype

UserOffset

| Kanal | Wert |
|---|---|
| 1 | 0,0000 ✓ |
| 2 | 0,0000 ✓ |
| 3 | 0,0000 ✓ |
| 4 | 0,0000 ✓ |
| 5 | 0,0000 ✓ |
| 6 | 0,0000 ✓ |

UserScale

| Kanal | Wert |
|---|---|
| 1 | 10,0000 ✓ |
| 2 | 1,0000 ✓ |
| 3 | 1,0000 ✓ |
| 4 | 1,0000 ✓ |
| 5 | 1,0000 ✓ |
| 6 | 1,0000 ✓ |

Einheiten

| Kanal | Einheit |
|---|---|
| 1 | N ✓ ▾ |
| 2 | mV/V ✓ ▾ |
| 3 | mV/V ✓ ▾ |
| 4 | mV/V ✓ ▾ |
| 5 | mV/V ✓ ▾ |
| 6 | mV/V ✓ ▾ |

Eingangstype

| Kanal | Eingangstype |
|---|---|
| Alle | 4 mV/V ✓ ▾ |

Danger-Zone

SaveAll | Browser Zeit setzten | reboot System

User-UnitText

✎ set UnitText slot 0 | ✎ set UnitText slot 1

Figure 5 - configuration part 2

Recorded measurements can be found in archive (archive.html). Here the CSV files are listed and the following actions with the files can be executed:

🗑 delete

⬇ download

◘ reduce

👁 display

*Delete*

The CSV-file will be deleted (Please note, no copy is provided).

*Download*

This will download a copy.

*Reduce*

Reduce means that the number of entries is decreased. In addition, the summarized values are stored in a new CSV file with the prefix "red_". The reduction takes place by combining time-related values and storing only their minimum and maximum values. The reduction can only be applied to CSV files that have more than 4000 entries.

*Display*

By clicking on Display, the user reaches the detail view (messung_detail.html) of the selected CSV file. The data of the CSV file is displayed here.

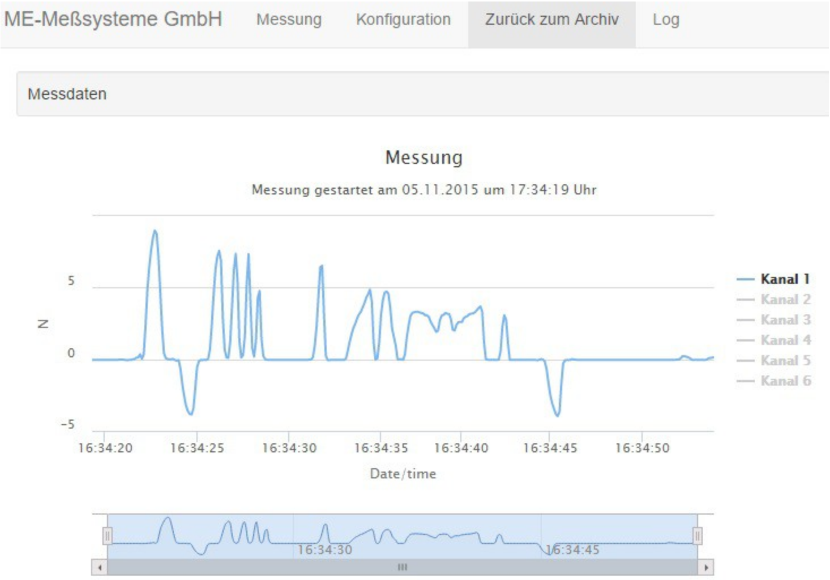
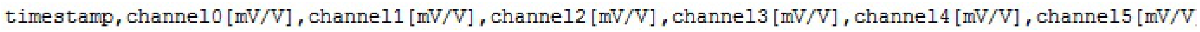


Figure 6 - Measurement detailed view

**Attention:** If CSV files are too large, browser of mobile devices may not represent the Web page. At this point, a reduced version of the CSV file should be created to allow the representation. The reduced versions of the CSV files are always so large that there should be no display problems on any devices.

*CSV-Format*

Each of CSV file created by the program has a file header and from one to n entries. The file header has the following format:



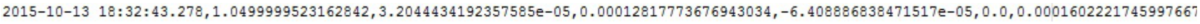
```
timestamp,channel0[mV/V],channel1[mV/V],channel2[mV/V],channel3[mV/V],channel4[mV/V],channel5[mV/V
```

Figure 7 - CSV-Header

In this case, the file header says that the first column contains a time stamp. The following columns contain the numbers of the individual channels of the GSV-6CPU. In addition, the channels still contain the set units of the respective channel in square brackets.

The timestamp is internally a Unix timestamp and is stored in the following format in the CSV file: "<year> - <month> - <day> <hours>: <minutes>: <seconds>.<milliseconds>“

The measured values are stored in float (textual). An example of a complete line:



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

Figure 8 - CSV-line

## Applied third-party software / libraries

Here is given an overview of the software used.

### PyPy

PyPy is an interpreter for Python code written in Python. The integrated JIT compiler allows certain parts of a Python code to be executed many times faster.

### Twisted (Lib)

Twisted is an event driven network software framework, which can be accelerated by PyPy. For the GSV-6 application, the framework provides the web server and, together with "Autobahn.ws", provides the websocket and serial connections.

### Autobahn.ws (Lib)

Autobahn.ws is a software framework that implements and provides the Websocket standard and a subprotocol that is based on the Websocket standard with the name WAMP. WAMP provides remote procedure calls and publish & subscribe services via Webockets.

### Crossbar.io

### Highcharts (Highstock)

## Installation

The installation is described, for example, on a Raspberry PI 2 with preinstalled Raspbian1 (version 2015-05-05). It is assumed that the EDIMAX EW-7811Un or Wi-Pi WLAN USB adapter already exists and is plugged in. It is also assumed that the user "pi" is used. It is intended to follow this installation guide linearly.

### Installation procedure

Firstly, PyPy is installed, a variant that accelerates Python code in parts and the package manager pip[1]. The following packages are then installed, which are required by the program:

- Twisted
- AutobahnPython
- PySerial
- numpy

In order to "mount" USB memory automatically  in the Linux operating system, the "usbmount" program is required.

Then, crossbar.io is installed, start-up scripts for crossbar as well as the main program serial2ws are installed and the WLAN adapter is configured to act as an access point.

If a configuration file is opened with nano, it can be saved with the key combination strg + o and can be closed with the key combination ctrl + x.

In addition, all commands are entered in the console.

The installation requires two folders (downloads and install), which must be created in advance:

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

## PyPy

PyPy is preinstalled in Raspian (2015-05-05), but in an older version (2.2.1), so a newer version (4.0.0) should be installed. This is done with the following commands in the console

```
cd ~/downloads
```

[1] https://github.com/pypa/pip

```
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
```

Note: Lines 6 and 7, as well as lines 8 and 9, are each one line.

### Pip

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

### Twisted

Twisted is not installed with the help of pip, because for the installations a line in the Setup.py must be escaped. It is commented in python with #.

```
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
```

Note: Lines 2 and 3 are one line

```
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

Since PyPy is used in this project, a customized variant of numpy must be installed with the following command (note: this takes a while> 15 min.):

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

If a standard Python was installed, the following command would suffice:

```
    pip install numpy
```

### usbmount

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

Then the configuration must be adapted. For this, the line FS_MOUNTOPTIONS in the usb, mount.conf is adapted:

```
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

For the installation of crossbar.io, a few dependencies are required in the Linux system. These are installed with the apt-get tool and then crossbar is installed via pip:

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

### Set the time zone

In order for serial2ws.py to use the correct time, the time zone has to be configured.

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

### Get project data from github.com

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

### Configuration file

The defaults.conf is the configuration file in which the following options can be set:

| Option | Description |
|---|---|
| router | Sets the web address for the Corssbar instance (e.g. ws://127.0.0.1:8001/ws/) |
| web | Sets the port over which the configuration web page can be reached (e.g. 8000) |
| boot_wait | Sets a time x, which is waited after the program start (e.g. 10 == 10 Sek.) |
| port | Sets the serial port (e.g. `/dev/ttyAMA0 or 4 (for Comport 5 under Windows))` |
| baudrate | Sets the baud rate of the serial port |
| csvpath | Sets the storage location where the CSV files are placed |
| log | Sets the log level (available: CRITICAL, ERROR, WARNING, INFO, DEBUG, TRACE) |

**Tip:** the cavpath must be present, otherwise the program stops with an error message.

The options can also be passed as parameters when starting from the command line. Example:

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

### Start-up-script

Copy the start-up-scripts from the scripts folder to /etc/init.d.

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

Mark scripts as executable:

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

In order to start crossbar and serial2ws during the boot process, this could be done via init.d. However, this guide describes how to start both programs using the rc.local file. To do this, open the /etc/rc.local file with nano:

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

And adjust the file as follows:

```
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
```

Crossbar and serial2ws are now started during the ramp-up of the Raspberry Pi. Furthermore, crossbar.io and serial2ws can be started and stopped as a daemon, with the following commands:

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

sudo /etc/init.d/serial2ws start
sudo /etc/init.d/serail2ws stop
```

### Set up WlAN access point

In order to provide a WLAN-Accesspoint for Raspberry Pi a program is required, that provides this function. Moreover, a DHCP server is required, which assigns an IP to the connected client. This section of the description has been created from two sources (source1[2] and source2[3]).

In this guide, hostapd is used for the access point and dnsmasq is used as DHCP server. However, the attention must be paid to ensure that the WLAN adapter is compatible [4] with the hostapd daemon1.

This guide describes the installation for a compatible WLAN adapter (for example Wi-Pi with the Ralink RT5370 chipset) and for a WLAN adapter with the Realtek RTL8188CUS chip.

For WLAN adapters with Realtek RTL8188 chip, a patched version of hostapd is used, which can be obtained in two versions: First, the pre-compiled variant from the git repo is used or secondly the leased version on the Raspberry Pi built (compiled). In both cases, however, hostap and dnsmasq should be installed via apt-get. The installed hostapd version is then replaced by the patched version.

### Installation

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

The following steps are only necessary if a WLAN adapter with the Realtek RTL8188 chip is used. If the adapter is compatible, please jump to the DHCP server configuration.

### Option 1, use of pre-compiled version

For this, the pre-compiled binary must be copied to the /usr/sbin folder. In order to maintain the original version in case of problems, it will be renamed.

```
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
```

### Option 2, compile the leased hostapd

To do this, all dependencies (Lib's) must be firstly installed:

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

[2] https://wiki.debianforum.de/WLAN-Access-Point_mit_hostapd_und_USB-Stick

[3] http://elinux.org/RPI-Wireless-Hotspot

[4] https://w1.fi/hostapd/

The git repo is then checked out

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

The next step is to create the leased variant of hostapd:

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

Finally, the binary must be copied to the /usr/sbin folder as in option1:

```
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
```

## Configuration

### DHCP-Server

Now that the required tools have been installed, dnsmasq and then hosdtapd are configured.

The dnsmasq configuration file should be ensured with the following command in advance :

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

Then open the new dnsmasq.conf with nano:

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

The following text is inserted:

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

Save and close the file.

### WLAN-Adapter

This completes the configuration for the DHCP server. The configuration of the WLAN adapter follows. During the boot process, a fixed IP address must be assigned to the WLAN adapter. The IP address depends on the IP address range of the pre-configured DHCP server. In this case, the IP address 192.168.9.1 is selected, for which the /etc/network/interfaces file is opened and edited:

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

Then change the line "iface wlan0 inet dhcp" as follows:

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

Also extract the following lines with the # character:

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

Now the following lines should be appended to the end of the interfaces file:

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

Finally, save and close the file.

Now the configuration of the hostap daemon follows. This program uses the WLAN adapter to set up an access point that allows devices to connect to a standard WLAN router. The configuration file must be adapted as follows.

Open configuration file:

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

The configuration must be then modified as follows:

Please note that you use a self-selected password by replacing <accesspoint_password> with your password. In addition, the "driver" line must be adjusted with the value "rtl1871xdrv" if you are using a mi Realtek chip WLAN adapter

```
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
```

Then the hostap daemon must be informed which configuration should be loaded during the boot process. For this, the file /etc/default/hostapd is opened and edited:

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

It should be modified as follows:

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

There is no need to set up the hostapd and dnsmasq programs automatically because they are started together with the network (see adaptation of the/etc/network/interfaces file).

Finally, restart the system:

```
sudo reboot
```

Now give the Raspberry Pi enough time to start up and the Accesspoint with the name "ME_AP" should be visible. If you are connected to a device, you should reach the GSV-6 website at http://192.168.9.1:8000.

Partially, there was the problem that no IP was assigned to the WLAN-Adapter(Interfaces) during the boot process. This can be avoided by adapting the/etc/ifplugd configuration where the file is modified as follows[5]:

Open the file with nano:

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

And adapt the following lines:

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

[5] http://rpi.vypni.net/wifi-ap-rt5370-on-raspberry-pi/