pi_ager install on Raspberry Pi 3/Pi 4/ Pi zero w under Pi OS (32-bit) with desktop and recommended software or Pi OS Lite

- For Pi 4/3: Download and install Raspberry Pi OS with desktop and recommended software from https://www.raspberrypi.org/software/operating-systems/

- For Pi zero: Download and install Raspberry Pi OS Lite from https://www.raspberrypi.org/software/operating-systems/

- Enable SSH for remote access
  ```
  sudo touch /boot/ssh
  ```

- Setup WLAN configuration
  Generate file wpa_supplicant.conf in /boot:
  ```
  country=DE
  ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
  update_config=1
  network={
      ssid="WLAN SSID"
      psk="WLAN PASSWORT"
  }
  ```

- Edit config.txt in /boot to support I2C and SPI devices:
  ```
  # Additional overlays and parameters are documented /boot/overlays/README
  # Use Pi-Ager Pins 11/13 GPIO 17/27 for I2C
  dtoverlay=i2c-gpio,bus=3,i2c_gpio_sda=17,i2c_gpio_scl=27
  # Use Pi-Ager Pin 16 for MCP3204
  dtoverlay=spi1-1cs,cs0_pin=16
  ```

  at the end of config.txt add the following lines to support bluetooth and Nextion TFT displays via serial port /dev/serial0 :
  ```
  [all]
  start_x=0
  enable_uart=1
  dtoverlay=miniuart-bt
  force_turbo=1
  ```

- Add in /boot/cmdline.txt at the end of line this to enable USB camera with fswebcam :
  ```
  dwc_otg.fiq_fsm_mask=0x3
  ```

- Reboot system

- Edit /etc/modules to load i2c-dev at boot, add this line :
  ```
  i2c-dev
  ```

- Add file :
  ```
  sudo touch /etc/modprob.d/raspi-blacklist.conf
  ```

- Get a copy from Pi-Ager repository to your local system:

```
git clone –b entwicklung https://github.com/Tronje-the-Falconer/Pi-Ager
```

All project file are now in the folder ./Pi-Ager/

- Copy setup.txt from local repository to /boot/ and edit it as needed.

- Copy /etc/modprobe.d/Pi-Ager_i2c_off.conf.on from local repository to /etc/modprobe.d/

- Reboot system

- Install lighttpd:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install lighttpd
sudo systemctl status lighttpd
```

```
sudo nano /etc/lighttpd/lighttpd.conf
```

and change Parameter

```
server.document-root = "/var/www/html"
```

to

```
server.document-root = "/var/www"
```

```
sudo usermod -G www-data -a pi
sudo chown -R www-data:www-data /var/www
sudo chmod -R 755 /var/www
```

- Reboot system

For testing the web server, generate html-page:

```
sudo nano /var/www/test.html
```

with content:

```
<html>
<head><title>Test-Seite</title></head>
<body>
<h1>Das ist eine Test-Seite.</h1>
</body>
</html>
```

Enter your IP Address (or localhost) into the browser followed by /test.html

In addition we need .htcredentials to contain user and password.

Fort that we use the Online-Tool https://websistent.com/tools/htdigest-generator-tool/

```
Username: pi-ager
REALM: Pi-Ager
Password: raspberry
```

Caution! All entries are case sensitive!

Open this file now

```
sudo nano /var/.htcredentials
```

and fill in the string output from the generator tool.

Save file with “STRG+o”, “RETURN” and close with “STRG+x”

Now we have to setup the password authentification in lighttpd:

```
sudo nano /etc/lighttpd/conf-available/05-auth.conf
```

The following lines are added under server.modules += („mod_auth“) :

```
auth.backend                  = "htdigest"
auth.backend.htdigest.userfile     = "/var/.htcredentials"

auth.require                  = ( "/settings.php" =>
                        (
                         "method" => "digest",
                         "realm" => "Pi-Ager",
                         "require" => "user=pi-ager"
                        ),
                         "/admin.php" =>
                        (
                         "method"  => "digest",
                         "realm"   => "Pi-Ager",
                         "require" => "valid-user"
                         ),
                         "/webcam.php" =>
                        (
                         "method"  => "digest",
                         "realm"   => "Pi-Ager",
                         "require" => "valid-user"
                         ),
                         "/notification.php" =>
                        (
                         "method"  => "digest",
                         "realm"   => "Pi-Ager",
                         "require" => "valid-user"
                         )
                       )
```

Then we activate this modul:

```
sudo lighty-enable-mod auth
```

In addition we have to edit :

```
sudo nano /etc/lighttpd/conf-available/15-fastcgi-php.conf
```

add at the end of the line

```
"broken-scriptfilename" => "enable"
```

a “,” and in a new line

```
"allow-x-send-file" => "enable"
```

Save end exit nano.

Now enable these modules:

```
sudo lighty-enable-mod fastcgi
sudo lighty-enable-mod fastcgi-php
```

Now reload the the webserver:

```
sudo service lighttpd force-reload
```

Now continue to install additional modules:

- Install Git
  ```
  sudo apt install git
  ```
- Install smbus
  ```
  sudo apt-get install python3-smbus
  ```
- Install sqlite3:
  ```
  sudo apt install sqlite3
  ```
- Install DHT sensor support
  ```
  sudo pip3 install Adafruit-DHT
  ```
- Install SHT1x sensors
  ```
  sudo pip3 install pi-sht1x
  ```
- Install libgd-dev (needed for new version of fswebcam)
  ```
  sudo apt install libgd-dev
  ```
- Install fswebcam:
  ```
  sudo apt install fswebcam
  ```
- Install influxdb
  ```
  sudo pip3 install influxdb
  ```
- Install php 7.3
  ```
  sudo apt install php7.3-common php7.3-cgi php7.3 php7.3-sqlite3
  ```
- Install additional modules for php7.3:
  ```
  sudo apt install php7.3-apcu php7.3-fpm php7.3-mbstring php7.3-phpdebug
  ```
- Install wiringpi:
  ```
  sudo apt install wiringpi
  ```
- Install wiringpi new version with Pi4 support :
  ```
  cd /tmp
  wget https://project-downloads.drogon.net/wiringpi-latest.deb
  sudo dpkg -i wiringpi-latest.deb
  ```
- Copy gpio to /usr/local/bin
  ```
  sudo cp /usr/bin/gpio /usr/local/bin
  sudo chmod 4755 /usr/local/bin/gpio
  ```

- Install PiShrink
  wget https://raw.githubusercontent.com/Drewsif/PiShrink/master/pishrink.sh
  chmod +x pishrink.sh
  sudo mv pishrink.sh /usr/local/bin

- Nextion serial client (HMI Dislplay support)
  sudo pip3 install nextion

- php zip support:
  sudo apt update
  sudo apt install php-zip

- Install lsof command:
  sudo apt update
  sudo apt install lsof

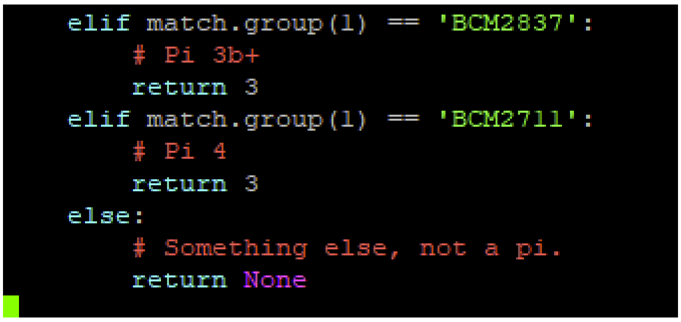
- Install Locale en-GB and de-DE UTF-8 using
  sudo raspi-config

- Enable Serial Interface, disable login, needed for HMI Nextion Display
  sudo raspi-config

- Install zip and unzip:
  sudo apt install zip unzip

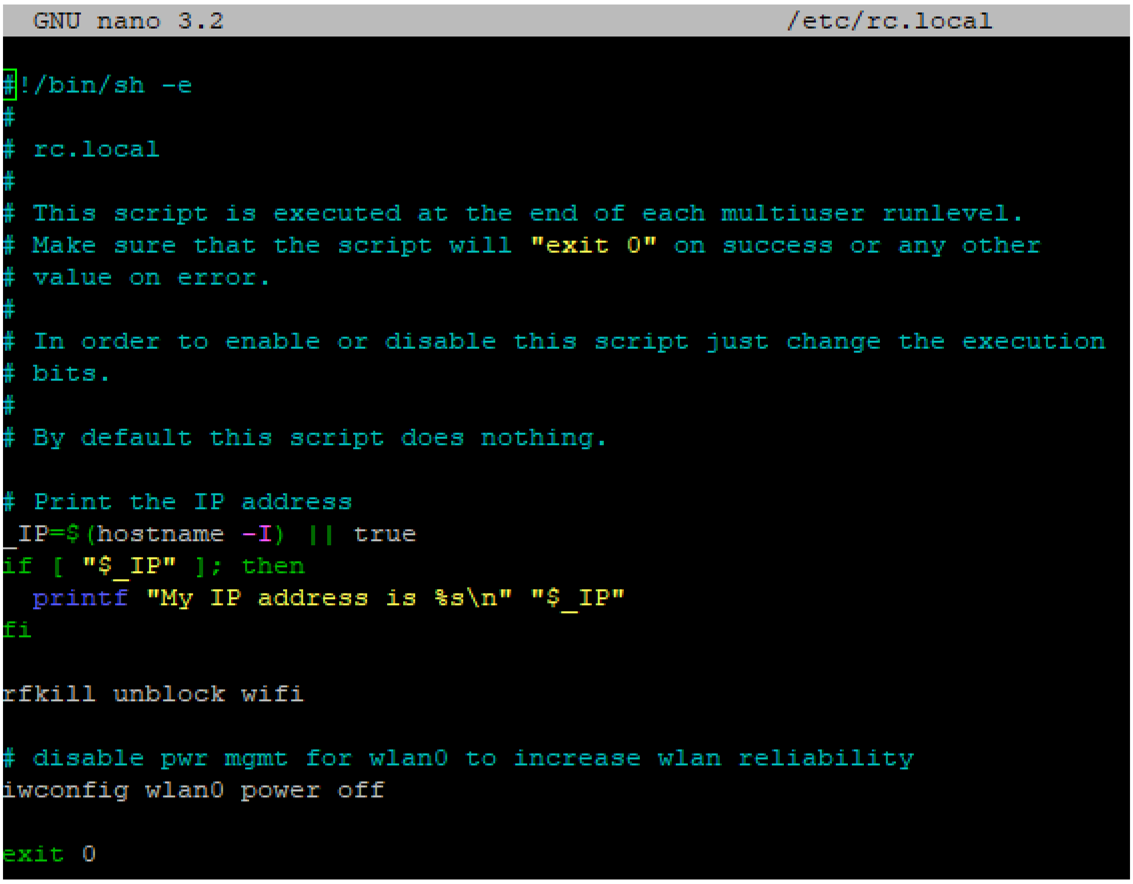
- Workaround for Adafruit_DHT for Pi4:
  In "/usr/local/lib/python3.7/dist-packages/Adafruit_DHT/platform_detect.py", you can add the followings at line #112 in the elif ladder, so it should workaround the issue.

```
elif match.group(1) == 'BCM2711':
    return 3
```



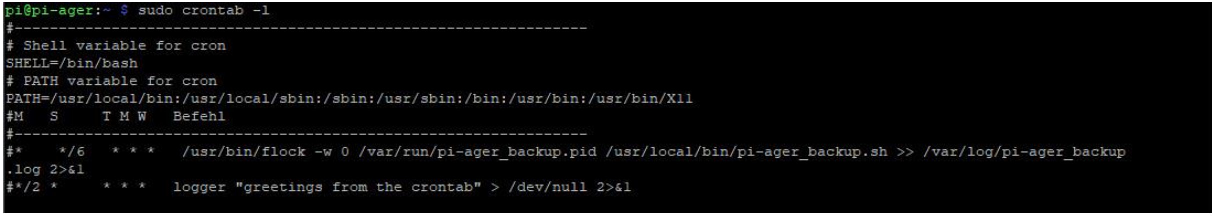
```
    elif match.group(1) == 'BCM2837':
        # Pi 3b+
        return 3
    elif match.group(1) == 'BCM2711':
        # Pi 4
        return 3
    else:
        # Something else, not a pi.
        return None
```

- Unblock wifi for Pi4, add rfkill unblock wifi and disable power management for wlan0:
  cd /etc
  sudo nano rc.local



```
  GNU nano 3.2                                              /etc/rc.local

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

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

rfkill unblock wifi

# disable pwr mgmt for wlan0 to increase wlan reliability
iwconfig wlan0 power off

exit 0
```

- Generate/edit crontab to prepare for automatic enable pi-ager_backup.sh



```
pi@pi-ager:~ $ sudo crontab -l
#-----------------------------------------------------------------
# Shell variable for cron
SHELL=/bin/bash
# PATH variable for cron
PATH=/usr/local/bin:/usr/local/sbin:/sbin:/usr/sbin:/bin:/usr/bin:/usr/bin/X11
#M   S     T M W   Befehl
#-----------------------------------------------------------------
#*    */6   * * *   /usr/bin/flock -w 0 /var/run/pi-ager_backup.pid /usr/local/bin/pi-ager_backup.sh >> /var/log/pi-ager_backup
.log 2>&1
#*/2 *     * * *   logger "greetings from the crontab" > /dev/null 2>&1
```

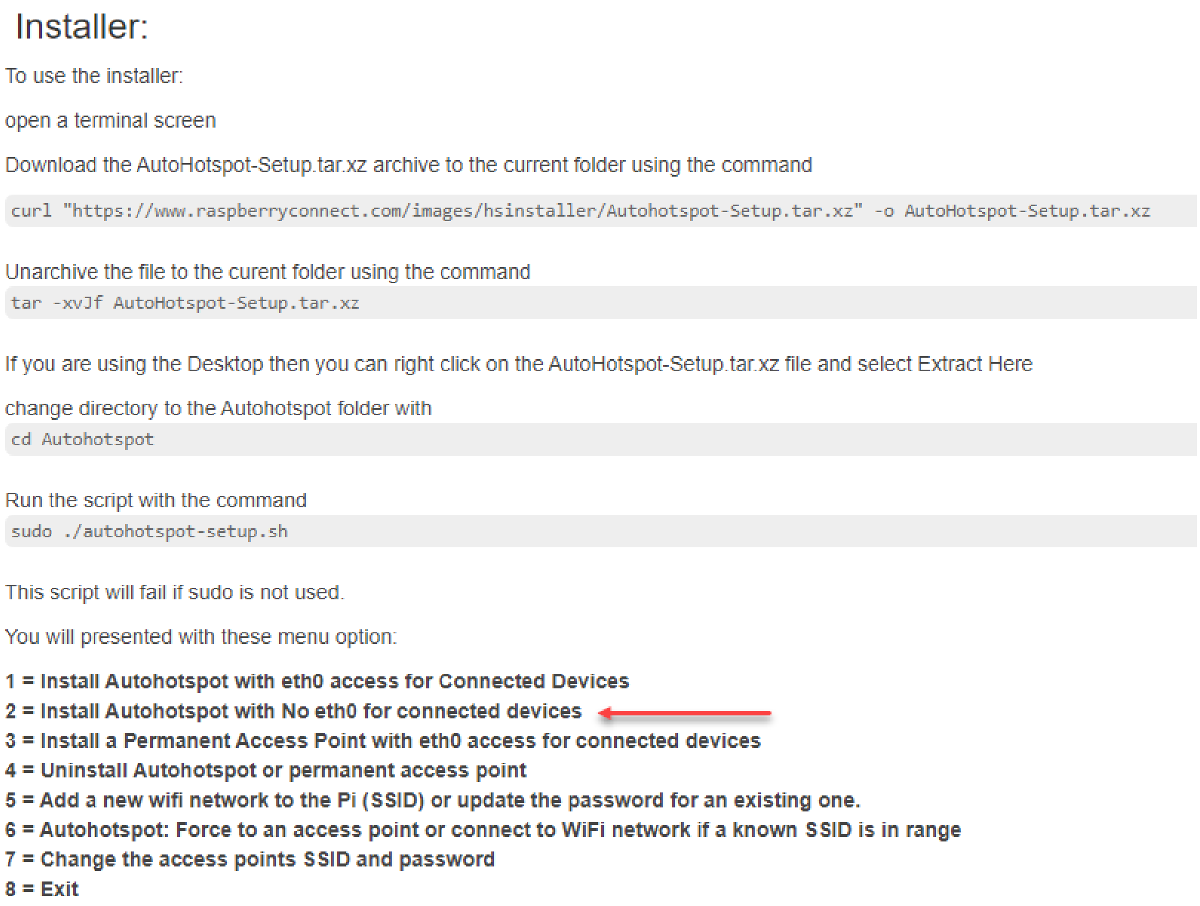
Use visudo to edit /etc/sudoers, so that the www-data User (User of Website) can execute /var/sudowebscript.sh :
sudo visudo
and then in sudoers following
...
#User privilege specification
root   ALL=(ALL:ALL) ALL
...
adding:

www-data ALL=NOPASSWD:/var/sudowebscript.sh, /var/show_wifi_connections.sh, /var/updatessid.sh

Save and exit.

- Install Raspberry Pi automatic Hotspot
  Information howto install can be found here :
  https://www.raspberryconnect.com/projects/65-raspberrypi-hotspot-accesspoints/183-raspberry-pi-automatic-hotspot-and-static-hotspot-installer

cd ~



Installer:

To use the installer:

open a terminal screen

Download the AutoHotspot-Setup.tar.xz archive to the current folder using the command

```
curl "https://www.raspberryconnect.com/images/hsinstaller/Autohotspot-Setup.tar.xz" -o AutoHotspot-Setup.tar.xz
```

Unarchive the file to the curent folder using the command

```
tar -xvJf AutoHotspot-Setup.tar.xz
```

If you are using the Desktop then you can right click on the AutoHotspot-Setup.tar.xz file and select Extract Here

change directory to the Autohotspot folder with

```
cd Autohotspot
```

Run the script with the command

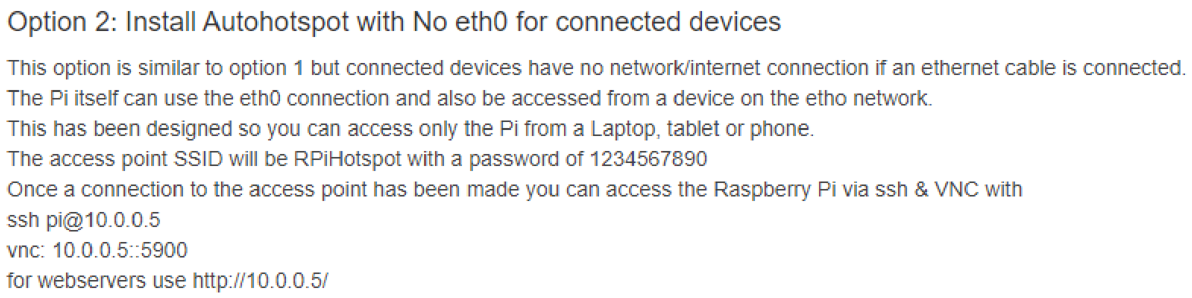
```
sudo ./autohotspot-setup.sh
```

This script will fail if sudo is not used.

You will presented with these menu option:

1 = Install Autohotspot with eth0 access for Connected Devices
2 = Install Autohotspot with No eth0 for connected devices
3 = Install a Permanent Access Point with eth0 access for connected devices
4 = Uninstall Autohotspot or permanent access point
5 = Add a new wifi network to the Pi (SSID) or update the password for an existing one.
6 = Autohotspot: Force to an access point or connect to WiFi network if a known SSID is in range
7 = Change the access points SSID and password
8 = Exit

Select option 2:

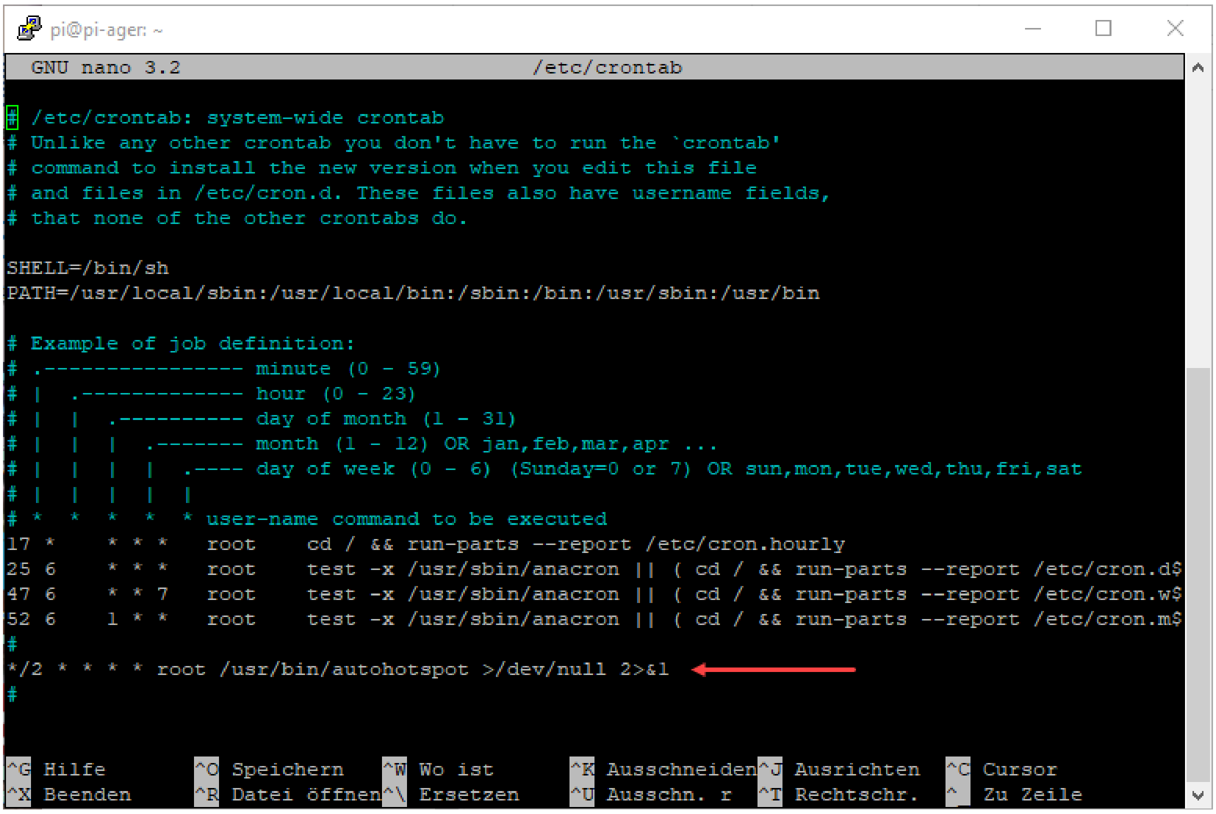


Option 2: Install Autohotspot with No eth0 for connected devices

This option is similar to option 1 but connected devices have no network/internet connection if an ethernet cable is connected.
The Pi itself can use the eth0 connection and also be accessed from a device on the etho network.
This has been designed so you can access only the Pi from a Laptop, tablet or phone.
The access point SSID will be RPiHotspot with a password of 1234567890
Once a connection to the access point has been made you can access the Raspberry Pi via ssh & VNC with
ssh pi@10.0.0.5
vnc: 10.0.0.5::5900
for webservers use http://10.0.0.5/

Then exit and reboot

Setup a timer for autohotspot so that connectivity is checked every 2 minutes:
sudo nano /etc/crontab



```
  GNU nano 3.2                                 /etc/crontab

# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.d$
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.w$
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.m$
#
*/2 * * * * root /usr/bin/autohotspot >/dev/null 2>&1
#

^G Hilfe      ^O Speichern  ^W Wo ist     ^K Ausschneiden^J Ausrichten  ^C Cursor
^X Beenden    ^R Datei öffnen^\ Ersetzen  ^U Ausschn. r  ^T Rechtschr.  ^_ Zu Zeile
```

From your local pi-ager repository copy /usr/bin/autohotspot to /usr/bin/ and change rights to 755. This version of autohotspot script checks also if this script is already running and stops execution if an instance of this script is active.

- Install Bluetooth modules to support Bluetooth Temp./Hum Sensor from Xiaomi
- sudo pip3 install bluepy requests
- sudo apt install bluetooth libbluetooth-dev
- sudo pip3 install pybluez pycryptodomex
- cd /opt
- sudo git clone https://github.com/JsBergbau/MiTemperature2.git
- copy MiCallback.sh from repository /opt/MiTemperature2 to Pi and enable execute rights for all
- sudo chmod +x MiCallback.sh
- copy my_thermometer.txt from repository /opt/MiTemperature2 to Pi and enable write rights for all
- sudo chmod +w my_thermometer.txt
- Now copy all files and folders from your local git repository /var/www to /var/www/
- from local repository /opt/pi-ager/ to /opt/pi-ager/
- from local repository /var/sudowebscript.sh to /var/ and change rights to 555
- from local repository /var/show_wifi_connections.sh to /var/ and change rights to 555
- from local repository /var/updatessid.sh to /var/ and change rights to 555
- sudo chown –R www-data:www-data /var/www
- sudo chown root:root /var/www/
- sudo usermod –G gpio –a www-data
- sudo chmod 666 /var/www/logs/logfile.txt
- sudo chown –R root:root /var/www/logs
- sudo chmod 755 /var/www/logs/

- sudo chmod 664 /var/www/config/pi-ager.sqlite3
- sudo chown -R www-data:www-data /var/www/config/
- sudo chmod 775 /var/www/config/
- sudo chmod 555 /var/sudowebscript.sh
- from local repository /usr/local/bin/*.sh copy all to /usr/local/bin/
  (pi-ager_backup.sh, pi-ager_image.sh,setup_pi-ager.sh)
  Set +x mode to the scripts :
  sudo chmod +x /usr/local/bin/*.sh

- from local repository /lib/systemd/system copy the following files to
  /lib/systemd/system/ :
  pi-ager_main.service
  setup_pi-ager.service

- from local repository /usr/bin copy the following file to /usr/bin/. This is a newer version of fswebcam with re-get frame on error.
  fswebcam
  Set +x mode to fswebcam:
  sudo chmod +x /usr/bin/fswebcam

- from local repository /usr/share/man/man1/fswebcam.1.gz copy the following file to /usr/share/man/man1/
  fswebcam.1.gz

- Enable setup_pi-ager.service to initialize system with data from /boot/setup.txt after next reboot:
  sudo systemctl enable setup_pi-ager
  sudo reboot