

mjpg-streamer Installation

Installation des mjpg-streamer.

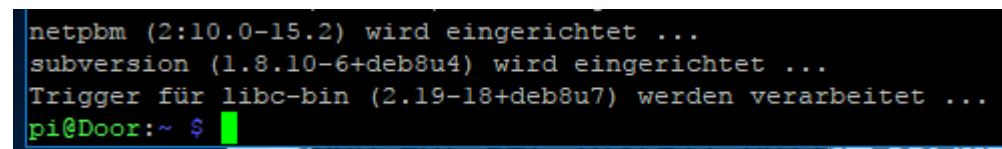
15. März 2016

<https://www.doorpi.org/forum/thread/36-installation-des-mjpg-streamer/?postID=6271&highlight=MJPEG-Streamer#post6271>

Quellcode

`sudo apt-get -y update && sudo apt-get -y upgrade`

`sudo apt-get install build-essential libjpeg-dev imagemagick subversion libv4l-dev
checkinstall`



```
netpbm (2:10.0-15.2) wird eingerichtet ...  
subversion (1.8.10-6+deb8u4) wird eingerichtet ...  
Trigger für libc-bin (2.19-18+deb8u7) werden verarbeitet ...  
pi@Door:~ $
```

`sudo modprobe bcm2835-v4l2`

Sollte es hier zu Problemen kommen ist wahrscheinlich die Kamera nicht aktiviert.

daher mit „sudo raspi-config“ ins Konfig und da nochmals aktivieren

weiter gehts mit:

`raspistill -o /tmp/test.jpg`



```
pi@Door:~ $ cd /tmp/  
pi@Door:/tmp $ ls -l  
pulse-DJkzTEkJgm4i  
pulse-PKdhtXMmrl8n  
ssh-UFtlFFaSBviE  
ssh-UKwU9DGBBA14  
systemd-private-702277bb10f04  
test.jpg  
pi@Door:/tmp $
```

`cd /tmp`

`ls -l`

schauen ob das Bild auch gemacht wurde

`test.jpg`

`cd`

svn checkout svn://svn.code.sf.net/p/mjpg-streamer/code/ mjpg-streamer-code

```
A   mjpg-streamer-code/uvc-streamer/Makefile
A   mjpg-streamer-code/.deps
U   mjpg-streamer-code
Ausgecheckt, Revision 182.
pi@Door:~ $
pi@Door:~ $
```

mit diesem Befehl landet man auf jeden Fall im richtigen Verzeichnis, war bei mir mal ein Problem beim ganzen probieren, daher hiermal eingefügt

cd /home/pi/mjpg-streamer-code/mjpg-streamer

Als nächstes erstellen wir einen Patch, nehmen uns aber nicht den Patch aus der offiziellen Anleitung sondern

<https://www.doorpi.org/forum/thread/723-mjpg-streamer-geht-nicht-im-mjpg-format/?postID=6442#post6442>

Sirlcy Anfänger Mittwoch 17:40 , hat mir eine kleine Anleitung geschrieben, was der patch von ihm genau macht ?? (bekomme ich auch noch raus)

Quellcode

sudo nano input_uvc_patch

1. Die Datei (input_uvc_patch.txt) im Anhang runterladen oder das blaue kopieren, in notepad++ in windows reinkopieren, dann mit dem Namen input_uvc_patch.txt abspeichern und dann mit filezilla zum raspi kopieren. (blauer Text)

```
--- plugins/input_uvc/input_uvc.c    (revision 174)
+++ plugins/input_uvc/input_uvc.c    (working copy)

@@ -405,9 +405,13 @@

    if(pcontext->videoIn->formatIn == V4L2_PIX_FMT_YUYV) {

        DBG("compressing frame from input: %d\n", (int)pcontext->id);

        pglobal->in[pcontext->id].size = compress_yuyv_to_jpeg(pcontext->videoIn, pglobal->in[pcontext->id].buf, pcontext->videoIn->framesizeIn,
gquality);

+        /* copy this frame's timestamp to user space */
+        pglobal->in[pcontext->id].timestamp = pcontext->videoIn->buf.timestamp;

    } else {

        DBG("copying frame from input: %d\n", (int)pcontext->id);

-        pglobal->in[pcontext->id].size = memcpy_picture(pglobal->in[pcontext->id].buf, pcontext->videoIn->tmpbuffer, pcontext->videoIn->
buf.bytesused);

+        pglobal->in[pcontext->id].size = memcpy_picture(pglobal->in[pcontext->id].buf, pcontext->videoIn->tmpbuffer, pcontext->videoIn->
tmpbytesused);

+        /* copy this frame's timestamp to user space */
+        pglobal->in[pcontext->id].timestamp = pcontext->videoIn->tmptimestamp;
```

```

    }

#ifdef 0

@@ -418,8 +422,6 @@

    prev_size = global->size;

#endif

- /* copy this frame's timestamp to user space */
- pglobal->in[pcontext->id].timestamp = pcontext->videoin->buf.timestamp;

/* signal fresh_frame */

pthread_cond_broadcast(&pglobal->in[pcontext->id].db_update);

Index: plugins/input_uvc/v4l2uvc.c
=====
--- plugins/input_uvc/v4l2uvc.c (revision 174)
+++ plugins/input_uvc/v4l2uvc.c (working copy)
@@ -450,6 +450,8 @@

    */

    memcpy(vd->tmpbuffer, vd->mem[vd->buf.index], vd->buf.bytesused);
+    vd->tmpbytesused = vd->buf.bytesused;
+    vd->tmptimestamp = vd->buf.timestamp;

    if(debug)

        fprintf(stderr, "bytes in used %d \n", vd->buf.bytesused);

Index: plugins/input_uvc/v4l2uvc.h
=====
--- plugins/input_uvc/v4l2uvc.h (revision 174)
+++ plugins/input_uvc/v4l2uvc.h (working copy)
@@ -28,6 +28,7 @@

#include <stdio.h>

#include <stdint.h>

#include <string.h>

#include <fcntl.h>

```

```
#include <unistd.h>

@@ -105,6 +106,8 @@

    int framecount;

    int recordstart;

    int recordtime;

+   uint32_t tmpbytesused;
+   struct timeval tmptimestamp;

};
```

Datei in nano mit rechtsklick ablegen dann mit Strg x und j abspeichern

4. Dann den Patch von **Sirlcy** ausführen mit:

patch -p0 < input_uvc_patch

```
pi@Door:~/mjpg-streamer-code/mjpg-streamer $ patch -p0 < input_uvc_patch
patching file plugins/input_uvc/input_uvc.c
patching file plugins/input_uvc/v4l2uvc.c
patching file plugins/input_uvc/v4l2uvc.h
Hunk #2 succeeded at 106 with fuzz 1.
pi@Door:~/mjpg-streamer-code/mjpg-streamer $
```

5. Weiter in der Anleitung "Installation Software MJPG-Streamer"

<https://www.doorpi.org/forum/thread/36-installation-des-mjpg-streamer/?postID=6271&highlight=MJPG-Streamer#post6271>

Wir führen anschließend diese Befehle aus und verlassen danach das Verzeichnis wieder.

Quellcode

make USE_LIBV4L2=true clean all

dauert wieder eine ganze Zeit

```
make[1]: Leaving directory '/home/pi/mjpg-streamer-code/mjpg-streamer/plugins/inputs'
cp plugins/input_testpicture/input_testpicture.so .
make -C plugins/input_file all
make[1]: Entering directory '/home/pi/mjpg-streamer-code/mjpg-streamer/plugins/inputs'
gcc -O2 -DLINUX -D_GNU_SOURCE -Wall -shared -fPIC -o input_file.so input_file.c
make[1]: Leaving directory '/home/pi/mjpg-streamer-code/mjpg-streamer/plugins/inputs'
cp plugins/input_file/input_file.so .
pi@Door:~/mjpg-streamer-code/mjpg-streamer $
```

sudo make DESTDIR=/usr/local install

```

make[1]: Leaving directory '/home/pi/mjpg-streamer-code/mjpg-streamer/plugins/input_file'
cp plugins/input_file/input_file.so .
pi@Door:~/mjpg-streamer-code/mjpg-streamer $ sudo make DESTDIR=/usr/local install
install --mode=755 mjpg_streamer /usr/local/bin
install --mode=644 input_uvc.so output_file.so output_udp.so output_http.so input_testpicture.so input_file.so /usr/local/lib/
install --mode=755 -d /usr/local/www
install --mode=644 -D www/* /usr/local/www
pi@Door:~/mjpg-streamer-code/mjpg-streamer $ cd
pi@Door:~ $

```

cd

Nun geben wir folgendes ein, nur um zu schauen das auch alles richtig installiert wurde.

Quellcode

`sudo modprobe bcm2835-v4l2`

Hier dürfen keine Fehlermeldungen erscheinen.

```

pi@Door:~/mjpg-streamer-code/mjpg-streamer $ cd
pi@Door:~ $ sudo modprobe bcm2835-v4l2
pi@Door:~ $

```

Jetzt kann der Streamer gestartet werden, mit folgendem Befehl:

Quellcode

`mjpg_streamer -i "/usr/local/lib/input_uvc.so -d /dev/video0 -n -r 1024x768 -f 24 -q 80" -o "/usr/local/lib/output_http.so -n -w /usr/local/www -p 9000"`

Hier wird ein kleiner Webserver gestartet der unter Deiner Adresse des raspberypi zu erreichen ist aber mit der Port Nummer 9000.

```

pi@Door:~ $ mjpg_streamer -i "/usr/local/lib/input_uvc.so -d /dev/video0 -n -r 1024x768 -f 24 -q 80" -o "/usr/local/lib/output_http.so -n -w /usr/local/www -p 9000"
MJPEG Streamer Version: svn rev: 3:172M
i: Using V4L2 device.: /dev/video0
i: Desired Resolution: 1024 x 768
i: Frames Per Second.: 24
i: Format.....: MJPEG
o: www-folder-path...: /usr/local/www/
o: HTTP TCP port.....: 9000
o: username:password.: disabled
o: commands.....: disabled

```

Bsp.:

192.168.178.87:9000 'meine IP, natürlich die eueres PI nehmen

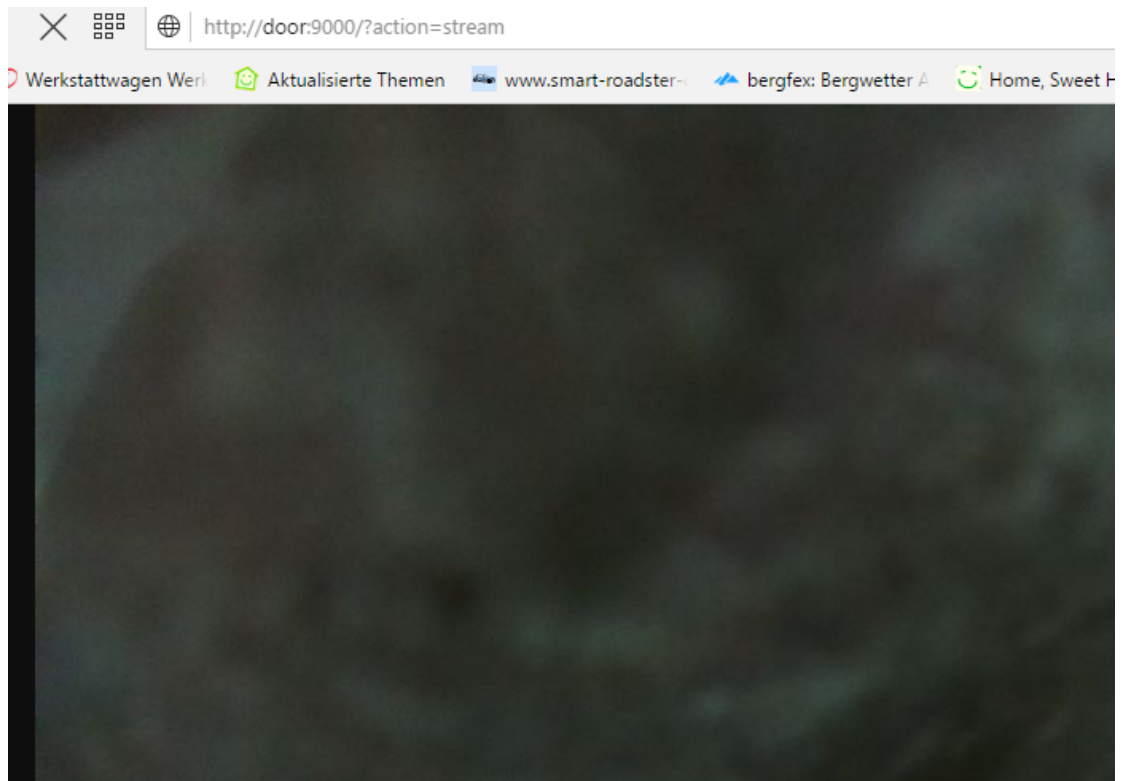


ganz unten auf der Seite sollte ein aktuelles Bild zu sehen sein

```
pi@Door:~ $ mjpg_streamer -i "/usr/local/lib/input_uvc.so -d
ib/output_http.so -n -w /usr/local/www -p 9000"
MJPG Streamer Version: svn rev: 3:172M
i: Using V4L2 device.: /dev/video0
i: Desired Resolution: 1024 x 768
i: Frames Per Second.: 24
i: Format.....: MJPEG
o: www-folder-path...: /usr/local/www/
o: HTTP TCP port.....: 9000
o: username:password.: disabled
o: commands.....: disabled
```

oder im win explorer mit dem angegeben link:

<http://door:9000/?action=stream>



mit

str c

beendet man den streamer im putty fenster

```
0: Commands.....: disabled
^Csetting signal to stop
i: cleaning up resources allocated by input thread
force cancellation of threads and cleanup resources
o: cleaning up resources allocated by server thread #00
done
pi@Door:~ $
```

Wenn man den mjpg_streamer beim booten automatisch gestartet haben möchte der sollte folgende Datei anlegen:

Quellcode

```
sudo nano /etc/init.d/mjpg_streamer
```

das was nun folgt in den Editor einfügen (alles was blau ist) in der Beschreibung aber zum kopieren bereitgestellt

<https://www.doorpi.org/forum/lexicon/entry/10-installation-software-mjpg-streamer/>

```
#!/bin/sh
```

```

# /etc/init.d/mjpg_streamer

### BEGIN INIT INFO

# Provides:      mjpg_streamer

# Required-Start:  $all

# Required-Stop:  $all

# Default-Start:  2 3 4 5

# Default-Stop:   0 1 6

# Short-Description: MJPG_Streamer_autostart

### END INIT INFO


start()

{

    modprobe bcm2835-v4l2

    sleep:2

    echo "Starting mjpg-streamer..."

    /usr/local/bin/mjpg_streamer -i "/usr/local/lib/input_uvc.so -d /dev/video0 -n -r 1024x768 -f 24 -q 80" -o
"/usr/local/lib/output_http.so -n -w /usr/local/www -p 9000" >/dev/null 2>&1 &

}


stop()

{

    echo "Stopping mjpg-streamer..."

    kill -9 $(pidof mjpg_streamer) >/dev/null 2>&1

}


case "$1" in

    start)

        start

        ;;

    stop)

        stop

        ;;

    restart)

```



```

stop

start

;;

*)

echo "Usage: $0 {start|stop|restart}"

;;

esac

exit 0

```

```

GNU nano 2.2.6      Datei: /etc/init.d/mjpg_streamer      Verändert
echo "Stopping mjpg-streamer..."
kill -9 $(pidof mjpg_streamer) >/dev/null 2>&1
}

case "$1" in
  start)
    start
    ;;
  stop)
    stop
    ;;
  restart)
    stop
    start
    ;;
  *)
    echo "Usage: $0 {start|stop|restart}"
    ;;
esac

exit 0

```

mit str x speichern und dann mit "j" bestätigen

das hier nicht vergessen:

sudo chmod +x /etc/init.d/mjpg_streamer

Mit diesem Befehl wird die eben erstellte Autostart-Datei in den Autostart eingetragen:

Quellcode

sudo update-rc.d mjpg_streamer defaults

Danach kannst Du es bequem per:

sudo service mjpg_streamer start

starten.

dann mit

`sudo service mjpg_streamer status`

überprüfen

Beenden kannst Du es mit:

`sudo service mjpg_streamer stop`

Kontrolle

`sudo service mjpg_streamer status`

```
pi@raspberrypi:~$ sudo service mjpg_streamer status
mjpg_streamer.service - LSB: MJPG_Streamer_autostart
Loaded: loaded (/etc/init.d/mjpg_streamer)
Active: active (running) since Mi 2017-05-03 18:51:03 CEST; 16min ago
Process: 11432 ExecStart=/etc/init.d/mjpg_streamer start (code=exited, status=0/SUCCESS)
CGroup: /system.slice/mjpg_streamer.service
└─11435 /usr/local/bin/mjpg_streamer -i /usr/local/lib/input_uvc.so -d /dev/video0 -n -r 1024x768

Mai 03 18:51:02 Door mjpg_streamer[11435]: MJPG-streamer [11435]: Desired Resolution: 1024 x 768
Mai 03 18:51:02 Door mjpg_streamer[11435]: MJPG-streamer [11435]: Frames Per Second.: 24
Mai 03 18:51:02 Door mjpg_streamer[11435]: MJPG-streamer [11435]: Format.....: MJPEG
Mai 03 18:51:02 Door mjpg_streamer[11435]: MJPG-streamer [11435]: www-folder-path....: /usr/local/www/
Mai 03 18:51:02 Door mjpg_streamer[11435]: MJPG-streamer [11435]: HTTP TCP port.....: 9000
Mai 03 18:51:02 Door mjpg_streamer[11435]: MJPG-streamer [11435]: username:password.: disabled
Mai 03 18:51:02 Door mjpg_streamer[11435]: MJPG-streamer [11435]: commands.....: disabled
Mai 03 18:51:02 Door mjpg_streamer[11435]: MJPG-streamer [11435]: starting input plugin /usr/local/lib/i
Mai 03 18:51:02 Door mjpg_streamer[11435]: MJPG-streamer [11435]: starting output plugin: /usr/local/lib
Mai 03 18:51:03 Door systemd[1]: Started LSB: MJPG_Streamer_autostart.
Hint: Some lines were ellipsized, use -l to show in full.
pi@raspberrypi:~$
```

Reboot:

Kontrolle ob auch alles richtig startet

`sudo service mjpg_streamer status`

alles idetisch wie oben :-)

Sicheurngsimage:

Step7 JessyP bis mjpg Stream

nicht gemacht

sollte ein Snapshot gewünscht sein so soll diesnahr einer Anleitung noch gemacht werden

<https://www.computerhilfen.de/info/raspberry-pi-kamera-installieren-fswebcam.html>

```
sudo apt-get install fswebcam
```