Aus einzelnen Caputres ein MJPEG-Stream erzeugen

Das Dekomprimieren eines RSTP-Streams (z. B. im Formst H.264 oder H.265) benötigt extrem viele Systemressourcen. Wenn man mehrere dieser Streams in einem Camera Surveillance System auf einem SBC wie dem Raspberry verarbeiten will, stößt man sehr schnell an die Grenzen dieser Systeme. Ein MJPEG-Stream in voller HD-Auflösung hingegen verbraucht am Raspberry nur sehr wenige Ressourcen. Leider stellen die meisten Kameras die MJPEG-Streams nur in geringer Auflösung zur Verfügung.

Abhilfe schafft, wenn man aus den Kamera-Images (also nicht den Streams, sondern in Captures im JPG-Format) wieder einen MJPEG-Stream in voller Auflösung zusammensetzt und an MotionEye als Stream weitergibt. Damit verbraucht man extrem wenig Ressourcen, hat aber die volle HD-Auflösung des Streams. Nachteil ist allerdings, dass man nur eine sehr geringe Framerate erzielt, weil man ja jedes Mal hintereinander das Capture von der Kamera abholen muss und dann an den Stream anhängen muss. Damit erzielt man nur eine Wiederholungsrate von vielleicht 1-2 Bilder pro Sekunde. Für eine Überwachungskamera reicht das aber in aller Regel aus.

Vorbereitung

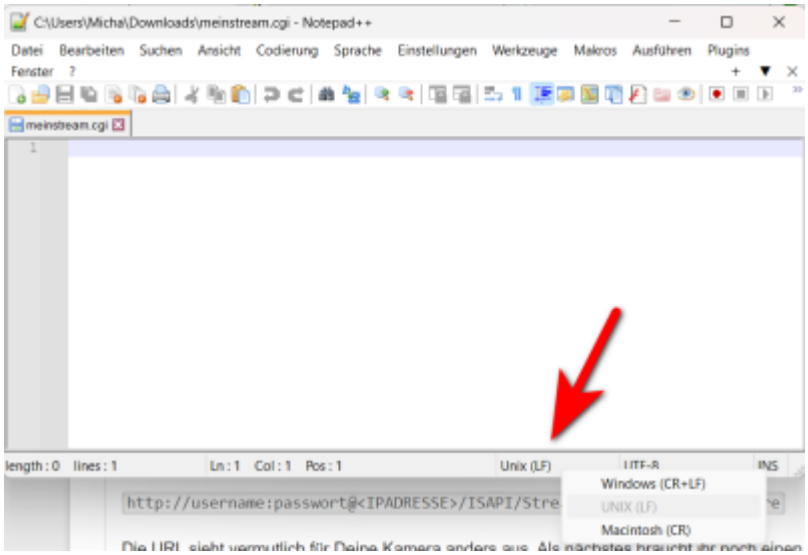
Du brauchst als erstes die URL, von der Du Standbilder in Deiner gewünschten Auflösung von der Kamera abholen kannst. Für eine Hikvision Kamera ist das zum Beispiel:

```
http://username:passwort@<IPADRESSE>/ISAPI/Streaming/Channels/1/picture
```

Die URL sieht vermutlich für Deine Kamera anders aus. Als nächstes braucht ihr noch einen guten Texteditor, der Unix-Zeilenenden verarbeiten kann. Dazu nutzt ihr zum Beispiel Notepad++.

Installation

Erzeugt eine neue Datei auf einem Windowsrechner mit dem Editor - die Dateiendung muss `.cgi` sein - zum Beispiel `meinstream.cgi` Vergesst nicht den Editor auf Unix-Zeilenendung umzustellen! Unter Notepad++ macht ihr das unten mit Rechtsklick in der Statusleiste:



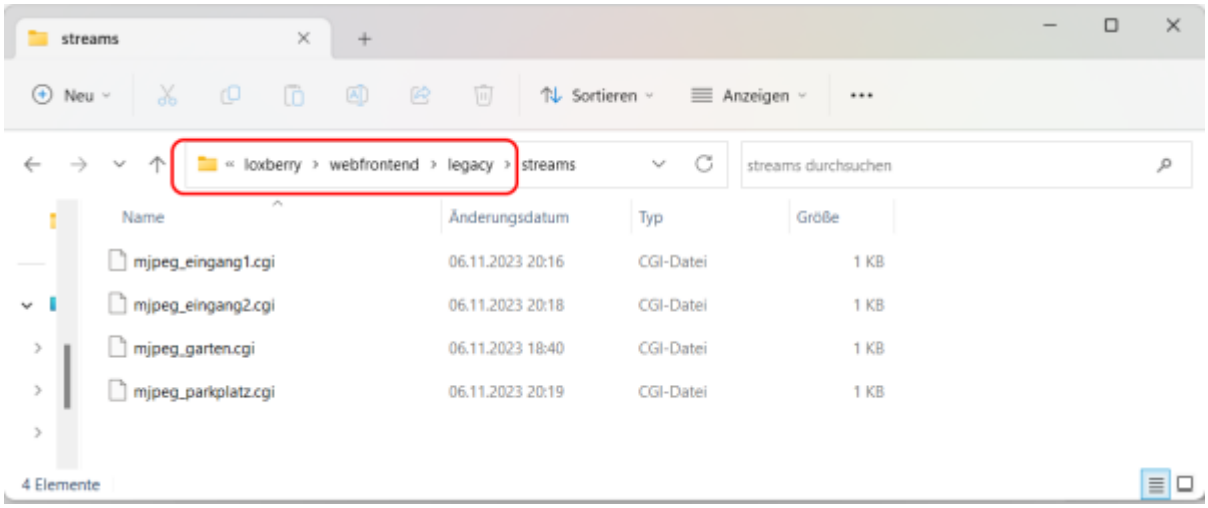
Nun kopiert ihr folgenden Inhalt in die Datei:

meinstream.cgi

```
#!/bin/bash

echo "Content-Type: multipart/x-mixed-replace;boundary=boundary12345"
echo "Cache-Control: no-cache"
echo ""
while :
do
    echo "--boundary12345"
    curl -is
"http://username:passwort@<IPADRESSE>/ISAPI/Streaming/Channels/1/pictur
e"
    sleep 0.2
done
```

Die URL hinter `curl -is` in der drittletzten Zeile müsst ihr natürlich gegen eure URL ersetzen. Diese Datei speichert ihr ab und tranferiert sie über den Windows Exporer auf dem LoxBerry in das Verzeichnis \\loxberry\webfrontend\legacy Ihr könnt Euch auch noch einen Unterordner anlegen, wenn ihr möchtet. Alternativ transferiert ihr die Datei über das Filemanager Widget. Wenn ihr mehrere Kameras habt, legt ihr Euch einfach pro Kamera eine Datei an.



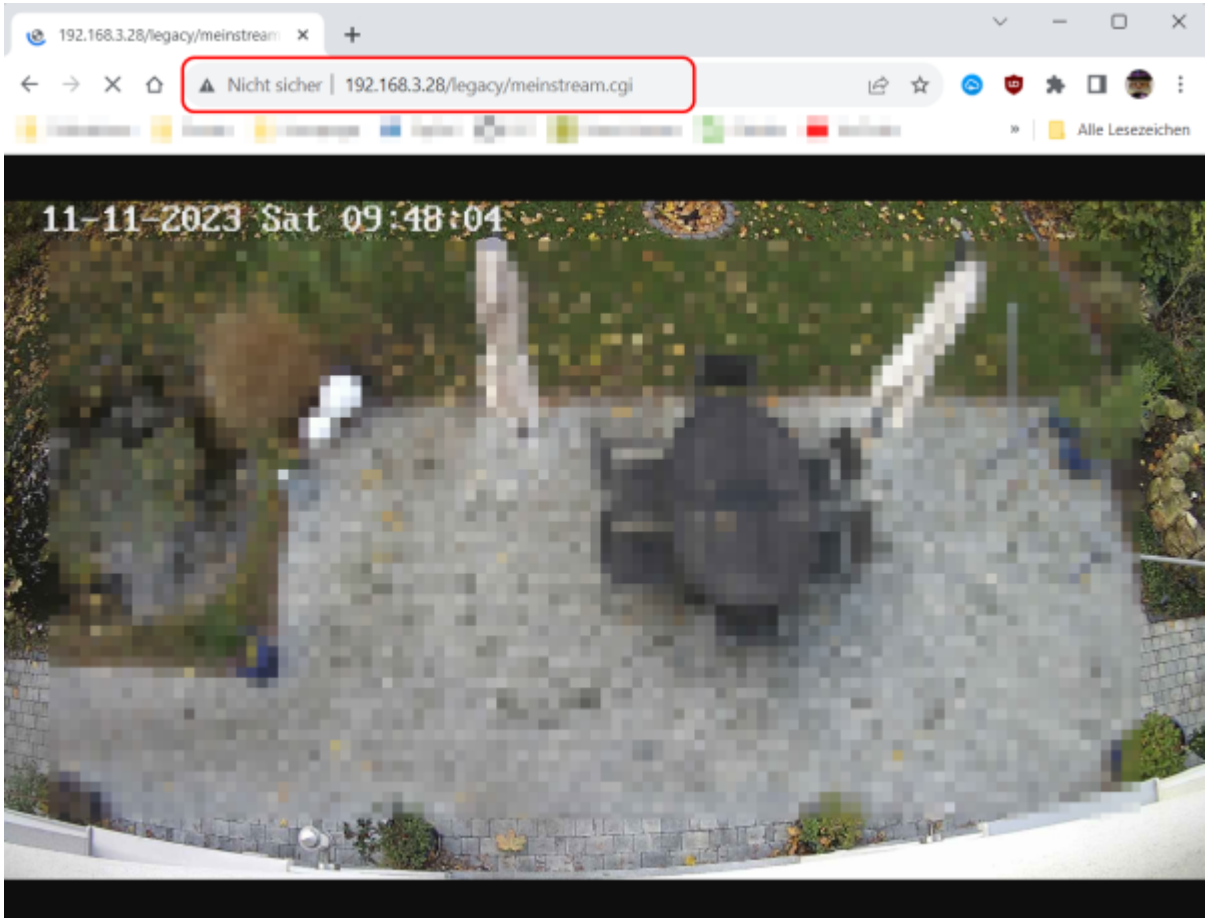
Nun loggt ihr Euch als letzten Schritt noch per Putty/SSH auf eurem LoxBerry ein: Eine SSH-Verbindung mit putty aufbauen / Shell-Zugriff Ihr müsst nun noch folgenden Befehl ausführen (den Dateinamen und anpassen und ggf. das Unterverzeichnis noch ergänzen!):

```
chmod a+x ~/webfrontend/legacy/meinstream.cgi
```

Fertig.

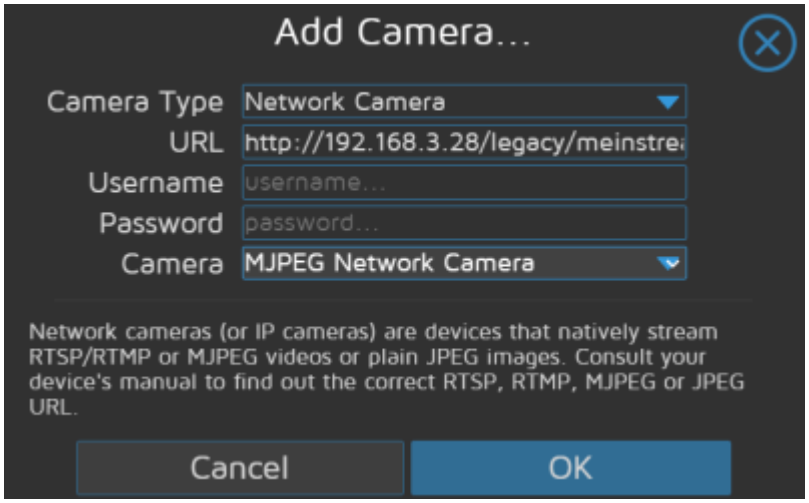
Stream aufrufen

Den Stream könnt ihr im Browser nun unter folgender URL erreichen:
`http://<ipadresse>/legacy/meinstream.cgi`



Einrichtung in MotionEye

Fügt in MotionEye eine Netzwerkkamera hinzu und verwendet dazu die obige URL ohne Benutzername und Passwort.

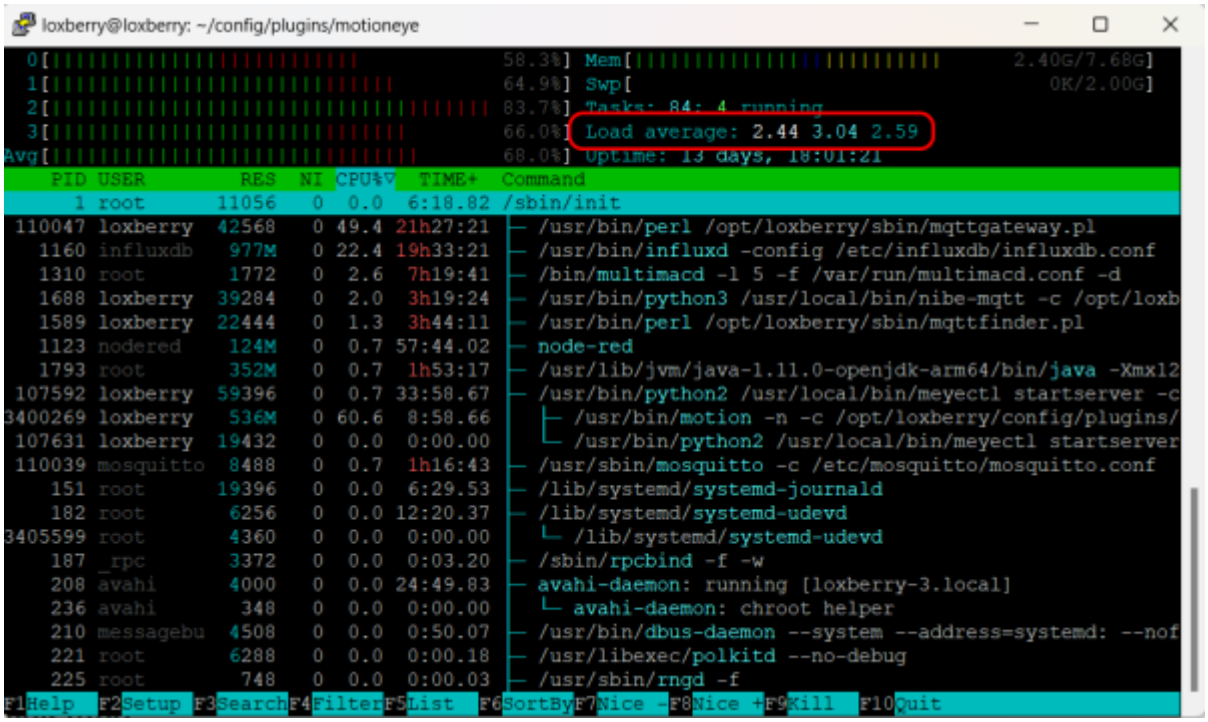


MotionEye erkennt die Kamera als MJPEG Network Camera. Das wars's! Nun könnt ihr die Kamera ganz normal nutzen und zum Beispiel bei Bewegungserkennung eine Aufzeichnung starten.

Für die volle Auflösung vergesst nicht diesen Tipp zu befolgen: Obwohl ich einen Stream in voller Auflösung eingerichtet habe, hat der MJPEG-Stream in MotionEye nur 640x480 Pixel. Warum?

Systemleistung

Ich habe bei mir auf diese Weise 5 Kameras eingebunden - die Streams haben eine Größe von 1920x1080 Pixeln. MotionEye läuft dabei auf einem Raspberry Pi 4 ohne Lüfter (aber mit großem Kühlkörper) völlig problemlos. Die CPU-Auslastung liegt bei ca. 50%.



From:
https://wiki.loxberry.de/ - **LoxBerry Wiki - BEYOND THE LIMITS**

Permanent link:
https://wiki.loxberry.de/plugins/motioneye_plugin/mjpegstream_aus_einzelcaptures?rev=1699693297

Last update: **2023/11/11 10:01**