

Plugin-Daten	
Autor	<a href="#">Michael Schlenstedt</a>
Logo	
Status	STABLE
Version	1.0.3.0
Min. LB Version	3.0.0.1
Release Download	<a href="https://github.com/mschlenstedt/LoxBerry-Plugin-MotionEye/archive/refs/tags/loxberry-plugin-motioneye-1.0.3.0.zip">https://github.com/mschlenstedt/LoxBerry-Plugin-MotionEye/archive/refs/tags/loxberry-plugin-motioneye-1.0.3.0.zip</a>
Beschreibung	Das MotionEye Plugin macht den LoxBerry zum Kamera Surveillance System - inkl. Bewegungserkennung und Aufzeichnung. Alle Videostreams werden als MJPEG-Stream zur Verfügung gestellt.
Sprachen	EN
Diskussion	<a href="https://www.loxforum.com/forum/projektforen/loxberry/plugins/331116-plugin-motioneye-kamera-surveillance-system-f%C3%BCr-den-loxberry">https://www.loxforum.com/forum/projektforen/loxberry/plugins/331116-plugin-motioneye-kamera-surveillance-system-f%C3%BCr-den-loxberry</a>

# MotionEye Plugin

Version History...

[Änderungshistorie](#)

## Aufgabe des Plugins

Das MotionEye Plugin installiert [die Software MotionEye](#) auf dem LoxBerry. Damit wird er zum vollwertigen Kamera Surveillance System. Zudem konvertiert die Software beliebige Kamera-/Video-Streams in MJPEG-Streams um, sodass auch Kameras, die kein MJPEG bieten, mit dem Türsteuerungsbaustein verwendet werden können. Die Konvertierung übernimmt dabei die Software [ffmpeg](#). Ffmpeg kann so ziemlich alle gängigen Codecs und Videoformate decodieren.

MotionEye bietet dabei folgende Features:

- kompatibel mit den meisten **USB Kameras** und dem Raspberry PI Kamera Modul
- Support für **IP (network) Kameras**
- **Motion Detection** mit Email Notifications, Zeitplan und **Schalten von virtuellen Eingängen am Miniserver**
- **Aufzeichnung bei Ereignis oder permanent, JPEG** Dateien für Images, **AVI/MP4** Dateien für Videos
- **Aufzeichnung aus Loxone heraus starten** (zum Beispiel per Bewegungsmelder)
- Uploading zu **Google Drive** und **Dropbox** sowie auf **externe Datenträger** und **Netzwerk-Shares**

## Einige Worte zur Systemauslastung... Lesen!



### Ernstgemeinte Warnung!

Die Verarbeitung von Videostreams benötigt erhebliche Systemressourcen.



Voraussetzung für eine vernünftige Nutzung ist ein **Raspberry Pi 4** mit großem Kühlkörper **und eventuell** einem Lüfter. Zudem muss ein gutes, **leistungsstarkes Netzteil** verwendet werden. Handy-Ladegeräte sind keine Netzteile! Wieviele Videostreams gleichzeitig verarbeitet werden können, hängt zudem auch von der Auflösung, Bitrate und Frames/s der Streams ab.

MotionEye kann technisch gesehen mehr oder weniger jeden Stream in jeder Qualität in einen MJPEG-Stream wandeln, Bewegung auswerten, aufzeichnen. Und das auch für beliebig viele Kameras. Aber die Hardware begrenzt die Möglichkeiten stark! Die Konvertierung eines Videostreams benötigt sehr viel CPU-Last - je höher die Qualität des Streams ist (Auflösung, Frames/s, Bitrate), desto höher sind die Anforderungen an die Hardware.

Soll der Stream konvertiert werden (z. B. aus einem H.264 Stream), gilt es als ersten Schritt die Videoqualität *in der Kamera* so gering wie nur irgendwie möglich einzustellen. Man sollte die Videogröße und die Bitrate soweit herunternehmen, bis das Bild gerade noch akzeptabel ist. Diesen Stream dann in MotionEye konvertieren.

Am Besten werden überhaupt keine Streams konvertiert. Aktiviert wenn möglich den MJPEG-Stream in der Kamera und lasst ihn von MotionEye nur re-streamen für die Türsteuerungsbausteine in Loxone. Das verbraucht sehr geringe Ressourcen.

Wenn ihr H264 Streams von der Kamera verarbeiten wollt, verzichtet auf eine Konvertierung in MotionEye!

Verzichtet auf die Bewegungserkennung wenn möglich! Diese verbraucht "gigantische" Ressourcen und lässt sich sinnvoll nur in einer VM nutzen. Aber selbst dann sind die Ergebnisse eher ernüchternd... Nutzt wenn irgendwie möglich die Bewegungserkennung in eurer Kamera und triggert dann über HTTP z. B. einen Virtuellen Eingang in Loxone.

Nachdem man die Kameras in MotionEye aktiviert hat, sollte man die CPU-Auslastung und die CPU-Temperatur des LoxBerrys kontrollieren. Schließt dazu MotionExe im Browser (die Webseite verbraucht durch die Anzeige viele Ressourcen), öffnet dann das Terminal Widget und startet im Terminal "htop":

```

0[|||||] Mem[|||||] 37.7% Mem[|||||] 2.10G/7.68G
1[|||||] Swp[|||||] 24.0% Swp[|||||] 8K/2.00G
2[|||||] Tasks: 77; 1 running
3[|||||] Load average: 1.97 2.02 1.96
Avg[|||||] Uptime: 5 days, 17:59:12
34.6%

PID USER RES NI CPU% MEM+ Command
1 root 10972 0 0.0 5.26 /sbin/init
1160 influxdb 932M 0 18.8 10:52:39 /usr/bin/influxd -config /etc/influxd/influxdb.conf
110047 loxberry 42568 0 5.4 60:10:16 /usr/bin/perl /opt/loxberry/sbin/mqttgateway.pl
1310 root 1772 0 2.1 10:44:57 /bin/multimacd -l 5 -f /var/run/multimacd.conf -d
1688 loxberry 38276 0 7.0 10:03:56 /usr/bin/python3 /usr/local/bin/nibe-mqtt -c /opt/loxb
1589 loxberry 23140 0 1.3 10:11:14 /usr/bin/perl /opt/loxberry/sbin/mqttfinder.pl
1522 nobody 3268 0 0.7 0:00:09 /usr/bin/shellinboxd -b --no-beep --localhost-only -t -
1523 nobody 1968 0 0.0 0:00:00 /usr/bin/shellinboxd -b --no-beep --localhost-only -
2966466 root 4240 0 0.0 0:00:00 /usr/bin/shellinboxd -b --no-beep --localhost-only
2966467 loxberry 6892 0 0.0 0:00:48 ssh loxberry@localhost
1793 root 352M 0 0.7 35:52:79 /usr/lib/jvm/java-1.11.0-openjdk-arm64/bin/java -Xmx128M
110039 mosquitto 8488 0 0.7 22:14:34 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf
151 root 20164 0 0.0 2:11:28 /lib/systemd/systemd-journald
182 root 6256 0 0.0 4:09:97 /lib/systemd/systemd-udev
2966681 root 4360 0 0.0 0:00:00 /lib/systemd/systemd-udev
187 _rpc 3372 0 0.0 0:01:01 /sbin/rpcbind -f -w
208 avahi 4000 0 0.0 8:36:23 avahi-daemon: running [loxberry-3.local]
236 avahi 348 0 0.0 0:00:00 avahi-daemon: chroot helper
210 messagebu 4508 0 0.0 0:17:00 /usr/bin/dbus-daemon --system --address=systemd: --nofor
  
```

Achtet hier auf die CPU- und Speicherauslastung. Beobachtet auch die Load Average. Die drei Werte bedeuten: Aktuelle Auslastung, 5-Minuten-Durchschnitt, 15-Minuten-Durchschnitt. Ist die Loadaverage gleich der CPU-Anzahl, habt ihr eine Volllastung des Systems (100%). Ist sie kleiner, haben die CPUs noch Reserve. Ist die Load Average größer als die Anzahl der CPUs, schaffen die CPUs es nicht alle Anfragen in der gewünschten Zeit zu verarbeiten. Im Screenshot ist die Load Average ungefähr 2 und der Raspberry 4 hat 4 CPU Kerne. Das bedeutet eine Auslastung von 50%.

## Download

- Das Plugin-Archiv (ZIP) kann auf GIT-Hub heruntergeladen werden: <https://github.com/mschlenstedt/LoxBerry-Plugin-MotionEye/releases>
- Der Sourcecode ist auf GitHub verfügbar: <https://github.com/mschlenstedt/LoxBerry-Plugin-MotionEye>

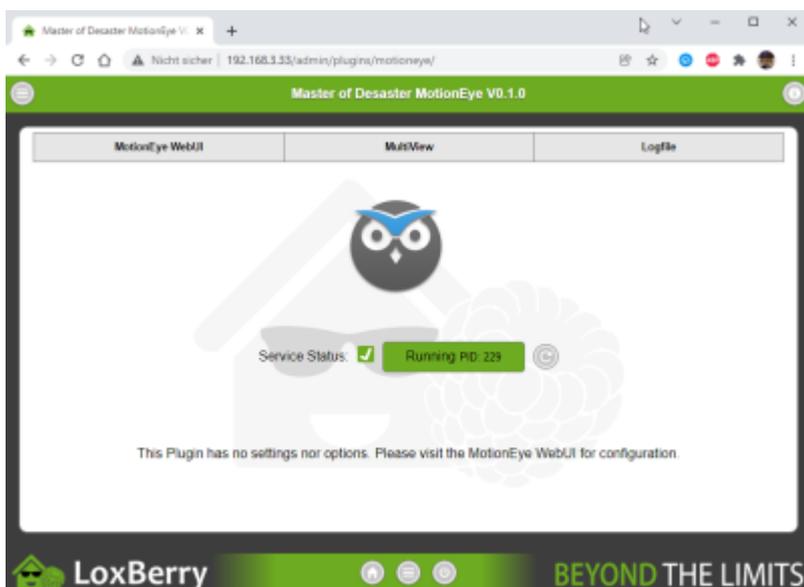
## Fehlerberichte

Fehlerberichte bitte direkt im [Repository auf GitHub](#) melden.

## Konfigurationsoptionen

Das Plugin bietet keine Konfigurationsoptionen an. Sämtliche Einstellungen werden in MotionEye durchgeführt.

Für eine Erklärung sämtlicher Optionen empfehle ich Euch die entsprechenden Tutorials auf YouTube: [https://www.youtube.com/results?search\\_query=MotionEye](https://www.youtube.com/results?search_query=MotionEye)



# Erkannte Bewegung an Loxone melden



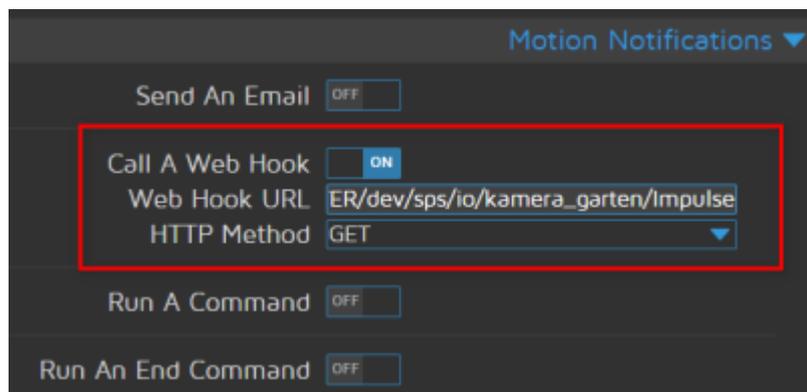
## Warnung

Die softwareseitige Bewegungserkennung benötigt **extrem hohe Systemleistung**. Ich empfehle dringend dieses Feature **nicht zu nutzen!**

Neben den Aktionen, die innerhalb von MotionEye konfiguriert werden können, wenn von der Kamera eine Bewegung erkannt wurde, kann auch eine externe Webseite bei Bewegungserkennung aufgerufen werden. So ist es möglich in Loxone einen virtuellen Eingang bei Bewegungserkennung zu schalten und so weitere Aktionen in LoxoneConfig zu programmieren (Licht an z. B.).

Dazu gebt ihr in MotionEye in der Kamerakonfiguration unter "Motion Notifications" eine sogenannte Web Hook URL an. Diese schaltet den Virtuellen Eingang innerhalb von Loxone. Bitte schaut in die Doku von Loxone bzw. ins Wiki, wie sich die URLs zusammensetzen: [REST Webservice](#) oder <https://www.loxone.com/dede/kb/webservices/>

Eine URL für den Eingang/Baustein "kamera\_garten" sieht z. B. so aus (Impuls): **http://LOXONE\_USER:LOXONE\_PASSWORT@IP\_MINISERVER/dev/sps/io/kamera\_garten/Impulse**



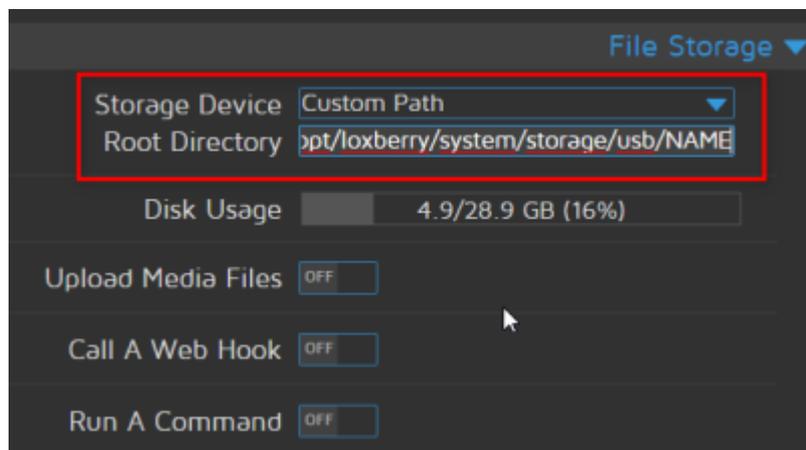
# Kamera Streams auf externem Datenträger/Netzwerkshare speichern



Ich empfehle dringend, die Streams ausschließlich auf einer SSD per USB oder per Netzwerk-Share auf einem NAS zu speichern. Nutzt auf keinen Fall die SD-Karte, auch ein USB-Stick kann permanente Speichervorgänge nicht dauerhaft verkraften.

Richtet zunächst im LoxBerry euer Netzwerkshare oder aber einen externen Datenträger über das Netshares- oder USB Storage-Widget ein: [Widget Netshares](#) oder [Widget USB Storages](#)

Im System werden sowohl die Netzwerkshares als auch die USB Datenträger unter dem Pfad **/opt/loxberry/system/storage/usb/NAME** bzw. **/opt/loxberry/system/storage/smb/NAME** eingebunden. Wenn ihr im Widget auf "Browse" geht, seht ihr den exakten Pfad. Der Pfad zu eurem Datenträger oder Netzwerkshare muss nun in MotionEye unter "Filestorage" als "Custom Path" eingetragen werden:



Achtung! Standardmäßig ist hier das Datenverzeichnis des Plugins auf eurer SD-Karte vorausgewählt. Ich empfehle Euch **dringend** keine großen Datenmengen permanent auf der SD-Karte zu speichern! Die Speicherkarte könnte dadurch Schaden nehmen.

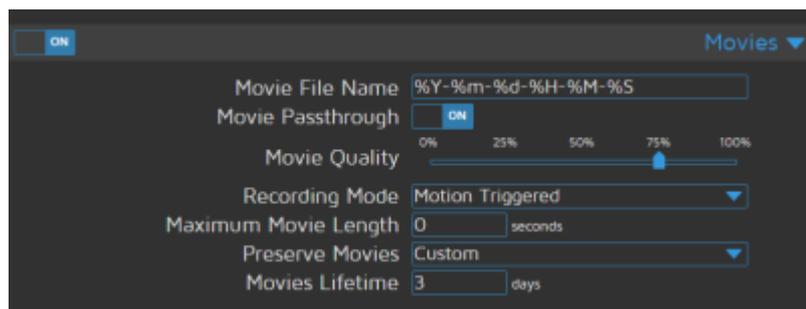
## Aufnahme per Loxone starten / per Mail versenden / abrufen

Mit Version 1.0.1 des Plugins können aus Loxone heraus Video- oder Fotoaufnahmen getriggert werden. So kann z. B. bei erkannter Bewegung über einen Bewegungsmelder die Aufzeichnung des Bildes für X Minuten gestartet werden.

Zunächst legt ihr einen passenden Pfad an, unter dem die Aufnahmen abgelegt werden sollen, siehe Kapitel oben.

### Movies

Wenn ihr Videoaufnahmen per Trigger speichern wollt, aktiviert ihr die Sektion "Movies":

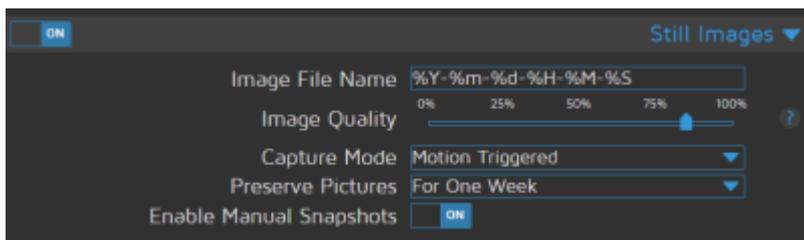


Es empfiehlt sich hier, **Movie Passthrough** zu aktivieren. Damit wird das Kamerabild 1:1 ohne

irgendwelche Konvertierung in ein MP4-File geschrieben. Das benötigt die geringsten Systemressourcen. Als Recording Mode wählt ihr Motion Triggered. Legt nun noch fest, wie lange die Aufnahmen gespeichert werden sollen. Nach dieser Zeit wird die Aufnahme automatisch gelöscht. Die Sektion "Motion Detection" weiter unten in den MotionEye-Einstellungen muss **nicht** aktiviert werden!

## Still Images

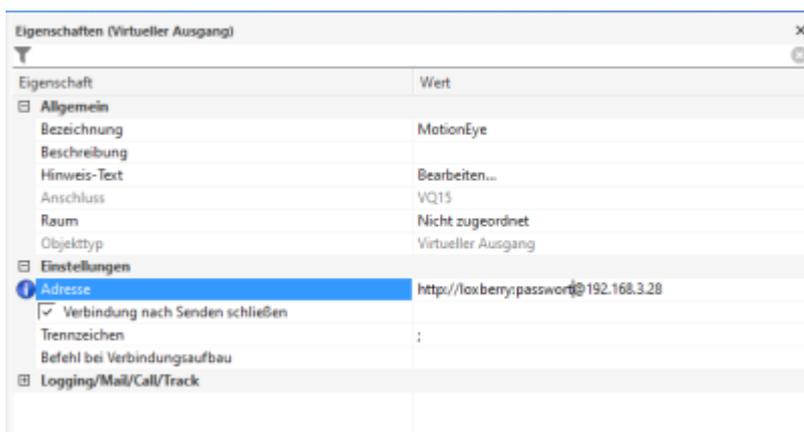
Wenn ihr lediglich ein oder mehrere Standbilder per Trigger speichern wollt, aktiviert ihr die Sektion "Still Images":



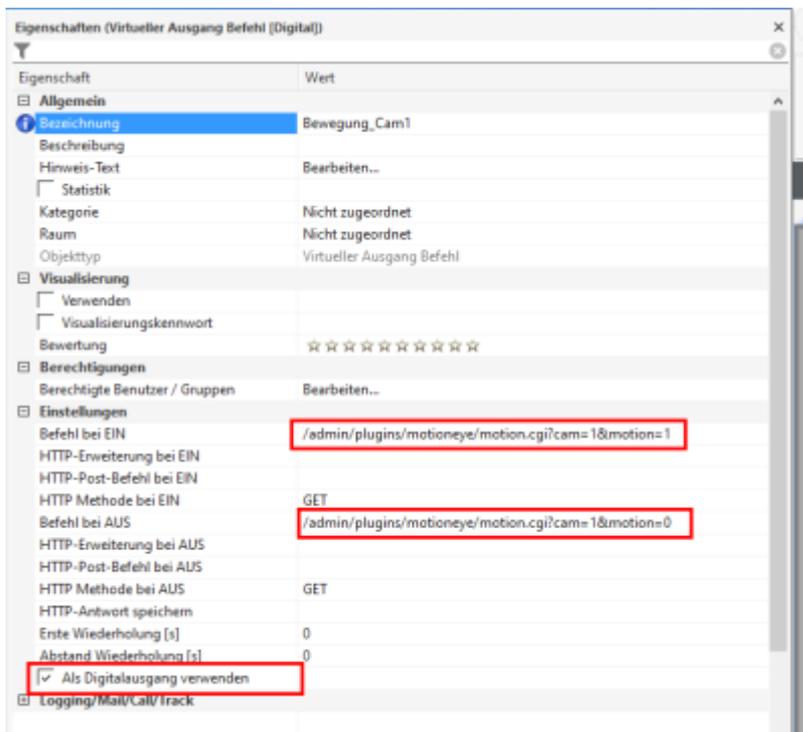
Als Capture Mode wählt ihr Motion Triggered oder Motion Triggered (One Picture) . Bei Motion Triggered wird pro Sekunde ein Bild gespeichert, solange die Bewegungserkennung aktiv ist. Im anderen Fall wird nur ein Bild zu Beginn gespeichert. Legt nun noch fest, wie lange die Aufnahmen gespeichert werden sollen. Nach dieser Zeit wird die Aufnahme automatisch gelöscht. Die Sektion "Motion Detection" weiter unten in den MotionEye-Einstellungen muss **nicht** aktiviert werden!

## Trigger über Loxone Config

In Loxone Config legt ihr Euch einen Virtuellen Ausgang an:



Die Adresse lautet: `http://loxberry:passwort@<ip_loxberry>`, z. B. `http://loxberry:passwort@192.168.3.28` . Die IP-Adresse, Username und Passwort müsst ihr natürlich wie immer anpassen. Unterhalb des virtuellen Ausgangs legt ihr nun einen Virtuellen Ausgang Befehl an:

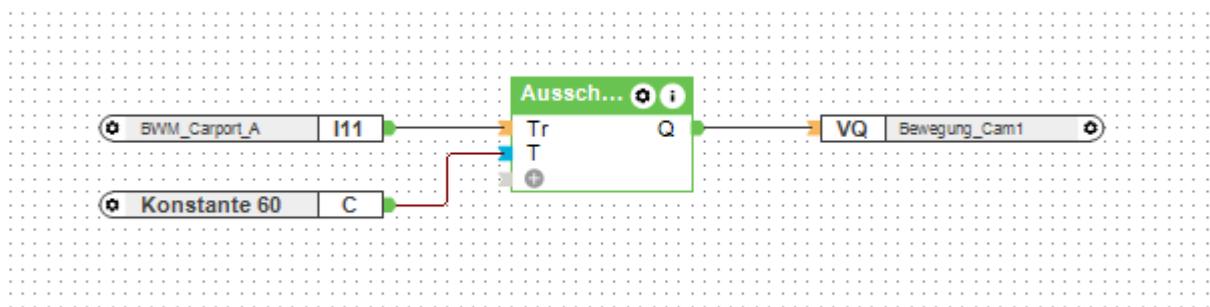


Befehl bei EIN: /admin/plugins/motioneye/motion.cgi?cam=1&motion=1

Befehl bei AUS: /admin/plugins/motioneye/motion.cgi?cam=1&motion=0

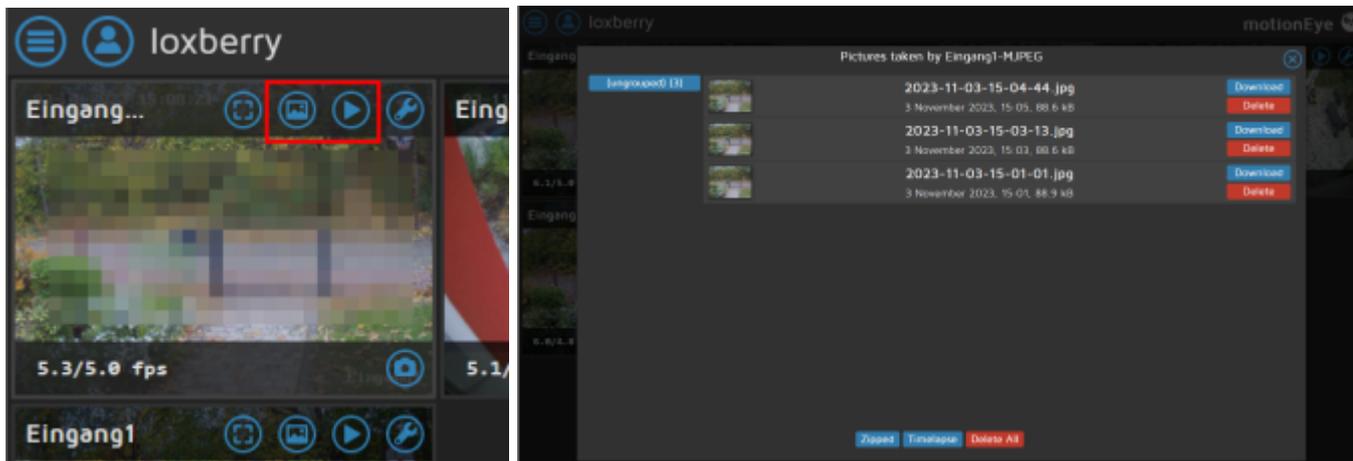
Als Digitaleingang verwenden: Aktiviert

Der Parameter cam enthält die Camera-ID, die ihr ansprechen wollt. Diese findet ihr in der MotionEye Weboberfläche in der Sektion "Video Device". Mit motion=1 aktiviert ihr die Aufzeichnung, mit motion=0 stoppt ihr die Aufzeichnung wieder. In dem ihr vor den Ausgang eine Ausschaltverzögerung hängt, könnt ihr die Dauer der Aufzeichnung bei Bewegungserkennung anpassen:

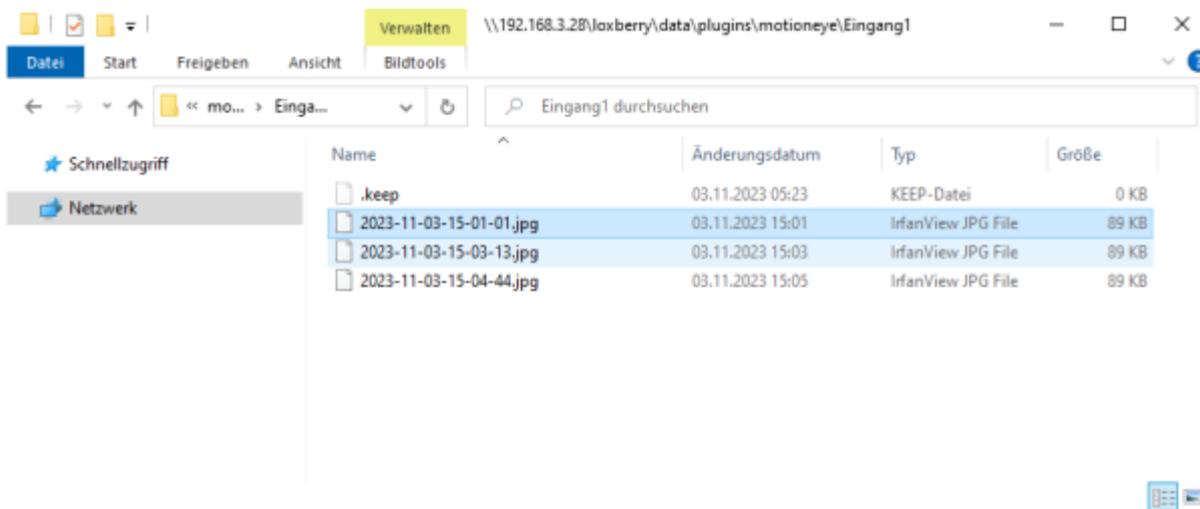


### Aufnahmen anschauen

Die Aufnahmen könnt ihr über die Weboberfläche von MotionEye erreichen und von dort auch direkt im Browser abspielen. Klickt dazu in der Weboberfläche auf das Picture- oder Play-Symbol. In der Übersicht lassen sich die Dateien auch herunterladen oder manuell löschen, zippen oder ein Timelapse Video erstellen.



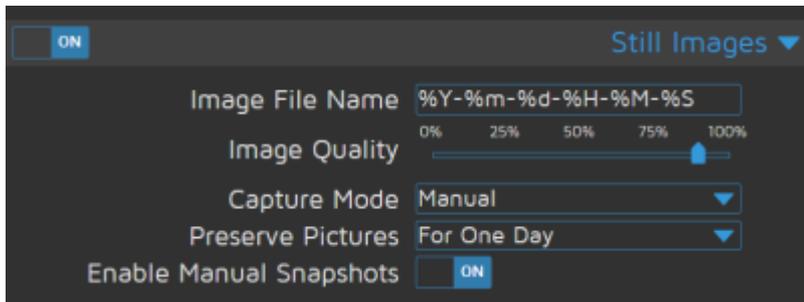
Eine andere Alternative ist per Netzwerkfreigabe auf den LoxBerry zuzugreifen und die Dateien direkt zu öffnen bzw. zu kopieren. Dazu öffnet ihr ein Explorer-Fenster und gebt zwei Backslashe gefolgt von der IP eures LoxBerrys ein, z. B. \\192.168.3.28. Wechselt in die Loxberry-Freigabe und geht in den Ordner \system\storage. Dort findet ihr eure Netzwerkfreigaben oder USB-Geräte und könnt direkt auf die einzelnen Dateien zugreifen. Solltet ihr die Dateien im Datenverzeichnis des Plugins speichern, ist der Pfad \data\plugins\motioneye.



### Aufnahme per Mail versenden

Möchte man die Aufnahme automatisch per Mail versenden, so ist das ebenfalls möglich (z. B. das Kamerabild beim Klingeln direkt per Email versenden). Natürlich muss zunächst Euer Mailserver im [LoxBerry Mail Widget](#) konfiguriert werden.

Aktiviert nun in der Sektion "Still Images" manuelle Captures - die Speicherzeit wählt ihr möglichst niedrig (außer ihr wollt die Bilder aufbewahren). Idealerweise speichert ihr die Bilder nicht auf der SD-Karte sondern auf einem externen Datenträger (z. B. USB-Stick).



Nun legt ihr wieder einen Virtuellen Ausgangsbefehl an (siehe oben):

Befehl bei EIN:

/admin/plugins/motioneye/motion.cgi?cam=1&mail=1&to=DeineEmail@example.com

Als Digitaleingang verwenden: Aktiviert

Mögliche Optionen/Parameter für den Befehl:

Option	Optional	Beispiel	Beschreibung
cam=		cam=1	Kamera ID, von der der Capture geholt wird
mail=		mail=1	Mail verschicken
to=		to=DeineEmail@example.com	Empfänger-Mailadresse. Mehrere Adressen mit Kommata trennen.
wait=	optional	wait=4	Wartezeit in Sekunden, bevor das Capture geholt wird. Hiermit kann etwas justiert werden, falls die Person noch nicht ganz im Bild ist oder das Kamerabild leicht verzögert ist.
text=	optional	text=Dieses ist eine Email	Body-Text der Email, Zeilenumbruch kann mit %0A eingefügt werden.
subject=	optional	subject=Eine neue Mail	Subjekt der Email
resize=	optional	resize=800x600	Bild-Größe, ohne Angabe wird die Größe nicht verändert. Die Größe kann in Breite x Höhe angegeben werden, alternativ nur die Breite (die Höhe wird dann im passenden Verhältnis skaliert) oder in % (z. B. 50%).



Achtet drauf, dass die Texte (Body-Text und Subject) zwingend URL-encoded sein müssen! Hierzu könnt ihr Konverter im Internet verwenden oder den gesamten Befehl einfach einmalig im Browser ausführen und dann aus der Adressleiste in Loxone kopieren. Der Browser wandelt den Aufruf automatisch in URL-encoded um.

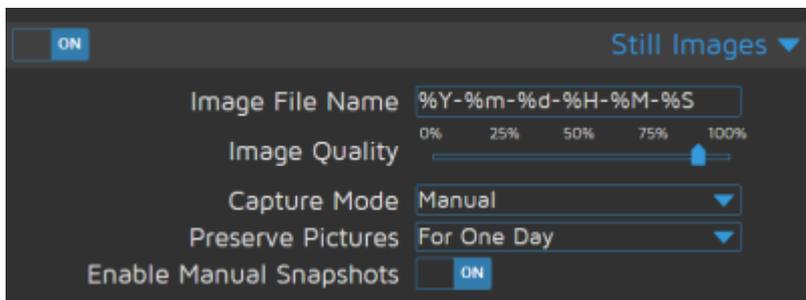
Solltet ihr Probleme mit der Größe des angehängten Bildes haben (zu klein), dann beachtet bitte den Hinweis aus den FAQ: [Obwohl ich einen Stream in voller Auflösung eingerichtet habe, hat der MJPEG-Stream in MotionEye nur 640x480 Pixel. Warum?](#)

### Aufnahme / Snapshot abrufen

Über das Plugin kann auch einfach ein aktueller Snapshot abgerufen werden. Das kann eine

Alternative für die URL im Kamerabaustein sein, wenn man aus Leistungsgründen den MJPEG-Stream nicht verwenden möchte oder aber das Bild noch einmal für Loxone resizen möchte (geht leider über MotionEye nicht). Loxone erzeugt aus den Einzelbildern, die abgerufen werden, dann wieder automatisch einen Stream.

Aktiviert in der Sektion "Still Images" manuelle Captures - die Speicherzeit wählt ihr möglichst niedrig (außer ihr wollt die Bilder aufbewahren). Idealerweise speichert ihr die Bilder nicht auf der SD-Karte sondern auf einem externen Datenträger (z. B. USB-Stick).



Nun legt ihr wieder einen Virtuellen Ausgangsbefehl an (siehe oben):

Befehl bei EIN: /admin/plugins/motioneye/motion.cgi?cam=1&show=1

Als Digitaleingang verwenden: Aktiviert

Mögliche Optionen/Parameter für den Befehl:

Option	Optional	Beispiel	Beschreibung
cam=		cam=1	Kamera ID, von der der Caputre geholt wird
show=		show=1	Aktuelles Kamerabild anzeigen
resize=	<i>optional</i>	resize=800x600	Bild-Größe, ohne Angabe wird die Größe nicht verändert. Die Größe kann in Breite x Höhe angegeben werden, alternativ nur die Breite (die Höhe wird dann im passenden Verhältnis skaliert) oder in % (z. B. 50%).

Solltet ihr Probleme mit der Größe des angehängten Bildes haben (zu klein), dann beachtet bitte den Hinweis aus den FAQ: [Obwohl ich einen Stream in voller Auflösung eingerichtet habe, hat der MJPEG-Stream in MotionEye nur 640x480 Pixel. Warum?](#)

## Multiview



Nur Kameras, für die das "Video Streaming" aktiviert wurde, erscheinen auch in der Multiview!

Das Plugin fügt jede Kamera, für die das Video Streaming aktiviert wurde, in eine simple Webseite ein, die über den [Webage-Baustein](#) in die Loxone Config integriert werden kann. So kann man sich mit einem Klick alle Kameras auf einen Blick aus der Loxone App heraus anschauen. Ein Klick auf das jeweilige Kamerabild öffnet den Kamerastream im Großformat.



## Weitere Artikel

- [LoxBerry als Überwachungskamera](#)
- [Aus einzelnen Caputres ein MJPEG-Stream erzeugen](#)

## FAQ - Häufig gestellte Fragen

### Installation

**F: Läuft MotionEye auch auf einem Raspberry 1, 2, 3?**

Nein. Du brauchst einen leistungsfähigen Rechner. Es ist mindestens ein Raspberry Pi 4 oder gleichwertig notwendig.

**F: MotionEye lässt sich nicht installieren. Ich erhalte im Log die Fehlermeldung dpkg: error processing archive /[...]bullseye\_motion\_4.5.1-1\_arm64.deb (--install): package architecture (arm64) does not match system (armhf)?**

Du hast Deinen LoxBerry mit dem falschen Image installiert. Du musst das 64Bit Image (AARCH64) verwenden. Dein LoxBerry läuft auf einem 32Bit Image.

### Leistung

**F: Ich habe mehrere RSTP-Kamerastreams eingerichtet und mein LoxBerry ist ausgelastet und reagiert träge.**

RSTP Streams zu dekodieren erfordert extrem viel Performance. Du kannst einen Trick verwenden und Snapshots Deiner Kamera in voller Auflösung zu einem MJPEG-Stream zu konvertieren. Dieser benötigt kaum Systemressourcen, um ihn in MotionEye weiter zu verarbeiten. Nachteil ist, dass die Bildwiederholungsrate sehr niedrig ist (1 Bild pro Sekunde in etwa). Für eine Überwachungskamera reicht das aber in aller Regel aus.

In diesem Artikel kannst Du nachlesen wie das geht: [Aus einzelnen Caputres ein MJPEG-Stream erzeugen](#)

## Videobild

**F: Obwohl ich einen Stream in voller Auflösung eingerichtet habe, hat der MJPEG-Stream in MotionEye nur 640x480 Pixel. Warum?**

Das ist ein Fehler in der Software Motion, auf der MotionEye basiert. Motion verwendet immer 640x480 und ignoriert die Größe des eingehenden Streams. Gebe bei der Option Extra Motion Options die echten Abmaße des eingehenden Kamerastreams wie folgt ein:

```
height 1080  
width 1920
```

Danach hat der MJPEG-Stream die volle Größe.

**F: Obwohl ich unter Video Streaming die Option "Streaming Image Resizing" aktiviert habe, ist die Streamgröße unverändert. Warum?**

Diese Option verändert nur die Streamgröße in der MotionEye WebUI und nicht die Größe des MJPEG-Streams oder des Snapshots. Schau hier wie Du die Größe trotzdem ändern (verkleinern) kannst: [Aufnahme / Snapshot abrufen](#)

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

Permanent link:  
[https://wiki.loxberry.de/plugins/motioneye\\_plugin/start](https://wiki.loxberry.de/plugins/motioneye_plugin/start)

Last update: **2024/06/09 18:42**