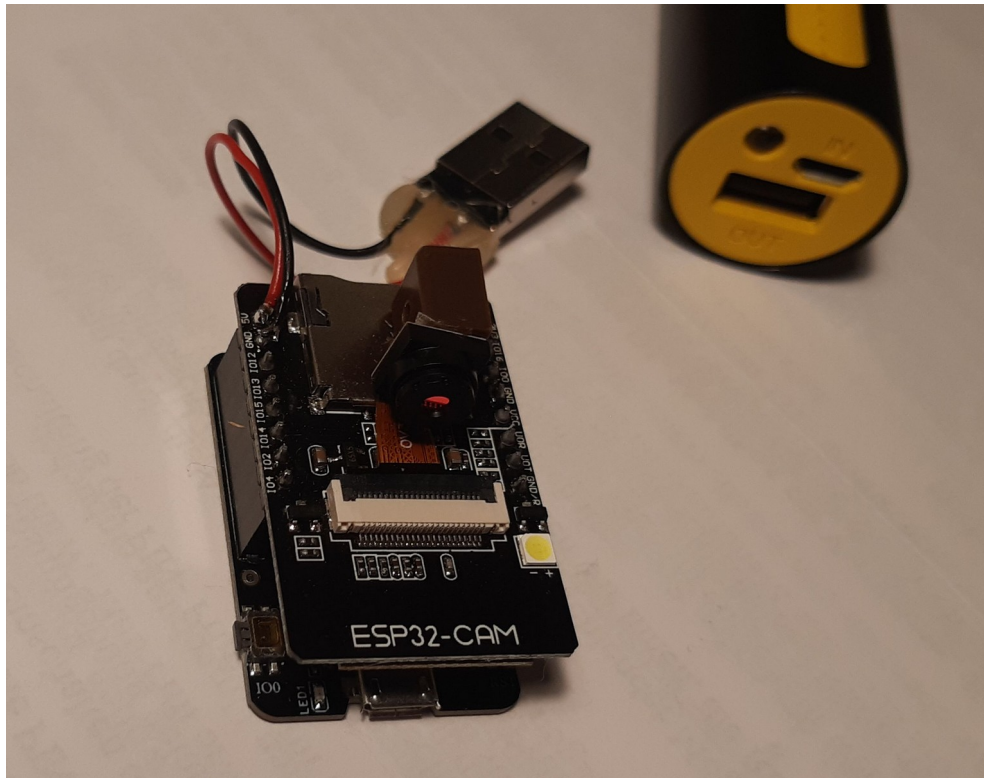


ESP32-CAM Modul mit Kamera OV3660



Stand: 14.03.2026

Inhaltsverzeichnis

ESP32-CAM Modul mit Kamera OV3660.....	1
Stand: 14.03.2026.....	1
Funktionsweise:.....	3
Sicherheit:.....	3
Lieferung:.....	3
Stromversorgung:.....	3
WLAN:.....	3
Inbetriebnahme:.....	4
Meine Anpassungen im Sketch:.....	5
Konfiguration ESP32-CAM OV3660 im Browser.....	6
Mögliche Konfigurationswerte zur Kamera OV3660.....	7
ESP32-CAM mit Kamera OV3660 steuern.....	10
Mein Script zur ESP32-Cam OV3660 Konfiguration.....	11
Streamen.....	11
mit VLC ein Video erstellen und als .mp4 abspeichern.....	12
Windows - Verbindungs- Test im Windows Gerätemanager:.....	13
Infos, Links:.....	13
ESP32-CAM:.....	13
Nachfolger: ESP32-S3-CAM.....	14

Funktionsweise:

Ein Videostream wird von der kleinen Kamera über WLAN übertragen und kann in einem Browser (Firefox, Edge, Chrom,..) oder VLC, ... angezeigt werden.
Zeitgleich kann immer nur eine Verbindung zur ESP32-CAM aufgebaut werden.
(Browser oder VLC, ...)

Sicherheit:

Es wird keine Verbindung ins Internet oder Cloud und auch keine APP benötigt, der Videostream wird unverschlüsselt im lokalen Netz übertragen.

Lieferung:

Das kleine ESP32-CAM Modul gibt es kostengünstig auf den bekannten Web Shops.
Bestellt: ESP32-CAM inkl. OV2640 Kamera, geliefert wurde das neuer Model:
ESP32-CAM-MB mit OV3660.

Stromversorgung:

Zur Stromversorgung der ESP32-CAM verwende ich ein Powerbank, welche in einem Waggon liegt und von einer Lok geschoben wird.

Das Modul benötigt ca. 200 - 300 mA je nach Stream (96x96 – 2048x1596) LED aus, und bis zu 700 mA bei je nach LED Beleuchtung.

WLAN:

Eine externe WLAN Antenne am ESP32-CAMModul ist nicht notwendig, der Empfang ist mit der integrierten Antenne bei mir gut.

Die WLAN IP-Adresse wurde auf der FritzBox fest eingestellt, reserviert.

Inbetriebnahme:

Installation der **Arduino-IDE** <https://www.arduino.cc/en/software/>

Verwendung von Beispiel Sketch aus der Arduino-IDE.

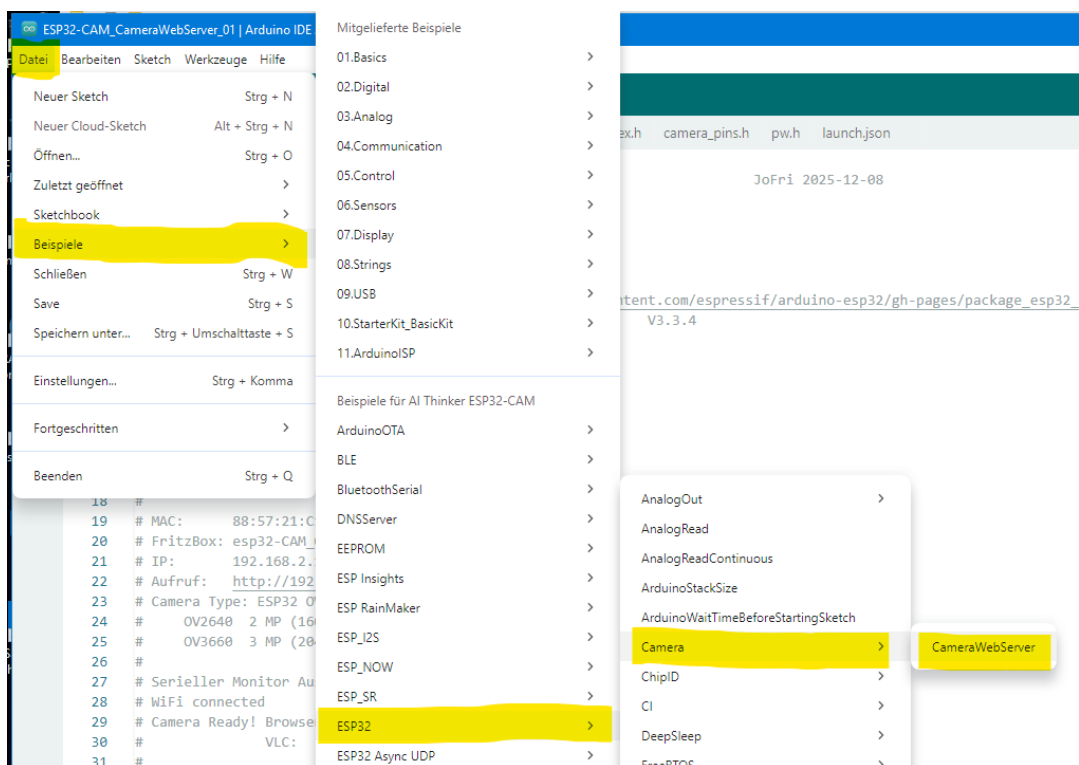
Zusätzlicher Boardverwalter-URLs:

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

Board Installation: **esp32** von Espressif System V3.3.4

Board Auswahl in IDE: **AI Thinker ESP32-CAM** und des **Ports**

Verwendet wurde aus Arduino-IDE 2.3.7 Beispiele – ESP32 – Camera – CameraWebServer



Unterstützte Camera Typen von ESP32-CAM

OV2640	2 MP (1600×1200)	CMOS bei den meisten ESP32-CAM-Modulen
OV3660	3 MP (2048×1536)	CMOS neu, höhere Auflösung
OV5640	5 MP (2592x1944)	

Meine Anpassungen im Sketch:

CameraWebServer.ino

```
const char *ssid = "*****";
const char *password = "*****";

// add für feste IP-Adresse - bei DHCP nächste 3 Zeilen löschen
IPAddress ip(192,168,2,114);      // IP Adresse im Heimnetzwerk
IPAddress gateway(192,168,2,1);   // Gateway (darf falsch sein)
IPAddress subnet(255,255,255,0);  // Subnetmask

config.xclk_freq_hz = 200000000;  // → 180000000; MHZ Taktfrequenz

s->set_saturation(s, -2);          // → 4 saturation
s->set_framesize(s, FRAMESIZE_QVGA);  // → FRAMESIZE_VGA
```

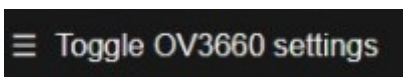
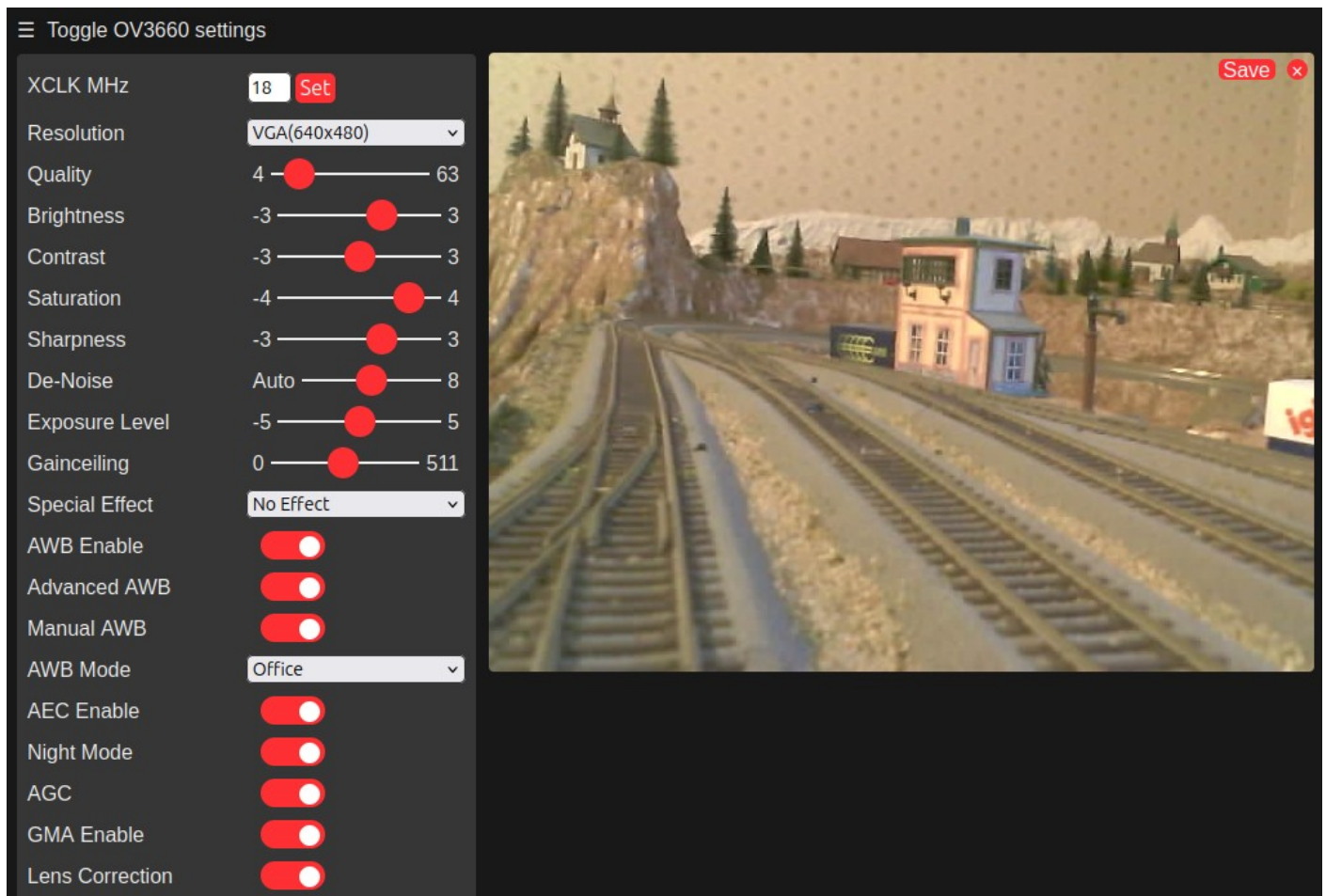
board_config.h Select camera model

```
// #define CAMERA_MODEL_ESP_EYE // Has PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
```

Konfiguration ESP32-CAM OV3660 im Browser

Die Konfigurationen des Moduls können über den Browser oder Script erfolgen.

Aufruf im Browser: <http://<IP-Adresse>/>



mit Klick auf die 3 Striche kann das Steuermenü aus und wieder eingeblendet werden.

Der Webserver der Kamera OV3660 erlaubt es mittels HTTP, Befehle an ihn zu senden, der generelle Aufruf lautet:

Befehl: <http://<ip>/control?var=<Variablenname>&val=<Wert>>

(meine Kamera hat die IP Adresse : 192.168.2.114)

Mögliche Konfigurationswerte zur Kamera OV3660

Befehls Tabelle für HTTP Requests zur ESP Kamera mit OV3660

fette Werte ist der Standard, **grünen Werte** sind meine Anpassung.

Diese Tabelle ist ein Auszug , nicht komplett.

Beschreibung OV3660 settings	Variablenname Command	Werte
XCLK Takt Frequenz	Xclk http://192.168.2.114/xclk? xclk=19	20 MHz 18 MHZ besser weniger max. 19 sonst ping timeout
	pixformat	PIXFORMAT_JPEG
Resolution Auflösung	Framesize http://192.168.2.114/control? var=framesize&val=7	19 QXGA (2028x1564) 18 P3MP (864x1564) 17 P HD (720x1280) 16 FHD (1920x1080) 15 UXGA(1600x1200) 14 SXGA(1280x1024) 13 HD (1280x720) 12 XGA(1024x768) 11 SVGA(800x600) 10 VGA(640x480) 9 HVGA (480x320) 8 CIF(400x296) 7 (320x320) 6 QVGA(320x240) 5 (240x240) 4 HQVGA(240x176) 3 (176x144) 2 (128x128) 1 QQVGA(160x120) 0 (96x96)
Quality Bildqualität	Quality http://192.168.2.114/control? var=quality&val=10	4 gut bis 63 schlecht 10 Voreingestellt
Brightness Helligkeit	Brightness http://192.168.2.114/control? var=brightness&val=1	-3 bis 3 1 Voreingestellt
Contrast Kontrast	Contrast http://192.168.2.114/control? var=contrast&val=0	-3 bis 3 0 Voreinstellung
Saturation Sättigung	Saturation http://192.168.2.114/control? var=saturation&val=1	-4 bis 4 -2 Voreingestellt
Sharpness Bildschärfe	Sharpness http://192.168.2.114/control? var=sharpness&val=1	-3 bis 3 1 Voreinstellung
DE-Noise ?	Denoise http://192.168.2.114/control? var=denoise&val=4	Auto 0 bis 8 4 Mitte

Exposure Level	ae_level http://192.168.2.114/control? var=ae_level&val=0	-5 bis 5 0 Voreinstellung
Gainceiling	Gainceiling http://192.168.2.114/control? var=gainceiling&val=250	0 bis 511 250 Mitte
Spezial Effect	special_effect http://192.168.2.114/control? var=special_effect&val=0	0 no Effekt 1 Negativbild 2 Schwarz/Weiß 3 Roter Teint 4 Grüner Teint 5 Blauer Teint 6 Sepia
AWB Enable Weiß Abgleich	Awb http://192.168.2.114/control? var=awb&val=1	0 Aus 1 Ein
Advanced AWB	Dcw http://192.168.2.114/control? var=dcw&val=1	0 Aus 1 Ein
Manual AWB	awb_gain http://192.168.2.114/control? var=awb_gain&val=0	0 Aus 1 Ein
AWB Mode	wb_mode http://192.168.2.114/control? var=awb&val=0	0 Auto 1 Sunny Sonnig 2 Cloudy Bewölkt 3 Office Büro 4 Home Zuhause
AEC Enable	Aec http://192.168.2.114/control? var=aec&val=1	0 Aus 1 An
Manual Exposure	aec_value	0 – 1536 490
Night Mode	Aec2 http://192.168.2.114/control? var=aec2&val=1	0 Aus 1 Ein
AGC	Agc http://192.168.2.114/control? var=agc&val=1	Aus 1 Ein
Gain	agc_gain	1 – 64 1
GMA Enable ?	raw_gma http://192.168.2.114/control? var=raw_gma&val=1	0 Aus 1 An
Lens Correction Linsen Korrektur	Lenc http://192.168.2.114/control? var=lenc&val=1	Aus 1 Ein
H-Mirror Horizontale Spiegelung	Hmirror http://192.168.2.114/control? var=hmirror&val=0	0 Aus 1 An
V-Flip	Vflip http://192.168.2.114/control?	0 Aus

Vertikale Spiegelung	var=vflip&val=1	1 An
BPC ?	Bpc http://192.168.2.114/control? var=bpc&val=1	0 Aus 1 An
WPC ?	Wpc http://192.168.2.114/control? var=wpc&val=0	0 Aus 1 An
Color Bar Farbbalken Test	Colorbar http://192.168.2.114/control? var=colorbar&val=1	0 Aus 1 An
LED Intensity	led_intensity	0 bis 255 0 LED aus
Zusätzliche Werte Bei mir nicht vorhanden ?		
Gesichtssuche aktivieren	face_detect	0 Aus 1 An
Gesichtserkennung aktivieren	face_recognize	0 Aus 1 An

Die zuletzt gesetzte Einstellung aus dem Konfigurationsmenü wird automatisch gespeichert.

Es kann **immer nur ein Client gleichzeitig** auf das Kameramodul zugreifen
Das bedeutet: Wenn der **Live-Stream**, das **Einzelbild** oder das **Konfigurations-Frontend**
geöffnet ist, kann jeweils **nur ein Browser-Tab** oder eine App wie **VLC** aktiv sein.
Andernfalls bleibt die Kamera „blockiert“ und die neue Verbindung schlägt fehl.

ESP32-CAM mit Kamera OV3660 steuern

Die Beispiele können einfach in die Adressleiste des Browser kopiert werden.
Vom Webserver kommt keine Antwort zurück, ob es geklappt hat oder auch nicht.

Befehl: <http://<ip>/control?var=<Variablenname>&val=<Wert>>
z.B: curl "http://192.168.2.114/control?var=framesize&val=11"

XCLK Taktfrequenz MHz :
<http://192.168.2.114/xclk?xclk=18> 18 MHz

Auflösung einstellen:
<http://192.168.2.114/control?var=framesize&val=6> 6 **QVGA(320×240)**
<http://192.168.2.114/control?var=framesize&val=10> 10 VGA(640×480)
<http://192.168.2.114/control?var=framesize&val=11> 11 SVGA(800×600)

Farbsättigung :
<http://192.168.2.114/control?var=saturation&val=4> 4 (-4 bis 4)

Bildqualität:
<http://192.168.2.114/control?var=quality&val=20> 20 (4 bis 63)

Helligkeit einstellen:
<http://192.168.2.114/control?var=brightness&val=2> 2 Normal

AWB Mode Manual einstellen:
http://192.168.2.114/control?var=awb_gain&val=1 Manual
http://192.168.2.114/control?var=wb_mode&val=0 Auto
http://192.168.2.114/control?var=wb_mode&val=3 Office

Bild auf den Kopf drehen:
<http://192.168.2.114/control?var=vflip&val=0>
<http://192.168.2.114/control?var=vflip&val=1> 1 Normal

Bild Horizontal drehen:
<http://192.168.2.114/control?var=hmirror&val=0> 0 Normal
<http://192.168.2.114/control?var=hmirror&val=1>

Mein Script zur ESP32-Cam OV3660 Konfiguration

```
#!/usr/bin/bash
#
# ESP32-Cam OV3660 Konfiguration
#
# Aufruf: ./ESP32-CAM_Konfig.sh
#
#

echo ESP32-Cam Erreichbarkeit prüfen
IPADRES="192.168.2.114"      # IP-Adresse von ESP32-Cam
ping $IPADRES -c 1          # 1 Versuch
ERRORLEVEL=${?}
echo $ERRORLEVEL
if [ $ERRORLEVEL -eq 0 ];
then
    echo " ESP32-Cam erreichbar: " $ERRORLEVEL;
else
    echo " ERRORLEVEL: " $ERRORLEVEL " ESP32-Cam nicht erreichbar";
    sleep 15
    exit;
fi

echo ESP32-Cam Konfiguration blind übertragen, es wird keine Rückmeldung
ausgegeben.
echo
    echo    Taktfrequenz auf 18 MHz setzen
    curl "http://192.168.2.114/xclk?xclk=18"

    echo
    echo " Auflösung einstellen: 6 QVGA(320×240), 10 VGA(640×480), 11
SVGA(800×600)          11 "
    #curl "http://192.168.2.114/control?var=framesize&val=6"
    #curl "http://192.168.2.114/control?var=framesize&val=10"
    curl "http://192.168.2.114/control?var=framesize&val=11"

    echo " Nachtmodus ein"
    curl "http://192.168.2.114/control?var=aec2&val=1"

    echo fertig

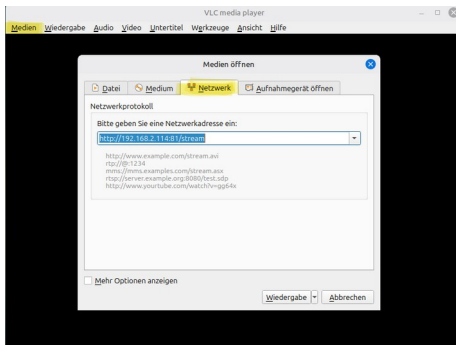
sleep 15
```

Streamen

- Browser, Einstellungen (Konfigurations-Frontend): <http://<IP-Adresse>/>
- Videolive-Stream: <http://<IP-Adresse>:81/stream>
- Einzelbild <http://<IP-Adresse>/capture>

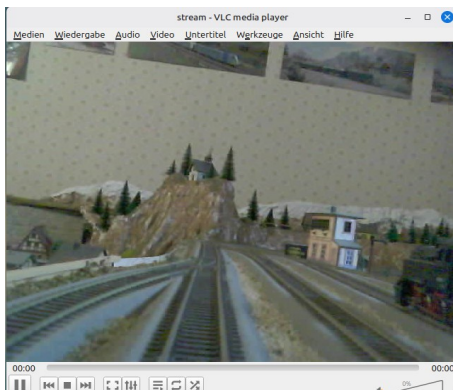
mit VLC ein Video erstellen und als .mp4 abspeichern.

VLC mit Standard Einstellungen



VLC - Medien - Netzwerkstream öffnen

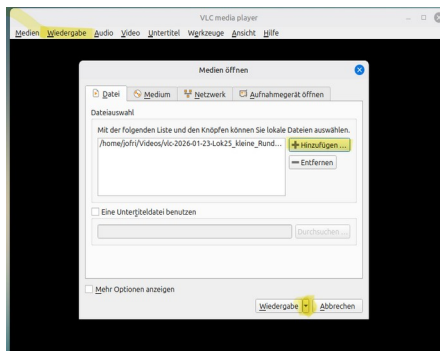
<http://192.168.2.114:81/stream>



Mit einen rechts Klick auf das Bild
die Aufzeichnung starten,

wir als .avi abspeichert.

die aufgenommene Datei in eine .mp4 konvertieren



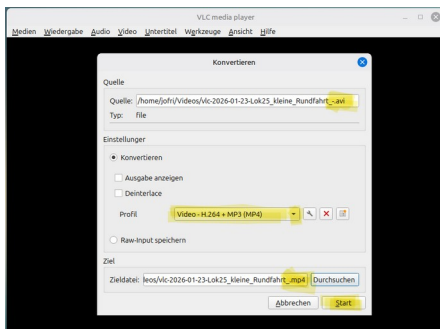
VLC Wiedergabe – Wiedergabe

Datei + Hinzufügen

.avi Videodatei auswählen

unten unter Wiedergabe (kleiner Pfeil nach unten)

Konvertieren auswählen



Quelle: der .avi Videodatei

Profil: Video – **H264 + MP3 (MP4)** auswählen

Zielfile: .mp4 Ausgabedatei

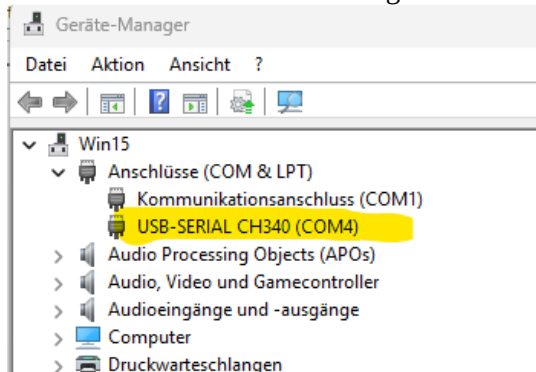
Start:

nun wird die .avi in eine kleinere .mp4

Datei konvertiert.

Windows - Verbindungs- Test im Windows Gerätemanager:

Windows: Test im Gerätemanager ob die ESP32-CAM erreichbar ist.



Die angeschlossene ESP32-CAM wird mit „USB-SERIAL CH340 und Port angezeigt.

Falls ein „?“ Angezeigt wird muss der Treiber für CH340 installiert werden.

Für ESP32-CAM CH340,CH341

https://www.wch-ic.com/downloads/CH341SER_EXE.html

Für ESP32-S3-CAM CH343

https://www.wch-ic.com/downloads/CH343SER_ZIP.html

Infos, Links:

Unter GitHub gleiches Sketch wie unter der
Arduino-IDE Beispiele – ESP32 – Camera – CameraWebServer

<https://github.com/espressif/arduino-esp32/blob/master/libraries/ESP32/examples/Camera/CameraWebServer/CameraWebServer.ino>

ESP32-CAM:

<https://www.fambach.net/esp32-cam-modul/>

<https://cool-web.de/esp8266-esp32/espressif-esp32-chip-varianten-im-vergleich.htm>

<https://cool-web.de/esp8266-esp32/esp32-cam-wlan-ip-kamera-anschliessen-einrichten-konfigurieren.htm>

<https://www.makerguides.com/de/stream-video-with-esp32-cam-de/>

<https://www.makerguides.com/de/programming-the-esp32-cam-de/>

<https://github.com/espressif/arduino-esp32/tree/master/libraries/ESP32/examples/Camera/CameraWebServer>

Nachfolger: ESP32-S3-CAM

<https://draeger-it.blog/esp32-s3-cam-der-leistungsstarke-nachfolger-der-esp32-cam/>

<https://prilchen.de/das-ultimate-esp32-cam-upgrade-freenove-s3-wroom-mit-3mp-ov3660/>

https://docs.freenove.com/projects/fnk0085/en/latest/fnk0085/codes/C/32_Camera_Web_Server.html

<https://docs.freenove.com/projects/fnk0085/en/latest/fnk0085/codes/C/Preface.html#ch343-importance>