VHS-Kurs „ESP32-CAM mit MicroPython“

Prof. Dr.-Ing. Ralph Wystup M.Sc. · Ordner `C:\VHS_Kurs` · Stand 17.11.2026, Fassung 1.8 der Anleitung (ersetzt LIESMICH_1.7.md; Installationsseite 1.4, Erklärung Download-Modus nach dem Einstecken)

Ordner

Ordner	Inhalt
`Teilnehmer\`	alles für die Kursrechner (kann so auf einen USB-Stick): oben die zwei Batch-Dateien und `LIESMICH_Teilnehmer_1.3.md`; `Kameraprogramme\` zum Lesen, `Werkzeuge\` (Hilfsprogramme), `pakete\` (Thonny 5.0.1, CH340-Treiber)
`Dozent\`	oben die Batch-Datei der Installationsseite; `Kameraprogramme\` (alles, was auf die Kamera kommt, mit `firmware.bin`), `Werkzeuge\` (Installationsseite, Konfiguration, Überwachung, Prüfwerkzeuge), `protokoll\`, `sicherung\`, `kameras_protokoll.csv`
`alt\`	überholte Fassungen

Für den Dozenten: Kameras vorbereiten

1. Einmal `Teilnehmer\Teilnehmer_1_PC_einrichten_1.6.bat` (installiert Thonny, Pakete, USB-Treiber, Thonny-Einstellung – nur was fehlt).
2. Thonny schließen. `Dozent\Dozent_Kameras_installieren_1.4.bat` doppelklicken – die **Installationsseite** öffnet sich im Browser (`http://127.0.0.1:8765/`; das schwarze Fenster offen lassen).
3. **Eine** Kamera per USB anstecken, Nummer eintragen (1, 2, 3 …), **„Kamera installieren“** drücken. Die Seite zeigt jeden Schritt:
   - Anschluss suchen und vorbereiten (die COM-Nummer wird automatisch gefunden),
   - Firmware: eine werksfrische Kamera bekommt MicroPython 1.27 mit Kameratreiber (`firmware.bin`, etwa 2 Minuten); der Flash-Modus wird automatisch über das USB-Brett geschaltet (am 17.11. geprüft), nur wenn das nicht geht, sagt die Seite „IO0 halten, einstecken“ an,
   - Programme aufspielen (Liste links auf der Seite, jede Datei per Klick ansehbar),
   - Probelauf wie in Thonny (Start, Stop, noch einmal Start) mit Verbindungsdaten; Zeile ins Protokoll `kameras_protokoll.csv`.
4. Kamera mit ihrem WLAN-Namen beschriften (`VHS-CAM-1` …), abziehen, nächste Kamera, nächste Nummer. Mitschrift jeder Installation: `Dozent\protokoll\installation_<Datum>.log`; Kameraliste `Dozent\kameras_protokoll.csv`.

Ohne Browser geht dasselbe auf der Kommandozeile: `python esp_kurs_kameras_konfigurieren_1.5.py` im Ordner `Dozent\Werkzeuge` (fragt nacheinander nach den Kameras).

Was auf jeder Kamera liegt (in Thonny im Dateibereich „MicroPython-Gerät“; Quellen in `Dozent\Kameraprogramme\`)

Es gibt **kein `main.py`**: die Kamera startet nach dem Einschalten nichts von selbst. Gestartet wird in Thonny mit dem grünen Start-Knopf.

Datei	Quelle	Zweck
kamera.py	esp_cam_kamera_1.6.py	eigenes WLAN VHS-CAM-<Nr>, Livebild unter http://192.168.4.1/; Ausgabe „AM HANDY EINGEBEN“, Verbindungsdaten, Zustand; bei Stop wird die Kamera abgeschaltet
blink.py	esp_cam_blink_1.2.py	Beispiel 1: weiße Blitz-LED (GPIO 4) blinkt – nach led_blickt.py (die rote LED hängt fest an VCC)
pwm.py	esp_cam_pwm_1.0.py	Beispiel 2: Blitz-LED per PWM heller/dunkler
led_handy.py	esp_cam_led_handy_1.1.py	Beispiel 3: weiße LED vom Handy schalten, OLED an IO14/IO15 – nach led_blink_Handy.py (ohne Kennwort im Programm)
ssd1306.py	ssd1306.py	OLED-Treiber (micropython-lib, MIT)
kamera.json	wird erzeugt	{"name": "VHS-CAM-<Nr>"}

Alle Bibliotheken, die die Programme importieren (asyncio, gc, json, network, time, ubinascii, machine, camera, _thread, usocket), stecken in der Firmware – auf dem PC ist dafür nichts zu installieren.

Die Installation endet mit einem **Probelauf wie in Thonny**: Start, Stop, Soft-Reboot, noch einmal Start – nur wenn beides läuft, meldet die Seite „FERTIG“.

Für die Teilnehmer (Ordner Teilnehmer\, siehe auch LIESMICH_Teilnehmer_1.3.md)

1. Einmal Teilnehmer_1_PC_einrichten_1.6.bat doppelklicken (Thonny, Pakete, USB-Treiber, Thonny-Einstellung; vorhandenes Thonny bleibt).
2. Kamera per USB anstecken, Teilnehmer_2_Thonny_mit_Kamera_starten_1.4.bat doppelklicken: bereitet den Anschluss vor, startet die Kamera per Software neu (sie steht nach dem Einstecken sonst im Bootlader) und öffnet Thonny mit dem Kameraprogramm.
3. **Grüner Start-Knopf** → oben „AM HANDY EINGEBEN“ (WLAN, Kennwort, Adresse), danach Zustandszeilen. **Stop** beendet.
4. Beispiele: in Thonny blink.py, pwm.py, led_handy.py auf der Kamera öffnen und Start drücken.

Warum nach dem Einstecken ein Reset nötig ist

Nach dem Einstecken meldet die Kamera in Thonny oft boot:0x3 (DOWNLOAD_BOOT …) waiting for download und „Device is busy or does not respond“. Das liegt nicht an der Einrichtung, sondern an der Elektronik des USB-Bretts ESP32-CAM-MB: Seine Steuerleitungen hängen über eine kleine Transistorschaltung an EN (Reset) und IO0 (Flash-Modus). Beim Einstecken schaltet Windows diese Leitungen kurz; liegt IO0 im Einschaltaugenblick auf 0, bleibt der eingebaute Bootlader im Download-Modus – noch bevor irgendein Programm läuft. Thonny kann die Leitungen beim Verbinden nur dauerhaft ein- oder ausschalten, aber keinen kurzen Reset-Puls geben („ein“ hält die Kamera im Reset, „aus“ lässt sie im Download-Modus). Deshalb gilt:

Nach dem Einstecken einmal neu starten: entweder Teilnehmer_2_Thonny_mit_Kamera_starten_1.4.bat benutzen (sie startet die Kamera per Software neu, ohne Taste) oder an der Kamera einmal **RST** drücken. Danach meldet sich MicroPython (>>>). Die Installationsseite macht diesen Neustart ebenfalls selbst.

(Gemessen am Dozenten-PC; wie andere PCs die Leitungen beim Einstecken schalten, hängt von deren Treiber ab.)

Gut zu wissen

- **CH340-Falle:** Öffnet ein Programm den Anschluss nach dem Einstecken direkt mit 115200 Baud, meldet Windows „Ein an das System angeschlossenes Gerät funktioniert nicht“, und der Wandler bleibt bis zum Neustecken gesperrt. Abhilfe: erst mit 9600 Baud öffnen – das machen alle Werkzeuge hier selbst (`esp_com_vorlauf_1.1.py`).
- **Kamera hängt** („Failed to capture initial frame“, `ESP_ERR_NOT_SUPPORTED`): USB-Kabel 5 s abziehen und wieder anstecken – RST allein hilft dann nicht, weil der Kamerasensor nicht stromlos wird. Das Kameraprogramm sagt das selbst an.
- **DTR/RTS:** Auf dem USB-Brett halten diese Leitungen die Kamera im Reset. Thonny ist deshalb auf `dtr = False, rts = False` eingestellt (`esp_thonny_einstellen_1.2.py`); sonst meldet Thonny „Device is busy or does not respond“.
- **Thonny hält den Anschluss**, solange es mit der Kamera verbunden ist – vor dem Installieren schließen.
- Kennwörter stehen nie in Programmen. Eigenes WLAN mit Kennwort: Datei `ap.json` mit `{"kennwort": "xxxxxxxx"}` (mind. 8 Zeichen) auf die Kamera legen, Heimnetz: `wlan.json` mit `{"ssid": "MeinNetz", "kennwort": "xxxxxxxx"}` – z. B. in Thonny anlegen und speichern.
- Thonny 5 sieht aus wie Thonny 4. Für Rechner ohne Internet legt `Dozent\Werkzeuge\esp_pakete_holen_1.2` Thonny 5.0.1 und den CH340-Treiber nach `Teilnehmer\pakete\` (im Repositorium nicht enthalten – Fremddateien).

Überholte Fassungen liegen im Unterordner `alt`.