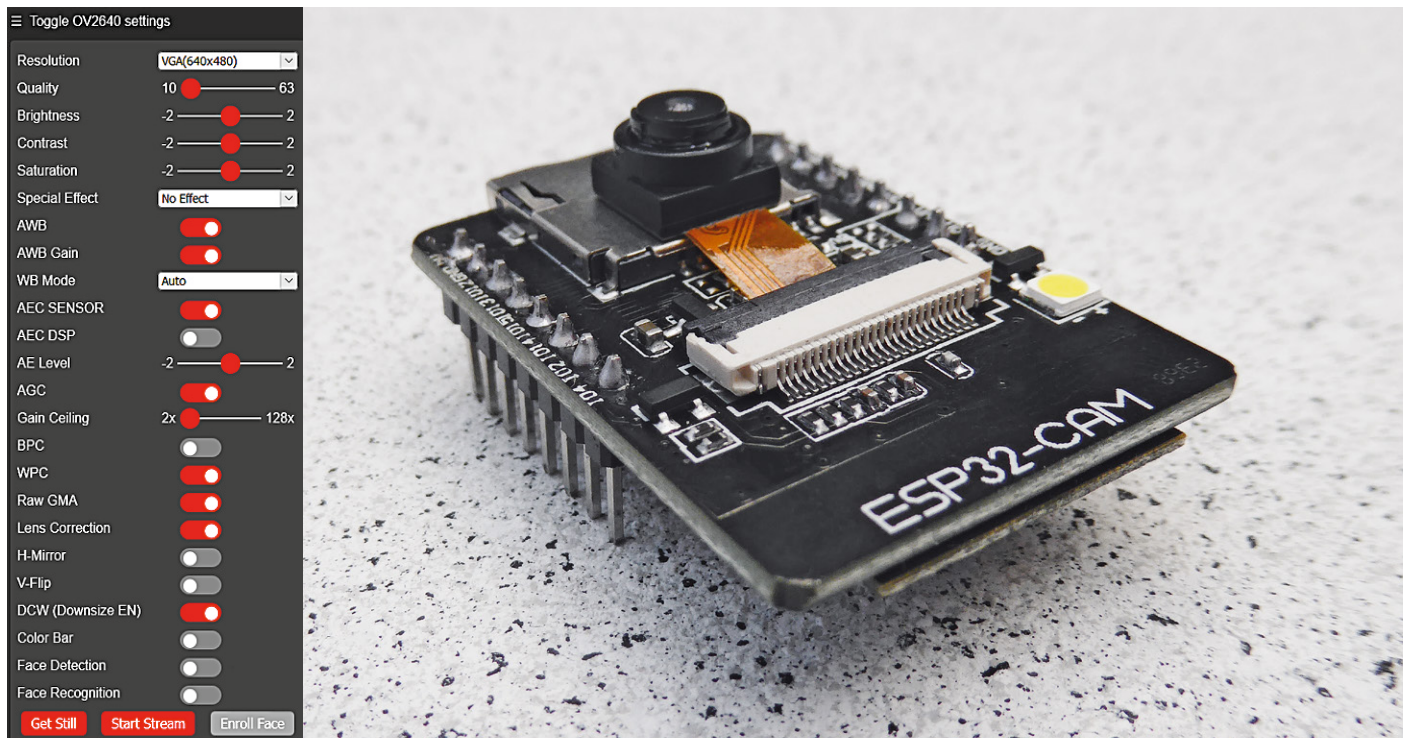


# Intelligente Kamera

## Wi-Fi-Kameramodul ESP32-CAM

Die Kombination eines Mikrocontrollers mit einer Kamera, die mit der Etablierung des Raspberry Pi in die Selbstbau-Szene eingezogen ist, erfreut sich ungebrochener Beliebtheit. Und spätestens mit dem Erscheinen der auf der leistungsstarken 32-Bit-Plattform ESP32 von Espressif basierenden, äußerst preiswerten Mini-Kameramodule ist diese Technik für jeden interessant, der sich mit Mikrocontroller-Anwendungen beschäftigt. Wir stellen die Plattform ESP32-CAM vor.



### Bilderfunk

Die ESP32-Plattform von Espressif ist derzeit der Renner unter den Mikrocontroller-Plattformen, viele Anwender bezeichnen sie inzwischen als den besseren Arduino. Leistungsfähige Prozessoreinheit, alle wichtigen Schnittstellen an Bord, viele GPIOs und vor allem die Möglichkeit, per WLAN und Bluetooth kabellos zu kommunizieren, bilden den technischen Hintergrund.

Mit dem von Espressif selbst kreierten „ESP-Eye“ (Bild 1) und der „WROVER Kit“-Entwicklungsplattform stehen zusammen mit der SDK „ESP-IDF“ zwei Entwicklungssysteme zur Verfügung, die Anwender in die Lage versetzen, Videoanwendungen schnell und einfach zu entwickeln. „ESP-Eye“ verfügt über eine 2-Megapixel-Kamera, ein MEMS-Mikrofon und mit 8 MB PSRAM sowie 4 MB Flash genügend Speicher für schnelle Bildverarbeitung. Zudem hat das Board einen Micro-USB-Port und ist so besonders einfach programmierbar.

Der preiswerte Ableger dieser Entwicklungsplattform ist das ESP32-CAM-Board, das von diversen An-

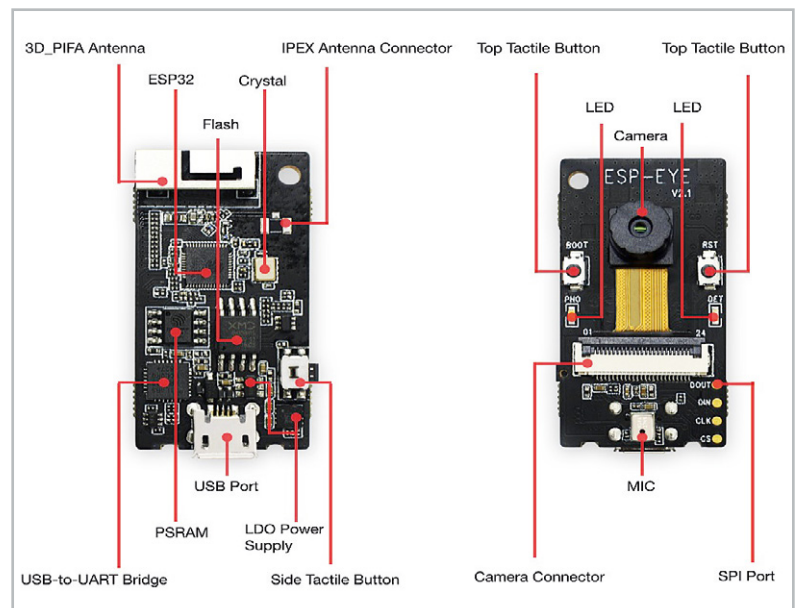


Bild 1: Das Kameraboard „ESP-Eye“ von Espressif ist eine komplette Entwicklungsplattform mit Mikrofon und USB-Port. Bild: Espressif



bietern mit ganz verschiedenen Ausstattungen, die sogar ein integriertes TFT-Display umfassen, kommt. Der bekannteste Anbieter ist dabei AI Thinker (Bild 2). Das Board hat einen digitalen Kameraanschluss für die bekannten 2-Megapixel-0V2640/7670-Kameras, trägt ein microSD-Kartenslot und verfügt über 520 kB SRAM sowie 4 MB PSRAM. Wie gesagt, kommen die ESP32-CAM-Module von zahlreichen, vornehmlich chinesischen Anbietern, die auch unterschiedliche Kameramodule einsetzen. Diese belegen unglücklicherweise den FPC-Folienkabelanschluss des Boards unterschiedlich, sodass der Anwender genau wissen muss, welches Kameramodell er vor sich hat. Darauf kommen wir noch.

Das ESP32-CAM-Modul ist ansonsten ein normales ESP32-Modul, allerdings verfügt es wegen der Belegung zahlreicher GPIOs mit den Kameraanschlüssen

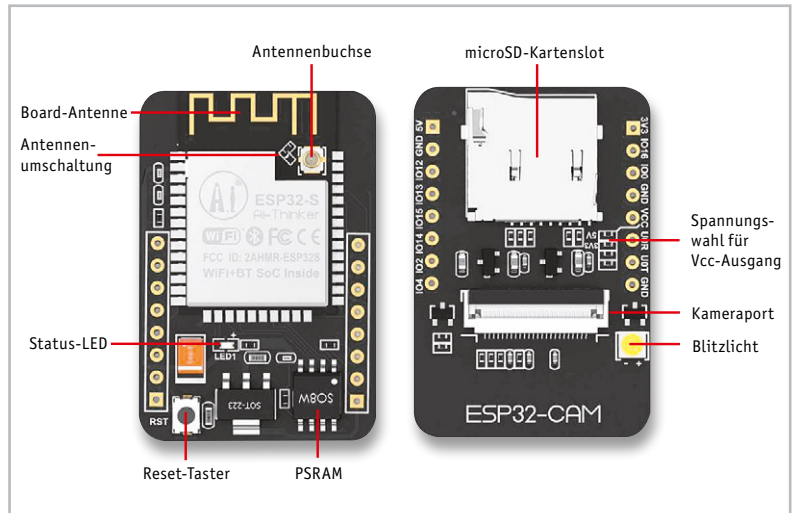


Bild 2: Das ESP32-CAM-Board von AI Thinker. Bild: AI Thinker

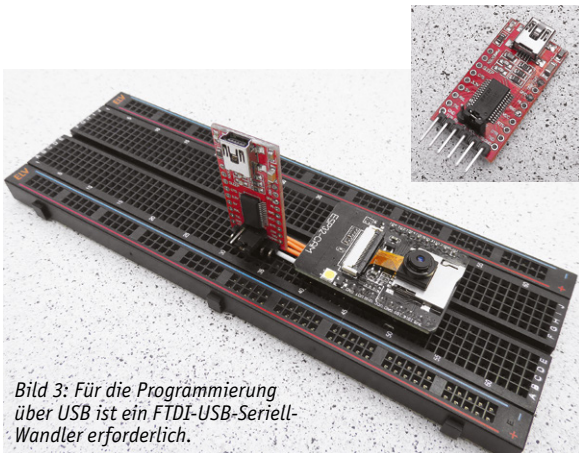


Bild 3: Für die Programmierung über USB ist ein FTDI-USB-Seriell-Wandler erforderlich.

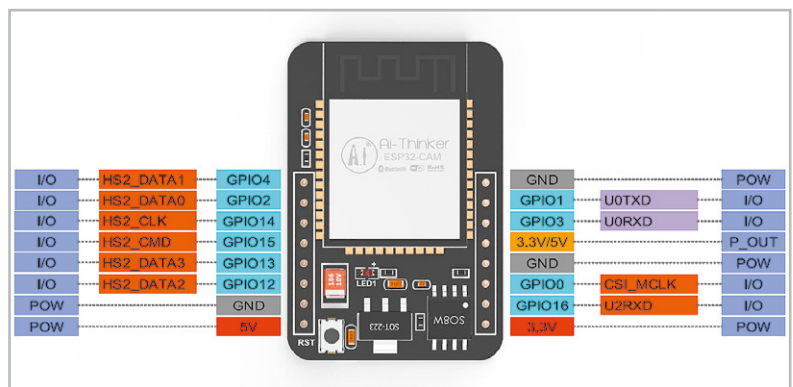


Bild 4: Die Anschlussbelegung des ESP32-CAM. Bild: AI Thinker

und denen des SD-Kartenslots nur noch über wenige freie GPIOs, darunter die für das Programmieren wichtige UART-Schnittstelle. Denn diese wird gebraucht, da das Board keinen USB-Port besitzt. Stattdessen erfolgt die Programmierung über ein externes FTDI-USB-to-Serial-Board, wie es im Musteraufbau in Bild 3 zu sehen ist.

Nutzt man die wenigen verbleibenden GPIOs geschickt aus, kann man mit dem kleinen Board tatsächlich ein ganzes videobasiertes Zugangssystem aufbauen, das etwa automatisch den Zugang über eine Gesichtserkennung oder ferngesteuert via Netzwerk/Internet ermöglicht. Ebenso kann man das Board von einem Bewegungsmelder aktivieren lassen, der dafür sorgt, dass das Geschehen vor der Kamera als Video-Stream ausgesendet oder als Video bzw. Einzelbildfol-

ge auf die SD-Speicherkarte aufgezeichnet wird. Die sich daraus ergebenden Möglichkeiten sind natürlich enorm, allerdings muss man sich auch im Klaren darüber sein, dass die Bildqualität der 2-Megapixel-Kamera unter heutigen Maßstäben begrenzt ist. Diese liefert eine maximale Auflösung von 1600 x 1200 Pixeln (UXGA) und ist per HTTP-Request mit zahlreichen Variablen genau an Einsatzort und Einsatzzweck anpassbar, wie man es etwa von konfigurierbaren Überwachungskameras kennt. Man kann sie entweder im Anwendungsprogramm oder aber auch, wie gesagt, per einfachem HTTP-Request über den Browser aufrufen.

Beispiel-HTTP-Request für die Einstellung der Auflösung auf QVGA:  
<http://IP der Kamera/control?var=framesize&val=4>

In Bild 4 ist die Pinbelegung des ESP32-CAM-Boards zu sehen. Hier fällt auf, dass das Board sowohl mit 3,3 V als auch 5 V zu betreiben ist und wahlweise, durch Umlöten eines Jumpers auf der Platine wählbar, 3,3 V oder 5 V für Peripherie ausgeben kann. In der Praxis, insbesondere bei Programmiervorgängen, hat sich der 5-V-Betrieb als die betriebssichere Variante erwiesen.

GPIO 4 ist standardmäßig durch den microSD-Kartenslot belegt. Nutzt man den nicht, kann auch er extern benutzt werden.

Übrigens ist man nicht unbedingt auf die 0V2640-Standardkamera, die sich im Lieferumfang des ESP32 CAM befindet, angewiesen. Die Kamera gibt es in verschiedenen Varianten, etwa mit längerem Anschlusskabel für abgesetzten Betrieb, oder mit anderen Objektiven, z. B. einem Fischaugen-Objektiv für einen breiteren Blickwinkel (Bild 5). Findige Elektroniker haben sogar eine Lösung gefunden, die Kamera mit einem C/CS-Mount-Wechselobjektiv aufzurüsten, wie in [1] gezeigt. So kann man Standardobjektive aus dem Überwachungskamerabereich einsetzen und das ESP32-CAM-Modul auf der kameraoptischen Seite deutlich aufwerten.



Bild 5: An das ESP32-CAM-Board können auch andere Kameras angeschlossen werden, hier ein Modell mit Fischaugen-Objektiv. Bild: Omnivision



Schließlich bietet Espressif auf seiner GitHub-Seite [2] Unterstützung in Form von Treibern für weitere Mini-Kameramodelle an, so die besonders für dunklere Standorte geeignete OV7725 von Omnivision. Sie eignet sich gut für den Überwachungskamerabetrieb.

Bleibt für die erste Vorstellung noch die Kommunikationsseite des Boards. Hier finden wir die übliche ESP32-Standardausstattung mit WLAN auf 2,4 GHz und Bluetooth. Für den Anschluss einer externen Antenne ist sogar eine Antennenbuchse vorhanden. Der Wechsel zwischen den beiden Antennensystemen erfolgt durch Umlöten eines Jumpers.

Die Firmware des Boards beinhaltet u. a. auch einen Webserver, der ein einfach bedienbares Web-Interface für die bequeme Konfiguration der Kamera via WLAN bietet. Außerdem ist hier der Start eines Videostreams, das Einlernen von Gesichtern für die Gesichtserkennung und der Start dieser Gesichtserkennung möglich.

## Erste Schritte

Wie alle ESP-Boards lässt sich auch das ESP32 CAM über die Arduino IDE programmieren. Die dazu benötigten Librarys und Beispielprogramme sind sowohl über den Installationslink in der Arduino IDE als auch über die Espressif-GitHub-Seite [3] zugänglich.

Zunächst muss der ESP32 in den Boardverwalter der Arduino IDE aufgenommen werden. Dazu setzt man den folgenden Link mit der ESP32-JSON-Datei in das Fenster „Zusätzliche Boardverwalter-URLs“ ein (Bild 6):

[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)

Danach geht man in den Boardverwalter der IDE und installiert nach Eingabe von „ESP32“ das Espressif-ESP32-Paket (Bild 7).

Nach einem Neustart der IDE findet man nun diverse ESP32-Boards in der Liste der unterstützten Boards (Bild 8), darunter auch das AI-Thinker-CAM-Board. In den Beispielen (Bild 9) sind nun auch zahlreiche Programmbeispiele für ESP32 gelistet, so auch unter „Camera“ das Programm „Camera Webserver“ (Bild 10).

Hier muss man lediglich noch das eigene Kameramodell definieren, wie in Bild 10 markiert, und die Zugangsdaten des zu nutzenden WLANs im Haus eintragen.

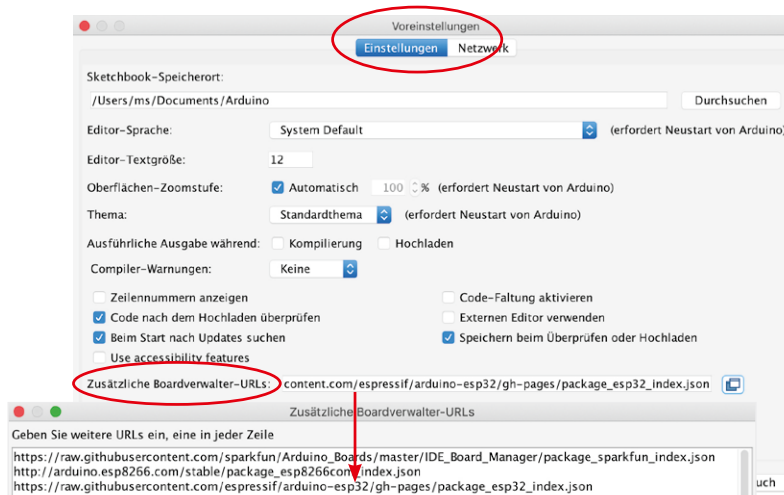


Bild 6: Zur Aufnahme der ESP32-Familie in die Arduino IDE muss zunächst die richtige Boardverwalter-URL eingetragen werden.

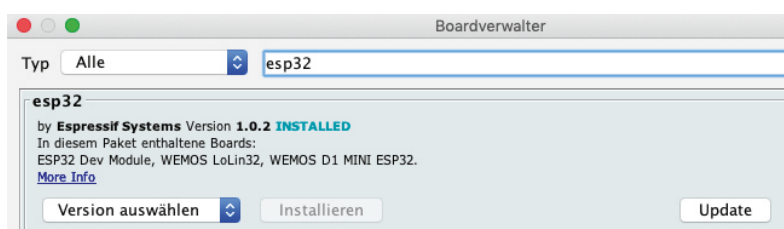


Bild 7: Über den Boardverwalter der IDE installiert man die ESP32-Boardfamilie ...

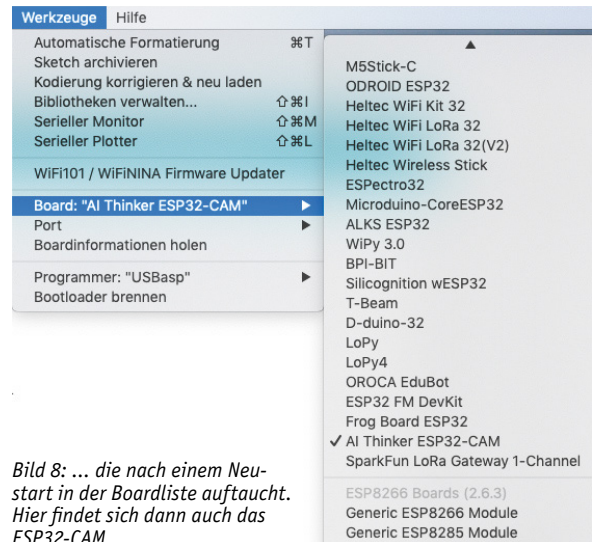


Bild 8: ... die nach einem Neustart in der Boardliste auftaucht. Hier findet sich dann auch das ESP32-CAM.

Bevor das Programm nun kompiliert und auf den ESP32 übertragen werden kann, muss man diesen mithilfe des bereits erwähnten FTDI-Adapters für das Programmieren präparieren. Dazu gehört auch, dass man den Port I00 auf GND legt, um den ESP32 in den Programmiermodus zu versetzen. Bild 11 zeigt die Schaltung dazu. Wichtig ist es, den Jumper auf dem FTDI-Adapter auf die gewählte Betriebsspannung einzustellen. Wie bereits erwähnt, kann man auch die 5-V-Betriebsspannung wählen, zahlreiche Anwender raten dazu, um Probleme bei der Programmierung zu umgehen. Auch spricht bei 3,3 V unter Umständen die Brownout-Detection des Boards an und blockiert den Betrieb.

Nach dem Anschluss des FTDI-Boards und dessen Verbindung mit dem USB-Port des Computers setzt man den ESP32 über seinen Reset-Taster zurück. Dies wird durch kurzes Aufleuchten der Blitzlicht-LED quittiert.

Nun kann man das Webserver-Programm kompilieren und via USB-Port auf den ESP32 übertragen. Ist dies erfolgreich verlaufen, meldet sich das ESP32-CAM-Board im seriellen Monitor mit seiner WLAN-IP-Adresse.

Über diese ist es per Browser erreichbar. Zunächst erscheint das bereits erwähnte Web-Interface (Bild 12), das zahlreiche Einstellungen erlaubt, und nach Anklicken von „Get Still“ (Standbild aufnehmen) bzw. „Start Stream“ erscheint das Kamerabild. In dem einfachen Webserver-Programm, das wir geladen haben, ist noch kein Speichern der Bilder implementiert, hierzu muss das Programm erweitert werden.

Über „Enroll Face“ ist die implementierte Gesichtserkennung erreichbar. Aktiviert man diese, wird man durch einen Anlernprozess geführt, bis der Webserver das Gesicht gespeichert hat. Betritt nun eine gespeicherte Person den Sichtbereich der Kamera und wird durch die Software erkannt, erfolgt eine Begrüßung. Ansonsten wird ein Alarm ausgelöst und die fremde Person mit „Intruder Alarm“ gemeldet.

Die Anzeige von Standbildern und Videostreams kann auch direkt mit dem URL-Aufruf des Webservers erfolgen, indem der Port und der zugehörige Befehl im Browser eingegeben wird. So erscheint nur das gestreamte Bild ohne das Web-Interface.

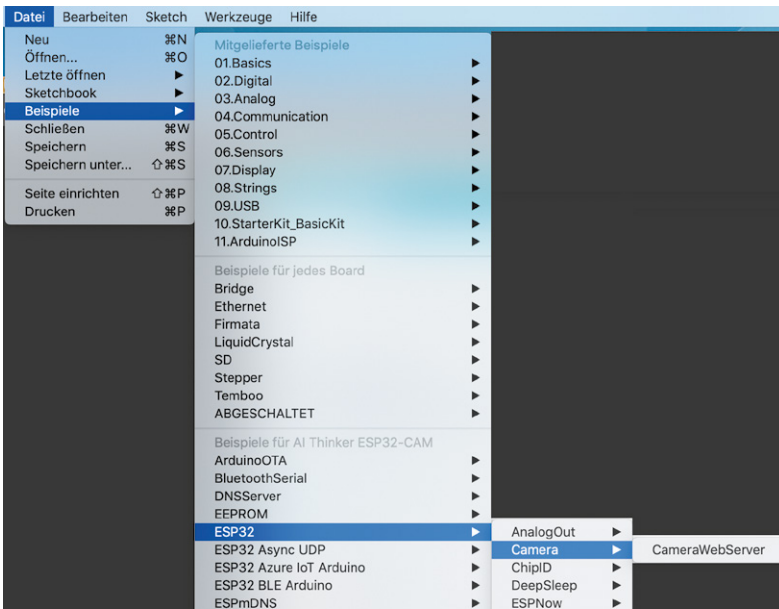


Bild 9: Mit der Boardfamilie werden auch die zugehörigen Programmbeispiele installiert.

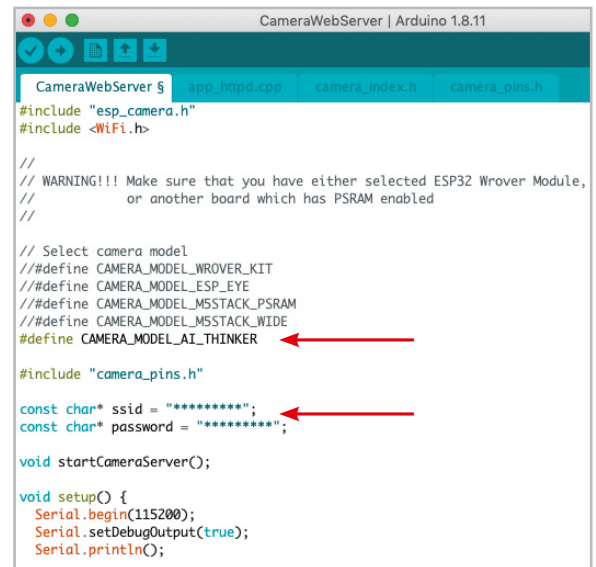


Bild 10: Im Beispielprogramm sind lediglich der eingesetzte Board-Typ und die Daten des eigenen WLANs einzutragen bzw. auszuwählen. Außerdem kann man hier die Kamerakonfiguration üben.

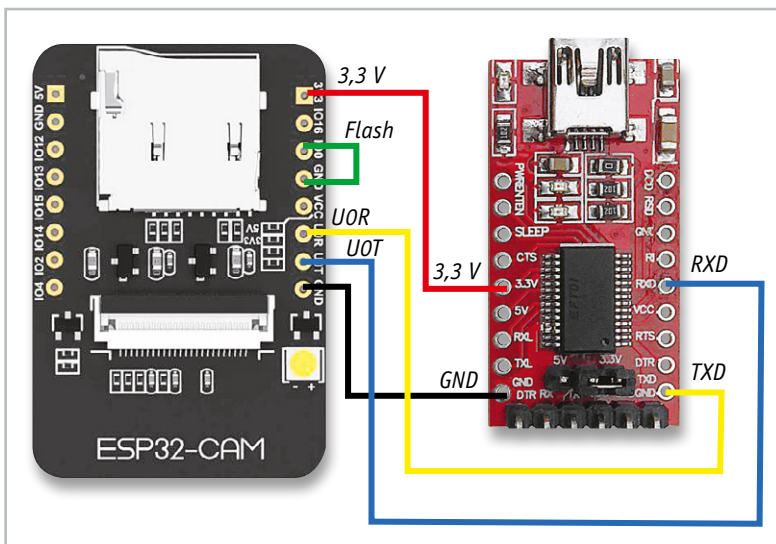


Bild 11: So werden FTDI-Board und ESP32-CAM zur Programmierung verbunden. Wichtig: Port 100 zum Programmieren auf GND legen!



## Weitere Infos:

- [1] C/CS-Mount-Objektiv am ESP32-CAM  
<https://www.youtube.com/watch?v=QIC6FtiUrOU>
- [2] GitHub-Seite von Espressif für ESP32-CAM:  
<https://github.com/espressif/esp32-camera>
- [3] GitHub-Seite von Espressif zur Arduino IDE:  
<https://github.com/espressif/arduino-esp32/tree/master/libraries/ESP32/examples/Camera/CameraWebServer>
- [4] Zugangssystem mit ESP32-CAM  
<https://robotzero.one/esp32-face-door-entry/>
- [5] Zeitrafferkamera mit dESP32-CAM:  
[https://www.youtube.com/watch?v=0\\_pewS4IPN4](https://www.youtube.com/watch?v=0_pewS4IPN4)
- [6] Alles rund um das ESP32-CAM-Board:  
<https://randomnerdtutorials.com/projects-esp32-cam/>
- [7] Espressif-Quellenverweise zu ESP32-CAM-Anwendungen:  
[https://www.espressif.com/en/news/ESP32\\_CAM](https://www.espressif.com/en/news/ESP32_CAM)

Alle Links finden Sie auch online unter: [de.elv.com/elvjournal-links](https://de.elv.com/elvjournal-links)

## Beispiele:

`http://IP der Kamera:81/stream`

→ Aufruf des Videostreams

`http://IP der Kamera:81/still`

→ Aufruf eines aktuellen Standbilds

Ist dieser „Probelauf“ gelungen, kann es an die Programmierung von Anwendungen gehen. In [4] ist z. B. ein Projekt für eine komplette Türsteuerung zu finden, in [5] ein Projekt für eine Zeitrafferkamera.

Die wohl wichtigste Anlaufstelle im Internet mit viel Wissen und allen nur denkbaren Projekten – u. a. findet man hier auch die bereits erwähnten Konfigurationsparameter – ist die Seite von Random Nerd Tutorials (RNT) [6]. Hier ist das Wissen um das Kameraboard so konzentriert zusammengefasst, dass Espressif es auf seiner ESP32-CAM-Unterseite [7] als Referenz empfiehlt. **ELV**

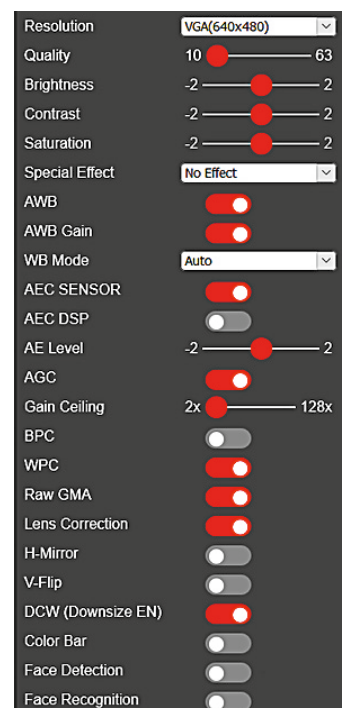


Bild 12: Nach Aufruf der URL des ESP32-CAM-Boards erscheint dessen Web-Interface.