



## Vorwort

### Über unsere Firma

KeeYees Technology Inc. befinden uns in Shenzhen, das Silicon Valley von China. Wir ist ein großer und professioneller Hersteller und Verkäufer von elektronischen Produkten, der sich der Forschung und Entwicklung von Open-Source Hardware, der Produktion und dem Marketing widmet. Alle unsere Produkte entsprechen den internationalen Qualitätsstandards und sind auf verschiedenen Märkten der Welt sehr beliebt. KeeYees ist die beste Wahl in verschiedenen elektronischen Modulen und Komponenten, die für Kunden aller Stufen geeignet sind, um Arduino- und Raspberry Pi Kenntnisse zu erlernen. Darüber hinaus verkaufen wir auch Produkte wie Zubehör für 3D-Drucker, Steckverbinder und Terminalsätze, DIY-Teile und Werkzeuge, um Ihre Arbeit oder Design-Herausforderungen von Zuhause, Schule bis hin zu industriellen Anwendungen zu unterstützen!

US Amazon Geschäft Startseite:

<https://www.amazon.com/shops/A2K4DGCC72N9AG>

UK Amazon Geschäft Startseite:

<https://www.amazon.co.uk/shops/A1F4U6XVWUBG1U>

DE Amazon Geschäft Startseite:

<https://www.amazon.de/shops/A1F4U6XVWUBG1U>

FR Amazon Geschäft Startseite:

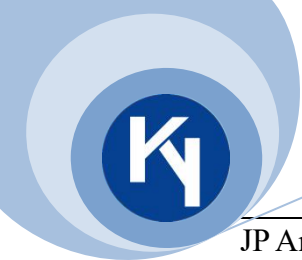
<https://www.amazon.fr/shops/A1F4U6XVWUBG1U>

IT Amazon Geschäft Startseite:

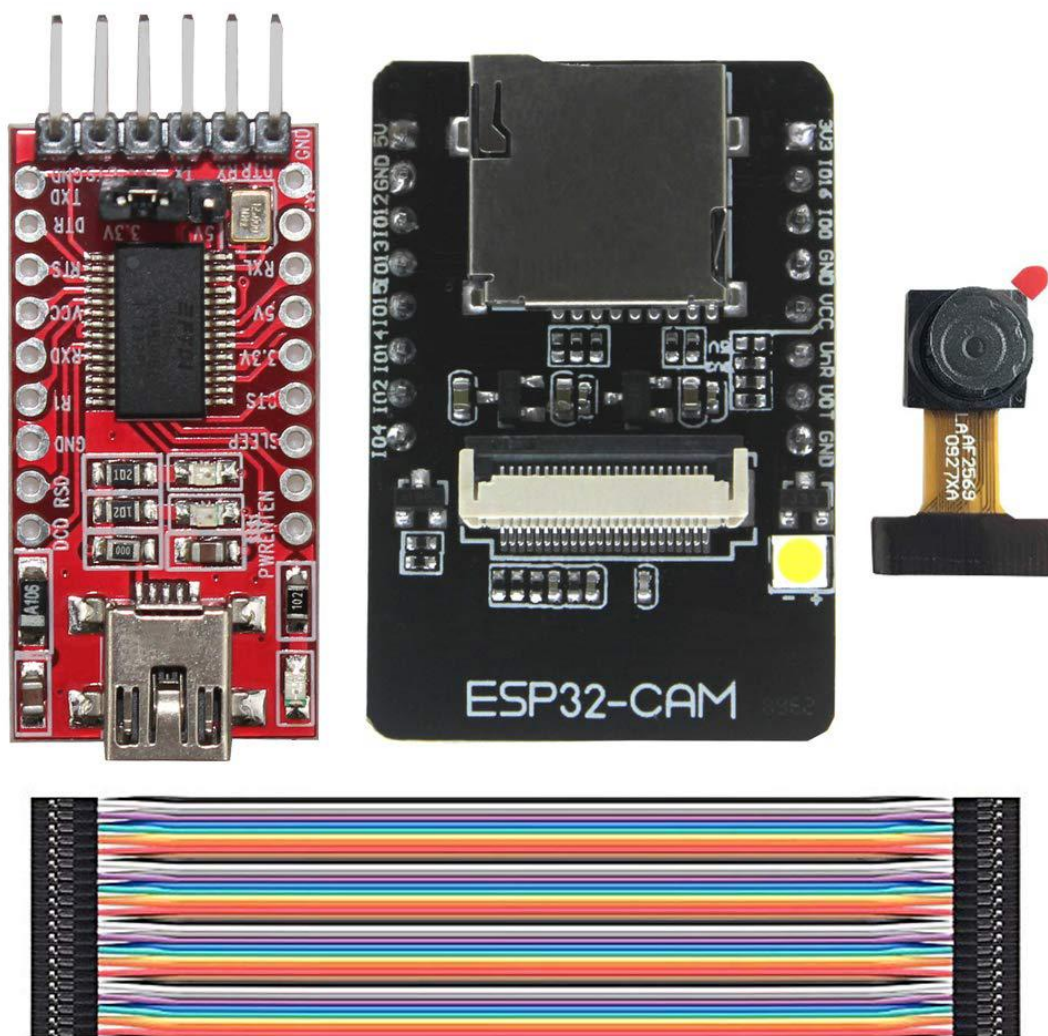
<https://www.amazon.it/shops/A1F4U6XVWUBG1U>

ES Amazon Geschäft Startseite:

<https://www.amazon.es/shops/A1F4U6XVWUBG1U>



## KeeYees ESP32-Cam WiFi und Bluetooth Kamera Modul

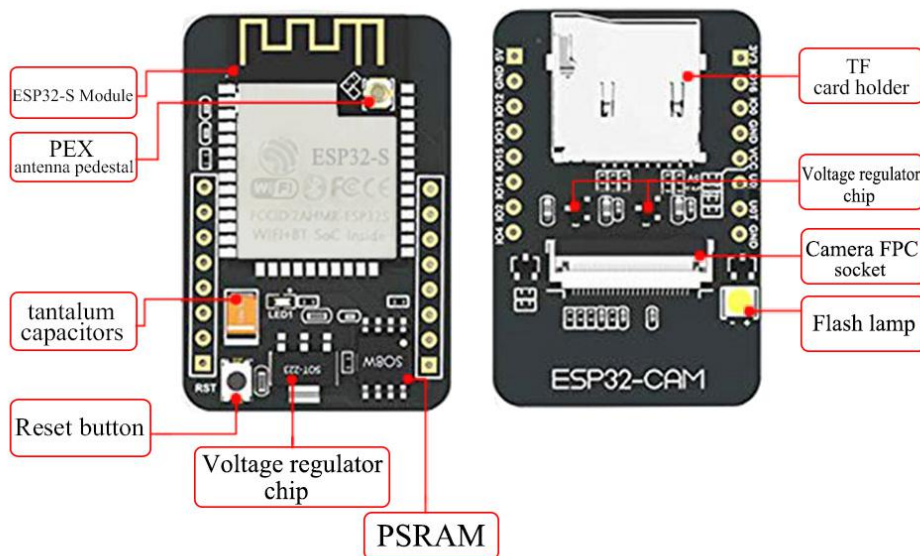




## Vorstellung der ESP32-CAM

Das ESP32-CAM ist ein kleines Kameramodul mit dem ESP32-S-Chip. Neben der Kamera OV2640 und mehreren GPIOs zum Anschließen von Peripheriegeräten verfügt sie über einen microSD-Kartensteckplatz, in dem mit der Kamera aufgenommene Bilder oder Dateien gespeichert werden können.

**KeeYees**





## Eigenschaften

- Das kleinste 802.11b /g/ n-WLAN-BT-SoC-Modul.
- 32-Bit CPU mit geringem Stromverbrauch kann auch den Anwendungsprozessor bedienen.
- Bis zu 160 MHz Taktrate, Rechenleistung bis zu 600 DMIPS.
- Eingebauter 520 KB SRAM, externer 4MPSRAM.
- Unterstützt UART / SPI / I2C / PWM / ADC / DAC.
- Unterstützt die Kameras OV2640 und OV7670 mit integrierter Blitzlampe.
- Bild-WiFi-Upload unterstützen.
- TF-Karte unterstützen.
- Unterstützt mehrere Schlafmodi.
- Eingebettetes Lwip und FreeRTOS.
- Unterstützt den Betriebsmodus STA / AP / STA + AP.
- Support Smart Config / AirKiss-Technologie.
- Unterstützung für lokale und Remote-Firmware-Upgrades der seriellen Schnittstelle (FOTA).



## ESP32-CAM Videokommunikation und Gesichtserkennung mit Arduino IDE

Die folgende Anleitung zeigt, wie Sie einen Video-Streaming-Webserver mit Gesichtserkennung und -erkennung mit Arduino IDE einrichten.

**Hinweis:** In diesem Tutorial verwenden wir das Beispiel aus der arduino esp32 Bibliothek. In diesem Tutorial wird nicht erläutert, wie Sie das Beispiel ändern.

### Erforderliche Teile

- ESP32-CAM mit OV2640
- FTDI Programmierer
- Female-Female Kabel Steckbrücken

Der ESP32-CAM wird ohne USB-Anschluss geliefert. Sie benötigen daher einen FTDI-Programmierer, um Code über die U0R- und U0T-Pins (serielle Pins) hochzuladen.

### ESP32-CAM Pinbelegung

Die folgende Abbildung zeigt die ESP32-CAM Pinbelegung (AI-Thinker-Modul).



Es gibt drei GND-Pins und zwei Pins für die Stromversorgung: 3,3 V oder 5 V.

**GPIO 1** und **GPIO 3** sind die seriellen Pins. Sie benötigen diese Pins, um Code auf Ihr Board hochzuladen. Zusätzlich, **GPIO 0** spielt auch eine wichtige Rolle, da es bestimmt, ob sich der ESP32 im Blinkmodus befindet oder nicht. Wenn **GPIO 0** ist mit GND verbunden, der ESP32 befindet sich im Blinkmodus.

Die folgenden Pins sind intern mit dem microSD-Kartenleser verbunden:

GPIO 14: CLK

GPIO 15: CMD

GPIO 2: Data 0

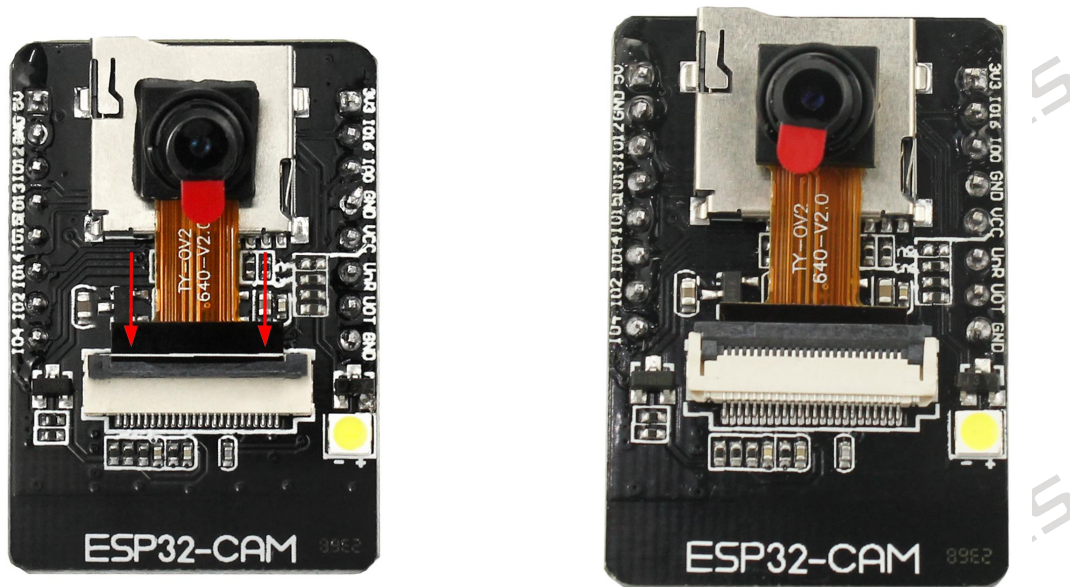
GPIO 4: Data 1 (auch an die On-Board-LED angeschlossen)

GPIO 12: Data 2





## Installieren das Kameramodul



1. Schnallen Sie das Kameramodul in die Unterseite der Schnalle im ESP32-Modul ein. Dies ist möglicherweise **etwas mühsamer** als Sie denken.
  2. Stellen Sie sicher, dass die weißen Linien auf der Unterseite des Kameramoduls nach dem Anbringen der Schnalle nicht sichtbar sind. Der Effekt ist der gleiche wie in Abbildung 2.
- \* Das Kameramodul fällt auch nicht herunter, wenn das ESP32-Modul geschüttelt wird.

## Videokommunikation Server



## Installieren Sie das ESP32-Add-On

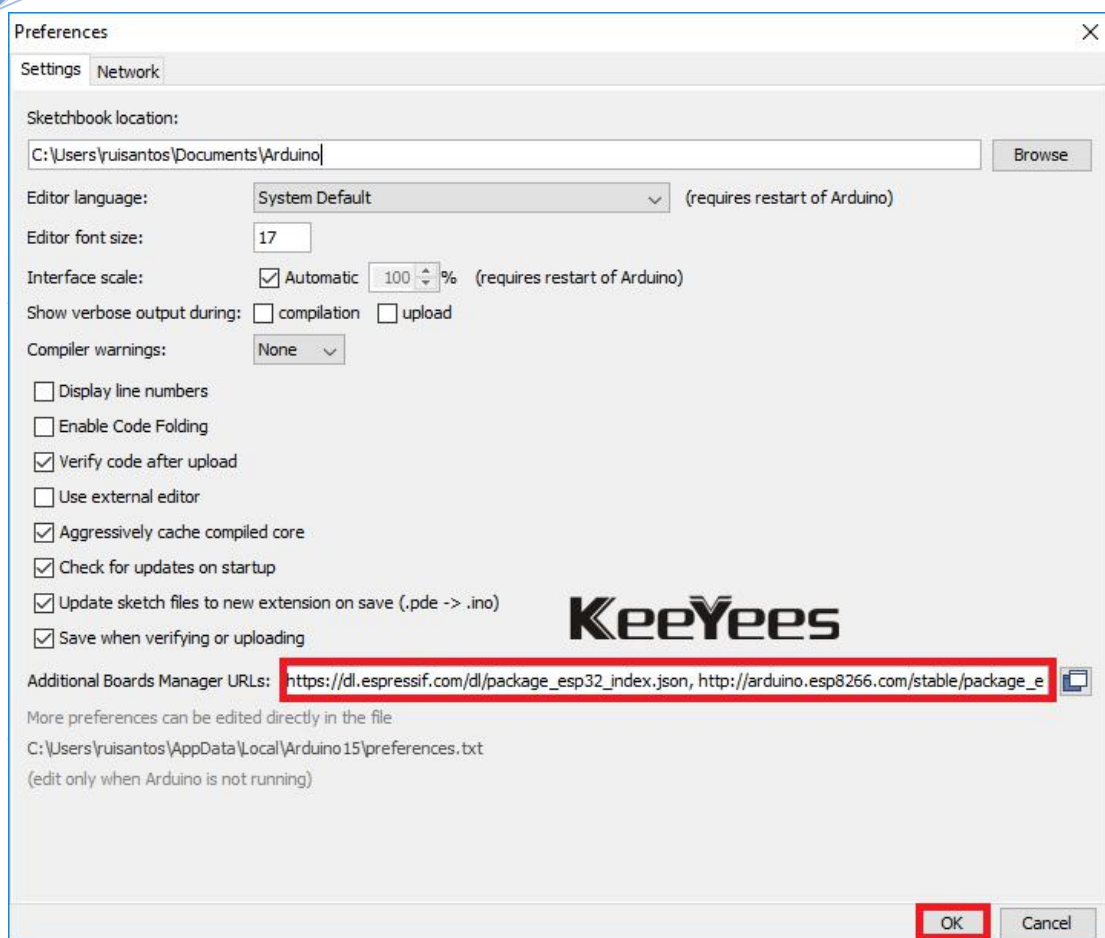
In diesem Beispiel verwenden wir Arduino IDE, um die ESP32-CAM-Karte zu programmieren. Daher müssen Sie Arduino IDE sowie das ESP32-Add-On installiert haben.

Öffnen Sie **Arduino IDE** und klicken Sie auf **File-> Preferences**, wie unten gezeigt.

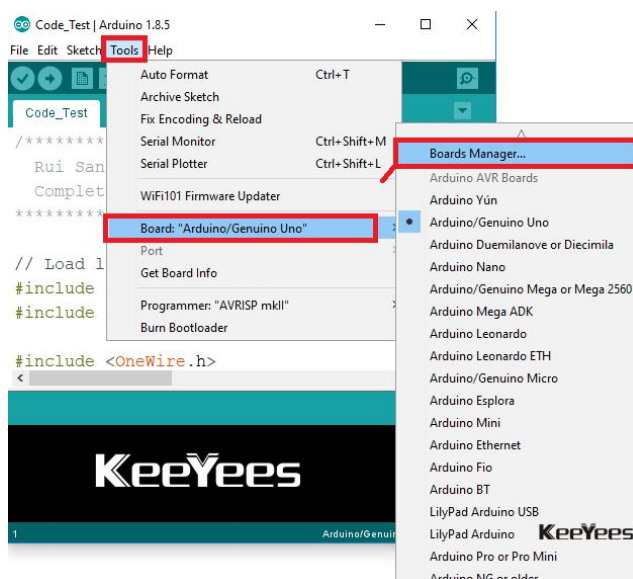


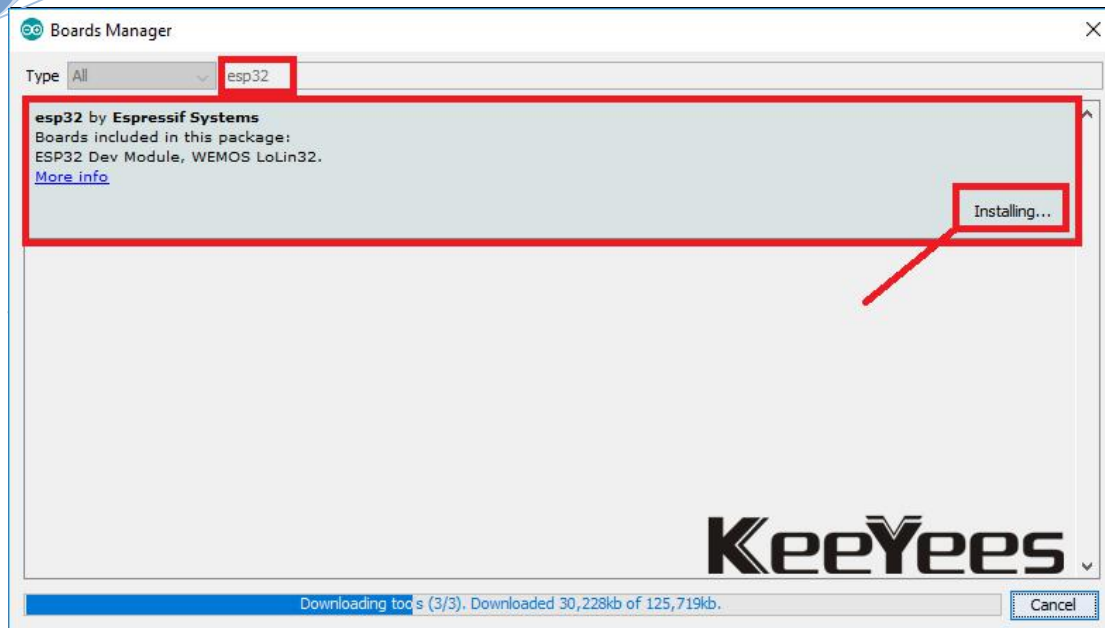
Geben Sie dann [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json) in das Feld **Additilnal Boards Manaper URLS** ein und klicken Sie auf "OK" (siehe unten).





Klicken **tools->board:->Boards Manager**, geben Sie dann **ESP32** in die Popup-Oberfläche ein und klicken Sie auf **Installieren**. Wie nachfolgend dargestellt:





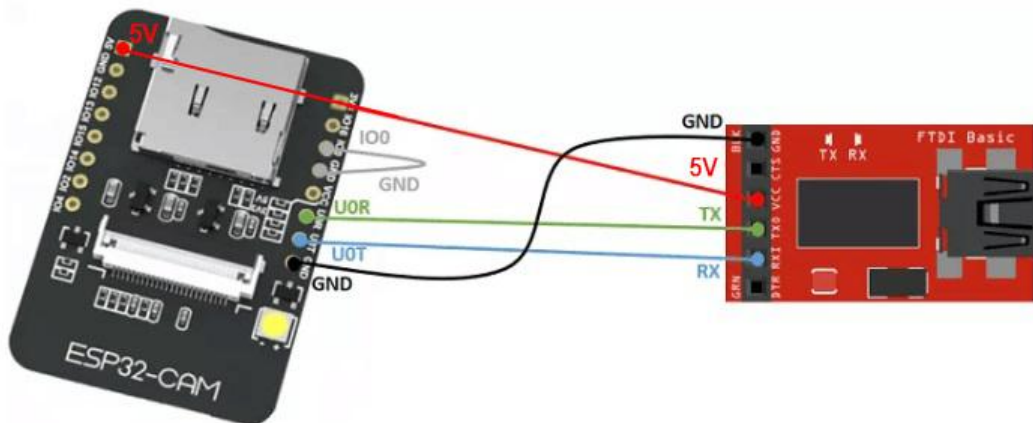
Nachdem Sie die obigen Schritte ausgeführt haben, können Sie das Modul jetzt verwenden.

## Schritte

1. Bereiten Sie eine Speicherkarte vor.
2. Setzen Sie die Speicherkarte und das Kameramodul OV2640 in den Kartenhalter ein.

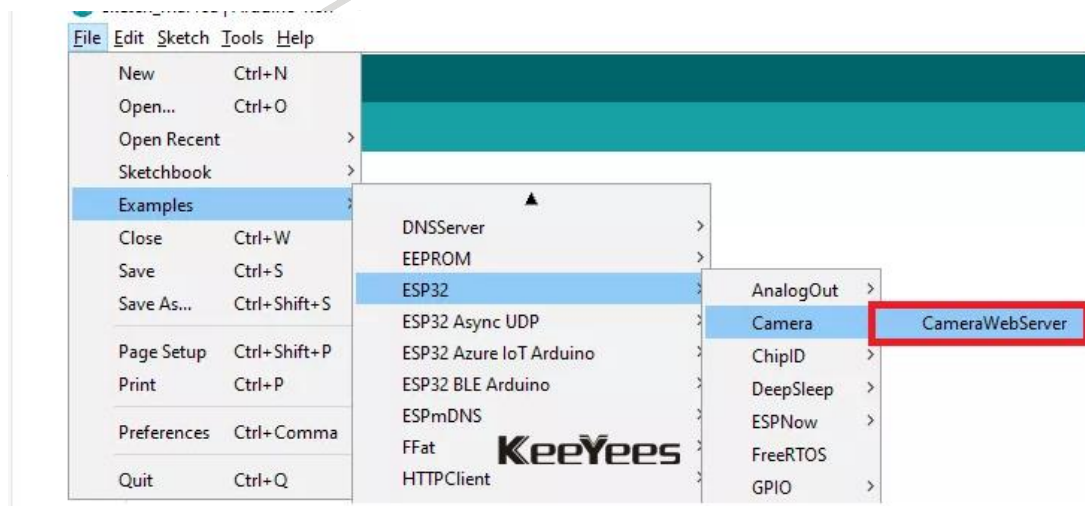
Schließen Sie die ESP32-CAM-Karte mit einem FTDI-Programmiergerät an Ihren Computer an.

Folgen Sie dem nächsten Schaltplan:

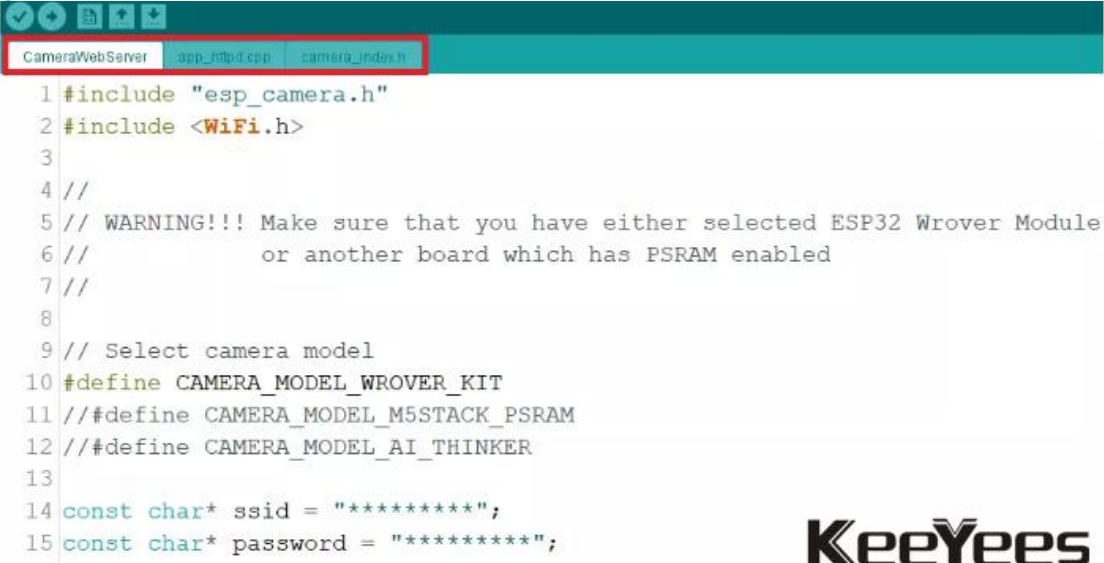


Wichtig: GPIO 0 muss mit GND verbunden sein, damit Sie Code hochladen können.

2. Gehen Sie in Ihrer Arduino IDE zu **File > Examples > ESP32 > Camera** und öffnen Sie das **CameraWebServer** Beispiel.



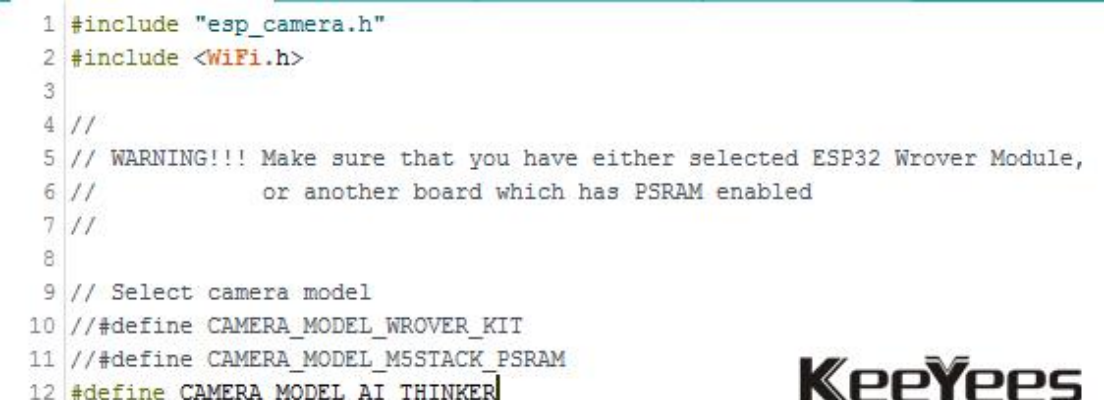
3. Der folgende Code sollte geladen werden.



```
1 #include "esp_camera.h"
2 #include <WiFi.h>
3
4 //
5 // WARNING!!! Make sure that you have either selected ESP32 Wrover Module
6 //           or another board which has PSRAM enabled
7 //
8
9 // Select camera model
10 #define CAMERA_MODEL_WROVER_KIT
11 // #define CAMERA_MODEL_M5STACK_PSRAM
12 // #define CAMERA_MODEL_AI_THINKER
13
14 const char* ssid = "*****";
15 const char* password = "*****";
```

KeeYees

4. Stellen Sie sicher, dass Sie das richtige Kameramodul auswählen. In diesem Fall verwenden wir das **AI-THINKER Model**. Kommentieren Sie die 10. Codezeile **#define CAMERA\_MODEL\_WROVER\_KIT** und kommentieren Sie die 12. Codezeile **#define CAMERA\_MODEL\_AI\_THINKER** aus.



```
1 #include "esp_camera.h"
2 #include <WiFi.h>
3
4 //
5 // WARNING!!! Make sure that you have either selected ESP32 Wrover Module,
6 //           or another board which has PSRAM enabled
7 //
8
9 // Select camera model
10 // #define CAMERA_MODEL_WROVER_KIT
11 // #define CAMERA_MODEL_M5STACK_PSRAM
12 #define CAMERA_MODEL_AI_THINKER
```

KeeYees

5. Sie müssen Ihre Netzwerkanmeldeinformationen in die folgenden Variablen einfügen:



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

```
const char* Passwort = "REPLACE_WITH_YOUR_PASSWORD";
```

Stellen Sie sicher, dass Sie die richtige SSID und das richtige Passwort eingeben.

6. Jetzt kann der Code auf Ihren ESP32 hochgeladen werden.

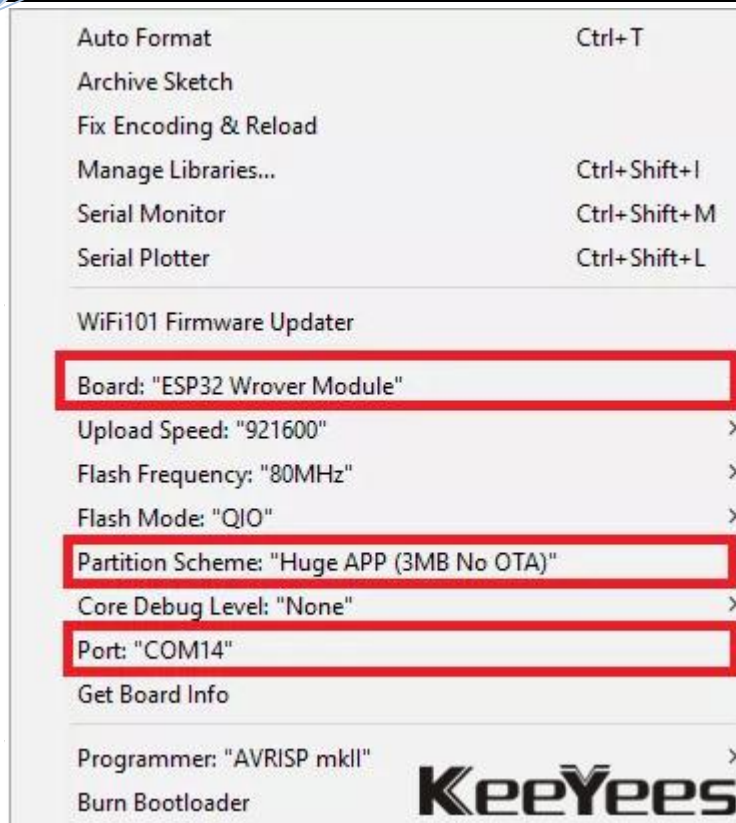
Führen Sie die folgenden Schritte aus, um den Code hochzuladen:

Gehen Sie zu Tools > Board und wählen Sie ESP32 Wrover Module

Gehen Sie zu Tools > Port und wählen Sie den COM-Port aus, an den der ESP32 angeschlossen ist.

Wählen Sie unter Tools > Partition Scheme die Option „Huge APP (3MB No OTA) “.

Klicken Sie dann auf die Schaltfläche zum Hochladen, um den Code hochzuladen.



**Wichtig: Wenn Sie den Code nicht hochladen können, überprüfen Sie, ob GPIO 0 mit GNDs verbunden ist und ob Sie die richtigen Einstellungen im Menü Extras ausgewählt haben. Sie sollten auch die integrierte Reset-Taste drücken, um Ihren ESP32 im blinkenden Modus neu zu starten.**

8. Wenn das in der folgenden Abbildung gezeigte Problem während des Herunterladens auftritt, klicken Sie auf der Entwicklungsplatine auf die Schaltfläche zum Zurücksetzen.





```
Uploading...
Global variables use 52696 bytes (16%) of dynamic memory, leaving 274984 by
esptool.py v2.6
Serial port COM18
Connecting.....
23
```

**Wichtig:** Wenn das Programm immer noch nicht auf die Entwicklungsplatine heruntergeladen werden kann, müssen Sie die Überbrückungskappe des FTDI232-Moduls auf 3,3 V stellen. Der 5-V-Pin auf der Entwicklungsplatine schaltet ebenfalls auf 3,3 V, steckt die Stromversorgung ein und aus und lädt das Programm herunter.

Nachdem der Download abgeschlossen ist, schalten Sie die Überbrückungskappe am FIDI232-Modul auf 5 V und das Netzkabel des ESP32-S-Moduls auf 5 V.

9. Trennen Sie nach dem Hochladen des Codes GPIO 0 von GND.

10. Abrufen der IP-Adresse

Öffnen Sie den seriellen Monitor mit einer Baudrate von 115200.

Drücken Sie die integrierte Reset-Taste von ESP32-CAM.

Die ESP32-IP Adresse sollte im seriellen Monitor ausgedruckt werden.



```
COM14
ets Jun 8 2016 00:22:57

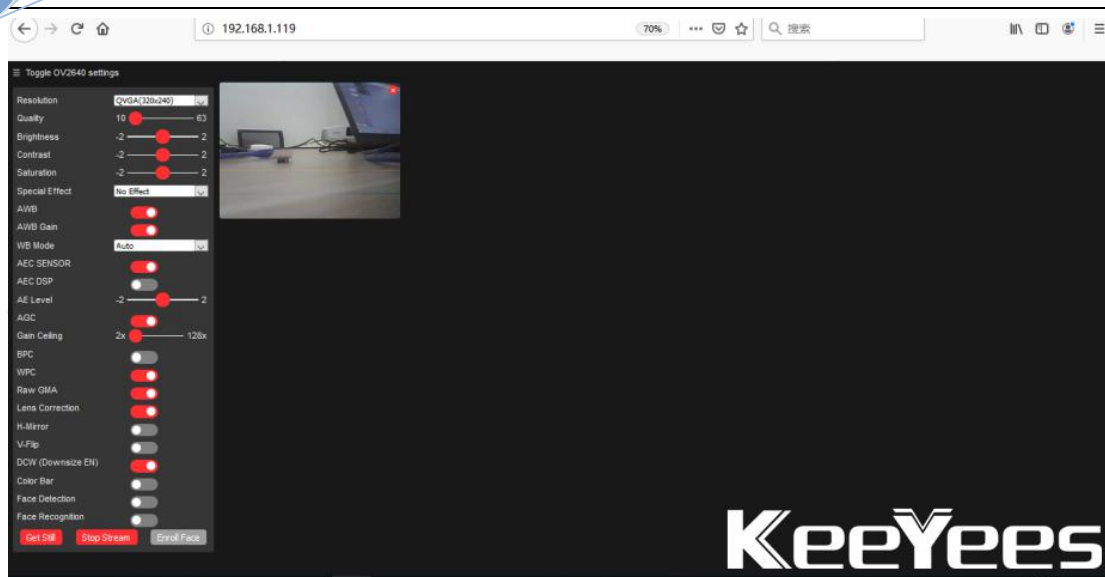
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
config: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1100
load:0x40078000,len:10088
load:0x40080400,len:6380
entry 0x400806a4

..
WiFi connected
Starting web server on port: '80'
Starting stream server on port: '81'
Camera Ready! Use 'http://192.168.1.91' to connect

KeeYees
[Autoscroll] [Show timestamp] Newline 115200 baud Clear output
```

## 11. Zugriff auf den Videokommunikation Server

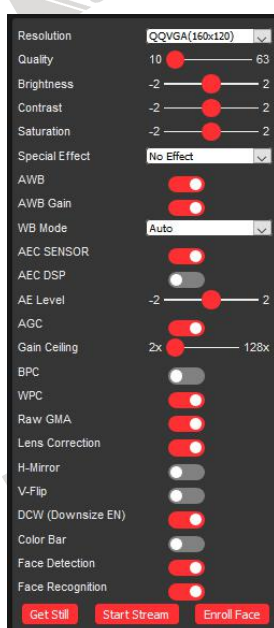
Jetzt können Sie auf Ihren Kamera-Streaming-Server in Ihrem lokalen Netzwerk zugreifen. Öffnen Sie einen Browser und geben Sie die IP-Adresse des ESP32-CAM ein. Klicken Sie auf die **Start Streaming**, um das Videokommunikation zu starten.



Sie können auch Fotos aufnehmen, indem Sie auf die **Get Still** klicken. Es gibt auch verschiedene Kameraeinstellungen, mit denen Sie die Bildeinstellungen anpassen können.

Schließlich können Sie die Gesichtserkennung und -erkennung durchführen.

Klicken Sie auf die **Face Detection** und **Face Recognition** Tasten (wenn die Auflösung größer ist, müssen Sie sie verkleinern).





Sie müssen ein neues Gesicht registrieren. Es werden mehrere Versuche unternommen, das Gesicht zu retten. Nach der Registrierung eines neuen Benutzers sollte er das Gesicht später erkennen (Betreff 0).

Und das ist es. Jetzt können Sie Ihren Videokommunikation Webserver mit Gesichtserkennung und -erkennung verwenden.

## Probleme, die Sie möglicherweise haben

### Problem 1

camera\_probe(): Detected camera not supported

SCCB\_Write(): SCCB\_Write Failed addr:0x30, reg:0xff, data:0x01, ret:-1

SCCB\_Write(): SCCB\_Write Failed addr:0x30, reg:0xff, data:0x01, ret:-1

SCCB\_Write(): SCCB\_Write Failed addr:0x30, reg:0x12 data:0x01, ret:-1

esp\_camera\_init(): Camera probe failed with error 0x20004

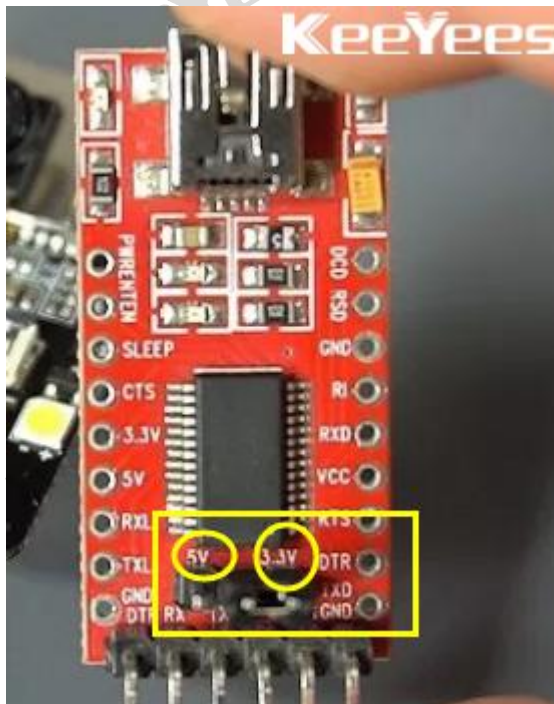
**Grund: OV2640 Kameramodul ist nicht richtig gesteckt!**



## Problem 2

brownout detector was triggered

Grund: Dieses Problem kann auftreten, wenn Sie eine Verbindung zu WiFi herstellen! Das FT232RL FTDI Mini-USB-TTL-Konvertermodul verfügt über eine Triggerfunktion. Wenn die Spannung zu niedrig ist, wird die Karte automatisch neu gestartet, um die Karte zu schützen.



Wie man es löst?

1. Stellen Sie den USB-TTL auf 3,3 V ein.
2. Schließen Sie es wie in allen Diagrammen gezeigt an die ESP32-CAM an.
3. Verbinden Sie Io0 und gnd. (Stellen Sie sicher, dass die Pins korrekt



sind. Es ist sehr wichtig.)

4. Schalten Sie den Code ein und laden Sie ihn hoch.

Testen Sie nun die ESP32-CAM

1. Entfernen Sie den IO0 und den GND-Jumper.
2. Ändern Sie den USB-TTL auf 5V (Ändern des Pins).
3. Ändern Sie die Spannung am ESP32-CAM auf 5V Pin.
4. Einschalten.
5. Öffnen Sie den seriellen Monitor.
6. Drücken Sie die Reset-Taste am ESP32-CAM.
7. Rufen Sie die IP-Adresse ab.

### Problem 3

rst:0x1 (POWERON\_RESET),boot:0x13 (SPI\_FAST\_FLASH\_BOOT)

configip: 0, SPIWP:0xee

clk\_drv:0x00,q\_drv:0x00,d\_drv:0x00,cs0\_drv:0x00,hd\_drv:0x00,wp\_drv:  
0x00

mode:DIO, clock div:1

load:0x3fff0018,len:4

load:0x3fff001c,len:1100

load:0x40078000,len:10088





load:0x40080400,len:6380

entry 0x400806a4

Gund: Keine IP-Adresse Wifi ist nicht erfolgreich verbunden.

Serial.println("checking wifi connecting");

Serial.print("connect fail,retry");

Stellen Sie sicher, dass die WLAN-SSID und das Passwort zu 100% korrekt sind.

```
CameraWebServer | Arduino 1.8.9
文件 编辑 项目 工具 帮助

CameraWebServer$ app_httpd.cpp camera_mdsh.h camera_pins.h

return;
}
Serial.println("begin");
sensor_t * s = esp_camera_sensor_get();
//initial sensors are flipped vertically and colors are a bit saturated
if (s->id.PID == OV3660_PID) {
  s->set_vflip(s, 1); //flip it back
  s->set_brightness(s, 1); //up the blightness just a bit
  s->set_saturation(s, -2); //lower the saturation
}
//drop down frame size for higher initial frame rate
s->set_framesize(s, FRAMESIZE_QVGA);

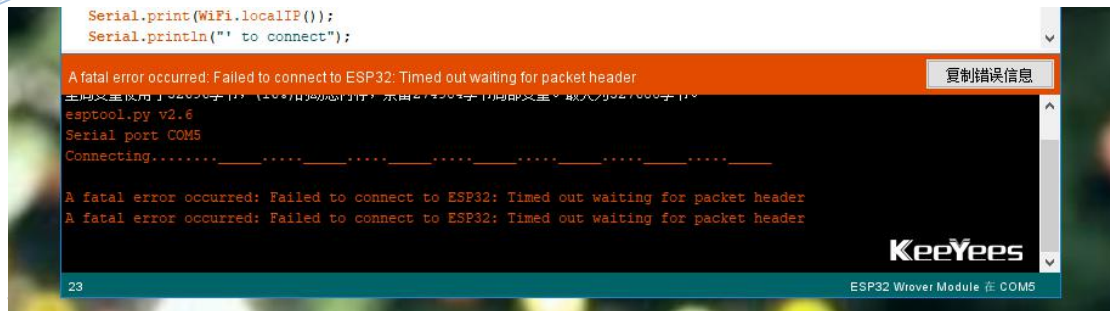
#if defined(CAMERA_MODEL_M5STACK_WIDE)
  s->set_vflip(s, 1);
  s->set_hmirror(s, 1);
#endif
//WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); //disable brownout detector
Serial.println("checking wifi connecting");
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print("connect fail,retry");
  Serial.print(".");
}
Serial.println("WiFi connected");

startCameraServer();

Serial.print("Camera Ready! Use 'http://'");
Serial.print(WiFi.localIP());
Serial.println(""); to connect";
```

#### Problem 4

Upload fehlgeschlagen



Stellen Sie sicher, dass die RXD- und TXD-Pins des ESP32-S-Moduls korrekt mit den FTDI232-Modulpins verbunden sind.

Drücken Sie die Reset-Taste, um zu bestätigen, dass das Licht an ist.



Nachdem der Upload erfolgreich war, trennen Sie GND und GPIO0. Zurück zum CameraWebServer, um neu zu starten.



```
CameraWebServer | Arduino 1.8.9
文件 编辑 项目 工具 帮助

CameraWebServer app_httpd.cpp camera_index.h camera_pins.h
Serial.print("Camera init failed with error 0x1x", err);
return;
}
Serial.println("begin4");
sensor_t * s = esp_camera_sensor_get();
//initial sensors are flipped vertically and colors are a bit saturated
if (s->id.PID == OV3660_PID) {
  s->set_vflip(s, 1); //flip it back
  s->set_brightness(s, 1); //up the brightness just a bit
  s->set_saturation(s, -2); //lower the saturation
}
//drop down frame size for higher initial frame rate
s->set_framesize(s, FRAMESIZE_QVGA);

#ifdef CAMERA_MODEL_M5STACK_WIDE
  s->set_vflip(s, 1);
  s->set_hmirror(s, 1);
#endif
//WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); //disable brownout detector
Serial.println("checking wifi connecting");
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.println("connect fail,retry");
  Serial.print(".");
}
Serial.println("WiFi connected");

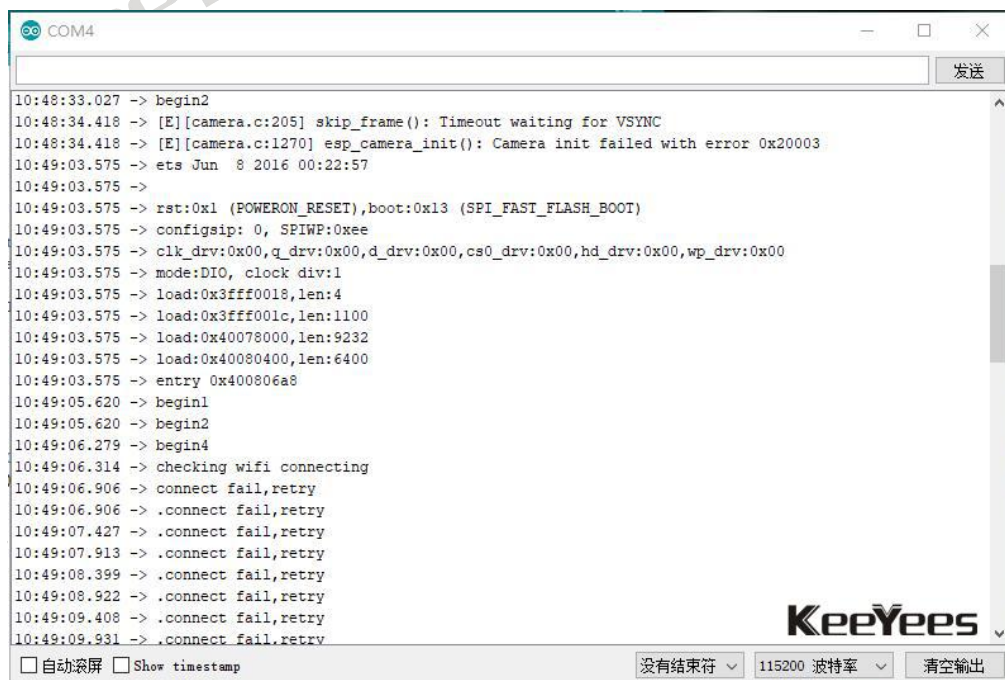
startCameraServer();

Serial.print("Camera Ready! Use 'http://");
Serial.print(WiFi.localIP());
Serial.println("' to connect");

上传成功。
Compiling sketch...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (134 compressed) at 0x00008000 in 0.0 seconds (effective 1536.0 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...

23 ESP32 Wrover Module @ COM5
```

Wenn dies immer noch fehlschlägt, ändern Sie den USB-TTL auf 5 V und die Spannung am ESP32-CAM auf 5 V. Dann zurücksetzen.



```
COM4
10:48:33.027 -> begin2
10:48:34.418 -> [E][camera.c:205] skip_frame(): Timeout waiting for VSYNC
10:48:34.418 -> [E][camera.c:1270] esp_camera_init(): Camera init failed with error 0x20003
10:49:03.575 -> ets Jun 8 2016 00:22:57
10:49:03.575 ->
10:49:03.575 -> rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
10:49:03.575 -> configsip: 0, SPIWP:0xee
10:49:03.575 -> clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
10:49:03.575 -> mode:DIO, clock div:1
10:49:03.575 -> load:0x3fff0018,len:4
10:49:03.575 -> load:0x3fff001c,len:1100
10:49:03.575 -> load:0x40078000,len:9232
10:49:03.575 -> load:0x40080400,len:6400
10:49:03.575 -> entry 0x400806a8
10:49:05.620 -> begin1
10:49:05.620 -> begin2
10:49:06.279 -> begin4
10:49:06.314 -> checking wifi connecting
10:49:06.906 -> connect fail,retry
10:49:06.906 -> .connect fail,retry
10:49:07.427 -> .connect fail,retry
10:49:07.913 -> .connect fail,retry
10:49:08.399 -> .connect fail,retry
10:49:08.922 -> .connect fail,retry
10:49:09.408 -> .connect fail,retry
10:49:09.931 -> .connect fail,retry

☐ 自动滚屏 ☐ Show timestamp 没有结束符 115200 波特率 清空输出
```